

**Unveiling Collaborative Group Identities
in Social Synthetic Environments
from Interaction Data**

**Thesis submitted for the degree of Doctor of Philosophy
Corrado Grappiolo**

Supervisors: Georgios N. Yannakakis and Julian Togelius

Abstract

One of the most fundamental — possibly culturally independent — characteristics of human societies is the ability to gather in groups and manifest group behaviours. This social property has implications on how individuals interact and *cooperate* with each other. The study of group behaviours is not only fundamental for better understanding phenomena such as social conflicts, but it can also contribute to the understanding of how the environment affects their evolution. Anthropologically, this study can also contribute to explain how and why humans, millions of years ago, started hunting together, farming together, up to, ultimately, form civilisations. One of the most difficult challenges related to group modelling resides on the fact that the societies are *complex systems*, in which the global patterns — e.g. groups — cannot be explained by the mere sum of its single units — e.g. individuals. Moreover, the group structures might even be dynamic: whilst new groups can form, others can merge into others, and so on.

The work presented in this dissertation aims to provide a computational framework capable of inferring the presence of group structures, and consequently assign group identities, to populations of socially driven individuals, by solely analysing the ongoing levels of cooperation of the interactions. Our group modelling framework is intended to be used in computer-mediated interaction scenarios, for simplicity called social synthetic environments, which can be effectively used to simulate aspects of real-life, yet by maintaining a customisable level of control of the phenomena under investigation. Examples of social synthetic environments are, among others, theoretical games e.g. prisoner’s dilemma or the ultimatum game, and cooperative computer games e.g. massively multiplayer online games or serious multiplayer games.

The proposed framework is composed of two pipelined modules. The first one, namely cooperation modelling, constitutes the interface between the framework and the social synthetic environment under investigation. Cooperation modelling is responsible for monitoring the ongoing interactions, evaluating their levels of cooperation, and maintaining up-to-date estimates of the whole society’s complex network. The network is built in a way that each node represents an individual of the society, and each link the directed amount of cooperation, from the individual who performs the action to the individual who receives it. The second module, namely group identity detection, leverages on the information held by the cooperation network and partitions it into groups, so that the *within*-group cooperation is maximised, and the *between*-group cooperation is minimised.

In this dissertation we investigated a plethora of computational methods for all the tasks and modules of the framework. More specifically, (1) we relied on the so-called *one-to-many* approach to gather and evaluate the ongoing interactions, in accordance with we observe how *one* individual treats *many* others, under the same interaction protocol, though these interactions may have not occurred at the same time; (2) we considered two methods for the evaluation of the interactions into normalised cooperation values; (3) we considered three cooperation network update rules, two of which widely used in e.g. reinforcement learning; (4) we relied on evolutionary computation for the network partitioning task. With respect to the latter, we examined three possible chromosome representations, two evolutionary approaches — namely single population and niching — and four mechanisms for niching activation.

We conducted thorough empirical investigations based on a benchmark experiment in social dilemmas where the impact of induced group on cooperation and reciprocity has been studied. From the authentic data of that benchmark study we extracted the reported behaviours that were most suitable for our investigation, and also used them to derive artificial societies of *believable* agents. The results obtained showed promising inferential performance of our group modelling framework. More specifically, we identified the most effective

configuration composed of (1) a *one-to-many* approach which aggregates an amount of current and past interactions relatively small with respect to the total volume of interactions; (2) a cooperation evaluation mechanism which classifies the interactions into discretised labels of *in-groupness* and *out-groupness*; (3) a cooperation network update rule extensively used in many other research fields, such as reinforcement learning, artificial neural networks, and ant colony optimisation; (4) a single population-based evolutionary algorithm which evolves its candidate solutions — representing network partitions into groups or *community structures*, each gene corresponding to a node of the network, and each allele value corresponding to the node’s community identity — via the well-established *modularity maximisation* technique. The most effective framework configuration showed good inference performance on the actual human-based behavioural data and across seven artificial society scenarios, though for the latter case it manifested a suboptimal speed of convergence.

This dissertation successfully leverages on the hypothesis that it is possible to infer the existence of group structures by solely focusing on the ongoing interactions. As a consequence, this work contributes to a plethora of fields such as evolution of cooperation, complex systems and network theory, and evolutionary computation. The limitations of the approach, and possible strategies to overcome them, are also proposed.

Contents

1	Introduction	1
1.1	Evolution of Cooperation	1
1.1.1	Game Theory and Evolutionary Dynamics	2
1.1.2	Social Dilemmas and Social Preferences	3
1.1.3	Social Dilemmas and Group Identities	4
1.2	Collaborative Computer Games and Crowdsourcing	4
1.3	Complex Systems and Complex Networks	5
1.4	Motivation and Aim of this Thesis	6
1.4.1	Hypothesis and Main Research Questions	7
1.4.2	Overview of the Framework, Challenges and Approaches	8
1.4.3	Case Studies and Main Findings	10
1.5	Summary of Contributions	11
1.5.1	List of Papers	11
1.6	Outline of this Dissertation	13
1.7	Summary	14
2	Related Work	15
2.1	Social Dilemmas and Cooperation	15
2.2	Agent-based Modelling and Evolutionary Dynamics	16
2.3	Group Behaviour and Identities	17
2.4	Community Structure Detection	18
2.4.1	Undirected and Unweighted Networks	18
2.4.2	Directed and/or Weighted Networks	19
2.5	Community Structure Detection via Evolutionary Computation	20
2.5.1	Node Representation	20
2.5.2	Edge Representation	20
2.5.3	Group Representation	21
2.5.4	Multiobjective Optimisation and Niching	21
2.6	Other Related Work	22
2.6.1	Overlapping Community Structure Detection	22
2.6.2	Social Network Studies	22
2.6.3	Cooperative and Multiplayer Computer Games	23
2.7	Summary	23
3	Tools	24
3.1	Network Theory	24
3.1.1	Note on Terminology	24
3.1.2	Network Types and Properties	25
3.1.3	Edge Degree	27
3.1.4	Random Walks on Graphs	27

3.2	Community Structure Detection	27
3.2.1	Modularity	28
3.2.2	Modularity Maximisation via Spectral Partitioning	29
3.2.3	Critiques against Modularity Maximisation	30
3.3	Evolutionary Computation	31
3.3.1	Evolutionary Population, Representation and Initialisation	31
3.3.2	The Fitness Function and the Evolutionary Loop	32
3.3.3	Parent Selection	33
3.3.4	Variation Operators	33
3.3.5	Survivor Selection	35
3.3.6	Additional Features: Niching	35
3.4	Reinforcement Learning	36
3.4.1	Value Function and Update Rules	36
3.4.2	The n -armed Bandit Problem	37
3.5	Summary	38
4	Benchmark Social Synthetic Environments	39
4.1	The Experiment by Chen and Li	40
4.1.1	Stage $S1$: Group Assignment	41
4.1.2	Stage $S2$: Online Chat	41
4.1.3	Stage $S3$: Other-other Allocation Game	41
4.1.4	Stage $S4$: Sequential Game	42
4.2	Chen and Li-based Simulation	43
4.3	Chen and Li Experiment's Derived Artificial Society	43
4.4	Summary	46
5	Interaction-based Cooperation Modelling	48
5.1	The <i>One-to-many</i> Approach	48
5.2	Cooperation Evaluation	49
5.3	Cooperation Network Update	51
5.3.1	Cooperation Update via Sample Average	51
5.3.2	Cooperation Update via Constant- α	51
5.3.3	Cooperation Update via Temporal Difference	52
5.4	Summary	52
6	Cooperation Modelling Empirical Evaluation	53
6.1	Performance Measure	53
6.2	Evaluation Protocol	54
6.3	Analysis of the <i>In/Out-group</i> Dichotomy	54
6.4	Preliminary Analysis of α	55
6.5	Preliminary Analysis of γ	57
6.6	Cooperation Modelling Analysis	57
6.7	Summary	60
7	Cooperative Group Identity Detection	67
7.1	Group Identity Detection via Modularity Optimisation	67
7.2	Modularity Maximisation via Evolutionary Computation	68
7.3	The Template Evolutionary Algorithm	69
7.4	Node Representation	69
7.4.1	Chromosome Initialisation	70
7.4.2	Crossover Operator	70
7.4.3	Mutation Operator	70

7.5	Group Representation	70
7.5.1	Chromosome Initialisation	71
7.5.2	Crossover Operator	71
7.5.3	Mutation Operator	71
7.5.4	Group Inversion Operator	72
7.6	Edge Representation	72
7.6.1	Chromosome Initialisation	73
7.6.2	Crossover Operator	73
7.6.3	Mutation Operator	73
7.7	Niching Approach	74
7.7.1	Niching Repair Procedure	76
7.7.2	Niching Activation Mechanisms	76
7.7.3	Reading Head Niching Activation	77
7.7.4	Roulette Selection Niching Activation	77
7.7.5	n -armed Bandit Niching Activation	77
7.8	Summary	78
8	Group Identity Detection Empirical Evaluation	83
8.1	Performance Measures	83
8.1.1	Normalised Mutual Information	84
8.1.2	Hungarian Algorithm Matching	84
8.2	Description of the Artificial Society Scenarios	85
8.2.1	Analysis of the Cooperation Modelling Phase	85
8.3	Chromosome Representation Analysis	87
8.3.1	Analysis of $m = 4$ Case	88
8.3.2	Analysis of $m = \{7, 11, 17\}$ Cases	90
8.4	Seeking for Improvements by Means of Niching	93
8.4.1	Analysis of Niching Based on the Exact Knowledge of m	93
8.4.2	Analysis of Niching Activation Mechanisms	96
8.5	Analysis of the Importance of the Fitness Function: Comparison with LinkRank	101
8.5.1	Analysis of the Chromosome Representation	103
8.5.2	Analysis of the Niching Activation Mechanism	103
8.6	Summary	107
9	Interaction-based Group Modelling Applied to the Experiment by Chen and Li	110
9.1	Generation of the Interaction Data	110
9.2	Cooperation Modelling Analysis	111
9.3	Group Identity Detection Analysis	112
9.4	Summary	114
10	Discussion	115
10.1	Recap, Limitations and Opportunities	115
10.1.1	The <i>One-to-many</i> Approach to Interaction Modelling	115
10.1.2	Cooperation Modelling	116
10.1.3	Group Identity Detection	118
10.1.4	Chromosome Representation	118
10.1.5	Niching Approach	120
10.1.6	Target Objective	121
10.1.7	Interaction-based Group Modelling as a Whole	122
10.1.8	Community Structure Detection	123
10.2	Extensibility	125
10.2.1	<i>One-to-many</i> Approach	125

10.2.2	Cooperation Modelling	125
10.2.3	Group Identity Detection	126
10.2.4	Interaction-based Group Modelling Framework as a Whole	127
10.3	Summary	127
11	Conclusions	128
11.1	Summary of Contributions	130
12	Chen and Li's Experiment Simulation	132
13	Cooperation Modelling	145
13.1	One-to-many Approach and the Other-other Game	145
13.2	Cooperation Evaluation and the Other-other Game	145
13.3	Cooperation Update via alpha Constant- α	146
13.4	Constant- α incremental analysis	146
13.5	Temporal Difference incremental analysis	146
13.5.1	Temporal Difference Does Not Maintain Normalised \mathcal{C} values	147
13.5.2	Analysis of the Sign of Temporal Difference Incremental	147
14	Group Identity Detection Empirical Evaluation — Secondary Results	149
14.1	Analysis of Niching Based on the Exact Knowledge of m	149
14.2	Analysis of the Importance of the Fitness Function: Comparison with LinkRank	150
14.2.1	Node/Edge/Group-based UCB1 algorithms	150
14.2.2	Node/Edge/Group-based ϵ greedy algorithms	171
14.2.3	Node/Edge/Group-based Roulette Wheel Selection algorithms	178
14.2.4	Edge/Group-based Reading Head algorithms	185

List of Figures

1.1	The schematic representation of the ultimatum game.	3
1.2	The schematic representation of our interaction-based group modelling framework.	9
1.3	Snapshots of the fairness of resource allocation adaptive game [85]. Figure 1.3(a) represents an in-game snapshot, whilst Figure 1.3(b) shows the adaptive levels generated via the experience-driven procedural content generation paradigm [224] throughout the whole game. The farther-most level corresponds to the first one; according to the player behaviour different level sizes, number of receiver agents and resources to be allocated were procedurally generated.	12
3.1	An example of an adjacency matrix and the corresponding unweighted and undirected network.	25
3.2	An example of a directed network; adjacency matrix and graph representation.	25
3.3	An example of a weighted network; adjacency matrix and graph representation.	26
3.4	An example of a directed and weighted network; adjacency matrix and graph representation.	26
3.5	An example of an undirected and unweighted network with scale-free property and four communities detected through the Clauset-Newman-Moore algorithm [34]. The nodes with the same colours belong to the same community structure.	28
3.6	Modularity, applied to the two subnetwork at the left (orange) and right (green) would return the same value, though if we focused on the <i>flow</i> of the edges we observe that they represent two different configurations. The LinkRank algorithm, based on a random walk approach and another modularity measure, aims to solve this ambiguity.	30
3.7	The general scheme of an Evolutionary Algorithm as a flow chart [49].	32
3.8	Schematic representation of (a) single point, (b) double point, and (c) uniform crossover operators.	34
4.1	A possible classification of social synthetic environments, based on their level of <i>realism</i> , that can implement in order to reproduce, simulate, or explain, real-life phenomena.	40
4.2	The schematic representation of the other-other game used by Chen and Li in their experiment [30]. A proposer P receives an endowment e . She now has to split it between two receivers, R_1 and R_2 , and cannot keep anything for herself. o_1 and o_2 are the splits for R_1 and R_2 respectively.	42
4.3	An example of how the other-other game triplets (P, R_1, R_2) are generated, based on a society of six participants. Individuals 1, 2 and 4 belong to the orange group, individuals 3, 5 and 6 to the green group. A random permutation is generated. Individual 5 will be the proposer for individuals 1 and 4, individual 1 will be the proposer for individuals 4 and 2, and so on. Individuals 5 and 2 will face an <i>out-out</i> interaction type; individuals 1 and 6 an <i>in-in</i> interaction type; individuals 3 and 4 will face an <i>in-out</i> interaction type.	42
5.1	The schematic representation of the cooperation evaluation task and the three possible ways considered in our dissertation: raw interactions (that is no cooperation evaluation); min-max normalisation of the interactions into continuous probabilities of <i>in-group</i> -ness; discretisation of the interactions into <i>in-group</i> and <i>out-group</i> labels/values.	49

5.2	The schematic representation of the cooperation network update task and the three possible update rules considered in our study. The weights of the edges, together with cooperation values $\{C_{i,P}, C_{j,P}, \dots, C_{k,P}\}$ used by <i>TD0</i> update rule, were omitted in order not to overcomplicate the Figure.	50
6.1	The probability distributions of observing <i>in-in</i> , <i>out-out</i> , and <i>in-out</i> interactions, under the <i>in/out-group</i> dichotomy, depending on the relationship between <i>in-group</i> sizes, <i>out-group</i> size, and hence the size n of society \mathcal{S}	55
6.2	The impact that α has on the maximisation of the <i>true</i> Q , when the min-max normalisation and constant- α network update rule are used, in presence of <i>CLAS</i> societies of size $n = 128$, $m = 2$, and the two partitions of Table 6.2 are considered.	56
6.3	The impact that γ has on the maximisation of Q , when the min-max normalisation and constant- α network update rule are used, with $\alpha = \{0.5, 1\}$, in presence of <i>CLAS</i> societies of size $n = 128$, $m = 2$, and the two partitions of Table 6.2 are considered.	58
6.4	Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{AD} society type and depending on two different sizes of t	59
6.5	Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{AD} society type and depending on two different sizes of t	60
6.6	Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{AD} society type and depending on two different sizes of t	61
6.7	Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{AD} society type and depending on two different sizes of t	62
6.8	Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{BC} society type and depending on two different sizes of t	63
6.9	Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{BC} society type and depending on two different sizes of t	64
6.10	Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{BC} society type and depending on two different sizes of t	65
6.11	Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{BC} society type and depending on two different sizes of t	66
7.1	An example of a node representation-based chromosome. The reference network is the one of Figure 3.1. The allele values of the chromosomes are represented by colours in the corresponding network.	69
7.2	An example of a chromosome adopting the group representation. The reference network is the one of Figure 3.1. The allele values of the object (left) part of the chromosome correspond to the group identities of the nodes, and are represented by colours in the corresponding network. The (right) group part of the chromosome is used for the recombination operators.	71

7.3	An example of uniform crossover operation, performed at the node part, which motivated Falkenauer to introduce the group representation [55]. The two parents, depicted on top of the figure, lead to the same partitioning into four groups. Under a group perspective, the generated offsprings should be clones of the parents. However, through node-based (uniform) crossover there is a risk that the generated offsprings (at the bottom of the figure) would lead to partitioning much different from the original parents, in this case both of them leading to two group structures.	79
7.4	An example of a group representation-based crossover operator. The two parent chromosomes are depicted at the top of the figure, the two generated offsprings are depicted at the bottom. The intermediate steps of the two crossover operators are depicted in between.	80
7.5	The six possible events in group evolution defined by Palla et al. [170] which inspired our group-based mutation operators. Each of the six mutation operations are presented in pseudocode in Algorithms 8 to 13.	81
7.6	An example of inversion operation for group representation. The inversion operator, applied to the dark green labels of the group part, transforms the topmost into the lowermost chromosome, leaves the node part unaltered.	81
7.7	An example of an edge representation-based chromosome. The reference network is the one of Figure 3.1. The allele value j of gene i is translated into the edge (i, j) . The subsequent conversion step will detect the chromosome subnetworks and then assign group identities. . .	82
7.8	An example of chromosome repair in an edge representation-based chromosome. Genes 1 and 5 are the only ones which allele values correspond to edges present in adjacency matrix of Figure 3.1. The other six genes are repaired; their allele values are chosen among their corresponding neighbouring nodes proportionally to their weights. Intuitively, in case of unweighted networks, each neighbouring node is chosen according to uniform distribution.	82
8.1	An example depicting how the hungarian algorithm, and consequently NHS , compute the string matching score between two strings. Figure 8.1(a) depicts the two strings that need to be aligned; Figure 8.1(b) is the related confusion matrix; Figure 8.1(c) corresponds to the optimal alphabet assignment computed by the Hungarian algorithm; Figure 8.1(d) depicts the calculation of matches and mismatches after the alphabet assignment; and finally Figure 8.1(e) normalises the alignment score NHS	84
8.2	The power law group size distribution of the 30 CLAS societies considered for three m values are examined ($m = \{7, 11, 17\}$).	86
8.3	The average modularity score, and its standard deviation, of the <i>true</i> groups over the ongoing cooperation network, inferred by the CM module, depending on the seven society types considered for the GID empirical evaluation.	87
8.4	Performance (NMI , NHS , and inferred number of groups) of the three chromosome representations compared to spectral partitioning for $m = 4$ equal size group types.	89
8.5	Performance (NMI , NHS , and inferred number of groups) of the three chromosome representation-based EAs compared to spectral partitioning for $m = \{7, 11, 17\}$ equal group size distribution.	91
8.6	Performance (NMI , NHS , and inferred number of groups) of the three chromosome representation-based EAs compared to spectral partitioning for $m = \{7, 11, 17\}$ power law group size distribution.	92
8.7	Performance (NMI , NHS) of N-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and spectral partitioning SP, for $m = 4$ equal size group types.	94
8.8	Performance (NMI , NHS) of N-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = \{7, 11, 17\}$ equal size group types.	95
8.9	Performance (NMI , NHS) of N-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = \{7, 11, 17\}$ power law group size distribution types.	96

8.10	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of the four node-based niching activation algorithms, compared to N-EA and SP, for $m = 4$ equal size group types.	97
8.11	Average performance (<i>NMI</i> , <i>NHS</i> and inferred number of groups) and standard deviation of the four node-based niching algorithms compared to SP and N-EA, for $m = \{7, 11, 17\}$ equal size society types.	99
8.12	Average performance (<i>NMI</i> , <i>NHS</i> and inferred number of groups) and standard deviation of the four node-based niching algorithms compared to SP and N-EA, for $m = \{7, 11, 17\}$ power law group size distribution society types.	100
8.13	Average ratio and standard deviation between the inferred modularity $Q(\mathcal{H})$, and the <i>true</i> modularity $Q(\mathcal{T})$, of N-EA, G-EA, E-EA and SP, for $m = \{4, 7, 11, 17\}$ equal size group distribution.	102
8.14	Average ratio and standard deviation between the inferred modularity $Q(\mathcal{H})$, and the <i>true</i> modularity $Q(\mathcal{T})$, of N-EA, G-EA, E-EA and SP, for $m = \{7, 11, 17\}$ power law group size distribution.	103
8.15	Comparison of the average performance and standard deviation of the node, group, and edge-based chromosome representation, single population evolutionary algorithms, depending on whether modularity (N-EA-Q, G-EA-Q, and E-EA-Q respectively) or LinkRank (N-EA-LR, G-EA-LR, and E-EA-LR respectively) is used as objective function, for $m = 4$ equal group size distribution.	104
8.16	Comparison of the average performance and standard deviation of the node, group, and edge-based chromosome representation, single population evolutionary algorithms, depending on whether modularity (N-EA-Q, G-EA-Q, and E-EA-Q respectively) or LinkRank (N-EA-LR, G-EA-LR, and E-EA-LR respectively) is used as objective function, for $m = \{7, 11, 17\}$ equal group size distribution.	105
8.17	Comparison of the average performance and standard deviation of the node, group, and edge-based chromosome representation, single population evolutionary algorithms, depending on whether modularity (N-EA-Q, G-EA-Q, and E-EA-Q respectively) or LinkRank (N-EA-LR, G-EA-LR, and E-EA-LR respectively) is used as objective function, for $m = \{7, 11, 17\}$ power law group size distribution.	106
8.18	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N-EA and node-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function.	107
8.19	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N-EA and node-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = \{7, 11, 17\}$ equal size types.	108
8.20	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N-EA and node-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = \{7, 11, 17\}$ power law group size distribution types.	109
9.1	average <i>True</i> modularity score and related standard deviation, of the 25 experimental runs of the CL simulation, depending on different temporal sizes of accumulation of the interactions.	112
9.2	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of SP, N-EA-280 and N-EA-2800, based on the networks obtained from a CM phase with $t = \{1, 5\}$	113
10.1	An example of two partition matchings for which <i>NHS</i> returns 0.66667 for both cases, whilst <i>NMI</i> returns 0.5158 in the first example and 0.5794 in the second example.	124
14.1	Performance (<i>NMI</i> , <i>NHS</i>) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 4$ equal size group types.	150

14.2	Performance (<i>NMI</i> , <i>NHS</i>) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 7$ equal size group types.	151
14.3	Performance (<i>NMI</i> , <i>NHS</i>) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 11$ equal size group types.	152
14.4	Performance (<i>NMI</i> , <i>NHS</i>) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 17$ equal size group types.	153
14.5	Performance (<i>NMI</i> , <i>NHS</i>) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 7$ power law group size types.	154
14.6	Performance (<i>NMI</i> , <i>NHS</i>) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 11$ power law group size types.	155
14.7	Performance (<i>NMI</i> , <i>NHS</i>) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 17$ power law group size types.	156
14.8	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 4$ equal size group types.	157
14.9	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 7$ equal size group types.	158
14.10	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 11$ equal size group types.	159
14.11	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 17$ equal size group types.	160
14.12	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 7$ power law group size distribution types.	161
14.13	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 11$ power law group size distribution types.	162
14.14	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 17$ power law group size distribution types.	163
14.15	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 4$ equal size.	164
14.16	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ equal size.	165
14.17	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ equal size.	166
14.18	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ equal size.	167

14.19	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ power law group size distribution.	168
14.20	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ power law group size distribution.	169
14.21	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ power law group size distribution.	170
14.22	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 4$ equal size.	171
14.23	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ equal size.	172
14.24	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ equal size.	173
14.25	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ equal size.	174
14.26	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ power law group size distribution.	175
14.27	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ power law group size distribution.	176
14.28	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ power law group size distribution.	177
14.29	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 4$ equal size.	178
14.30	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ equal size.	179
14.31	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ equal size.	180
14.32	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ equal size.	181
14.33	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ power law group size distribution. . . .	182
14.34	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ power law group size distribution. . .	183
14.35	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ power law group size distribution. . .	184

14.36	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 4$ equal size.	185
14.37	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ equal size.	186
14.38	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ equal size.	187
14.39	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ equal size.	188
14.40	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ power law group size distribution.	189
14.41	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ power law group size distribution.	190
14.42	Average performance (<i>NMI</i> , <i>NHS</i> , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ power law group size distribution.	191

List of Abbreviations

Throughout the dissertation we will make use of acronyms, abbreviations and mathematical symbols. Although we have made an effort not to overuse them, it might be likely that they could introduce confusions to the reader. In order to reduce this risk, this preliminary Section aims to introduce and briefly describe the most commonly used ones.

List of Acronyms

- GM: interaction-based group modelling framework;
- CM: cooperation modelling component, that is, the first module of the group modelling framework;
- GID: group identity detection component, that is, the second module of the group modelling framework;
- CL: Chen and Li's experiment simulation, that is, the authentic behavioural data recorded by Chen and Li in their social dilemma experiment on induced group identities [30]. In our empirical evaluations we generated (simulated) the pairings proposer-receiver1-receiver2 needed to extract specific behavioural data;
- CLAS: the Artificial Societies which agent stochastic behaviours are derived by Chen and Li experiment's authentic behavioural data;
- CSD: community structure detection, that is, the task of partitioning a given network into clusters, or communities, or groups;
- SP: spectral partitioning, a well-established algorithm for community structure detection;
- EC: evolutionary computation, a term which encompasses a large family of computational techniques which allow for the seeking of solutions to optimisation problems via stochastic processes;
- EA: evolutionary algorithm — part of a family of algorithms subset of evolutionary computation — which essentially implements the Darwinistic concepts of e.g. natural selection, survival of the fittest, mating and reproduction;
- N-EA/G-EA/E-EA: single population EAs, with *node/group/edge*-based chromosome representation, based on modularity [159] as fitness function. Depending on the empirical analysis, these algorithms will be also referred to as N-EA-Q/G-EA-Q/E-EA-Q;
- N-EA-LR/G-EA-LR/E-EA-LR: single population EAs, with *node/group/edge*-based chromosome representation, based on LinkRank [112] as fitness function;
- N-1N/G-1N/E-1N: EA with *node/group/edge*-based chromosome representation composed of a population of chromosomes which partition a given network into a fixed number of community structures;

- N-UCB1/G-UCB1/E-UCB1: evolutionary algorithms, with node/group/edge-based chromosome representation, which rely on *sealed* niches — i.e. niches which chromosomes of its population are *forced* to partition a given network into an exact number of community structures — and a niching activation mechanism based on the UCB1 algorithm used for the n -armed bandit problem [12]. Depending on the empirical analysis, these algorithms — which are based on modularity as fitness function — will be also referred to as N-UCB1-Q/G-UCB1-Q/E-UCB1-Q;
- N-UCB1-LR/G-UCB1-LR/E-UCB1-LR: evolutionary algorithms, with node/group/edge-based chromosome representation, which rely on sealed niches, a niching activation mechanism based on the UCB1 algorithm used for the n -armed bandit problem, and LinkRank as fitness function;
- N- ϵ GREEDY/G- ϵ GREEDY/E- ϵ GREEDY: evolutionary algorithms, with node/group/edge-based chromosome representation, which rely on sealed niches and a niching activation mechanism based on the ϵ -greedy algorithm used for the n -armed bandit problem [12]. Depending on the empirical analysis, these algorithms — which are based on modularity as fitness function — will be also referred to as N- ϵ GREEDY-Q/G- ϵ GREEDY-Q/E- ϵ GREEDY-Q;
- N- ϵ GREEDY-LR/G- ϵ GREEDY-LR/E- ϵ GREEDY-LR: evolutionary algorithms, with node/group/edge-based chromosome representation, which rely on sealed niches, a niching activation mechanism based on the ϵ -greedy algorithm used for the n -armed bandit problem, and LinkRank as fitness function;
- N-RHEAD/G-RHEAD/E-RHEAD: evolutionary algorithms, with node/group/edge-based chromosome representation, which rely on sealed niches, and a niching activation mechanism — called *reading head* — which selectively evolves, each generation, the most fit niche and predefined number of adjacent ones. Depending on the empirical analysis, these algorithms — which are based on modularity as fitness function — will be also referred to as N-RHEAD-Q/G-RHEAD-Q/E-RHEAD-Q;
- N-RHEAD-LR/G-RHEAD-LR/E-RHEAD-LR: evolutionary algorithms, with node/group/edge-based chromosome representation, which rely on sealed niches, a reading head niching activation mechanism, and LinkRank as fitness function;
- N-RWS/G-RWS/E-RWS: evolutionary algorithms, with node/group/edge-based chromosome representation, which rely on sealed niches, and a niching activation mechanism — called *roulette wheel selection* — which selectively evolves niches proportionally to the overall niching fitness landscape. Depending on the empirical analysis, these algorithms — which are based on modularity as fitness function — will be also referred to as N-RWS-Q/G-RWS-Q/E-RWS-Q;
- N-RWS-LR/G-RWS-LR/E-RWS-LR: evolutionary algorithms, with node/group/edge-based chromosome representation, which rely on sealed niches, a roulette wheel selection niching activation mechanism, and LinkRank as fitness function.

List of Symbols

- avg: sample average;
- sd: sample standard deviation;
- df: degrees of freedom;
- N : a generic network;
- V : the set of nodes or vertices of N ;
- E : the set of links or edges of N ;
- \mathcal{N} : a cooperation network of a given society, under the group modelling perspective;

- A : the adjacency matrix of V ;
- $A_{i,j}$: a generic entry of A , positioned in the i -th row and j -th column;
- \mathcal{S} : when used alone it represents a society of individuals; when used in combination with \mathcal{N} it represent the set of nodes of the cooperation network used by the group modelling framework;
- \mathcal{C} : cooperation network \mathcal{N} 's set of edges, or alternatively their mathematical representation through an *adjacency matrix*;
- $\mathcal{C}_{i,j}$: a generic entry of \mathcal{C} , positioned in the i -th row and j -th column;
- $\{a_1, a_2, \dots, a_n\}$: the n individuals of society \mathcal{S} , and alternatively, the nodes of cooperation matrix \mathcal{N} ;
- $\{v_1, v_2, \dots, v_n\}$: the n nodes/vertices forming V of N ;
- (i, j) : the (un)directed edge of \mathcal{C} (or more generically E) connecting node v_i with node v_j
- $\mathcal{T} = \{g_1, g_2, \dots, g_n\}$: the *true* group identities of society \mathcal{S} /network \mathcal{N} . The g_i values specify the group identity labels of each individual a_i /node v_i ;
- $\mathcal{H} = \{h_1, h_2, \dots, h_n\}$: the inferred group identities of society \mathcal{S} /network \mathcal{N} obtained through the application of the group modelling framework. The h_i values specify the inferred group identity labels of each individual a_i /node v_i ;
- Q : the modularity measure used for community structure detection;
- TGB : Temporal Group-based metrics of fairness of resource allocation [86];
- O_P : the set of resource distributions $\{o_i, o_j, \dots, o_k\}$ made by an individual $a_P \in \mathcal{S}$ within a distribution phase t .

Chapter 1

Introduction

There is a vast corpus of literature investigating social behaviours. Most of them highlight what it appears to be a universal feature of humans, that is the ability to identify themselves as being part of groups — independently on how randomly groups are formed or induced — and behave accordingly, that is by manifesting *in-group* favouritism. Being able to detect group identities is an important task, as it can bring insight on how these affect societies and the environment the latter live within. Moreover, they may also contribute to better understand e.g. the leading causes of the evolution of cooperative interactions and social conflicts. Nevertheless, the detection of *emergent* groups is a non-trivial task. Societies are *complex systems*, meaning that in order to represent their global patterns (groups) it is not possible to simply rely on the sum of their single units (individuals).

In this dissertation we aim to infer group structures, and consequently assign group identities, in *synthetic* environments populated by *socially-driven* individuals, through the analysis of the ongoing interactions. The term synthetic refers to the use of computer-based interfaces and simulation — such as human-based theoretical games and derived artificial societies of *believable* agents, or ultimately cooperative multiplayer games — for the creation of a competitive/cooperative interaction protocols. The term socially driven, hence, refers to the fact that the individuals/agents behave more or less cooperatively (altruistically [205, 32, 127]) depending on who they interact with. Under the assumption that the societies we consider are already composed of groups, we embark the conceptual and empirical investigation of a computational framework which measures the level of cooperation of the ongoing interactions, maintains up-to-date estimates of the society's cooperation network, and ultimately partitions the network into groups, with the objective to maximise *within*-group cooperation levels, and minimise *between*-group cooperation levels.

This Chapter firstly introduces the main concepts of evolution of cooperation (Section 1.1), cooperative computer games (Section 1.2), and complex systems and network theory (Section 1.3), as these areas collectively synthesise the key aim of this dissertation. Section 1.4 presents the research questions and correspondent hypotheses of our investigation, together with the description of the generic computational properties of our framework. The Chapter also strengthens the validity of our work by providing a list of contributions that this thesis resulted to, and outlines the research areas which would benefit from the dissertation (Section 1.5).

1.1 Evolution of Cooperation

One of the most intriguing, complex, yet basic questions related to human society are: when and why did humans start cooperating with each other? When and why did they start becoming altruistic, gathering up, hunting together and sharing food? Such events put the basis of the evolution of many other social phenomena, ranging from communication and languages up to farming, the birth of civilisations and wars.

A possible way to contextualise the evolution of cooperation is by focusing on the analysis of the *interactions*. If we viewed cooperation as an altruistic process [60], that is the degree to which an individual attempts to satisfy others' concerns [205, 32], then we would immediately identify the clash with the Darwin-

Table 1.1: An example of the payoff matrix for the prisoner’s dilemma game. The available actions for the two prisoners are shown in both column and row; the outcomes (i.e. years of imprisonment) are shown in round brackets. The first value is the outcome of the first prisoner, the second value is the outcome of the second prisoner.

		Prisoner A	
		Confess	Deny
Prisoner B	Confess	(1, 1)	(3,0)
	Deny	(0,3)	(2, 2)

istic theories of survival of the fittest, natural selection, and competition [38], even though certain theories of animal behaviours have given potential answers to the inconsistency between natural evolution and altruism, e.g. the selfish gene theory by Dawkins [40].

There exist many approaches to study *evolution of cooperation*, ranging from the more direct anthropological and archaeological studies — which could help with e.g. building a timeline of historical causal events explaining how farming arose from hunting — to those more indirect ones — based on e.g. computer-mediated environments and simulations — which could allow for the study and analysis of isolated facets of interactions. This dissertation deepens its roots in the latter approach.

1.1.1 Game Theory and Evolutionary Dynamics

Although originally designed for modelling economical interactions, game theory has developed into an independent field with solid mathematical foundations and many applications [212]. Game theory studies revolve around the concept of *strategic* action making performed by *rational individuals*. A rational individual (usually called *agent* when it is simulated or artificially implemented) is considered to be an entity which performs actions strategically, that is, in order to maximise his own payoff (goal).

Another fundamental component of game theory studies is the interaction protocol, or simply *game*. Games are usually played among two or more opposing participants and are represented by means of a *payoff matrix*, which defines the (monetary) outcomes of the opponents depending on all the possible combinations of their available actions. The term opponent suggests that the games are designed in a way to instigate a clash between *competitive* behaviours, which generally lead to immediate high and long term low payoffs, and *cooperative* behaviours, which generally lead to the opposite outcomes.

Possibly, the most known abstract/theoretical game is *prisoner’s dilemma*. In this two-opponent game, the participants play the role of prisoners being questioned by the police about a crime they might have committed. Communication between the prisoners is not available. They face two options: either *confess* the commitment of the crime or *deny* it. Depending on their combined actions they will face different amount of years of imprisonment. Table 1.1 depicts the dilemma as a payoff matrix.

The central aim of game theory is the search for the *Nash equilibrium* [155], that is a solution concept in which each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by changing only their own strategy (*stable strategy*) [168]. With respect to the example of Table 1.1, for instance, the Nash equilibrium would correspond, for both opponents, to the *Deny* action, despite the fact that their best outcomes would be achieved when both of them *Confess*.

Game theory has been extensively used to investigate how cooperation might evolve. Evolutionary game theory [193, 164] is an established research field which is centred on the repeated execution of theoretical games on agent populations. The agents are generally implemented by means of very simple behavioural rules of action selection; the outcomes of their strategies will dictate the success or failure of the agent within the population, meaning that they will either replicate or disappear from it, in order to simulate, rather abstractly, the theories of Darwinism [166, 13, 96]. Similarly to game theory, the goal of evolutionary dynamics studies is the search for an *evolutionary stable strategy*, that is the most consolidated and irreplaceable strategy within the population.

Arguably, the studies on evolutionary game theory can be seen as being part of the larger field of evolutionary dynamics which, intuitively, does not necessarily rely on game theory for agent replication and

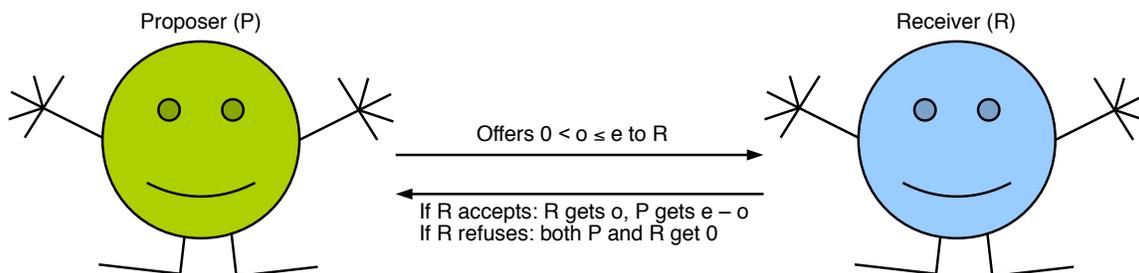


Figure 1.1: The schematic representation of the ultimatum game.

the evolution of behaviours through the population. The use of Computer-based simulation — e.g. Sugarscape [51] and Netlogo [206] among others — have been contributed to the definition of e.g. the balance between predators and preys in ecosystems [218], ethnocentrism [96, 217], and, clearly, cooperation [216].

It is intuitive that the use of evolutionary dynamics for the study of complex phenomena such as the evolution of cooperation has pros and cons. On one side, they can reproduce potentially infinite types of simulations, quickly, and at a very low cost; moreover, their level of abstraction can be customised, so that the emphasis of the generated dynamics could be put on different features. On the other hand, the studies on evolutionary dynamics might suffer from a lack of *realism* of the scenarios under investigation [94], and although their findings can still be interpreted as abstractions of real-life behaviours [165], their *in-vivo* validations are generally missing [94, 211].

1.1.2 Social Dilemmas and Social Preferences

Game theory games are not only studied mathematically by seeking Nash equilibria or synthetically by means of evolutionary dynamics; they are also investigated in controlled laboratory experiments with human participants, and it is in these specific cases that striking findings were obtained.

Consider for instance the bargain or *ultimatum game* depicted in Figure 1.1. In its most common setup, an individual, called the *proposer* P , initially receives an endowment e . P has to decide the amount $0 < o \leq e$ to be given to his opponent individual, called the *receiver* R , and to keep the remaining amount $e - o$ for herself. R has now two options: she can either accept the bargain, in which case she will gain o and P will gain $e - o$, or refuse it, in which case they will both gain zero.

Under the game theoretical assumption of rational individuals, the Nash Equilibrium for P would correspond to offering the smallest possible o , and for R to always accept whichever offer she receives, since it would always be greater than zero. However, when the game is played iteratively among individuals, the offers o are generally deviating from the Nash equilibrium, even significantly [60, 39]. Such counter-intuitive finding, at least under the rational individual assumption made in game theory, led to the birth of a new research field, called *social dilemmas* [60, 39], leaning more towards social and behavioural sciences than, e.g. game theory, mathematics and economics.

Similarly to the approaches of evolutionary dynamics behaviour, researchers investigating social dilemmas started pursuing the definition of behavioural rules which could explain why the recorded behaviours deviate from the Nash equilibria. Since the behaviours were observed especially when the games were iterated, there seemed to be an underlying relationship between profit maximisation, reciprocation of the treatments, and *altruism* or *cooperation*, as opposed to competition or selfishness [205, 32]. In other words, the participants tended to have *social*, rather than selfish *preferences* [60, 39].

One of the most well-studied type of social preference is *inequity aversion* or *fairness* [61, 21, 106]. Its formulation aims to explain the reluctant behaviours that individuals tend to have when they face situations of inequality of treatment with respect to the rest of the society/participants of the experiments. A remarkable finding is that these reluctant behaviours are manifested both under the forms of *envy* with respect to better treatments and *guilt* with respect to worse ones.

Similarly to the studies of evolutionary game theory, the studies on social dilemmas and especially inequity

aversion have taken significant advantages from the use of computer-based simulations by means of *agent-based modelling* [51, 142, 105]. Although disputable, a possible way to differentiate evolutionary dynamics simulations from agent-based modelling ones is to consider the former as means to *explain* real-life phenomena by assuming generic laws, whilst the latter can be interpreted as means to *explore* behavioural rules, and subsequently compare them with similar patterns observed in real-life [219]. Nevertheless, the use of agent-based modelling might still suffer from the same drawbacks that are observed in evolutionary dynamics studies [94].

1.1.3 Social Dilemmas and Group Identities

The formula for inequity aversion is “colour blind”, that is, it assumes that an individual perceives an equal/unequal treatment with respect to the equal/unequal treatment of any individual of the whole population. However, even if its formulation and fine parameter tuning could well explain the behaviours registered in social dilemma experiments with human participants, these are generally based on a rather small population — generally less than a hundred participants. In other words, inequity aversion might still fail to describe the behaviours observed in real-life, larger-scale populations.

As a matter of fact, human history is full of wars and conflicts, indicating that *global* behaviours appear to lean more towards competition than cooperation; competition might even have had a leading role for the emergence of civilisation [210], which might even suggest that cooperation is just a *local* phenomenon. Moreover, even if we just focused on smaller scales, for instance on our everyday life, we will still face flaws on the theories of (colour blind) inequity aversion: no matter which type of interaction we consider, we would almost always observe different qualitative and quantitative manifestations of it, depending on the parts involved.

Imagine for instance the gift exchange rituals we nearly all perform during e.g. religious events or birthdays. We all, more or less implicitly, give a *value* to what we receive or give, or alternatively *compare* and possibly *prefer* it to another one [221]. Possibly, we behave in this way because we also value/compare/prefer some individuals over others. We would not hesitate to book a holiday in an exotic place to celebrate a very special event for person we love, though we would probably think twice, before doing the same, as a birthday gift for our new work colleague whom we barely remember his/her name. Whilst a shoulder pat or ultimately a free coffee at the vending machine would suffice in the latter case, it might create non negligible troubles for the former.

These intuitive behaviours have also been found in social dilemma studies. For instance, Marzo et al. [147] conducted an experiment in which human participants interacted with each other through a variation of the ultimatum game. Two social categories were present, namely friends and strangers; the data gathered showed that the offers made by the proposers were generally higher towards friend than stranger receivers.

The friend vs. stranger behavioural pattern can be observed in many social scenarios, such as work unions vs. managers or politicians, countries against each other, supporters of football teams against each other. We could generalise the concept into a simple dichotomy: people who are more preferred, against people who are preferred less. In other words, people tend to form *groups*, identify themselves as being part of a group (*group identity*), and behave accordingly (*group behaviour*). In fact, individuals belonging to the same group (*in-group*) tend to cooperate more with each other than when they interact with people belonging to other groups (*out-group*). Furthermore, people tend to manifest group behaviours no matter how randomly these groups are formed or induced [39, 30, 18, 203] and *in-group* favouritism appears to be culturally independent [16].

1.2 Collaborative Computer Games and Crowdsourcing

One possible way to overcome, or at least reduce, the drawbacks arising from the use of evolutionary dynamics and/or agent-based modelling simulations [94], is the use of multiplayer computer games. Multiplayer computer games have the advantage to both implement more immersive scenarios than e.g. the payoff matrices

or the general text-based interfaces used in social dilemma experiments [30], and to aim for the generation of complex cooperative patterns [213]. Furthermore, we assist to the growing attention put upon the creation of cooperative games *per se* [185, 188, 213], the release of Massively Multiplayer Online Games (MMOGs), such as World of Warcraft (Blizzard Entertainment, 2004), Glitch (Tiny Speck, 2011), and the consequent growing attention given through the analysis of player behavioural data of those games [46, 47, 200, 201, 126]. On the same basis we also observe the rising consideration academia and industry put upon serious and educational games, especially those aiming at teaching non-soft skills, such as social conflict resolution [225, 32].

Another example of the potential of cooperative games come from the Foldit project [109], in which players are challenged with the task of protein folding. Foldit is also a solid example of collective collaborative intelligence implemented through crowdsourcing. Moreover, we remark the use of crowdsourcing as a powerful tool to gather generic perception preferences [222] over qualitative features, such as fairness of treatment [86], social conflicts [31], and emotions [223].

Although through cooperative computer games the level of realism of the scenarios under investigation might increase, the *emergent* dynamics might not always be the envisaged ones. For instance, some game mechanics might lead to having players occasionally teaming up in order to reach a common goal, though hardly ever cooperating with each other [47]. Moreover, the increase of agency given to the players might lead to the generation of highly complex dynamics [154], which understanding can be attempted by means of complex tools, such as *complex networks*.

1.3 Complex Systems and Complex Networks

The term *complex system* is generally used to refer to a class of constructs (systems) composed of many parts (units) interconnected with each other. The interconnections regulate dependencies among the units, which interact with each other and, depending on the complex system type, would also adapt to each other (self-organisation). The repeated interactions of its units usually lead to the formation of global patterns which cannot be explained by the mere sum of each single unit's characteristics (non-linearity). Examples of complex systems are the set of biological cells forming human bodies, brain neurones which lead to thoughts and emotions, and individuals, which lead to societies, culture, and social conflicts.

The most central computational/conceptual tool for the analysis and understanding of complex systems is the *network*. Graphically, a network is composed of a set of *nodes (vertices)*, which are used to represent e.g. the units of a complex system, and a set of *links (edges)*, which are used to depict the dependencies and interconnections between nodes. Networks can also be represented mathematically, through a so-called *adjacency* matrix. The analytical study of networks derives from graph theory, in which graphs are usually mere mathematical constructs. On the other hand, the term *complex networks*, especially used when it is related to complex systems, is used to imply the existence of non-trivial, non-random, *complex* patterns within their topology.

In the recent years, and especially the last decade, we have witnessed the establishment of a new research field, called *network theory* or *network science* [14]. In a nutshell, network science is an interdisciplinary research area which main aim is the analysis and understanding of complex networks, which are either directly built from — or aim to replicate/generate — real world data and phenomena. As a matter of fact, network science aggregates many disparate research fields, ranging e.g. from physics to social sciences, from biology to mathematics, from computer science to psychology. Complex systems and network science have been gaining more and more attention not only by research institutions and academic societies — for instance the Santa Fe Institute¹ or the Complex Systems Society² — but also in the media — see for instance ABC's documentary "How Kevin Bacon Cured Cancer"³.

A fundamental reason for the popularity of complex systems and complex networks is the continuous increase of pervasiveness of technology — both *hard* e.g. smart phones, and *soft* e.g. the Web 2.0 — which allows the collection of huge, previously unthinkable volumes of data and the generation of new forms of

¹<http://www.santafe.edu>

²<http://www.complexsociety.eu/>

³<http://www.abc.net.au/tv/documentaries/interactive/futuremakers/ep4/>

interactions. For instance, the Internet allows us to break the spatial boundaries and connect to each other independently of where we are located in the world. Its most popular platforms, e.g. *Facebook*, *Twitter*, *Second Life* or *Minecraft*, made social interactions easier, yet global patterns became more complex at the same time, see for instance the mobilisation of people forming the *Occupy* movement or the “*Arab Springs*” uprising events. Terms such as “social network” or “six degrees of separation” are less mysterious to non-experts. Moreover, through network science, it has been possible to *predict* the N1H1 pandemic spread and capture Saddam Hussein [14]. The studies of complex networks allowed to better understand the “small world effect” [208], the related scale-free property [15], and power law distributions [35] were better contextualised. Arguably, network science is providing a major *quantitative* contribution to research fields which traditionally relied upon *qualitative* approaches of investigation, such as behavioural and social sciences. Through the analysis of e.g. mobile phone geo-location it is possible to infer social networks of group evolution [170, 48].

Lately we have even observed an increasing trend on the analysis of time-dependent *dynamic* networks — i.e. networks in which links and/or nodes are added/removed/modified through time — and *multiplex* networks — i.e. networks with more than one type of edges, and/or possibly node attributes [154]. One of the most studied property of complex networks is their modularised structures into groups or clusters, which are usually referred to, in network science, as *community structures*. Although there is yet a lack of unanimous consensus on the definition of what it is, a community structure is generally considered to be a subset of nodes which are more similar to each other than the rest of the nodes in the network. Depending on the complex system under investigation, one might aim to partition the network into *disjoint* sets — which arguably constitutes the most widely used approach to *community structure detection* — or *overlapping* sets [171]. The task of community structure detection is known to be computationally difficult (NP-complete) [128], which means that the optimal solutions can only be approximated by heuristics.

1.4 Motivation and Aim of this Thesis

The seminal work revolving around the evolution of cooperation that we encountered led us to believe that group identities and behaviours, which appear to be cross-culture properties of individuals [16], play a key role within the evolution of societies. Moreover, we believe that e.g. cooperation and reciprocity are a form of manifestation of the existence of group structures, rather than being self-standing phenomena. Another way to formulate our belief is to consider societies as complex systems, in which the interactions can represent the links between individuals, and in which the group structures correspond to the global patterns. Furthermore, the dynamics occurring within the system lead to phenomena such as group evolution, e.g. growth, merge or death [170], social conflicts and wars [32, 210]. Cooperation can then be considered as a feature of the interactions; if properly analysed, for instance by assuming the existence of relevant reference individuals [86], it can lead to the inference of the system’s global patterns and complex dynamics [86, 85]. Possibly, through this interpretation, it might be possible to further link notions such as inequity aversion and group identities, for instance by finding *critical mass* limits beyond which “colour blind” altruism evolves into *in-group* favouritism.

Driven by this belief, we decided to embark the research for the definition of a computational framework capable of inferring group structures, and assign group identities, to populations of non-merely-greedy individuals, by solely analysing the levels of cooperation of the interactions. Our interaction-based group modelling framework was designed with a number of fundamental modelling properties. We consider the following properties due to our intention to apply the group modelling framework to cooperative computer games [213, 185, 188].

- **online modelling:** the framework must be capable of inferring the existing group structures in such a way that it would be possible to relate group-based events with the observed interactions. This property becomes more important not only in case of dynamic and *emergent* groups [213, 170], but also in case the underlying interaction protocols or environments are dynamic [32, 225, 224];
- **indirect modelling:** the framework should be able to infer the existing group structures without *directly* questioning each single individual about her group identity. The reasons are numerous. For

instance, the direct approach might influence their behaviours by either smoothing out or increasing their *in-group* favouritism. In presence of highly interacting cooperative computer games, an excessive querying might affect the player’s immersion/attention/frustration, with potential dramatic outcomes in terms of in-game behaviours. The group modelling framework could be used to infer group structures of past experiments, as a form of *delayed online modelling*⁴. Moreover, these past experiments might had been designed and implemented for different purposes, meaning that the information required by the *direct* approach might be missing. Finally, the group modelling framework could be applied as a modelling tool on agent-based modelling simulations aimed to investigate generative group behaviours; no matter how *realistic* an agent is, the querying of its group identity appears to be meaningless. Nevertheless, we do not exclude that, if properly designed, some forms of *direct approaches* might further empower our group modelling framework, as we will discuss in Chapter 10;

- **unsupervised modelling:** the framework cannot in principle make any assumptions on the number of groups and their size. This property becomes fundamental particularly when the societies and their underlying group structures are dynamic.

Although the group modelling framework is designed to be applied to cooperative computer games, for this current study we had to rely on the social dilemma experiment originally conducted by Chen and Li [30], the reasons why can be found in Chapters 4 and 10. Nevertheless, we claim that the properties of our group modelling framework make it generic and robust, which means that it could potentially be applied to both social dilemmas, as it is the case of this dissertation [30], cooperative computer games [32, 107, 225], and potentially even to massively multiplayer online games [47, 201, 126]. Moreover, the framework can be applied to both human-based experiments, agent-based simulations, and potentially even to evolutionary dynamics testbeds. Consequently, when the disambiguation between these fields of application is not required (or unimportant for the discussion being), we will refer to them with the term *social synthetic environments*. The field of application of our research will be initially based on resource exchange/allocation within social synthetic environments, being easier to quantify as opposed to, for instance, language-based interactions we can find in website forums or online chats.

We also want to stress that the main emphasis of our research is given on the semantic understanding and syntactic definition of the most suitable computational techniques that could lead to the most reliable implementations of our group modelling framework. This means, in other words, that the fine parameter tuning task, which was partly conducted during earlier investigations, will only have a secondary role in this dissertation.

1.4.1 Hypothesis and Main Research Questions

We formalise the aims of this dissertation as the following hypothesis:

By measuring the level of cooperation of the ongoing interactions among the individuals of a social synthetic environment, we will be able to detect *in-group* and *out-group* relationships, subsequently unveil the existing (cooperative) group structures, and hence detect the individual’s group identities.

The main research questions and the corresponding methodological approaches taken to address them, are presented below.

How can we measure the cooperation of the interactions?

We base our approach on the following definition of cooperation or altruism: the degree according to which a person aims to satisfy the others’ concerns [205, 32]. To tackle this question we will rely on what we call the *one-to-many* perspective [86]. Essentially, we aim to observe how *one* individual behaves towards *many*

⁴We decided to use this somewhat bulky term in order not to introduce ambiguities with the online modelling property previously described.

other individuals, providing that these interactions are consistent with each other, though they might not occur at the same time.

Through the *one-to-many* approach it is then possible to detect the presence of favouritism of treatments, or alternatively *unfair* behaviours [86]. This favouritism has clearly a correspondence with the definition of cooperation we decided to consider. At this point, the quantification of the treatments can occur, with the ultimate outcome to be able to interpret them as either being *in-group* or *out-group* types.

How do we extend the information held in local interactions to the global perspective of group structures?

By assuming that the social synthetic environment under investigation is a complex system, whose units represent the society’s individuals, we consequently obtain that the global patterns represent the society’s group structures and identities.

Hence, we will aim to build the society’s underlying complex network, in which the nodes represent the individuals of the society, and the links represent, quantitatively, the levels of cooperation of the interactions among the individuals.

How can we detect the groups and assign group identities?

We know that one manifestation of group identities is the fact that individuals belonging to the same group (*in-group*) tend to cooperate with each other more than when they interact with individuals belonging to other groups (*out-group*) [39].

Thanks to the representation of the social interactions through a complex network, and their quantitative representation of cooperation, we can now relate the concept of group identity detection to the community structure detection task extensively studied in network theory [77, 14].

How do we know that the approach taken would unveil the *true* groups?

This research question is highly relevant, given there is still not an unanimous consensus of what a community structure means, and that the notion of group can be explained through many disparate research fields, such as philosophy and psychology. Thankfully, however, we can generalise the investigation by leveraging on the fact that individuals manifest group identities no matter how randomly the groups are induced [202].

Therefore, we will answer this question empirically, by considering experiments in which the (induced) *true* group identities are known, and by comparing them with the inferential results of our group modelling framework.

1.4.2 Overview of the Framework, Challenges and Approaches

The hypothesis made in this dissertation, and the subsequent strategies we set in order to answer the research questions, need now to be formally expressed by computational means. Figure 1.2 depicts the design of our proposed group modelling framework. The framework is organised in two pipelined modules: the Cooperation Modelling (CM) component — which quantifies the levels of cooperation of the interactions and maintains up-to-date estimates of the society’s complex (cooperation) network — and the Group Identity Detection (GID) component — which performs community structure detection on the society’s complex network in order to retrieve the group identities of the individuals of the social synthetic environment. The first two research questions are mainly tackled by the CM module, whilst the third one is mainly tackled by the GID module. Intuitively, the fourth and last research question is tackled by the analysis of the whole framework — combination of the two modules.

The *one-to-many* approach for the evaluation of the levels of cooperation of the interactions corresponds to the interface between the group modelling framework and the social synthetic environments, and is performed by the CM module. Essentially, the aim of the *cooperation evaluation* task is to aggregate the interactions, compare them with each other, and output the comparison on a normalised scale of cooperation, where value *zero* corresponds to absence of cooperation, and value *one* corresponds to full cooperation. For the

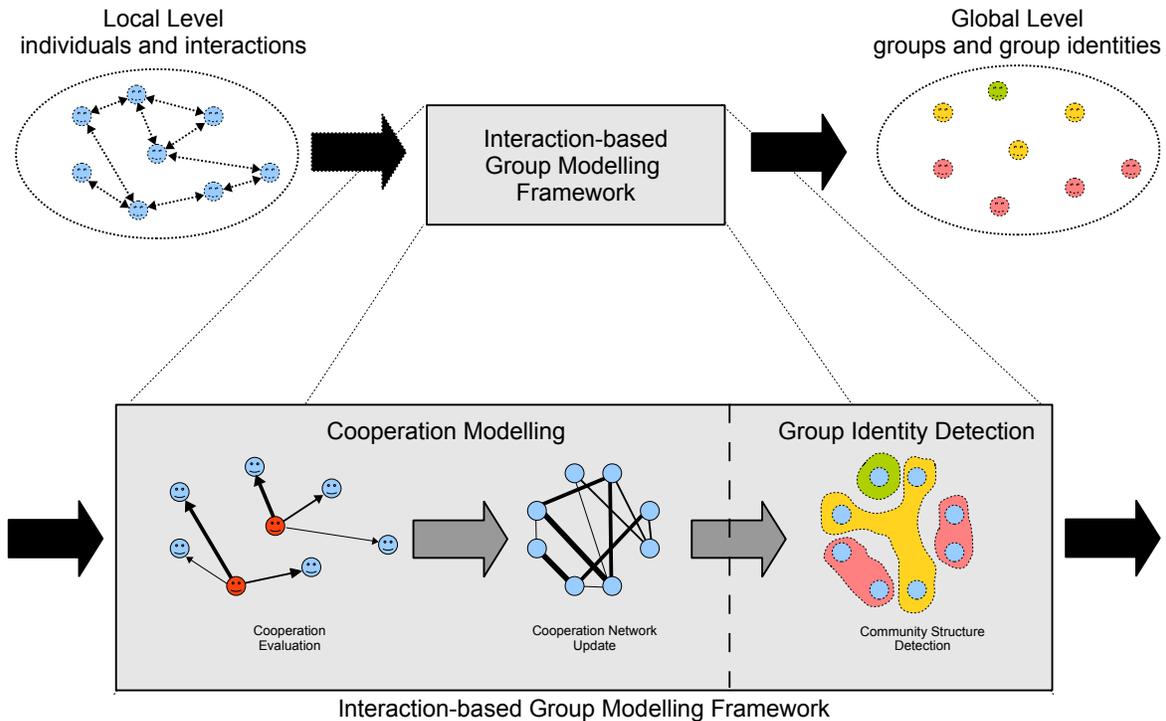


Figure 1.2: The schematic representation of our interaction-based group modelling framework.

cooperation evaluation task we will consider different temporal window sizes regulating the gathering of interactions, together with two possible approaches to interpret the interactions on a normalised scale.

The subsequent task, called *cooperation network update*, is intuitively in charge of using the just evaluated interactions in order to maintain up-to-date estimates of the society’s underlying complex network of cooperation. This cooperation network will have its nodes corresponding to the individuals of the society under investigation, and its links representing the ongoing cooperation values among individuals. The links will be both *directed* — i.e. they will represent the proposer-receiver roles of the interactions — and *weighted* — i.e. they will hold the ongoing cooperation values. For the cooperation network update task we will consider both straightforward update rules, that is sample average, and other computational methods, such as those used in reinforcement learning [199].

The validity of the CM module and its configurations, will be investigated separately from the subsequent GID module. This will allow us to better understand the robustness of the module against its main challenges, which are mainly represented by its effectiveness to highlight *in-group* vs. *out-group* boundaries, to successfully deal with *ambiguous* interaction behaviours, such as those occurring among individuals who all belong to different groups, and to provide accurate cooperation evaluations even in presence of different magnitudes of the interactions.

The cooperation network will evolve in terms of its edge weights, which are subsequently used to detect the *in-group* vs. *out-group* boundaries. The network’s initial condition — which we decided to set in order to represent the highest level of uncertainty — will assume equal cooperation levels among all the individuals of the society (i.e. *colour blindness*). Throughout the observation of the interactions, then, the CM module will refine the edge weights. Ultimately, in presence of perfect representation, each *true* group would be represented as a completely separated subnetwork — all *between-group* edge weights equal to zero, i.e. absence of cooperation — with full *within-group* connectivity — all nodes of the groups are connected with each other by an edge of weight equal to one. It is worth noting that the dynamic evolution of the cooperation network is endemic to our interaction-based approach to group modelling, even in case the underlying social

synthetic environment has *static* groups — i.e. they do not change throughout the execution of the whole modelling framework.

Intuitively, this characteristic feature of the CM module constitutes the main challenge of the GID framework. In fact, its implementation must be able to be generic across the different network typologies encountered, and also to be able to provide accurate inferences of the underlying group structures, even in presence of a high degree of uncertainty of the ongoing cooperation levels.

This thesis will consider evolutionary computation as the algorithmic approach for the community structure detection task. It is worth noting that evolutionary computation has already been provided extremely promising results in terms of community structure detection in sparse undirected and unweighted networks [174, 177, 204], and also in fully connected, directed and weighted networks [88]. The novelty of this thesis comes from the investigation of several different evolutionary algorithm setups for community structure detection of dynamic networks in terms of typology and topology. Moreover, the empirical investigations conducted aim to better understand what the best evolutionary approaches might be for the task at hand. For instance, what the impact of the use of specific chromosome representations is, and which process should we follow, in order to limit the huge search space in large networks. In order to have a clear insight, we will consider different group structure configurations across the social synthetic environments considered.

1.4.3 Case Studies and Main Findings

Despite the fact our group modelling framework and related properties were defined with the main aim to be applied to cooperative computer games, the empirical evaluation of its ability to unveil existing *true* group structures was conducted, in this dissertation, on the existing social dilemma experiment by Chen and Li, which had the main goal to understand group behaviours and identities in induced groups [30]. The reasons why the study reported in this dissertation could not be based on cooperative computer games are given in Chapter 4 and 10.

We conducted three main investigations, one based on the actual authentic human reported behavioural data (CL), for a total of 25 independent experimental runs, and two based on artificial societies of stochastic agents, which behaviours were derived from the authentic behavioural data (CLAS), for a total of 270 independent experimental runs. The CLAS-based investigations were focused on understanding the effectiveness of the CM and GID modules separately, whilst the CL-based investigation was used as a form of *validation* [94], of our whole framework, to model group structures of human-based societies.

The overall results gathered showed extremely promising effectiveness of our interaction-based approach across society partitions, noise in the interactions, and dynamic changes of the estimated cooperation networks. Moreover, among the evolutionary algorithms considered, we observed the unique evolutionary trend achieved by the somewhat simplest chromosome representation (namely N-EA), which evolves candidate solutions by making each of its gene correspond to a node of the network, and its allele values corresponding to a community structure identity label. N-EA generates a peculiar aggregative evolutionary search, that is, it generally starts by assuming a high number of groups, and then refines its solutions towards the *true* number of groups. This trend was not observed in any other evolutionary algorithms types examined, which instead rather quickly converged to a number of groups generically lower than the *true* ones. Although N-EA leads to the best overall performance registered, its aggregative approach manifests a generally slow convergence to high inference scores. We investigated the possibilities to improve its speed by means of niching approaches, in accordance with the promising results we gathered during our early work [88]. However, although in some cases we manage to accelerate the convergence to high inference scores, the niching approach failed to overall outperform N-EA.

The evolutionary experiments were put in comparison with a deterministic optimisation algorithm for community structure detection, that is spectral partitioning [160], which we consider to be a reliable performance benchmark algorithm, due to its deterministic nature, and especially because it has already been applied for community structure detection to a vast plethora of network problems [144, 59].

However, the GID module had to rely on cooperation networks estimated by the CM module. The investigation on the latter showed the clear advantage that the evaluation of cooperation of interactions in the forms of discretised *in/out-group* labels/values has with respect to the clear identification of *in-group*

vs. *out-group* boundaries. Moreover, we discovered that the best performance is achieved when the so-called constant- α rule is used for cooperation network update. This update rule has been used in many computational intelligence methods, among which we remark reinforcement learning [199], artificial neural network’s perception training [95], and artificial ant colony pheromone update [42].

1.5 Summary of Contributions

To the best of our knowledge, there has not been any other attempt to detect group structures, and assign group identities, by solely analysing the levels of cooperation of the interactions occurring among individuals of social synthetic environments.

Furthermore, the empirical research here conducted applies, for the first time, the interaction-based group modelling framework to authentic recorded empirical data of human behaviours based on a social dilemma interaction protocol. This constitutes a rather solid basis upon which the application of the group modelling framework to more complex social synthetic environments, such as cooperative computer games, can be attempted.

In addition, the investigations here conducted contribute to the field of network theory, especially for what it regards the community structure detection task and its related evolutionary community structure detection approach. We further consolidate the potential use of evolutionary computation beyond sparse undirected and unweighted networks, by showing the extremely good performance of e.g. N-EA across different societal scenarios and in presence of a highly uncertain, directed, weighted, and potentially fully connected cooperation networks.

Though only marginally, we also contribute to the discussion on what a community structure means, and what the best approaches for its detection might be. More specifically, we further strengthen the effectiveness of community structure detection via *modularity maximisation* [160, 158, 161, 128].

Other contributions regard the impact that the scenarios under investigation, and more specifically the number of groups and their size distribution have with respect to the *difficulty* to detect the *true* group structures [124]. Similarly, we also raise discussions around the use of performance measures for the evaluation of the closeness between inferred and *true* community structures.

Beyond the community structure detection task, the group modelling framework confirmed the soundness of the *one-to-many* approach for the understanding of *in-group* favouritism and *unfair* treatments [86]. Moreover, the interfacing of the group modelling framework with the social synthetic environment considered in this dissertation was achieved by applying the so-called *temporal group-based* fairness metric (*TGB*) [86]. *TGB* is possibly the first group-based metric of fairness which has also been validated against human perceptions of fairness treatment in a computer-game-based crowdsourcing experiment. This constitutes yet another solid basis for the application of our interaction-based approach and modelling framework to higher interacting environments such as cooperative computer games [32].

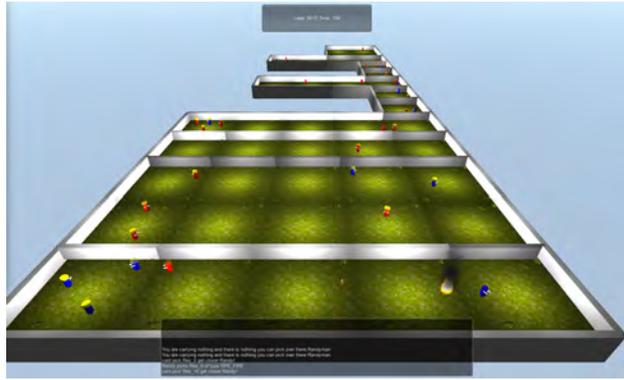
1.5.1 List of Papers

This thesis is the outcome of slightly more than three years of work focused on many disparate, yet interconnected fields. We here present the related list of publications and the contributions they gave to the dissertation. Published papers are presented according to their shared grounds, rather than their date of publication.

- **Towards Detecting Group Identities in Complex Artificial Societies [91]**. This work represents our very first attempt to prototype the group modelling framework. We based our investigation on artificial societies of *believable* agents interacting with each others by means of the ultimatum game. The main emphasis of the work was given on the *implicit* methods the cooperation evaluation task performed by the CM module. The agent dynamics and their *in/out-group* behaviours were based on the notion of friend, stranger and acquaintance which were inspired by the work of Marzo et al. [147] and Chang et al. [25].



(a) Fairness Game - in-game snapshot



(b) Fairness Game - level adaptation

Figure 1.3: Snapshots of the fairness of resource allocation adaptive game [85]. Figure 1.3(a) represents an in-game snapshot, whilst Figure 1.3(b) shows the adaptive levels generated via the experience-driven procedural content generation paradigm [224] throughout the whole game. The farthestmost level corresponds to the first one; according to the player behaviour different level sizes, number of receiver agents and resources to be allocated were procedurally generated.

- **Artificial Evolution for the Detection of Group Identities in Complex Artificial Societies [83].** This work extends [91] by assuming scale-free complex friendship networks influencing the behaviour of the agents. The main emphasis was here given to the use of evolutionary computation for the GID module.
- **Using Reinforcement Learning and Artificial Evolution for the Detection of Group Identities in Complex Adaptive Artificial Societies [89] and Interaction-based Group Identity Detection via Reinforcement Learning and Artificial Evolution [87].** These two studies extend [83] by considering dynamic societies and the group formation global pattern. The main emphasis was given to the cooperation network update task performed by the CM module.
- **Shifting Niches for Community Structure Detection [88].** This work takes a step aside with respect to the group modelling problem and rather focuses on the community structure detection problem. In here we investigated and compared different evolutionary algorithms for the detection of pre-existing communities built upon directed, weighted, and fully connected networks. We also investigated the impact that the conversion of directed and weighted networks into sparse undirected and unweighted ones might have on the overall inferential performance.
- **Towards Player Adaptivity in a Serious Game for Conflict Resolution [85].** This work was focused on the creation of an adaptive game, initially single player, which would allow the player to explore a plethora of resource allocation strategies in a computer-based environment. The game mechanics were essentially those of the *dictator* [20] or the *other-other* [30] social dilemma games, though the receiver agents were depicted as clearly belonging to two group structures, based on their body colour. The game was implemented following the principles of the experience-driven procedural content generation paradigm [224]. The experience-driven adaptation mechanism aimed to *influence* the player behaviour towards a *fair* resource distribution among the two groups of agents [86]. A snapshot of the adaptive game can be seen in Figure 1.3.
- **Towards Multimodal Player Adaptivity in a Serious Game for Fair Resource Distribution [92].** This work directly followed the previous one described by exploring the possibility to add multimodal adaptivity to the experience-driven procedural content generation framework⁵. More specif-

⁵A promotional video of the work can be found here: <http://www.youtube.com/watch?v=4EYvawpCC88>

ically, we aimed to augment the player experience modelling component by considering the player’s level of attention/distraction inferred from the his/her head poses and eye gazes tracked through webcam [11].

- **Validating Generic Metrics of Fairness in Game-based Resource Allocation Scenarios with Crowdsourced Annotations [86].** The (uni-modal) resource distribution game [85] relies on an equation of fairness of treatment which takes into account the existence of group identities among the receiver agents (i.e. *TGB*). Motivated by the promising results obtained in the game’s pilot test, we embarked the investigation aimed to understand whether the fairness formula does indeed correlate to the perception of fairness of the players. The research considered also four other ad-hoc measures of equality and the use of support vector machines to represent fairness. The metrics were validated against annotations retrieved by means of a crowdsourcing experiment. Moreover, *TGB* appears to be the metric which best correlates with the reported perceptions of fairness of treatment.
- **A Computational Approach Towards Conflict and Conflict Resolution for Serious Games [32].** This work puts the basis on the use of collaborative games for serious applications, that is conflict resolution teaching, and mostly contributes, to this thesis, by the notion we give to cooperation, competition, and how we intend to measure them among the interactions of our social synthetic environments.
- **Modelling Global Pattern Formations for Collaborative Learning Environments [84].** This work can be seen as the merging point between serious gaming via collaborative games and the need to model group identities for the better understanding of evolution of cooperation and social conflicts.
- **Towards Validating Game Scenarios for Teaching Conflict Resolution [31].** This work can be seen as the union of [86] and [32]. The crowdsourcing experiment was based on *Village Voices* [107], a computer-based implementation of the ultimatum game, which is also derived from our initial studies done in [85].
- **Controllable Procedural Map Generation via Multiobjective Evolution [207].** This work, finally, applied the same crowdsourcing principles to understand the power of procedural content generation in the Starcraft real time strategy game (Blizzard Entertainment, 1998).

1.6 Outline of this Dissertation

The remaining Chapters of the dissertation are organised as follows. The background notions aimed to better situate and understand our study are presented in Chapter 2 — where we present the most relevant related work — and Chapter 3 — where we bring the most fundamental computational concepts of e.g. network theory, community structure detection, evolutionary computation and so on. Chapter 4 provides a thorough description of the experiment conducted by Chen and Li [30], from which we extracted the behavioural data, and also implemented derived artificial societies (Section 4.3), for the empirical evaluations of our group modelling framework.

Chapter 5 brings a detailed description of the cooperation modelling component, together with the approaches for the cooperation evaluation task, and cooperation network update task, which were considered in this dissertation. Moreover, Chapter 6 will present a thorough empirical investigation of the CM module, and its possible configurations, based on the artificial societies of Section 4.3.

The second component of the group modelling framework, that is the GID module, is presented in Chapter 7, in which we will first provide the motivations why we decided to rely on evolutionary computation for the community structure detection. Subsequently, we will delve into the details of three possible chromosome representations, and further describe a possible evolutionary approach through the use of niching. Similarly to what has been done for the CM module, the subsequent Chapter 8 will present an extensive empirical analysis of the possible GID configurations previously described, based on seven artificial society scenarios of Section 4.3.

Chapter 9 will present the empirical application of the group modelling framework — composed by the best performing setups highlighted in the previous Chapters 6 and 8 — this time no longer based on the

derived artificial societies, rather on the authentic recorded behavioural data of the experiment by Chen and Li. As it can be understood, we gave a secondary importance to this investigation than those based on the artificial societies. The reason is dual: (1) the behavioural data is objectively smaller and simpler than those simulated via artificial societies; (2) we wanted to follow the simulation-validation pattern [94], in order to give a particular emphasis on the “final product” obtained from the best group modelling framework configuration.

Chapter 10 will bring discussions arising after the empirical evaluations. It will present weaknesses and benefits of our approach, limitations and improvements in terms of computational techniques, and highlight possible future work along which the research could continue. Finally, Chapter 11 wraps up the main conclusions and summarise the whole dissertation.

Further integrative material can be found in the Appendices. More specifically, Appendix 12 will list the authentic recorded behaviours of the experiment by Chen and Li. Appendix 13 will extend Chapters 5 and 6 by providing mathematical equation steps which were omitted in order not to overcomplicate the discourse on cooperation modelling. Appendix 14 integrates Chapter 8 by presenting all the secondary results of the evolutionary community structure detection task.

1.7 Summary

This Chapter introduced the dissertation by firstly describing the main research topics revolving around the group identity modelling problem. Then, it presented our research hypothesis, and related research questions, which has led us to the definition of a computational framework capable of inferring the presence of group structures by measuring the levels of cooperation of the ongoing interactions occurring among socially-driven individuals. The Chapter then presented the key contributions of this dissertation, plus a list of previous publications which led to the definition of this work.

In the next Chapter we will provide an overview of the most relevant work related to the study conducted in this dissertation.

Chapter 2

Related Work

This Chapter surveys the state of the art of research areas that we believe are related to the main aims of this dissertation. In particular, we provide background information on evolution of cooperation, under the main perspective of human-based experiments (Section 2.1), synthetic studies via agent-based modelling and evolutionary dynamics (Section 2.2), whilst the main studies on group identities will be presented separately in Section 2.3. The Chapter provides also an overview of the most important and relevant studies on community structure detection (Section 2.4), with particular dedication on the use of evolutionary computation for community structure detection (Section 2.5). Finally, other related work which could not be directly classified in the previous Sections are reported in Section 2.6.

2.1 Social Dilemmas and Cooperation

Social dilemmas [39, 134, 60, 115] are essentially experimental studies on abstract/theoretical games — e.g. *prisoner’s dilemma*, the *ultimatum game*, the *dictator game*, and the *public good game* — in which the involved participants face a decision making problem. In a typical game theory setting, the participants have a goal of profit maximisation; this is pursued by performing actions, which affect one or many other participants (opponents). The actions may be numerous, though they usually fall into two main categories: *competitive* actions, in which the immediate return (payoff) is generally high for the individual who performed it (proposer) and generally low for the affected individuals (receivers), or *cooperative* actions, in which the immediate payoff for the proposer is generally lower, whilst the payoff of the receivers is generally higher.

In game theory the assumption made is that all participants are *selfish*, or *greedy*, or more simply *rational*, meaning that they would tend to maximise their own payoff only. These studies generally have a mathematical connotation, and aim to find a mathematical function, or a *strategy*, which allows for the generation of an optimal sequence of actions. A strategy that forbids the exploration of other actions, which would ultimately lead to lower payoffs, is called *stable* strategy or Nash equilibrium [155, 168].

Unarguably, social dilemma studies are a consequence of the application of abstract theoretical games to human participants, which highlighted how the actions taken (behaviours) were significantly deviating from the Nash equilibria [20, 61, 67, 106]. This deviation in behaviour towards others is usually referred to as *social preference* [60].

Rabin [182] is acknowledged as one of the main researchers investigating social preferences. In his work he mainly focuses around the prisoner’s dilemma game and some of its variations, with the aim to define, mathematically, the altruistic behaviours observed. His study in [182] introduces the first widely acknowledged formalisation of the term *reciprocal fairness*, which closely links to the notion of collaboration [205, 127, 32]. A reciprocal individual responds kindly to actions that are perceived to be kind, and in a hostile manner to actions that are perceived to be hostile [60]. Other similar work aimed to the formalisation of reciprocal fairness, equity theory and reciprocity was performed by e.g. Leventhal [131], Levine [132], Falk and Fishbacher [54], and Charness and Rabin [26].

A more popular type of social preference is *inequity aversion* [61, 21]. In their work, Fehr and Schmidt [61] assume that inequity averse persons want to achieve an equitable distribution of material resources [60]. In other words, the individuals are altruistic towards other individuals, providing that the others' material payoffs are below an equitable benchmark (*guilt*), though they feel *envy* if the payoffs of the others exceed the equitable level [60].

2.2 Agent-based Modelling and Evolutionary Dynamics

Social dilemma studies have benefitted not only from experiments based on human participants, but also from agent-based modelling simulations [51, 142, 105]. Some of the main advantages of the use of agent-based modelling is the fact that their level of *believability* can be adjusted, so that particular aspects of social interactions and leading phenomena can be investigated in a more controlled manner, with more scaling capabilities, and in a quicker way, though at a cost of loss of *realism* [94]. There exist two main approaches to implement the behaviours of the artificial agents: either by using established functions (model-based approach), e.g. inequity aversion, or alternatively by retrieving behaviours, e.g. by means of machine learning techniques, from preexisting human-based experiments (data-driven approach).

Among social preference studies, the ultimatum game (see Figure 1.1) is possibly the most explored. For instance, De Jong et al. [41] used adaptive artificial agents which implement inequity aversion. Similar work was conducted by Xianyu [219], who also regulated the interactions through an underlying communication network. More emphasis on the network structure, as one of the leading causes for the evolution of reciprocal altruism was made by Shutters [192]. In his work, the agents have a fixed policy, though they evolve and replace less fit agents in the society, so that the *emergence* of specific strategies can occur.

Similarly to Shutters, Chang et al. [25] generated a set of agents with different behaviours, three static and one adaptive. They let the agents interact with each other in what they called the social ultimatum game [110, 24], a variant of the ultimatum game in which each agent can choose the opponent to play with. Though their study was not aimed to the direct implementation of agent-based simulations, similarly to Chang et al., Gal et al. [74] used a machine learning approach to mine player behaviours recorded through experiments on *Colored Trails* [73], a computer game with similar mechanics to the ultimatum game.

A modified version of the dictator game called trust game was investigated by Figueiredo et al. [63]. In their work, which closely resembles De Jong's work in [41], agents are implemented with trust and reputation equations and are let to interact with each other, with the coupling being generated by the agent themselves. Similar work was performed by Sutcliffe and Wang [197] though their interaction protocol is not based on an underlying network.

As opposed to the studies of social dilemmas via agent-based modelling, in which the game theoretical aspect is focused on understanding why individuals are being reciprocally altruistic, the research usually referred to as evolutionary dynamics [102] aims to investigate how generally simple behavioural rules spread through e.g. an underlying communication network, a lattice, and so on. Arguably, evolutionary dynamics simulations are generally more abstract than those obtained via agent-based modelling in social dilemmas, though it is not uncommon to find difficulties with the categorisation of one experiment in either of the two types.

Among all, we cannot mention the work by Axelrod and Hamilton's on evolution of cooperation [13] and ethnocentrism [96], which unarguably constitute the milestones of the research on evolutionary dynamics. Nowak is another key contributor to the research on evolutionary dynamics: his most recognised work is probably the investigation of the popular "five rules" for the evolution of cooperation [165, 166]. In those studies, five rules, namely kin selection, direct reciprocity, indirect reciprocity, network reciprocity, and group selection, are studied in order to understand how they affect evolution of strategies of agents playing the prisoner's dilemma.

The work on evolutionary dynamics is solely tangential to the research we present in this dissertation; however, the above mentioned studies are listed in order to provide a more extensive overview of the topic of evolution of cooperation and related studies.

2.3 Group Behaviour and Identities

Among the studies on social dilemmas, those centred on the investigation of group identities constitute the line of research the most relevant to our dissertation. These studies focus on the ability of individuals to identify themselves as being part of a group [39], hence behaving differently depending on whether they interact with people belonging to their own group (*in-group*) or other groups (*out-group*).

Following the standard approaches in game theory, Akerlof and Kranton [2] proposed a utility function of group identities. Their function is then tested against real-world examples, that is gender discrimination, labour market, household division of labour, and the economics of social exclusion and poverty. Charness et al. [27] investigate whether individuals who are members of a group, and who identify with it, behave differently in strategic environments than players who do not. They rely on two versions of the prisoner's dilemma game under two conditions, that is face-to-face and anonymous. Further extension to this work was conducted by Sutter [198], who applied the paradigms of Charness et al. [27] to non-strategic decisions, even when no *out-group* exists.

Similarly to Charness et al., the face-to-face approach was investigated by Marzo et al. [147], who aimed to understand how the existence of friendship structures affect the levels of altruism of human participants in Colored Trails. Their findings show that people tend to be more generous in offers towards friends than strangers, whilst no clear evidence was found in terms of responder behaviour. Their findings were used for the implementation of artificial societies in some of our previous work [91, 83, 87]. Kim [111] conducted a study on group behaviours by setting up an ultimatum game experiment between participants from Washington and Lee University and Virginia Military Institute. His findings were consistent to those obtained by Marzo et al.

Tajfel [203, 202] is one of the main contributors to the theories of group behaviour. He not only investigated *in-group* favouritism (social identity), but also showed that people manifest group behaviours even when these are generated randomly [18]. Chen and Li [30] took inspiration from Tajfel and conducted a social dilemma experiment aimed to understand whether and how induced group identities affect individual behaviours. Their work has been used in our dissertation as the experimental protocol of our studies and will be described in detail in Section 4.1. Moreover, their experimental protocol has been of inspiration in similar other studies. For instance, Kranton et al. [116] relied on the same “painters preference” group assignment (see Section 4.1.1) and the *other-other* allocation game (see Section 4.1.3), Morita and Servátka [153] relied on allocation games, whilst Tremewan [209], instead, studied coalition formation in *divide the dollar* games. Fiedler et al. [62], on the other hand, utilised a similar experimental protocol to conduct an experiment focused on social distance, i.e. the perceived distance or dimension of closeness between interacting individuals or groups.

With respect to the studies on group behaviours based on non randomly assigned groups, Goette et al. [78, 79] investigate whether the social aspect of organisations has the benefit of fostering non-selfish cooperation and the punishment of norm violation within the group. Their work is based on soldiers of the Swiss Army, grouped together in platoons during their four-week phase of officer training, who interacted with each other by means of the prisoner's dilemma game. Another example is the work of Bernhard et al. [16], which focuses on tribes of Papua New Guinea using a form of the ultimatum game. One of their finding is, once again, *in-group* favouritism, meaning that providers tend to transfer more money to the participants belonging to their own tribe. Although not strictly centred on group behaviours, though similarly to Bernhard et al. [16], Jackson et al. [104] focused on how tribal populations in India treat each other in terms of trust and cooperation.

Other studies focused on several aspects of intra-group decisions were performed by Bornstein [22], by investigating how within group decisions lead to different strategies compared to those observed in single-participant ultimatum games. Abbink et al. [1] study how conflict in contest games is influenced by parties being groups instead of individuals, and investigate the existence of punishment between members of a party.

2.4 Community Structure Detection

This Section covers the most prominent work on Community Structure Detection (CSD) conducted by studies not focused on evolutionary computation, for which we decided to dedicate a separate discussion in Section 2.5. In a nutshell, the CSD task can be seen as an unsupervised problem of partitioning a given network into k clusters — with knowledge about k generally missing [144] — by solely leveraging on the information held by the topology of the network. CSD is a rather new research field and one of the most important among those on network theory [14, 77, 157, 3]. As a result, research on the very semantical definition of a community is still ongoing; as a consequence, the establishment of a solid and generally accepted approach and algorithm for CSD is still missing. The studies outlined in the following Sections aim to give an overview of the most established work and current directions of CSD research. The reader is referred to [144, 14, 157, 3, 121], among others, and Section 3.1 in this dissertation, for an introduction to the main concepts of complex networks and community structure detection.

2.4.1 Undirected and Unweighted Networks

Since CSD is an unsupervised task, one of the most consolidated approaches is the evaluation of a given partition through a quality measure, e.g. the clustering coefficient [77, 157, 3]. One of the most common measures of the goodness of a network partitioning is *modularity* [161], which aims to compare the *within community* edge density of a given network partitioning with the *expected* edge density of a random graph with the same average degree distribution of the given network (see Section 3.2). Newman [160] proposed the use of modularity as a target objective to be maximised via spectral partitioning [162], which arguably constitutes one of the most established approaches for CSD of mid-size networks, e.g. up to 10^4 nodes¹. A description on how spectral partitioning is used for CSD will be provided in Section 3.2.2. When networks scale up and become large, e.g. more than 10^4 nodes, other simpler approaches, generally based on agglomerative greedy optimisation are used, such as the Clauset-Newman-Moore algorithm [34] for fast community structure detection in large networks via modularity maximisation. Another similar approach was proposed by Blondel et al. [19], which instead creates a hierarchical agglomeration of nodes into communities in an incremental fashion.

Modularity maximisation has been also used in combination with e.g. extremal optimisation [45], ant colony optimisation [145], hierarchical clustering [77, 101], and particle swarm optimisation [220]. Other methods for community structure detection involve e.g. canonical clustering approaches [137] and k-means [75]. Pons and Latapy [178] rely on random walks for CSD in large networks, while Sobolevsky et al. [194] present a generic, fitness independent algorithm for community structure detection in undirected and unweighted networks. Their approach, which was intended to compare modularity with the minimum description length measure, is however very structured, which involves merging two communities, and splitting a community into two by moving nodes between two distinct communities. Steinhäuser and Chawla [195] present a relevant study to the core of this dissertation. They compare four algorithms for CSD, based on networks with known *true* community structures, by relying on five performance measures.

Modularity has also been extended to more complex network types, such as multiplex networks [154] — i.e. networks in which edges are of more than one type. Furthermore, modularity has also been extended in a hierarchical, multi-scale form by Lambiotte [119], which has also been applied to brain networks by Meunier et al. [151].

Despite the fact that modularity [161] is possibly the most widely used and extended measure for CSD, its effectiveness and generalisability has been questioned by many researchers. Possibly, the most important contribution on this line of research came by Fortunato and Barthelemy [68]. In their work they highlighted that the definition of community implied by modularity is actually not consistent with its optimisation, which may favour network partitions with groups of modules combined into larger communities. The *resolution limit* of modularity does not depend on particular network structures, but stems solely from the comparison between the number of links of the interconnected communities and the total number of links of the

¹This statement was excerpted from Barabási’s keynote talk the International School and Conference on Network Science (NetSci) held in Copenhagen in June 2013. <http://netsci2013.net/>

network [17]. As a consequence, modification of modularities, by e.g. defining internal parameters like a resolution parameter [122] or split penalty [29] were proposed.

Beyond modularity, Radicchi et al. [183] proposed the notion of *weak community*, that is a community for which the sum of all its within community degrees is larger than the sum of all its *out-degrees*. Their work also reflected on the *community score* measure, defined by Angiulli et al. [6] for undirected and unweighted networks, which has been extensively used for CSD via evolutionary computation, as we will show in Section 2.5.

Cruz et al. [36] proposed an entropy-based objective function to be maximised for the detection of community structures. Aldecoa and Marín [4] propose a new measure of the quality of a community structure partition in undirected and unweighted networks, called *surprise*, though it is non-trivial how to extend it and generalise it to weighted networks. Lancichinetti et al. [125] developed a statistical method aimed at discriminating between a given community and structures arising as topological fluctuations due to e.g. noise, which could also be used as an objective function for CSD. Similar work was also conducted by Gfeller et al. [76], where they introduce a method to identify the nodes lying “between clusters” through a measure called *clustering entropy*.

2.4.2 Directed and/or Weighted Networks

The work on modularity [161] led Newman and colleagues to the investigation of directed networks [128]. Gómez et al. [81] extended modularity to signed networks [5] — i.e. a form of weighted networks with semantic representation which can be seen as similar to social balance theory [98] — by using two target measures, namely modularity maximisation, and *frustration minimisation*.

Random walks have appear to be a promising approach to CSD, especially in directed networks [144], arguably due to the semantic notion of *flow* [186] of the network obtained by its edge directionality. A possible way to combine random walks with measures of network modularity comes from Kim et al. [112], who take inspiration from the PageRank [169, 118] algorithm, for the definition of their *LinkRank* measure. The definition of LinkRank was motivated by the apparent inability of modularity [128] to properly evaluate edge directionality and hence network flow (see also Section 3.2.3). Rosvall and Berstrom [186] rely on random walks for the detection of communities in large citation networks extracted from cross related academic research fields.

The intuitive extension of modularity [81, 161] led to its application also to weighted networks [158]. Beyond modularity, furthermore, Lu et al. [140] propose two metrics, called *intra-centrality* and *inter-centrality*, to characterise nodes in communities.

Farmer and Fotheringham [59] used spectral partitioning for the maximisation of what appears to be the most generic definition of modularity, obtained by the combination of the directed [128] and weighted [158] ones. In this dissertation we followed a similar approach for the definition of a baseline performance upon which we will test our evolutionary approaches (see Chapter 8).

Most of the research on weighted and directed networks occur through the studies on *dynamic networks*. Loosely speaking, a dynamic network is a network which changes its topology during time. The research on dynamic networks is generally composed of two main tasks: community structure detection on the given networks, and identification of salient points in time in which the network topology — hence the related communities — differ significantly. Modularity maximisation is also used by Duan et al. [44], in an algorithm for community structure detection in dynamic directed and weighted networks. Liu et al. [139] applied modularity maximisation to a very large directed and weighted network of 100 million nodes by means of a greedy aggregative algorithm similar to the one proposed by Blondel et al. [19]. Greene et al. [93] used the CSD method by Bondel et al. for the study of community evolution in dynamic social networks, though undirected and unweighted.

The vast majority of the listed work — including our previous work [88] — relies on the benchmark synthetic network model initially proposed by Girwan and Newman [77]. In a nutshell, this model is based on networks of 128 nodes divided to four communities of equal size. The difficulty of that benchmark is adjusted by regulating the amount of edges connecting communities. Motivated by the analysis of real life networks and other related phenomena, which showed that it is rather common to observe community

sizes and their connectivity following a power law distribution [35], many studies decided to investigate and propose new, arguably more *realistic* benchmark scenarios. On that end, we report the work of Lancichinetti et al. [124], Lancichinetti and Fortunato [120], and Orman et al. [167]. As we will show later in Section 8.2, for our empirical investigation we followed the guidelines of the benchmark problems of both Girvan and Newman [77] and Lancichinetti et al. [124].

2.5 Community Structure Detection via Evolutionary Computation

Evolutionary computation has been extensively used in many disparate research fields, such as clustering [100] and data mining [69]. These fields of research closely relate to the CSD research, though it appears that the study of evolutionary computation for this problem is slightly less popular. Moreover, the vast majority of the existing work on evolutionary CSD is focused on undirected and unweighted networks. These are just two of the many reasons which motivated us to further investigate genetic search in our previous work [88] and this dissertation at large. Moreover, one of the most central phase of an evolutionary algorithm design, which would significantly affect its computational performance, is the choice the most appropriate chromosome representation [10]. As we will show in Chapter 7, we considered three distinct representations, which we will call *node*, *edge*, and *group*.

2.5.1 Node Representation

Probably the most intuitive chromosome representation for evolutionary CSD, derived from standard clustering problem applications, consists of chromosomes of length n , where n is the number of nodes of the network. Each gene corresponds to a node of the network, and its allele value represents the node’s community identity. Figure 7.1 depict an example of a chromosome with node representation, and the corresponding community structures of a given network. This type of representation is also called *string* or *object* representation [100, 55]. We will however refer to it as *node* representation, in order to better situate it, at least syntactically, with the other two representations considered in this study (see Chapter 7).

Tasgin et al. [204] propose an algorithm with constrained crossover operators similar to the group crossover operators [55], a mutation operation which allows genes to take, as new allele values, only those available among the corresponding node’s neighbours, and modularity as fitness function. Gog et al. [80] used this representation combined with modularity, in collaborative evolutionary algorithms. The work was then extended by Chira et al. [33], in which they propose a new fitness function, which takes into account also features such as the size of the community and node neighbourhood.

In our study we will apply the node representation, together with modularity and LinkRank, to directed and weighted networks. Moreover, we will further investigate their use through niching, given the promising results that were obtained in our previous study [88].

2.5.2 Edge Representation

Another chromosome representation is the *locus-based* [172]. This representation considers chromosomes of length n , similarly to the node representation. However, the allele value j of gene i no longer represents a community identity label, rather the edge of the network connecting nodes i and j . For consistency, moreover, most of the evolutionary approaches for CSD which are based on this representation impose the constraint that an allele value is valid only if the corresponding edge exists in the network under investigation. We will also refer to the locus-based representation as *edge* representation. Figures 7.7 and 7.8 depict, respectively, a chromosome with edge representation and its imposed constraint.

One of the most promising and effective uses of this representation comes from Pizzuti’s GaNet [174, 177], which also relies on uniform crossover and a mutation operator allowing genes/nodes i to take allele values j only if node j is present in the underlying network. GaNet, moreover, evolves its chromosomes by maximising the community score fitness function. The work on GaNet led to the definition of GaNet+ [175], designed to detect overlapping communities, and community structures in signed networks [5].

Mazur et al. [148] studied GaNet depending whether community score or modularity was used. Similar use of the locus-based representation and modularity was conducted by Wang et al. [214], though the algorithm they implemented differs from GaNet especially in its mutation operator, and the fitness function they considered was a modularity-like measure [141].

Other related work based on the edge representation involve He et al. [97], who explore the use of ensemble learning and modularity; Lipczak et al. [135], who utilises an agglomerative hierarchical clustering representation which might bring potential advantages with respect to large networks; and Firat et al. [64], whom representation is similar to k -menoids. Furthermore, Liu et al. [138] propose the use of evolutionary computation as a replacement of Spectral Partitioning [160], making it thus one of the most effective evolutionary CSD applications to large networks.

In our dissertation we will extend the application of the edge representation for the CSD task in directed and weighted networks, will explore its potential combination with a niching approach, and will examine the impact modularity and LinkRank have on algorithms based on this chromosome representation.

2.5.3 Group Representation

Falkenauer [57, 58, 55, 56] is possibly the most important exponent investigating the use of an extended form of the node chromosome representation, especially targeted for grouping tasks, such as bin packing (Falkenauer and Delchambre [58]) or job scheduling (Falkenauer and Bouffouix [57]). The proposed representation — which we will refer to as *group* representation — and related crossover operator, which lead to the definition of the so-called group genetic algorithm (GGA) [55], were also used for solving the graph colouring task [65] by Eiben et al. [50].

Due to the similarity that the graph colouring task has with community structure detection, we decided to consider a variation of Falkenauer’s GGA in our dissertation. To the best of our knowledge, there has never been any attempt, except from the one presented in this dissertation, to apply the group representation to community structure detection problems. Moreover, and similarly to the node and edge representation, we will investigate single-population based algorithms, algorithms adopting niching methods, and the impact two fitness function, i.e. modularity and LinkRank, have on the algorithms who adopt this chromosome representation.

2.5.4 Multiobjective Optimisation and Niching

Multiobjective optimisation for CSD has been investigated in the last few years. The clear, intuitive potential advantages of its use is the combination of more than one fitness function, e.g. modularity and community score, to compensate the drawbacks of the two measures with each other, or to define target objectives which can take into account many more complicated aspects of community structures.

Once again, Pizzuti [176] relied on the locus-based representation to evolve chromosomes which were maximising the community score [6] and minimising a fitness function designed by Lancichinetti et al. [123] for overlapping communities. Similar work was also conducted by Shi et al. [191]. Other related work examples of multiobjective optimisation are those of e.g. Liu et al. [136] for separated and overlapping communities; Dy et al. [43] for overlapping community detection; Folino and Pizzuti [66] for dynamic networks; and Gong et al. [82], who relied on immune systems. The studies presented in this dissertation do not consider multiobjective optimisation, though we consider them as a viable future line of investigation.

In general, niching methods are evolutionary computation approaches in which multiple explicitly or implicitly separated populations are kept and evolved concurrently, allowing for the seek of multiple local optima at once [180, 181]. Niching methods have been extensively studied for clustering problems, see [143, 187, 196, 190] among others. However, to the best of our knowledge, the only use of niching for CSD was attempted by our early study on fully connected, directed and weighted networks [88]. The promising results obtained, especially considering the rather structured niching method used, motivated us to further investigate niching in this dissertation.

2.6 Other Related Work

This Section covers a plethora of work which relevance to our dissertation is secondary, though it can still contribute to better situate or motivate our study, or inspire future research paths.

2.6.1 Overlapping Community Structure Detection

The general outcome obtained by canonical community structure detection is that the network is partitioned in such a way that each node will belong to one and only one community. However, there exist many examples in real life, and especially in social network studies, in which nodes might belong to more than one community. These so-called *overlapping* communities are the focus of several studies. The most important contribution came from Palla et al. [171], who proposed an overlapping CSD method based on clique percolation. Similar work was also conducted by Friggeri et al. [70], who proposed a measure of *cohesion* among the nodes.

Modularity has also been extended to overlapping communities by Nicosia et al. [163], and Evans and Lambiotte [53]. Although the research on overlapping CSD is tangential to our research — the experimental setup we considered assumes non overlapping groups, see Chapter 4 — we do not exclude the future investigation in a line of study similar to ours. This is particularly motivated the promising results of its application through evolutionary computation [175], and more generally by its application in social contexts [170].

2.6.2 Social Network Studies

This Section presents relevant work mostly centred on social networks which are not necessarily investigated through game theory or social dilemma interactions. Aspects of social networks usually revolve around concepts such as homophily [149], i.e. the principle that a contact between similar people occurs at a higher rate than dissimilar people. Typical aspects investigated in social networks are religion, ethnicity, social classes, and so on.

Within social network studies, social balance theory [98, 133] has a prominent role. Social balance theory suggests that opponent relationship types, e.g. like vs. dislike, or friend vs. enemy, can be balanced among two people if both individuals like/dislike each other. By representing the relationship types, in a network, by signed weights, e.g. +1 for like and -1 for dislike, the multiplication of the weight edges can be used to determine whether e.g. three individuals² have a balanced or unbalanced relationship [7].

Within maintaining the closeness to our investigation, *social balance theory* has been studied by Antal et al. [7], who evolved network weights, in an evolutionary dynamics fashion, in order to simulate the tendency that humans have to reduce unbalanced triads. Similar evolutionary dynamics work was done by Khanafiah and Situngkir [108], though they relied on two measures, called *global balance index* and *local balance index*, to regulate the evolution of the network. Social balance theory was also extended to social groups by Newcomb [156]; with this respect we remark the work of Hummon and Doreian [103], who conducted experiments, similar to the previous ones described, though the main target of investigation was the group-based balance. Beyond the evolution of networks, social balance theory has also been studied with the aim to predict the ± 1 signs of nodes [130]. Although not directly connected, we remark the work of Amelio and Pizzuti [5], who investigated a multiobjective evolutionary computation approach to community structure detection in signed networks, which intuitively appears to share properties and notions with social balance theory and groups.

Other social phenomena investigated via social networks, which can also be used to characterise group behaviours, are *norms*, intended as a consolidated common behaviour among individuals. Sen and Sen [189] investigated the role that social network topology might have on norm emergence. They considered two network topologies, that is scale-free and ring networks, and let a population of reinforcement learning agents to learn which action, among five available, they should play. The learning task was achieved by providing positive reward if the two interacting agents were performing the same action, and negative reward otherwise.

With respect to modelling community structures in real-life situations, Palla et al. [170] adopt a data-driven approach for the detection of real-life community structures, of phone call networks, by means of clique

²The triad concept of social balance theory is reflected in the common sentence “the enemy of my enemy is a friend”. Translated into edge signs we obtain $(-1) * (-1) * 1 = 1$, that is, a balanced triad.

percolation. As we will show in Section 7.5, the social group evolution dynamics considered in their work inspired the mutation operators implemented in our evolutionary algorithms based on group chromosome representation. Similarly, Eagle et al. [48] rely on phone data to detect community structures of reciprocal friendship, and cross-validate it against the self-reports gathered from the participants of their experiment.

2.6.3 Cooperative and Multiplayer Computer Games

As stated in Chapter 1, our research on the definition of an interaction-based group modelling framework is particularly focused on its subsequent application to cooperative multiplayer computer games. Arguably, games would allow for the implementation of a higher level of interaction realism compared to those considered in canonical social dilemma experiments. The application of our framework to highly interactive environments might possibly shorten the gap between the empirical studies on evolution of cooperation and those observed in real life [211, 94, 150], see also Figure 4.1.

This Section aims to bring the attention to some of the most relevant work on collaborative games and multiplayer games, which have a focus on social interaction analysis and group modelling. El-Nasr et al. [188] defined metrics of collaboration for collaborative games. Their approach involves expert knowledge and is based on well-established console games such as *Guitar Hero* (Activision, 2006). Similar work was also done by Rocha et al. [185], though they did not define any metrics but just described, broadly, cooperative mechanics commonly found in games. Vogiazou and Eisenstadt [213] studied the influence of communication to the emergence of group behaviours; their qualitative findings, however, were not backed-up empirically. Nevertheless, their work showed how direct communication might not be required for the evolution of complex patterns and group formation. Ducheneaut et al. [46, 47] use longitudinal gameplay data collected in *World of Warcraft* (Blizzard Entertainment, 2004) to understand the relationships occurring between players grouping together in order to solve common guilds. Similar research efforts, augmented with discussions on social dilemmas and evolution of cooperations, were also conducted by Chen [28]. Szell and Thurner [201] investigated social dynamics in an economy based computer game set in space. The participants can interact with each other by trading resources, can actively form and leave groups. Their work focused on the analysis of the social network built from the private messages exchanged among the players. Interestingly, they did not conduct any community structure detection investigations aimed to understand whether there is, or how it might be defined, any form of correlation between built up groups and exchanged messages. Their work is then further extended together with Lambiotte [200]. More particularly, they analyse multiplex networks of six different types of one-to-one interaction. Three of them carry a positive connotation (friendship, communication, trade), and the other three a negative connotation (enmity, armed aggression, punishment).

2.7 Summary

This Chapter presented some of the most relevant work which could better help with situating the research presented in this dissertation. As we could observe, terms such as cooperation, groups, and networks, are used in many disparate research fields, ranging from e.g. psychology, sociology, economics mostly for the former, and also physics and mathematics for the latter. Moreover, the use of evolutionary computation for community structure detection has been mainly used in undirected and unweighted network problems. The Chapter also discussed studies which, though not directly related to the fundamentals and specific implementations of our framework, can be considered as topics of possible future research. These will be partly resumed in Chapter 10.

The next Chapter presents the most fundamental theoretical and computational concepts which are needed to better understand our work.

Chapter 3

Tools

This Chapter introduces the theoretical backgrounds of our dissertation. Section 3.1 will first present the most important concepts of network theory, with a particular emphasis on the community structure detection task. Section 3.3 will provide an overview of evolutionary computation. Finally, Section 3.4 will describe, briefly, the reinforcement learning problem and common approaches for solving it.

3.1 Network Theory

Networks and network problems have become ubiquitous as data from many different disciplines can be naturally mapped to graph structures [157, 3, 144, 14]. Many real-life phenomena can be schematically depicted through networks; food webs, friendship relationships, spread of diseases, the Internet, are just some examples. Moreover, real-life networks tend to carry “complex” patterns inside their topology. For instance, it has been observed that in many real-life network the nodes have links which form a power law distribution [15, 35], which broadly means that they are composed of many nodes with few connections and few nodes with many connections. These patterns have an important role in phenomena such as the spread of information, diseases, and so on.

The term *complex networks* is generally used to refer to networks with non-trivial topological features, such as those observed in real life. This term is used in order to delineate a syntactical and semantical separation from *random graphs* [52], which focus on a purely mathematical point of view, and never meant to serve as a model of real systems [14, 52].

Network science or *network theory* [14] is the research field mostly focused on the analysis of complex networks. This rather young field is gaining more and more attention, due to the growing access of data which can be used to represent complex phenomena through networks. In this Section we will solely introduce the main mathematical concepts of networks, their main topological structures, and, particularly, one very important complex pattern they tend to exhibit, that is, community structures. For a more thorough understanding of complex networks we remand to [157, 14, 3] among all.

3.1.1 Note on Terminology

In the scientific literature the terms network and graph are used interchangeably [14]. Yet, there is a subtle distinction between the two terminologies: the terms *network*, *node*, and *link* often refers to real systems. In contrast, the terms *graph*, *vertex*, and *edge* are used when the mathematical representation of networks play a key role [14]. Yet, this distinction is rarely made, so these two terminologies are often used as synonyms of each other [14]. The same will occur in this dissertation, though we will aim to selectively choose them depending on whether the emphasis is given on the complexity of the system (network terminology) or on the underlying mathematical concepts (graph terminology), see Table 3.1.

Table 3.1: Network and graph terminology [14].

Network Terminology	Graph Terminology
network	graph
node	vertex
link	edge

	1	2	3	4	5	6	7	8
1	0	0	0	1	1	0	0	0
2	0	0	1	0	0	0	1	0
3	0	1	0	0	0	0	0	1
4	1	0	0	0	1	0	0	1
5	1	0	0	1	0	1	0	1
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	0
8	0	0	1	1	1	0	0	0

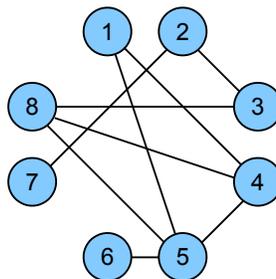


Figure 3.1: An example of an adjacency matrix and the corresponding unweighted and undirected network.

3.1.2 Network Types and Properties

A network N is generally defined as

$$N = (V, E) \tag{3.1}$$

where $V = \{v_1, \dots, v_n\}$ is the network's set of nodes or vertices, of size $|V| = n$, and E is the set of all links or edges connecting nodes in V . We will refer to a single edge, connecting nodes v_i and v_j , as (i, j) . If (i, j) exists, then v_i and v_j are said to be *neighbours*. A network can be depicted in two ways: through a graph structure, or through a matrix notation, called *adjacency matrix* $A = n \times n$. Item $A_{i,j}$ represents the edge between v_i and v_j in the adjacency matrix; if $(i, j) \in E$ then $A_{i,j} = 1$, otherwise $A_{i,j} = 0$. Figure 3.1 depicts a network of eight nodes in both ways.

Depending on the information carried on by the edges, there can exist several different network types. A network is said to be *directed* if the edges specify a direction of information between nodes: (i, j) means that v_i connects to v_j , though the opposite might not necessarily hold. This also means that A might no longer be symmetric, that is, if $A_{i,j} = 1$ that does not necessarily mean that $A_{j,i} = 1$. Graphically, an edge can be depicted with an arrow indicating its direction. Figure 3.2 depicts an example of a directed network.

	1	2	3	4	5	6	7	8
1	0	0	0	1	1	0	0	0
2	0	0	0	0	0	0	1	0
3	0	1	0	0	0	0	0	0
4	1	0	0	0	1	0	0	1
5	1	0	0	0	0	1	0	0
6	0	0	0	0	0	0	0	0
7	0	1	0	0	0	0	0	0
8	0	0	1	0	1	0	0	0

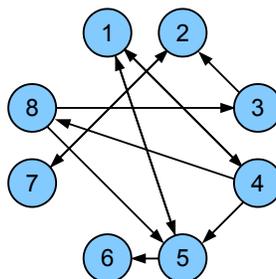


Figure 3.2: An example of a directed network; adjacency matrix and graph representation.

	1	2	3	4	5	6	7	8
1	0	0	0	75	25	0	0	0
2	0	0	100	0	0	0	25	0
3	0	100	0	0	0	0	0	50
4	75	0	0	0	50	0	0	25
5	25	0	0	50	0	100	0	75
6	0	0	0	0	100	0	0	0
7	0	25	0	0	0	0	0	0
8	0	0	50	25	75	0	0	0

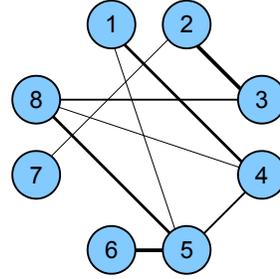


Figure 3.3: An example of a weighted network; adjacency matrix and graph representation.

	1	2	3	4	5	6	7	8
1	0	0	0	75	0	0	0	0
2	0	0	100	0	0	0	75	0
3	0	100	0	0	0	0	0	50
4	50	0	0	0	50	0	0	75
5	25	0	0	50	0	50	0	0
6	0	0	0	0	100	0	0	0
7	0	25	0	0	0	0	0	0
8	0	0	0	25	75	0	0	0

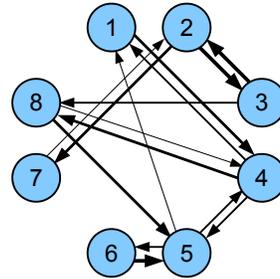


Figure 3.4: An example of a directed and weighted network; adjacency matrix and graph representation.

A network is said to be *weighted* if its edges carry a quantitative information about the strength of the connection between the nodes. The weights can be discrete, continuous, can range from minus infinite to plus infinite, and so on. Mathematically, entry $A_{i,j}$ will hold, directly, the edge weights. Graphically, instead, one can adopt different approaches; for instance, the weights can be explicitly reported, alternatively, the magnitude of the edges can be inferred by the thickness of the edges, and so on. Figure 3.3 depicts an example of weighted networks, through its adjacency matrix, and graphically through edge thickness.

Networks can also be directed and weighted (see Figure 3.4); nodes can have self-linking edges, or could even be fully connected, meaning that each node is connected with all other nodes in N (either including or excluding self-edges). On a complexity scale, it is intuitive that the undirected and unweighted case is the simple type, whilst the complete directed and weighted typology is the most complex one — compare Figures 3.1 and 3.4.

In general, the graph representation provides a more immediate perception of the complexity of the network under investigation, though it is generally through the adjacency matrix that it is possible to unveil, mathematically, the network’s complex structures.

Depending on the complex system under investigation, different network topologies might lead to different representation of information. For instance, an unweighted and undirected network is sufficient to represent the (reciprocal) friendship network of the *Facebook* platform; on the other hand, directed networks can better represent the followers network of the *Twitter* platform, since these are not necessarily reciprocal. Weighted networks can be used to represent, for instance, the amount of time friends spend together. Directed and weighted network can be used to represent, for instance, the air traffic among flight destinations.

Networks can be even more complex. For instance, if a network is composed of more than one type of edges then it is said to be a *multiplex* network. A network can also carry node attributes. A network can be *dynamic*, meaning that its topology changes throughout its existence. A network can also be *bipartite*, which means that are composed of two clear sets of nodes, and the edges connect only nodes belonging to different

sets.

3.1.3 Edge Degree

Several network properties can be extracted from their edge degree distribution. Mathematically, the degree distribution k_i , of node v_i , in an undirected and weighted network is calculated as

$$k_i = \sum_j A_{i,j} \quad (3.2)$$

in a directed network, instead, we can define, for node v_i , an *in-degree*

$$k_i^{in} = \sum_j A_{j,i} \quad (3.3)$$

and an *out-degree*

$$k_i^{out} = \sum_j A_{i,j} \quad (3.4)$$

Intuitively, in case of undirected networks, we would obtain $k_i = k_i^{in} = k_i^{out}$. Also, Equations (3.3) and (3.4) can be applied to weighted networks as well.

3.1.4 Random Walks on Graphs

The concepts described in this Subsection will help with better understanding the forthcoming *LinkRank* measure [112] of Section 3.2.3. The concepts are presented based on undirected and unweighted networks; nevertheless, their application to directed and/or weighted networks is straightforward.

Generally, a random walk is a mathematical concept formalising a procedure consisting of a sequence of random steps [144]. A random walk is built on a graph by starting from a random node and moving, randomly through the network, by following the node's edges and hence visiting neighbouring nodes. Let us assume that at time t the random walker is at node v_i . Hence, at $t + 1$, the random walker will move to the neighbouring node v_j with probability $1/k_i$. This defines the transition matrix P of the random walk [144], in which element $P_{i,j}$ is defined as follows:

$$P_{i,j} = \begin{cases} \frac{A_{i,j}}{k_i} & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Additionally, we can assign a probability π_i of visiting each node v_i in V . In general, random walks are considered to be Markov Chains [144], where the set of possible states corresponds to the node of the graph. This also means that, given the initial probability distribution of visiting all the nodes $\pi = [\pi_1, \dots, \pi_n]$, we can ultimately obtain the *stationary distribution* π_s . π_s is a probability distribution that does not change over time (after a sufficient amount of time), as $\pi_s = \pi P^t$, $\forall t$.

3.2 Community Structure Detection

As previously stated, many real life networks manifest the so-called power law link distribution, meaning that most of their nodes will have few links, whilst few nodes will have many links. These networks are also said to be *scale-free*. Figure 3.5 depicts an example of an undirected and unweighted network with scale-free property.

Another common property of complex network is the fact that nodes that are *alike* tend to be connected more with each other than with others. This feature, which has correspondence with the notion of clustering used in data mining, is usually referred to, in network theory, as *community structure*. As a matter of fact, there is still a lack of unanimous consensus on what a community structure might mean, and hence how

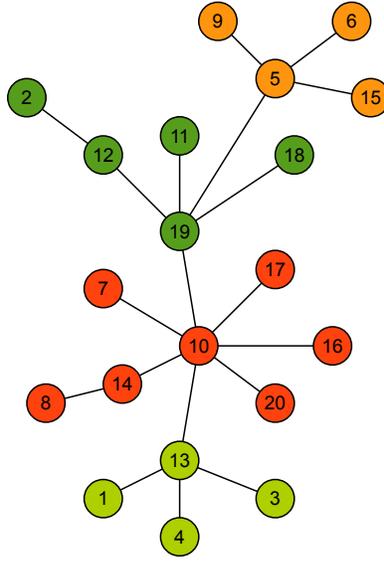


Figure 3.5: An example of an undirected and unweighted network with scale-free property and four communities detected through the Clauset-Newman-Moore algorithm [34]. The nodes with the same colours belong to the same community structure.

it can be detected. For instance, a community can be characterised by the node degree distributions, both in a *strong* [161] or *weak* [183] interpretation; it can relate to the *flow* of information specified by the edge directionality [186], and so on. As a consequence, the research on *Community Structure Detection* (CSD), or even how community structures can be quantitatively evaluated, is still on its early stages.

Computationally, CSD can be interpreted as an unsupervised clustering problem in which, except for some cases, the number of existing communities is unknown. Moreover, it has been proven that CSD is a difficult task, proven to be NP-complete over the set of networks of a given size [129].

Due to the *in-group* and *out-group* relationships of social group identities (see Chapter 1), and especially those considered in our empirical evaluations (see Chapter 4), in this dissertation we will solely focus on community structures obtained by *strict* node partitioning, meaning that a node will belong to one and only one community (or have one and only one community identity). Other forms of partitioning, for instance into *overlapping* communities [171, 43], is left to future investigations.

3.2.1 Modularity

Among the approaches to evaluate the goodness of a network partition into community structures, *modularity* is one of the most popular one. Loosely speaking, a network manifests modular community structures if its partitioning leads to a high connectivity among nodes *within* the community, whilst there is low connectivity among nodes *between* communities. Figure 3.5 depicts a network which can be partitioned in four community structures according to this interpretation.

Modularity Q aims to compare the *within*-community edge density of a given network partitioning with the expected edge density of a random graph with the same average degree distribution of the given network. The measure was initially introduced by Newman and Girvan [161, 144] for undirected and unweighted network partitions. Semantically, it can be expressed as follows:

$$Q = (\text{fraction of edges within communities}) - (\text{expected fraction of edges})$$

More formally, given an undirected and unweighted network $N = (V, E)$, and a partitioning into community structures, modularity Q^{uu} is calculated as follows:

$$Q^{uu} = \frac{1}{2w} \sum_{i,j} \left[A_{i,j} - \frac{k_i k_j}{2w} \right] \delta(c_i, c_j) \quad (3.5)$$

where A is N 's adjacency matrix, c_i (c_j) is node v_i (v_j)'s community structure identity, k_i is the degree of v_i calculated via Equation (3.2), $w = \frac{1}{2} \sum_i k_i$, and δ is the Kronecher function, which returns one if $c_i = c_j$ and zero otherwise. Modularity typically ranges between zero and one, though negative values are also possible. The term $\frac{k_i k_j}{2w}$ represents the expected number of links between node i and j .

Further work on modularity led to its application on weighted networks [158], and to its definition on directed networks [8, 129]:

$$Q = \frac{1}{m} \sum_{i,j} \left[A_{i,j} - \frac{k_i^{in} k_j^{out}}{m} \right] \delta(c_i, c_j) \quad (3.6)$$

where $m = \sum_{i,j} A_{i,j}$, k_i^{in} and k_j^{out} are calculated according to Equations (3.3) and (3.4), whilst A , c_i , c_j and δ are defined as for the undirected and unweighted case. Furthermore, similarly for the undirected and unweighted case, Equation (3.6) had also been applied to directed and weighted networks [59, 88].

3.2.2 Modularity Maximisation via Spectral Partitioning

In their original work, Newman and Girvan [161] used hierarchical clustering to partition a network into community structures, and subsequently relied on Q to determine when to break the nested partitions. Subsequently and more directly, Q has been used as an objective function for CSD. The approach, called *modularity maximisation*, is amongst the most established one. Many are the algorithms used for modularity maximisation; spectral partitioning is possibly one of the most consolidated ones [160, 162].

Modularity maximisation via spectral partitioning aims to split a given network, recursively, into two subnetworks, as long as the modularity computed on the whole N is being maximised. The partitioning is obtained by calculating the eigenvectors of the so-called *modularity matrix* B . Given an undirected and unweighted network with adjacency matrix A , the generic term $B_{i,j}$ of the modularity matrix is obtained as follows:

$$B_{i,j} = A_{i,j} \frac{k_i k_j}{2w}$$

The eigenvectors \mathbf{s} of B , which have length n , can be used to detect the partitioning: by making each feature s_i correspond to node v_i , and assigning community identity ± 1 depending on their sign, through modularity maximisation spectral partitioning will chose the eigenvector which maximises

$$Q^{uu} = \frac{1}{4w} \mathbf{s}^T B \mathbf{s}$$

Once the best \mathbf{s} is found, a subsequent refinement of the solution can be done [129]. The idea is the same as for the selection of the best \mathbf{s} : in an iterative process, each feature of \mathbf{s} is switched to the other community, with an order dictated by the highest delta Q^{uu} obtained after the switch; once all features are switched, the intermediate solution which maximises the relative Q^{uu} is chosen. Then, the two partitions of the current solution can be further split, recursively via spectral partitioning and fine tuning, so that more than two communities can be detected. Intuitively, the recursive process ends once no further split would improve the overall Q^{uu} . In other words, spectral partitioning is a *greedy*, *divisive*, and *deterministic* approach to CSD.

Spectral partitioning has also been used for CSD in directed networks. The term $D_{i,j}$ of modularity matrix D is calculated as follows:

$$D_{i,j} = A_{i,j} \frac{k_i^{in} k_j^{out}}{m}$$

Modularity matrix D , however, might no longer be symmetric, which means that the eigenvectors cannot be computed. The problem is solved by using $(D - D^T)$, meaning thus that Q is computed as follows:

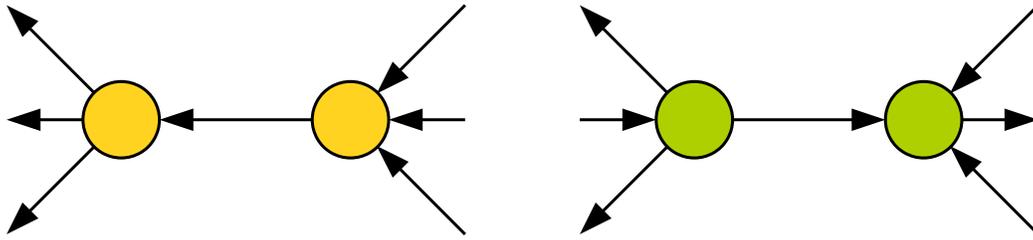


Figure 3.6: Modularity, applied to the two subnetwork at the left (orange) and right (green) would return the same value, though if we focused on the *flow* of the edges we observe that they represent two different configurations. The LinkRank algorithm, based on a random walk approach and another modularity measure, aims to solve this ambiguity.

$$Q = \frac{1}{4m} \mathbf{s}^T (D - D^T) \mathbf{s}$$

the recursive application of spectral partitioning, together with the local fine tuning, can now be done. Although this approach is not a universally good one — there would hardly be any significant research on CSD via modularity maximisation otherwise — spectral partitioning works rather well for mid-sized networks¹. Moreover, due to its popularity [162], deterministic approach, and the fact that it has been successfully used for directed and weighted networks [59], it will constitute the baseline performance upon which we will compare the evolutionary approach for CSD we investigated in our dissertation, as it will be extensively presented in Chapter 8.

3.2.3 Critiques against Modularity Maximisation

Despite being one of the most adopted approach, modularity has been criticised under many aspects. One claim against its use, or at least one which highlights its weaknesses for certain network types, is the so-called resolution limit [122]. In a nutshell, modularity, hence CSD via its maximisation, seems to suffer networks composed of either small sized *true* communities — which would tend to be aggregated together to form bigger ones — or those with large sized communities — which would tend to be split into smaller ones. A solution proposed by Reichardt and Bornholdt [184] requires the addition, to the modularity definition, of a resolution parameter λ , which would allow for the steering of the optimisation process towards expected community sizes:

$$Q^\lambda = \sum_S \left[\frac{k_{in}^S}{2m} - \lambda \left(\frac{k_{tot}^S}{2m} \right)^2 \right] \quad (3.7)$$

where the sum runs over all the detected communities, k_{tot}^S is the sum of the degrees of vertices in module S , k_{in}^S is twice the number of internal edges of module S , and $m = \sum_{i,j} A_{i,j}$. High values of λ lead to smaller modules, and low values of λ lead to bigger modules [184, 122]. Clearly the resolution-dependent modularity would have an advantage over Newman's, providing however the existence of a priori knowledge about the community structures in the network.

Another weakness of modularity, this time when applied to directed networks, was highlighted by Kim et al. [112]. More specifically, they showed that modularity seem not to be able to properly evaluate the *flow*/patterns of edge directions, see Figure 3.6 as an illustrative example. Their alternative approach to CSD in directed networks, called *LinkRank*, is inspired by the *PageRank* algorithm [169, 118] — mostly known to be the algorithm behind the *Google* search engine. PageRank and LinkRank assume the existence of a random user who clicks, randomly, hyperlinks on web pages / moves randomly through the nodes of a

¹see the footnote of Chapter 2

networks. Ideally, a page that is referenced by many others should be an *important* one, hence have higher chances to be reached by the random walker. Similarly, the hyperlink between an important page to another one should have more importance than a link, to that page, from another non important page.

Intuitively, LinkRank relies on the stationary probability distributions π_s of the given directed network. However, since it is possible that in the network there might exist dangling nodes which would interrupt the random walk — i.e. nodes v_i for which $k_i^{out} = 0$ — or similarly that there might exist nodes which cannot be reached — i.e. nodes v_j for which $k_j^{in} = 0$ — the stationary distribution is obtained as follows:

$$\pi_s = \pi_s G \tag{3.8}$$

where G is the so-called *Google matrix*, in which entry $G_{i,j}$ is calculated as follows:

$$G_{i,j} = \alpha \frac{A_{i,j}}{k_i^{out}} + \frac{1}{n} (\alpha a_i + 1 - \alpha) \tag{3.9}$$

where $(1 - \alpha)$ is the *teleportation probability*, by which the random walker stops following the hyperlinks and opens a random webpage, a_i is equal to one only if v_i is a dangling node, otherwise it is zero, whilst $A_{i,j}$ and k_i^{out} were previously defined in Equation (3.4). $\frac{A_{i,j}}{k_i^{out}} = 0$ if $k_i^{out} = 0$. By adding a_i and α to the definition of G , the random walker would not be trapped in any part of the network during the random walk process [112].

Ultimately, Kim et al. proposed a new modularity measure based on the stationary distributions, Q^{lr} , defined as follows:

$$Q^{lr} = \sum_{i,j} [\pi_i G_{i,j} - \pi_i \pi_j] \delta(c_i, c_j) \tag{3.10}$$

which, semantically, it corresponds to

$$Q^{lr} = \begin{aligned} & \text{(fraction of time spent walking} \\ & \text{within communities by a random walker)} - \\ & \text{(expected value of this fraction)} \end{aligned}$$

Intuitively, Q^{lr} can also be applied on directed and weighted networks [112]. Although this new modularity might indeed give more importance to the directionality of the edges, it now depends on parameter α , which is independent on the network structure, which also implies it might introduce other difficulties with respect to the CSD task, similarly to the parameter λ of the multi-resolution modularity Q^λ — see Equation (3.7).

3.3 Evolutionary Computation

Evolutionary computation is a term which encompasses a large family of computational techniques which allow for the seeking of solutions to optimisation problems via stochastic processes. It takes inspiration of the main concepts of Darwinism, that is offspring creation, natural selection and survivor of the fittest. Evolutionary computation has been used in many different research fields. It has not only provided successful and surprisingly original outcomes, but also showed extreme robustness against uncertain and dynamic environments [9, 49].

In this Section we will only present the key concepts which are required in order to better understand our research on evolutionary community structure detection presented in Chapter 7.2. We will rely on the typical structure of an Evolutionary Algorithm (EA) [49], depicted schematically in Figure 3.7 and via pseudocode in Algorithm 1.

3.3.1 Evolutionary Population, Representation and Initialisation

An EA is typically based on a *population* of candidate solutions of the optimisation problem. Each candidate solution is called an *individual* or *chromosome* of the population. However, since the term “individual” is

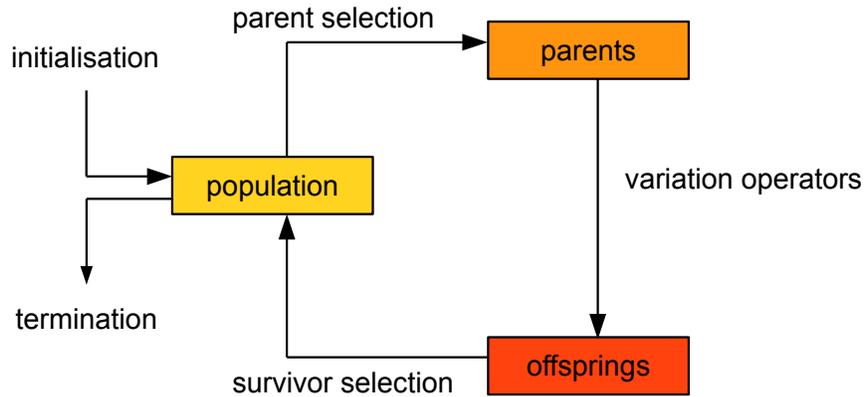


Figure 3.7: The general scheme of an Evolutionary Algorithm as a flow chart [49].

Algorithm 1 Generic Evolutionary Algorithm

- 1: initialise population with random candidate solutions
 - 2: evaluate each candidate
 - 3: **repeat**
 - 4: select parents
 - 5: recombine pair of parents
 - 6: mutate the resulting offsprings
 - 7: evaluate new candidates
 - 8: select individuals for next generation
 - 9: **until** termination condition is satisfied
-

also used to refer to the member of the social synthetic environments under investigation, in this dissertation we will mostly use the term chromosome.

The most fundamental design consideration for an EA is arguably the definition of its *chromosome representation* [10]. All other aspects of evolutionary computation, such as the variation operators and fitness evaluations, necessarily depend on the way the chromosomes represent their related candidate solution. For instance, they can represent *trees* [9], as it is commonly done in evolutionary computation’s genetic programming subfield. Alternatively, they could represent *strings* [9], composed of a sequence of *genes*, which *allele* values correspond to characters selected from an *alphabet*. In this dissertation, the chromosomes of our all EAs considered will adopt the string representation. The chromosomes of the same population can have equal or different length (in our dissertation we will consider both cases), and can represent candidate solutions more or less directly [9, 49]. It is usually said that the chromosome represents the *genotype* of an individual (solution), and that its translation into an object which is used for fitness calculation represents the *phenotype*.

The EA’s evolutionary process begins by randomly initialising its chromosome population. Depending on the problem the EA aims to solve, and possibly on the existence of some a priori knowledge about the global optima configuration, the initialisation process of each chromosome can undergo some form of additional heuristics which would restrict, or more loosely orientate, the subsequent evolutionary search.

3.3.2 The Fitness Function and the Evolutionary Loop

All possible chromosomes can be imagined as belonging to a so-called *search space*. This needs to be explored, so that the chromosome representing the global optima of the problem could be found. In order to achieve this, the EA needs an evaluation procedure of its current population and, based on it, prosecute its evolutionary search. The evaluation procedure, also called *fitness function*, expresses quantitatively how *well* a candidate

solution fulfils the objective. The evaluation of the whole search space through the fitness function leads to a *fitness landscape*, composed of peaks corresponding local optima solutions. The globally optimal solution(s) corresponds to the chromosome(s) which fitness score leads to the highest peaks in the fitness landscape. Intuitively, the more accurate the fitness function can represent the objective of the evolutionary search, the more feasible the exploration of the search space will be. However, there is also the risk of a fitness function (even an accurate one) making the fitness landscape deceptive, thereby hindering effective search.

Once the population is initialised, and each chromosome is evaluated, the evolutionary loop begins. Each iteration of the loop is called an evolutionary *generation*. Within each generation, new candidate solutions (*offsprings* or *children*) will be generated from existing ones (*parents*) and based on their relative fitness scores.

Independently of its content, however, the loop still requires a termination condition. A possible termination condition can be a maximum number of generations, the lack of improvement of the highest fitness scores for a consecutive number of generations, or also the reach of the highest fitness score limit, providing that such limit is known. Another common stopping criteria, which is also the one we will use in our dissertation, is the reach of a certain number of fitness evaluations.

3.3.3 Parent Selection

According to the theories of natural evolution, better fit individuals would have higher chances to reproduce, with the consequence that the resulting children will inherit their parent's good genes, which will ultimately lead to improved candidate solutions.

This *Parent Selection* procedure can be implemented in many possible ways, providing however that the individual's fitness score is taken into account. Among all, we highlight two important features that could be implemented: (1) the probability for an individual to be selected could be proportional to its fitness score, so that the genetic search could prosecute towards local optima (2) all the individuals could have a non-zero chance to be selected, so that the algorithm would not stagnate to local optima. Recall in fact that evolutionary computation is a stochastic process, meaning thus that current suboptimal solutions can still lead to global optima. Although in two different contexts, in this dissertation we will use two popular probability-based techniques, namely *roulette wheel selection* and *rank selection*, hereby described.

Given a population P of n chromosomes, roulette wheel selection will assign, to individual c_i with fitness f_i , probability

$$p_i^w = \frac{f_i}{\sum_j^n f_j}$$

On the other hand, rank selection will first determine the ordered rank, by decreasing fitness², of the individuals in P ; subsequently, it will assign probability

$$p_i^r = \frac{n + 1 - r_i}{\sum_{j=1}^n j}$$

where r_i is the rank index of chromosome c_i . Table 3.2 presents a comparative example of the two probability distributions based on an arbitrary population of chromosomes, with arbitrary representation and fitness function.

3.3.4 Variation Operators

The role of variation operators is to create new individuals from old ones [49]. In the corresponding search space this amounts to generating new chromosomes-candidate solutions. Variation operators can be classified according to their *arity*, that is the number of chromosomes they take as input parameter. In our dissertation we will focus on two, possibly most widely used operators: the *binary recombination* or *crossover* operator, and the *unary mutation* operator.

²Obviously, in case of minimisation problems, rank selection will sort the individuals by ascending fitness score.

Table 3.2: A fictional example of an evolutionary population at a certain evolutionary loop. The first three individuals, that is the current elite, will survive the selection process. The remaining seven individuals will be replaced by new offsprings. The last two columns report the different probabilities of parent selection — which would lead to the generation of the offsprings — depending on whether roulette wheel or rank selection is used.

Rank Index	Individual	Fitness Score	Parent Selection Probability	
			Roulette Wheel (p^w)	Rank (p^r)
1	J	0.98	0.1614	0.1818
2	E	0.8	0.1318	0.1636
3	B	0.78	0.1285	0.1455
4	A	0.69	0.1137	0.1273
5	C	0.63	0.1038	0.1091
6	F	0.56	0.0923	0.0909
7	H	0.53	0.0873	0.0727
8	D	0.4	0.0659	0.0545
9	I	0.36	0.0593	0.0364
10	G	0.34	0.056	0.0182

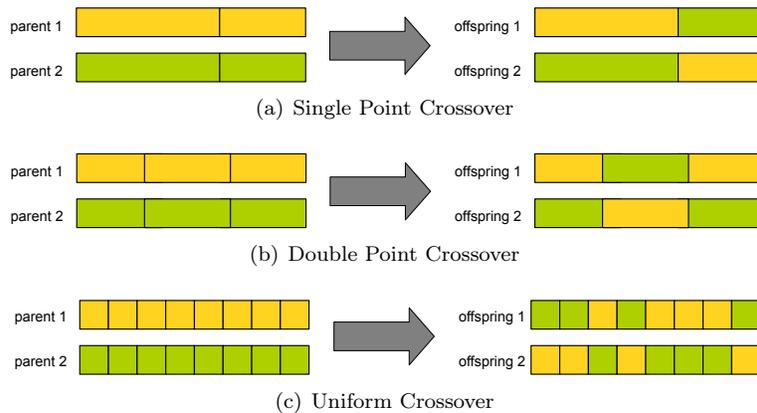


Figure 3.8: Schematic representation of (a) single point, (b) double point, and (c) uniform crossover operators.

Recombination Process

The *binary* recombination process, also called *crossover*, aims to simulate the mating process present in nature, so that fit parents will transfer their “good” genes to their children. In our dissertation we will rely on crossing over two parents for the generation of two offsprings, though this approach is not the only one possible.

Depending on the chromosome representation used, the crossover operator can be implemented in several different ways. Two of the most generic ones, called *single point* and *double point* crossover, generate offsprings by copying alternate segments of sequential genes of the parents. Another possibly even more generic recombination operation is *uniform* crossover, which assigns genes of the first/second parent to either the first/second offspring independently and uniformly. The three crossover operators, applied to string-based chromosomes, are schematically depicted in Figure 3.8.

Finally, the crossover operators might even be probabilistic, meaning that two parents, once selected, will reproduce and generate offsprings with probability $0 < p_{\text{cross}} < 1$, otherwise they will not mate. In the latter case, many options are available. In all our implementations two parents will mate, via uniform crossover, and generate two offsprings with probability $p_{\text{cross}} = 0.8$; with probability $1 - p_{\text{cross}} = 0.2$, instead, they will

Algorithm 2 Generic Mutation Operation

```
1: sample  $r \in [0, 1)$  uniformly
2: while  $r < p_{\text{mut}}$  do
3:   perform a mutation operation on the chromosome
4:   sample  $r \in [0, 1)$  uniformly
5: end while
```

not mate and the parent selection procedure will reoccur.

Mutation Process

Mutation is a *unary* operator which takes a chromosome, modifies its genes, and produces another chromosome. Typically, this operator is applied on newly generated offsprings after crossover, though it can even exist without any previous recombination operation.

The mutation operator is clearly representation dependent. For instance, if the genes have allele values belonging to a predefined alphabet (e.g. string-based representation), mutation would necessarily change the genes of a chromosome by sampling new values from the available alphabet. Algorithm 2 depicts the pseudocode of the generic mutation operation we implemented in the EAs considered in this dissertation. The process is based on a mutation probability $0 < p_{\text{mut}} < 1$, which allows the change of more than one gene per chromosome, though this accumulated probability decays exponentially. In all our implementations, moreover, we will consider $p_{\text{mut}} = 0.8$.

3.3.5 Survivor Selection

Once the offspring *pool* has been filled up, these will replace the previous individuals, the evolutionary generation will terminate and a new one will begin. One last common feature of EAs, which also resembles the Darwinist notion of survival of the fittest, is implemented by allowing the most fit individuals of the previous generation survive the replacement process, and still be present in the genetic population of the next evolutionary iteration. These survivor individuals are generally those which — offsprings not included — scored the highest fitness values. This feature is also called *elitism*. For instance, Table 3.2 depicted a population of ten chromosomes; three of them belong to the *elite* subpopulation, and these will be also present, together with the newly generated offsprings, to the next evolutionary loop. In general, given a population of m chromosomes, and an elite of e individuals, the offspring pool needs to have a minimum size of $m - e$, so that the new population for the next generation will still have m chromosomes.

3.3.6 Additional Features: Niching

Depending on the characteristic of the optimisation problem there is the risk that the search space which needs to be explored might become intractable. This is particularly often true if the designed EA is intended to work for a vast range of similar optimisation problems, meaning that the chromosome representation is very generic so that the search space becomes huge. If the search process fails to explore the entire search space and prematurely converges on a subspace which does not contain the global optimum, this optimum will likely never be reached. In other words, there might be a need for enforcing *diversity* in the population.

One common way to tackle the diversification of candidate solutions, and the concurrent exploration of different regions of the search space, is *niching* [143]. Simply put, niching approaches consider the genetic population as being composed of many more-or-less independent subpopulations, which mate and evolve within each others, so that multiple local optima can concurrently evolve [181, 180]. Clearly, the way the niches can be implemented are countless. For instance, niches can be *temporally* or *spatially* characterised; they can be constrained, so that each niche will correspond to disjoint subregions of the search space; alternatively they can share some information, or even exchange chromosomes [88], a process which some of its variants

take the name of *island model* [215]. Ultimately, niches could even represent completely different facets of the problem solutions; they might or might not share the fitness function [187], and so on.

3.4 Reinforcement Learning

Broadly speaking, *reinforcement learning* is a term which encompasses a family of problems for which the search for solutions are based on the *trial and error* or *learning from experience* paradigms [152, 199]. The computational techniques used in order to solve reinforcement learning problems generally rely on sequential decisional making processes [199]. These processes are usually related to the notion of *rational agent*, that is an entity, situated in an *environment*, which perceives it or *observes* it through *sensors* and performs actions through *actuators*. In a reinforcement learning problem, the agent receives a quantitative feedback, or *reward*, based on the goodness of the action performed in a particular environment-agent configuration, also called *state*. This would allow the agent to *try* different actions given observations and record its success or failure (*error*); through the repeated observation of the same (or similar) states (*experience*), the agent would ultimately find the state-action mapping which would lead to the best global outcome. Depending on the problem at hand, and the related knowledge in possess to the agent, the finding process could be seen as *planning* or *learning*.

More specifically, given a *state transition function*, which defines successive states after each agent's actions, and a *goal state*, which defines the envisaged outcome of the reinforcement learning problem, the duty of the agent is to *learn* a function mapping (current) states/observations to (current) actions, so that the goal state can be sequentially reached. This function, called *policy*, is modified through the interactions with the environment; the best policy for a given problem, which might be learnt or not, is called *optimal policy*. An agent which performs actions based a policy is also said to have a *behaviour*.

The rational agent needs to perform actions by taking into account both *past* experience — so that it can avoid the visit of already encountered states — and *future* experience, so that it can reach the goal state. In order to take care of past experience, the agent might need to maintain the whole history of visited states and actions performed, which might result problematic if not intractable. However, if the problem's state transition function is proven to have the Markov property, then the agent can perform actions solely based on the current state, in which case we will have a *reactive* or *memoryless* agent. A reinforcement learning problem with Markov property is also called Markov decision process. In order to take care of future experience, so that the goal state can be reached, the agent has to necessarily rely on the observations of the environment and, especially, the feedback of its performed actions. More specifically, the agent assumes that there exists a *reward function* which is defined so that a relational preference over states can be defined. Therefore, a simple *greedy* policy which, at a certain *timestep* and a given state, choses the action which maximise the *discounted accumulated future rewards* would lead to the goal state. The future rewards need to be discounted, so that their accumulation has a limit and will avoid unwanted behaviours in which the agent enters an infinite loop among the same states.

3.4.1 Value Function and Update Rules

The mapping between discounted cumulated future rewards and policy can be made through a *state value function* V , which allows for the quantitative representation of the preference over states, or an *action-state value function* Q , which allows for the quantitative representation of the preference over action-state pairs³. Through the learning from experience paradigm, at each *timestep* t , the rational agent will perform actions a in state s , receive reward r , observe the next state s' , and hence update the corresponding $V(s)$ or $Q(s, a)$. The iterative process can be *episodic*, meaning that an agent will have to solve a set of problem instances (*episodes*), and in each episode the agent will refine its value function. Ultimately, the value function will converge and, possibly, the *optimal greedy* policy will be found.

³The terms V and Q used in reinforcement learning should not be confused with the corresponding terms used in network theory.

Depending on the amount of knowledge in possess to the agent, the value function update can be done in many different ways. Three are the main approaches: those based on dynamic programming, *Monte Carlo* methods, and a combination of the two, called *temporal difference* learning, which arguably constitutes the most important approach for solving reinforcement learning problems.

In a nutshell, dynamic programming methods are used when the agent has a full knowledge of the *world*, composed of the environment, the state transition function, and the reward function. These methods allow for incremental updates of the value function each time the immediate reward, after performing a certain action in a certain state, is observed (*bootstrapping*). Monte Carlo methods do not require full knowledge about the world and instead can *learn* from experience by generating (*sampling*) actions, which define the episodes, and subsequently update the value function of the whole visited state chain or action-state combination chain. Finally, temporal difference, which is a combination of the two previous methods: it allows an agent to learn from experience — similarly Monte Carlo methods — and perform immediate updates of the value function — similarly to dynamic programming methods. The discussion of their similarities and differences goes beyond the scope of our dissertation. However, we here highlight two update rules which are also considered in our study.

The first one, called constant- α Monte Carlo, is defined as follows:

$$V(s) \leftarrow V(s) + \alpha [r - V(s)] \tag{3.11}$$

where r is the immediate reward obtained after performing action a in state s , and $0 < \alpha < 1$ is a constant step-size parameter dictating the incremental update of $V(s)$.

The other update rule considered, called TD(0), is the simplest temporal difference method and is defined as follows:

$$V(s) \leftarrow V(s) + \alpha [r + \gamma V(s') - V(s)] \tag{3.12}$$

where s' is the state encountered by the agent after performing action a in state s , and γ is a discount factor of the next state's value function.

3.4.2 The n -armed Bandit Problem

One of the main difficulties a rational agent faces when attempting to solve a reinforcement learning problem, especially when it does not have full knowledge about the world, is the balancing between the *exploration* of the state space, in order to eventually find more preferred states which would lead it to the goal state, and the *exploitation* of already visited configurations, for which an estimate of the discounted cumulated future reward is known.

An example of this dilemma, and the search for a tradeoff between exploration and exploitation, is the *n -armed bandit problem*. An individual (rational agent) faces n slot machines and can play only one at a time. The action of the agent corresponds to the lever/bandit selection, and the rewards are the payoffs for hitting the jackpot. Through repeated plays the agent maximises its winnings by selecting the best levers.

The agent might chose to exploit its current knowledge, and hence greedily play the machine which has returned, on average, the highest payoff, or explore, by choosing a random machine, for which the estimated average payoff is not clearly known.

Many techniques and approaches have been investigated for this class of problems. In this dissertation we will consider two algorithms, namely ϵ -greedy and UCB1 [12], which pseudocode can be found in Algorithms 17 and 18 respectively⁴. In a nutshell, the ϵ -greedy algorithm is centred on the exploration probability $0 < \epsilon \leq 1$, which is function of the elapsed timesteps. At each timesteps, the exploration phase will occur with probability ϵ , which means that a random lever would be uniformly selected. Intuitively, the exploitation — activation of the most successful lever — will occur with probability $1 - \epsilon$. On the other hand, UCB1 is a deterministic algorithm, which allows for the triggering of levers based both on their past amount of activations and their successfulness.

⁴Although algorithms 17 and 18 deal with niches rather than slot machines, their pseudocode definition is invariant.

3.5 Summary

This Chapter presented the most essential concepts, theories and techniques which are needed to better grasp the computational aspects of our interaction-based group modelling framework. Section 3.1 introduced the main notions of network theory, whilst the community structure detection task is outlined in Section 3.2. This task will be implemented, in our group modelling framework, by means of evolutionary computation, which main computational features were introduced in Section 3.3. As we will show later in Chapters 5 and 7, our modelling framework relies on techniques inspired by reinforcement learning, which was briefly introduced in Section 3.4.

The next Chapter describes, in details, the social synthetic environment benchmarks used for the validation of our interaction-based approach to group modelling.

Chapter 4

Benchmark Social Synthetic Environments

This Chapter presents the scenarios used to empirically evaluate our interaction-based group modelling framework. Broadly speaking, the ideal scenario upon which we can test the validity of our approach is one in which a society of individuals, belonging to non-overlapping groups, interact with each other within computer-based environments. For simplicity, We will refer to these scenarios as social synthetic environments.

Before delving into the description of the social synthetic environments we considered in our dissertation, an overall discourse on the requirements these should fulfil in order to better evaluate our approach, their scarcity amongst research studies related to ours, and the decisions we took to filter them out, should be given, in order to establish the benchmark scenarios for future work, and possibly identify other existing scenarios which we had unwillingly missed.

Three are the requirements a social synthetic environment should fulfil in order to be used as testbed for the empirical evaluation of our group modelling framework framework:

- the group structures should affect the behaviour of the individuals, that is, the individuals should have a *group identity* (group identity requirement);
- it must be possible to retrieve the existing *true* group structures in the society, since our aim is to *unveil* them from the observation of the interactions (ground truth requirement);
- an individual of the society should interact, through the same protocol, with both individuals belonging to her own group (*in-group*) and individuals belonging to other groups (*out-group*), though not necessarily at the same time (interaction requirement);

Let us use Figure 4.1 as a reference of domain of application of our theories. These items were retrieved from the introductory discourses made in Chapter 1. With respect to the “Cooperative Computer Games” box, the three requirements would necessarily exclude the use of well-established massively multiplayer online games/social environments. For instance, *World of Warcraft* (Blizzard Entertainment, 2004), *Second Life* (Linded Research, 2003) or *Minecraft* (Mojang, 2001), which are essentially open virtual worlds in which human-controlled avatars interact with each other without a *strict* purpose [47], cannot be considered, mainly due to the lack of the ground truth requirement. Similarly, smaller-scaled collaborative computer games, explicitly designed to instigate *emergent* group behaviours, such as *Village Voices* [107], or the open environment designed by Vogiazou and Eisenstadt [213], cannot be used, since they still mainly lack of the ground truth requirement. Multiplayer sport games, such as *FIFA* (EA Sports 1993) or *Pro Evolution Soccer* (Konami 2001) franchises, or even first person shooter titles, such as *Call of Duty* (Activision 2003) and *Counter Strike* (Valve, 1999) franchises, essentially organise players into teams with the only aim to defeat opponent ones. Therefore, these category of games, which mainly lack of the interaction requirement, cannot

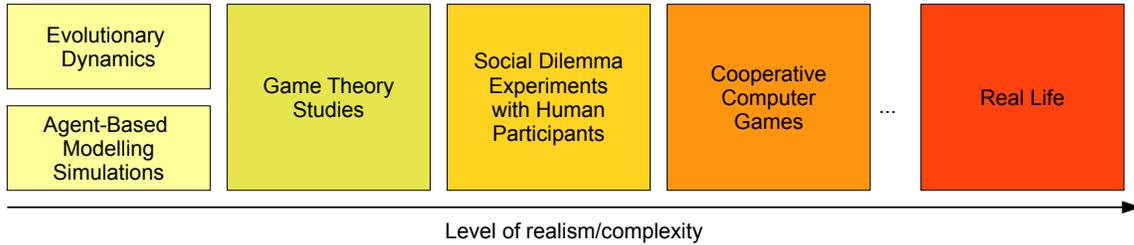


Figure 4.1: A possible classification of social synthetic environments, based on their level of *realism*, that can implement in order to reproduce, simulate, or explain, real-life phenomena.

Table 4.1: A recap of the six sessions of the experiment by Chen and Li and the number of repetitions of them (with different participants).

Treatment	Stage				Number of Repetitions
	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>	
<i>original</i>	painting	yes	yes	yes	15
<i>random within</i>	random	yes	yes	yes	2
<i>random between other</i>	random	yes	yes	yes	2
<i>random between same</i>	random	yes	yes	yes	2
<i>no chat</i>	painting	no	yes	yes	4
<i>no help</i>	painting	no	no	yes	2
<i>control</i>	no	no	no	yes	9

be considered for the empirical evaluation of our interaction-based approach. Therefore, with respect to Figure 4.1, we had to take a step back and look at the “Social Dilemma Experiments with Human Participants” category.

Although, as we have described in the previous Chapters 1 and 2, there exist a huge corpus of experiments focused on social preferences and inequity aversion, we had to discard the vast majority of them, since they did not focus on group behaviours and identities, that is, they lacked of all three requirements. Nevertheless, among these, two of them provided a severe contribution to our work.

Marzo et al.’s experiment [147] can be seen as the crossroad between game theory experiments and cooperative computer games. In their work, they investigate social preferences among individuals which categorise each other as either friends or strangers. They relied on *Colored Trails* [73], a computer game upon which the interactions were resembling the ultimatum game. As a matter of fact, we heavily relied on their findings, in our previous work [91, 83, 87], to build artificial societies of *believable* agents. Nevertheless, for the investigations of this dissertation, we could not rely on their empirical data, due to the lack of the interaction requirement.

4.1 The Experiment by Chen and Li

The second work was conducted by Chen and Li [30], who aimed to investigate the effects of induced group identities on social preferences, such as distribution and reciprocity preferences. They made use of a wide range of game theory games [26] which were played by the participants belonging to groups, minimally constructed in several ways. Their work is the one which most fits our three requirements, and, as a consequence, has been used as the benchmark scenario for the empirical evaluations of our thesis. The procedure we adopted to extract the most relevant data is presented in Section 4.2. Additionally, we decided to further exploit their empirical data; the result is the implementation of artificial societies which is described in Section 4.3.

The experimental sessions were divided into five treatments and one control session. In the treatment

sessions, the experiments were divided into four stages: group assignment ($S1$); collective problem solving via online chat ($S2$); *other-other* allocation game, where each participant allocates tokens to two other participants ($S3$); and finally a set of two-person sequential games ($S4$). The subject of the control session, instead, participated only in the fourth stage.

Table 4.1 summarises the different features of the six sessions covered by the experiment. Further details of each single feature are given to their relative sections.

4.1.1 Stage $S1$: Group Assignment

In accordance with classical social psychology experiments [203], Chen and Li relied on the use of artwork to induce group identities. In the *original* treatment subjects reviewed five pairs of painting by two modern artists, Klee and Kandinsky, with one painting within each pair by Klee, and the other one by Kandinsky. The participant provided pairwise preference without being told the artist of each painting. Based on the reported painting preferences, subject were divided into two groups, the Klee group and the Kandinsky group. Subjects were privately informed about their group identity and the number of people in their group. The groups remained the same throughout the experiment.

Subjects of the *random within* and *random between* treatments were assigned to two groups randomly, by allowing the participants to select sealed envelopes, from a stack, which were containing either a maize or blue slip.

At the end of the first stage, after being categorised into two groups, subjects of the *original* treatment were given the answer key to the artists. Participants of the *random within* and *random between* treatments were given the five pairs of paintings along with the answer keys; they had five minutes to study the paintings and prepare them for the second stage.

4.1.2 Stage $S2$: Online Chat

Subjects of the *original*, *random within* and *random between* treatments subsequently participated in a second task that involved group communication via a chat program. The task was to answer two questions on which artist made each of two additional paintings. The participants voluntarily exchanged information, exclusively with *in-group* members, to help one another obtain the correct answers. The subjects were free to submit answers individually after the chat.

In order to test the impact of the chat component on behaviour, the experimenters introduced a new treatment, *no chat*, in which the subjects underwent the same experimental procedure as the *original* treatment, except, intuitively, from this stage $S2$, which they did not take part into.

4.1.3 Stage $S3$: Other-other Allocation Game

In this stage of the *original*, *random within*, *random between* and *no chat* treatments, every subject was asked to allocate a given number of tokens e between two other anonymous participants. No one was allowed to allocate tokens for herself. Figure 4.2 depicts the schematic representation of an occurrence of the other-other game. The stage had five rounds, in which the total number of tokens increased from 200 to 400 with an increment of 50 tokens in each round. During each round, every subject decided how to allocate tokens under three cases: if both receivers came from her own group (*in-in*), if both came from the other group (*out-out*), and if one came from her own group and the other from a different group (*in-out*).

At the end of stage $S2$, a random sequence of ID numbers was generated by the computer to decide the provider-receivers triplets of the other-other game (P, R_1, R_2). Everyone allocated tokens between the two participants whose IDs directly followed hers in the sequence. Figure 4.3 depicts the schematic representation of a sequence of six participants and the six triples based on which the other-other games are played.

In order to understand the effects of other-other allocation games on the strength of group identity, the experimenters added a *no help* treatment, were both stages $S2$ and $S3$ were taken out.

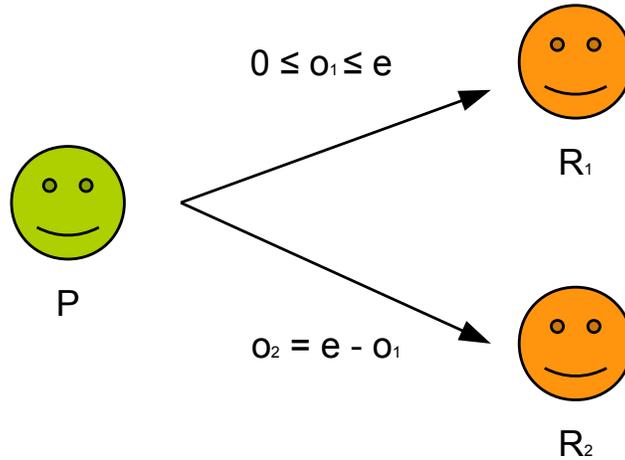


Figure 4.2: The schematic representation of the other-other game used by Chen and Li in their experiment [30]. A proposer P receives an endowment e . She now has to split it between two receivers, R_1 and R_2 , and cannot keep anything for herself. o_1 and o_2 are the splits for R_1 and R_2 respectively.

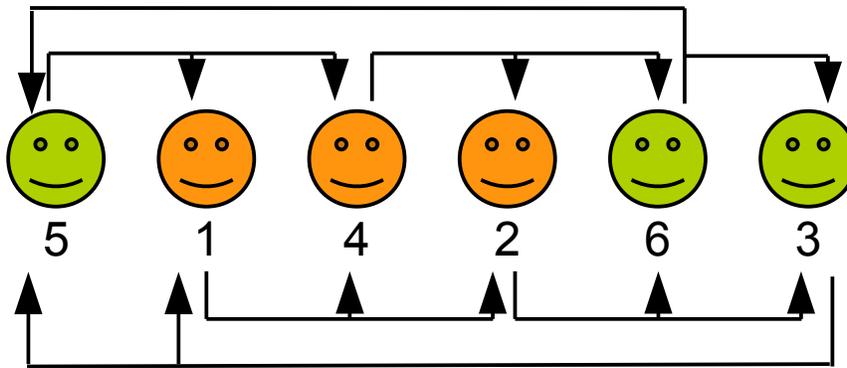


Figure 4.3: An example of how the other-other game triplets (P, R_1, R_2) are generated, based on a society of six participants. Individuals 1, 2 and 4 belong to the orange group, individuals 3, 5 and 6 to the green group. A random permutation is generated. Individual 5 will be the proposer for individuals 1 and 4, individual 1 will be the proposer for individuals 4 and 2, and so on. Individuals 5 and 2 will face an *out-out* interaction type; individuals 1 and 6 an *in-in* interaction type; individuals 3 and 4 will face an *in-out* interaction type.

4.1.4 Stage S_4 : Sequential Game

In this stage, subjects of all the five treatments and the control group made decisions in a series of two-person sequential move games selected from Charness and Rabin [26], as well as an extension of some of the games. Stage S_4 was aimed to investigate the impact of group identity on social preferences and economic outcomes, whilst the first three stages were designed to induce and enhance group identity.

Participants of the *random within* treatment made decisions for both *in-group* and *out-group*; participants of the *random between* treatment, instead, in two of the four sessions of this stage, were matched with only *in-group* members and made one (*in-group*) decision in each game. Similarly, in the remaining two sessions, they were matched only with *out-group* members and made one (*out-group*) decision in each game.

4.2 Chen and Li-based Simulation

Chen and Li conducted experiments based on societies of 16 individuals divided into two groups (for the treatment sessions only) of either equal size, for the two *random within* and *random between* treatments, or arbitrary size, for the other three treatments. They conducted 36 independent experimental sessions, in order to conduct their statistical analysis, as it is summarised in Table 4.1.

Their work and dataset is extremely relevant for our empirical evaluations. However, in order to rely on behavioural data that could fulfil all three requirements presented at the beginning of this Chapter, we had to focus solely on a portion of the whole dataset, and hence ignore some of its entries. In particular:

- the data related to the *control* session cannot be used due to the lack of the group identity requirement;
- the data related to the *no help* treatment cannot be used to the lack of the interaction requirement;
- the data of stage *S4* only partly satisfies the interaction requirement, since the participants did not play all the possible roles of all possible allocation games considered in their study;

We will refer to the modified version of the authentic experiment as Chen-Li (CL) simulation. The CL simulation is hence composed of:

- 25 experimental sessions of the *original*, *random within*, *random between other*, *random between same*, and finally *no chat* treatments;
- the behavioural data will be the one recorded for stage *S3* only, that is, the five rounds of the other-other game, across three interaction types, and for the five endowment e amounts;
- the triads (P, R_1, R_2) will be simulated by generating uniformly random sequences of individual IDs;

The full extracted behaviours of the 25 CL simulations are listed in Appendix 12. As it can be noted, in the *original* treatment, one session considers 15 individuals (see Table 12.1) and another one 14 individuals (see Table 12.4), whilst the remaining 23 sessions consider 16 individuals. This partial information is also present in the original behavioural data and was due to the fact that, in those sessions, there were some individuals who had already taken part into the experiment. Chen and Li decided to discard the repeated data, which was hence impossible to retrieve. In order not to introduce behavioural artefacts, therefore, we decided to consider, for those three simulations, a smaller population size.

4.3 Chen and Li Experiment’s Derived Artificial Society

Despite being the only reliable social synthetic environment which fulfils the requirements for the validation of our interaction-based approach, the CL simulations are nevertheless rather simple, both in terms of society size (16 individuals), number of groups (two), and possibly nuances of group behaviours, or at least for some of the treatments considered, see the details in Appendix 12.

For these three reasons, we decided to expand the CL simulation by means of Artificial Societies (CLAS), for which the agent behaviours are extracted from the empirical data of the authentic experiment. As a matter of fact, the CLAS will constitute the core benchmark scenarios upon which we will empirically evaluate our interaction-based approach. Hence, the study based on CL simulations will be presented and considered as a secondary/consequent result.

At the beginning of each CLAS simulation, a society \mathcal{S} of n agents $\{a_1, a_2, \dots, a_n\}$ are instantiated; their behaviours will be sampled, uniformly, among the five treatments of the CL simulations (i.e. excluding the *no help* treatment and *control* sessions), and will not change throughout the experiment. Given m , the predefined number of groups in \mathcal{S} , each agent a_i is then initialised with a *true* group identity g_i sampled within $[1, m]$. Its group identity will not change throughout the experimental run, meaning thus that the group structures in \mathcal{S} are *static*. This condition also reflects the experimental setup of the authentic experiment by Chen and

Algorithm 3 CLAS - other-other game iteration

```
1: generate a permutation  $perm = \{i, \dots, k\}$ ,  $|perm| = n$ , of indices corresponding to the individuals  $\mathcal{S} = \{a_1, a_2, \dots, a_n\}$ 
2: for  $j \in [1, n]$  do
3:    $e \leftarrow 200 + 50 * U[0, 4]$ 
4:    $P \leftarrow perm[j]$ 
5:    $R_1 \leftarrow perm[(j + 1) \bmod n]$ 
6:    $R_2 \leftarrow perm[(j + 2) \bmod n]$ 
7:   if  $(is\_in\_group(P, R_1) = false) \ \& \ (is\_in\_group(P, R_2) = true)$  then
8:      $[o_2, o_1] \leftarrow P \rightarrow \text{playother} - \text{othergame}(e, R_2, R_1)$ 
9:   else
10:     $[o_1, o_2] \leftarrow P \rightarrow \text{playother} - \text{othergame}(e, R_1, R_2)$ 
11:   end if
12: end for
```

Algorithm 4 CLAS agent - play other-other game

```
1: #  $R_1$  and  $R_2$  are ordered so that out-in interactions are not possible
2:  $interactionType \leftarrow in-in$ 
3: if  $(is\_in\_group(P, R_1) = false)$  then
4:    $interactionType \leftarrow out-out$ 
5: else
6:   if  $(is\_in\_group(P, R_2) = false)$  then
7:      $interactionType \leftarrow in-out$ 
8:   end if
9: end if
10: retrieve  $\mu, sd$  given the interaction type and  $e$ 
11:  $o_1 \leftarrow gaussian(\mu, sd)$ 
12:  $o_1 \leftarrow \min(0, o_1)$ 
13:  $o_1 \leftarrow \max(o_1, e)$ 
14:  $o_2 \leftarrow e - o_1$ 
15: return  $[o_1, o_2]$ 
```

Li, which induced the group identities in stage $S1$ and maintained them unaltered until the end of stage $S4$. We will refer to the ordered sequence of *true* group identities g_i as $\mathcal{T} = \{g_1, g_2, \dots, g_n\}$.

In order to add noise in the observed interactions, the agents will receive endowments which will be uniformly sampled within the five available (unlike the CL simulations which are presented in a loop fashion). Given the triad (a_p, a_i, a_j) , proposer agent a_p will stochastically split e into o_i (for a_i) and o_j (for a_j) according to normal distributions. More specifically, from the CL simulation data (see Appendix 12) we extracted sample average (avg) and sample standard deviation (sd) for each combination of interaction type (3), magnitude of the endowments e (5), and treatment considered (5), that is $3 * 5 * 5 = 75$ in total. Table 4.3 reports, in details, their values, used to generate the offer o_i for the first receiver individual a_i (the split for the second receiver a_j can be trivially calculated once o_i is sampled, as $o_j = e - o_i$). Note that, similarly to the original experiment, in case of *in-out* interactions, the receivers are ordered so that the first one corresponds to the *in-group* individual and the second one to the *out-group* individual.

The pseudocode of an interaction of the other-other game performed by a CLAS is presented in Algorithm 3. Algorithm 4, in addition, presents the pseudocode implemented in each artificial agent of the society, which defines its behaviour for determining the o_i and o_j values of the other-other games played.

All the artificial societies we have considered have a population of $n = 128$ agents. This decision was taken so that it could be easier to bridge our research and related results with those obtained in network theory, focused on community structure detection, based on synthetic networks [161]. Depending on the group

Table 4.2: Sample averages and standard deviations extracted from Chen and Li experiments, divided by treatment, other-other game round, and interaction type based on the triad's group identities. The allocations regard the first receiver only, as the average allocation for the second receiver will be obtained by subtracting the endowment e minus the reported average. Clearly, the standard deviations will be the same among the two receivers.

in-in interactions											
Treatment	N	round 1 ($e = 200$)		round 2 ($e = 250$)		round 3 ($e = 300$)		round 4 ($e = 350$)		round 5 ($e = 400$)	
		avg	sd								
<i>original</i>	237	100.29	22.49	128.89	25.63	155.48	28.42	183.32	44.87	204.85	40.12
<i>random within</i>	32	98.28	26.29	125.62	26.57	157.66	39.82	185.16	38.02	206.53	29.19
<i>random between others</i>	32	102.84	17.43	134.06	24.41	148.22	26.44	183.44	31.63	196.72	37.36
<i>random between same</i>	32	100	0	133.44	26.71	146.87	28.22	188.28	41.63	207.81	36.16
<i>no chat</i>	64	101.42	18.05	135.09	33.15	154.61	28.91	179.30	48.78	205.48	52.60

out-out interactions											
Treatment	N	round 1 ($e = 200$)		round 2 ($e = 250$)		round 3 ($e = 300$)		round 4 ($e = 350$)		round 5 ($e = 400$)	
		avg	sd								
<i>original</i>	237	95.95	30.96	123.98	38.54	146.26	50.71	175.67	52.95	204.80	50.62
<i>random within</i>	32	97.5	44.43	119.53	39.01	144.91	55.75	187.37	44.87	205.47	45.67
<i>random between others</i>	32	98.69	29.46	126.72	30.92	148.75	27.09	172.66	45.17	202.69	47.72
<i>random between same</i>	32	89.16	39.41	118.19	47.91	146.62	45.28	179.69	54.04	205.47	53.03
<i>no chat</i>	64	96.91	31.02	132.12	37.47	155.45	55.47	183.26	61.34	195.47	57.48

in-out interactions											
Treatment	N	round 1 ($e = 200$)		round 2 ($e = 250$)		round 3 ($e = 300$)		round 4 ($e = 350$)		round 5 ($e = 400$)	
		avg	sd								
<i>original</i>	237	137.51	48.38	172.95	56.29	198.26	74.57	238.97	83.63	266.24	98.69
<i>random within</i>	32	143.56	50.89	184.37	63.42	214.06	75.12	238	92.84	278.91	102.8
<i>random between others</i>	32	140	48.66	173.25	58.55	212	62.38	245.09	70.8	272.91	79.03
<i>random between same</i>	32	155.44	40.99	183.28	59.93	232.66	64.11	267.03	72.34	298.12	81.42
<i>no chat</i>	64	133.8	46.11	167.75	61.25	203.17	69.83	230.61	80.26	253.3	90.76

Algorithm 5 Preferential Attachment for Group Identity Assignment

```
1: #  $|\mathcal{S}| = n$ ,  $m =$  number of groups in  $\mathcal{S}$ 
2: #  $\mathcal{T}$ ,  $|\mathcal{T}| = n$  holds the true group identities
3: #  $c$  and  $p$  are working arrays of size  $m$  elements initialised to zero
4:  $k \leftarrow 1$ 
5: for  $i \leftarrow 1$ ;  $i \leq 2m$ ;  $i \leftarrow i + 2$  do
6:    $\mathcal{T}[i] \leftarrow k$ 
7:    $\mathcal{T}[i + 1] \leftarrow k$ 
8:    $c[k] \leftarrow c[k] + 2$ 
9:    $k \leftarrow k + 1$ 
10: end for
11: Update Preferential Attachment ( $c$ ,  $p$ )
12: for  $i \leftarrow 2m + 1$ ;  $i < n$   $i \leftarrow i + 1$  do
13:   sample  $r$  uniformly  $\in [0, 1)$ 
14:    $k \leftarrow 0$ 
15:   while  $r \geq 0$  do
16:      $k \leftarrow k + 1$ 
17:      $r \leftarrow r - c[k]$ 
18:   end while
19:    $\mathcal{T}[i] \leftarrow k$ 
20:    $c[k] \leftarrow c[k] + 1$ 
21:   Update Preferential Attachment ( $c$ ,  $p$ )
22: end for return  $\mathcal{T}$ 
```

Algorithm 6 Update Preferential Attachment

```
1: for  $i \leftarrow 1$ ;  $i \leq m$ ;  $i \leftarrow i + 1$  do
2:    $p[i] \leftarrow \frac{c[i]}{\sum_{j=1}^m c[j]}$ 
3: end for
```

modelling component we are focusing on, the number m of groups will variate: $m = 2$ for the analysis of the cooperation modelling component (see Chapter 6), whilst, for the analysis of the group identity detection component, we will consider $m = \{4, 7, 11, 17\}$ groups. We also define three methods to instantiate the group sizes. For $m = 2$ we will chose their sizes arbitrarily (their motivations are presented in Section 6.3). For the GID evaluation phase, instead, we will consider both groups of equal size (with ± 1 agent difference for $m = \{7, 11, 17\}$), or group sizes following a power law distribution (for $m = \{7, 11, 17\}$ only).

The power law group size distributions are built by following the principles of preferential attachment [14], which pseudocode is presented in Algorithms 5 and 6. As it could be easily understood, in lines 6-11 of Algorithm 5, we added the constraint which imposed the groups to have a minimum size of two agents, so that the interaction requirement, presented at the beginning of this Chapter, is satisfied.

4.4 Summary

This Chapter described, in details, the social synthetic environments used as benchmark scenarios for the empirical investigation of our interaction-based group modelling framework. We highlighted the main problems we faced with respect to their seek, and then introduced the experiment conducted by Chen and Li [30] (Section 4.1), originally aimed to investigate group identities and group behaviours in social dilemma experiments. We also described how and why we decided to utilise the authentic behavioural data, in order to obtain both simulations of the original human-based experiment (Section 4.2), and derive artificial societies of stochastic agents (Section 4.3).

The next Chapter will delve into the computational definition of the first module of our group modelling

framework, namely the cooperation modelling component.

Chapter 5

Interaction-based Cooperation Modelling

Cooperation Modelling (CM) is the first module of our proposed interaction-based group modelling framework, see Figure 1.2. Its duty is dual: (1) evaluate the levels of cooperation of the interactions occurring among individuals of a social synthetic environment, and (2) maintain an accurate representation of the ongoing society’s cooperation network.

Section 5.1 describes the *one-to-many* perspective used by CM in order to process the interactions; Section 5.2 introduces possible strategies for cooperation evaluation, whilst Section 5.3 describes possible cooperation network update rules. The empirical evaluation of CM is instead found in Chapter 6.

5.1 The *One-to-many* Approach

Initially, the module needs to evaluate the *raw* interactions in terms of cooperation. Cooperation (or altruism) can be interpreted as the degree to which an individual attempts to satisfy others’ concerns [32, 205]. In addition, we know that individuals belonging to the same group (*in-group*) tend to cooperate more with each other than when they interact with individuals belonging to other groups (*out-group*) [39].

A further way to analyse the *in-group* vs. *out-group* (*in/out-group*) dichotomy of interactions is by observing the individuals’ *fairness* of treatment [86]. The *Temporal Group-based Fairness* metrics (*TGB*) [86] allows to quantitatively measure how fairly or equally a proposer individual has allocated resources among a society of receiver individuals, which belong to two distinct groups. *TGB* is defined as follows:

$$TGB = \begin{cases} 0 & \text{if } |D| = 0 \\ 1 - \frac{1}{|D|} \sum_{t=1}^{|D|} |\mu_A^t - \mu_B^t| & \text{otherwise} \end{cases} \quad (5.1)$$

where $|D|$ is the total number of resource distribution *phases*, t represents the single distribution phase, A and B are the two group labels/identities existing in the society, $|A|, |B| > 0$, whilst $\mu_A(\mu_B)$ represent an estimate of the amount of resources acquired by group $A(B)$ [86]. *TGB* requires that the value of the resources being distributed are normalised to the $[0, 1]$ interval; as a consequence, the μ values will also be normalised; therefore, *TGB* ranges between 0 and 1. *TGB* is maximised (absolute fairness) when $\mu_A = \mu_B$, whilst it is minimised (absolute unfairness) when either $\mu_A = 0$, $\mu_B = 0$, or $|D| = 0$.

The length of t , which then defines $|D|$, is adjustable. For instance, it could correspond to each single resource allocation, a variable length of time encompassing a fixed number of resource distributions, a fixed length of time encompassing a variable number of allocations, a unique segment of time which would encompass all the resource distributions, and so on.

For a resource distribution task, given a society \mathcal{S} of n individuals $\{a_1, a_2, \dots, a_n\}$, within which a proposer a_P is defined, and her k receiver individuals are partitioned into two groups, four possible group-based interactions can be identified, as listed in Table 5.1: a_P can belong to either A, B (cases 3 and 4), or another group C (cases 1 and 2). *TGB* cannot be applied neither in case 1 nor in case 3, since it requires two distinct groups of receiver individuals. Moreover, since we aim to analyse the levels of fairness of treatment under

Table 5.1: The possible interaction cases between a provider a_P and a society of individuals divided into two or more groups. The table lists the possible applications of TGB [86].

Case	a_P 's Group Identity	A and B	Interaction	TGB Applicable
1	C	$A \equiv B$	<i>out-out</i>	no
2	C	$A \cap B = \emptyset$	<i>out-out</i>	yes
3	A or B	$A \equiv B$	<i>in-in</i>	no
4	A or B	$A \cap B = \emptyset$	<i>in-out</i>	yes

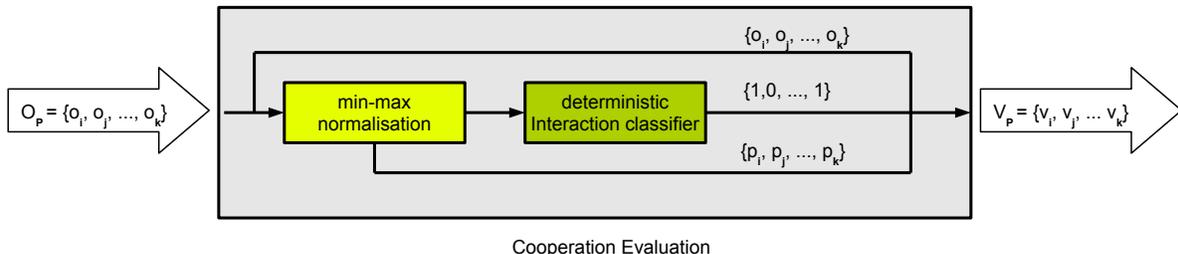


Figure 5.1: The schematic representation of the cooperation evaluation task and the three possible ways considered in our dissertation: raw interactions (that is no cooperation evaluation); min-max normalisation of the interactions into continuous probabilities of *in-group*-ness; discretisation of the interactions into *in-group* and *out-group* labels/values.

the *in/out-group* dichotomy, for case 2, a_P would perceive A and B as being both *out-group*, hence, TGB cannot be applied. In conclusion, under the *in/out-group* dichotomy, we can apply TGB and measure the level of fairness of treatment only in presence of *in-out* interactions.

Let us now focus on case 4, and let us assume $a_P \in A$. As a consequence, we would expect to have a_P being more altruistic or cooperative towards A 's individuals rather than B 's [39]. By applying TGB , hence, we would generally observe unfair treatments, that is $TGB < 1$, which further implies that $\mu_A^t > \mu_B^t$.

On the other hand, for case 3, we could expect to observe behaviours which tend to yield to absolute fairness, since a_P should not, in principle, hold any form of favouritism among her *in-group* individuals (we do not assume the existence of neither subgroups nor overlapping groups). However, for cases 1 and 2, we cannot make any a priori assumption about a_P 's behaviours.

Therefore, in order to detect the boundaries between *in-group* and *out-group* individuals, we must be able to observe *in-out* interactions, since, by means of TGB , the relationship $\mu_A \leq \mu_B$ can help us identifying where a_P 's preference goes to. However, depending on the interaction protocol under investigation, it might happen that the conditions for the application of TGB are not met. A way to overcome this problem is the enlargement of the temporal window t . As a consequence, TGB would be applied to interactions between *one* individual towards *many* others, which have not necessarily occurred at the same time. We will refer to the gathering of interactions through this temporal enlargement as *one-to-many* approach. Clearly, in order to apply TGB , the gathered interactions must be of the same *type*.

5.2 Cooperation Evaluation

TGB requires knowledge about the groups existing among the society; however, since that knowledge constitutes the goal of our research, we cannot apply TGB to define *in-group* and *out-group* boundaries: for instance, if we aimed to partition the receivers to two distinct groups, so that TGB is minimised, we would end up with one group composed of the individual which received the smallest amount of resources, whilst all the remaining receivers would be assigned to the other group. Alternatively, we could put constraints on the group sizes, though this could potentially lead to erroneous group-based fairness estimations.

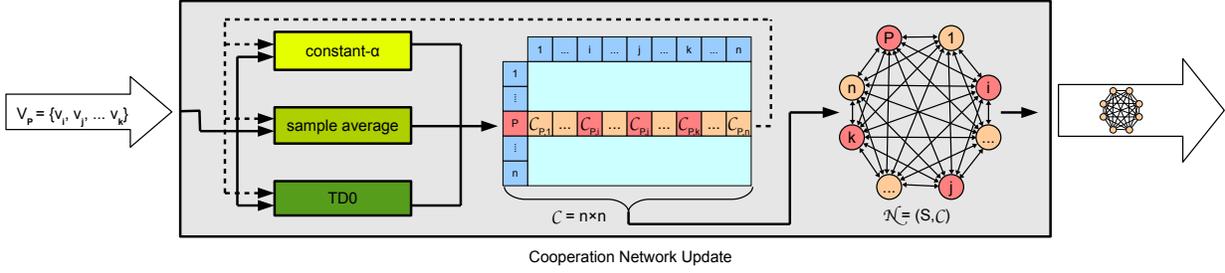


Figure 5.2: The schematic representation of the cooperation network update task and the three possible update rules considered in our study. The weights of the edges, together with cooperation values $\{C_{i,P}, C_{j,P}, \dots, C_{k,P}\}$ used by *TD0* update rule, were omitted in order not to overcomplicate the Figure.

Nevertheless, we can leverage on the fact that unfair behaviours are indications of *in-group* favouritism, and represent, probabilistically, the uncertainty that a receiver individual a_i might be either *in-group* or *out-group* with a provider individual a_P . Moreover, thanks to the *one-to-many* approach, we could enlarge the scope of the probability estimation beyond the single proposer-receiver interaction, up to whole society, for instance, when $t = |D| = 1$.

More formally, given the set of resource distributions $O_P = \{o_i, o_j, \dots, o_k\}$, made by a_P within the same distribution phase t , the simple scaling of the o_i values to the $[0, 1]$ interval via min-max normalisation:

$$p_i = \frac{o_i - \min(O_P)}{\max(O_P) - \min(O_P)} \quad (5.2)$$

allows us to transform/interpret the *raw* interactions into probability values. The single p_i values would hence represent the probability that receiver a_i is *in-group* with provider a_P , given the current distribution phase t , composed of the O_P resource distributions to other receiver individuals.

Moreover, given a threshold τ , the probability values could now be used to partition the receivers into two distinct groups:

$$d_i = \begin{cases} 0 & p_i < \tau \quad (\text{out-groupness}) \\ 1 & \text{else} \quad (\text{in-groupness}) \end{cases} \quad (5.3)$$

In summary, cooperation evaluation can be considered as the task which gathers *raw* interactions under the *one-to-many* perspective, and outputs either continuous or discrete information about the *in-group* or *out-group* nature/likelihood of the interactions between a proposer individual and many receiver individuals:

$$V(O_P) = \{v_i, v_j, \dots, v_k\}_P = \begin{cases} \{o_i, o_j, \dots, o_k\}_P & o_x \in \mathbb{R} \\ \{p_i, p_j, \dots, p_k\}_P & p_x \in [0, 1] \\ \{d_i, d_j, \dots, d_k\}_P & d_x \in \{0, 1\} \end{cases} \quad (5.4)$$

Equation (5.4) and Figure 5.1 depict the schematic representation of cooperation evaluation. We decided to include also the case in which the *raw* interaction are left unaltered, since this condition will be investigated in Chapter 6.

Cooperation evaluation assumes independence among the set of O_P interactions, that is $t = |D| = 1$. This is convenient not only because the societal dynamics might change during the whole modelling task, but also because there might exist an unknown ordering of the interactions, for instance in case a_P could have full control over her resource allocation task.

5.3 Cooperation Network Update

This task is in charge of enlarging the scope of the cooperation values, *beyond* the interactions gathered through the *one-to-many* perspective, up to the perspective of the society as a whole. In other words, it aims to follow the paradigms of complex systems in order to infer about global patterns. This is achieved through a *cooperation matrix*

$$\mathcal{C} = n \times n \quad (5.5)$$

of the whole society \mathcal{S} , in which value $\mathcal{C}_{i,j}$ represents the up-to-date cooperation value between proposer individual a_i and receiver individual a_j . Vector V_i , output of the Cooperation Evaluation task applied to the interactions of individual a_i , see Equation (5.4), will be used to update the i -th row of the \mathcal{C} matrix. Through \mathcal{C} we can obtain an overall view of the ongoing cooperation across \mathcal{S} ; in fact, an individual a_i , who has not interacted with an individual a_j yet, can still be put in relationship with each other, by interpreting \mathcal{C} 's relative cooperation network

$$\mathcal{N} = (\mathcal{S}, \mathcal{C}) \quad (5.6)$$

where, intuitively, the individuals $a_i \in \mathcal{S}$ represent the nodes of the network, and \mathcal{C} represents the set of edges/adjacency matrix of the cooperation network. \mathcal{N} is directed — since $\mathcal{C}_{i,j}$ and $\mathcal{C}_{j,i}$ represent different proposer-receiver interaction roles — possibly weighted — depending on the incoming v_i values and on the cooperation network update rule in use — and potentially even fully connected (complete) — depending on whom, during the resource allocation/cooperation modelling task, each individual in \mathcal{S} interacts with.

We hereafter present the three possible cooperation network update rules which were considered in the dissertation.

5.3.1 Cooperation Update via Sample Average

The simplest and most straightforward cooperation update rule is sample average. The update of $\mathcal{C}_{i,j}$ is as follows:

$$\mathcal{C}_{i,j} \leftarrow \frac{u_{i,j} \mathcal{C}_{i,j} + v_{i,j}}{u_{i,j} + 1} \quad (5.7)$$

and, consequently,

$$u_{i,j} \leftarrow u_{i,j} + 1 \quad (5.8)$$

where $v_{i,j}$ is the cooperation value of vector V_i describing the interaction between provider a_i and receiver a_j , and $u_{i,j}$ is a variable keeping the count of how many times a_i has interacted with a_j .

5.3.2 Cooperation Update via Constant- α

Equation (5.7) can be rewritten as follows (see Appendix 13.3 for the details):

$$\mathcal{C}_{i,j} \leftarrow \mathcal{C}_{i,j} + \frac{1}{u_{i,j} + 1} (v_{i,j} - \mathcal{C}_{i,j}) \quad (5.9)$$

When sample average is used, the incremental factor $1/(u_{i,j} + 1)$ of the terms inside the brackets is reduced each time provider a_i interacts with receiver a_j , due to Equation (5.8). An intuitive alternative would be the use of a *constant* incremental factor $0 < \alpha \leq 1$:

$$\mathcal{C}_{i,j} \leftarrow \mathcal{C}_{i,j} + \alpha (v_{i,j} - \mathcal{C}_{i,j}) \quad (5.10)$$

Equation (5.10) corresponds to e.g. the constant- α Monte Carlo update rule for non stationary environments used in reinforcement learning [199], the pheromone update rule used in artificial ant colony

optimisation metaheuristic [42] (see Appendix 13.3 for the details), and the delta update rule of perception training with a constant learning rate [95]. This not only suggests that this update rule might provide a substantial modelling advantage in presence of non-stationary (dynamic) group behaviours, but also avoids to keep the $U = n \times n$ matrix of each single $u_{i,j}$ values, which are required by the sample average update.

5.3.3 Cooperation Update via Temporal Difference

By following the reinforcement learning path introduced by Equation (5.10), another possible update rule is the one used in temporal difference learning — more specifically TD0 [199]:

$$\mathcal{C}_{i,j} \leftarrow \mathcal{C}_{i,j} + \alpha (v_{i,j} + \gamma \mathcal{C}_{j,i} - \mathcal{C}_{i,j}) \quad (5.11)$$

where $0 < \gamma \leq 1$, which is analogous to the discount rate of accumulated future rewards, aims to consider, for the update of $\mathcal{C}_{i,j}$, its reciprocal cooperation value $\mathcal{C}_{j,i}$. Equation (5.11) can be interpreted as a *collaboration* update rule, providing that collaboration is interpreted as reciprocal cooperation [127, 32, 205]. Intuitively, when $\gamma \rightarrow 0$, Equation (5.11) converges to Equation (5.10).

To conclude this Section, Figure 5.2 depicts a schematic representation of the cooperation network update task. As it can be easily understood, this task is independent on the techniques used for cooperation evaluation, and consequently on the nature of the interactions of the society. Semantically, under the complex systems perspective, we can interpret the cooperation network update task as the transition from the society-system’s *local* level, represented by the single interactions among specific individuals, to the system’s *global* level, represented by the society cooperation network, its multiple *in-group* and *out-group* boundaries, and hence its partitioning into group structures. The latter aspect constitutes the final stage of our interaction-based group modelling approach, which is introduced in Chapter 7.

5.4 Summary

This Chapter presented the first module of our interaction-based group modelling framework, called Cooperation Modelling (CM). Given a social synthetic environment, CM observes the ongoing interactions and processes them, in order to build the overall society’s cooperation network.

By assuming that groups exist in the society, and that individuals identify themselves as being part of groups, the necessary condition for the application of the CM component is the observation of interactions between a proposer individual and both individuals belonging to her group (*in-group*), and individuals belonging to other groups (*out-group*) — namely *in-out* interaction types — in accordance with the principles of the temporal group-based metrics of fairness of resource distribution [86]. In order to increase the chances to observe *in-out* interactions, CM adopts the so-called *one-to-many* approach, which consists to gather the interactions one individual performs with many others, which have not necessarily occurred at the same time, though they are still of the same type.

Ultimately, the gathered interactions can be processed (cooperation evaluation) in order to detect, for each individual in the society, her own *in-group* and *out-group* boundaries. When combined, the boundaries will allow for the creation and update of the whole society’s cooperation network. The Chapter presented three possible methods for the evaluation of the cooperation levels of the interactions, and three methods for the update of the society’s cooperation network. The empirical evaluation of the CM configurations is reported in the next Chapter 6.

Chapter 6

Cooperation Modelling Empirical Evaluation

This Chapter presents a thorough analysis of how the different configurations of the Cooperation Modelling module (CM), presented in Chapter 5, would affect the task of community structure detection, which is held by the subsequent group identity detection module, introduced in Chapter 7.

First we will investigate the generic properties of the society structures which might affect the modelling (Section 6.3); then we will isolate the discourse and focus on the preliminary investigation of the parameters α and γ , used by constant- α and TD0 update rules, Equations (5.10) and (5.11); finally, we will present the analysis of the performance of the whole module across a combination of cooperation evaluation configurations, cooperation network update rules, and frequency of update.

6.1 Performance Measure

In a nutshell, our overall interaction-based group modelling framework aims to build a cooperation network, from interactions, and perform community structure detection on it, so that the existing *true* groups, which affect the interactions, can be unveiled.

Hence, we could imagine the cooperation modelling component as the module responsible of approximating or inferring an existing underlying complex network with community structures. This means that the more the inferred cooperation network deviates from a random network, the easier its complex (community) structures can be detected, and the better it will be for the whole task of the group modelling framework.

Given a society $\mathcal{S} = \{a_1, a_2, \dots, a_n\}$ of n individuals, and its partitioning into m *true* groups

$$\mathcal{T} = \{g_1, g_2, \dots, g_m\} \tag{6.1}$$

where $g_i \in [1, m]$ represents the group identity/label of each individual $a_i \in \mathcal{S}$, the performance measure for the evaluation of CM's computation, that is its network $\mathcal{N} = (\mathcal{S}, \mathcal{C})$, will be the modularity score Q , calculated on \mathcal{C} given \mathcal{T} :

$$Q(\mathcal{C}, \mathcal{T}) = \frac{1}{m} \sum_{i \neq j} \left(\mathcal{C}_{i,j} - \frac{k_i^{out} k_j^{in}}{m} \right) \delta(g_i, g_j) \tag{6.2}$$

where m , k_j^{in} , k_i^{out} and δ are defined according to Equation (3.6). We will also refer to this calculation as the *true* modularity score. Intuitively, since Q is a measure of goodness of network partitioning into communities [161], the CM configuration which registers the highest $Q(\mathcal{C}, \mathcal{T})$ scores, would consequently facilitate the subsequent community structure detection task.

Table 6.1: The two artificial society types used for the investigation of CM’s performance.

Society	n	m	$ G_1 $	$ G_2 $
\mathcal{S}_{AD}	128	2	22	106
\mathcal{S}_{BC}	128	2	53	75

6.2 Evaluation Protocol

The evaluation will be based on simulations, of the experiment performed by Chen and Li, by means of artificial societies (CLAS, see Section 4.3). Please recall that the artificial societies are static, meaning that the group identities, and hence the group behaviours of the agents, will not change throughout the whole simulation runs. We here focus on artificial societies of $n = 128$ agents. The simulations will involve each agent playing 400 iterations of the other-other game.

Depending on their partitioning, two society structures are investigated. Both are based on $m = 2$ groups, as reported in Table 6.2. The motivation of these society types is presented in the subsequent Section.

6.3 Analysis of the *In/Out-group* Dichotomy

Section 5.1 described the necessary condition for the application of *TGB* for the identification of *in-group* and *out-group* boundaries, that is the observation of *in-out* interactions. Therefore, this Section aims to investigate whether there exist relationships between *in-group* and *out-group* sizes which could potentially affect the overall Cooperation Modelling component.

Given an occurrence of the other-other game, a provider agent a_P , belonging to group G of size $|G|$, will interact with one *in-group* and one *out-group* individual with probabilities

$$p(in) = \frac{|G| - 1}{n - 1}, \quad p(out) = 1 - p(in) = \frac{n - |G|}{n - 1} \tag{6.3}$$

furthermore, the probability of observing, in one occurrence of the other-other game, *in-in*, *out-out*, and *in-out* interactions will be respectively:

$$\begin{aligned} p(in - in) &= p(in) p(in) \\ p(out - out) &= p(out) p(out) \\ p(in - out) &= 2 p(in) p(out) \end{aligned} \tag{6.4}$$

The probability distributions depend on the *in-group* size $|G|$, as depicted in Figure 6.1. Furthermore, we can identify four intervals of $|G|$ in which the most likely interaction that is observed changes, as it is also summarised in Table 6.2.

Intervals *B* and *C* are those for which *in-out* interactions are the most likely, hence we would expect to see, in general, high *true* modularity values of the estimated \mathcal{N} networks. On the other hand, interval *A* corresponds to scenarios for which the *true* modularity values are generally low, since the *out-out* interactions, which could have any arbitrary level of fairness of treatment, might lead to erroneous classifications of *out-group* individuals as *in-group*, see the considerations made in Section 5.1 on this matter. Finally interval *D* might be the easiest interval to deal with, since we intuitively expect to see, for *in-in* interactions, absolute fairness of treatment, which could even be filtered out automatically if the cooperation evaluation task is based on the min-max normalisation, see also Appendix 13.2.

Based on the four intervals, we are going to consider, for our empirical evaluation, two society types, composed on $m = 2$ groups of different sizes, as reported in Table 6.2. Society \mathcal{S}_{AD} relates to the *A* and *D* scenario intervals, whilst Society \mathcal{S}_{BC} relates to the *B* and *C* interval. Although a higher number for m would allow for more combination of intervals, it is easy to understand that their empirical evaluations would still relate to the $m = 2$ case, since we assume that an individual does not distinguish among different *out-group* types.

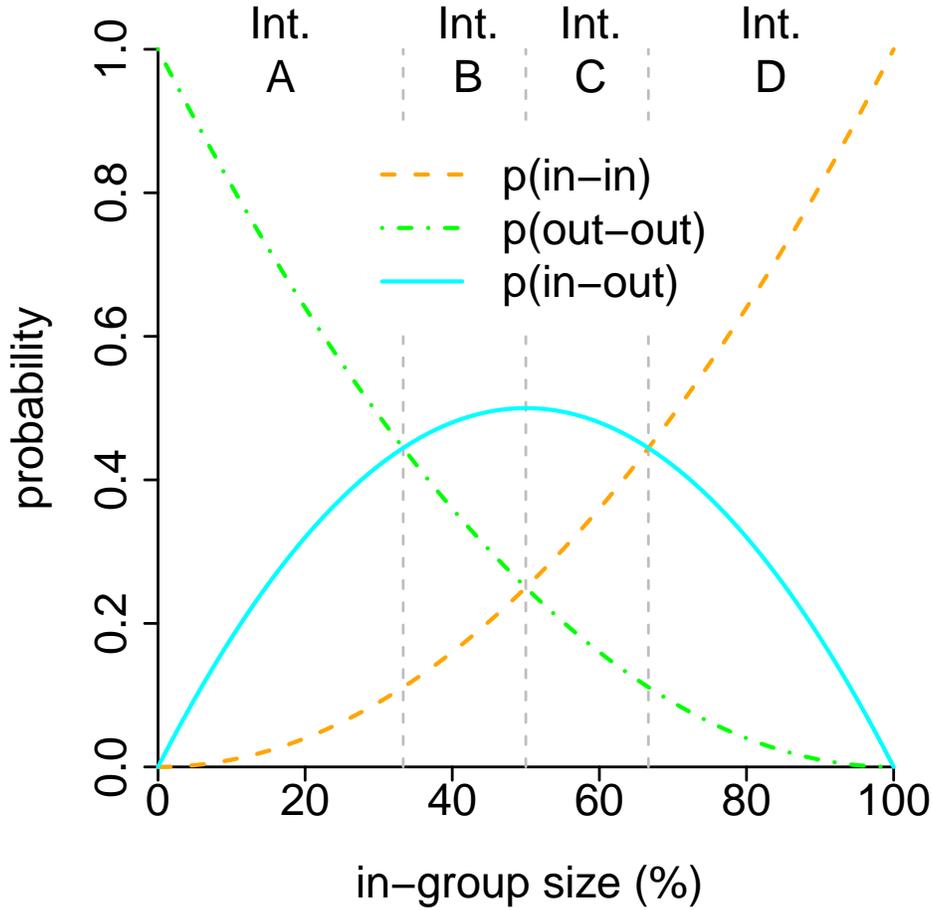


Figure 6.1: The probability distributions of observing *in-in*, *out-out*, and *in-out* interactions, under the *in/out-group* dichotomy, depending on the relationship between *in-group* sizes, *out-group* size, and hence the size n of society \mathcal{S} .

6.4 Preliminary Analysis of α

In order to isolate the role of the incremental factor α , used by constant- α and TD0 update rules — see Equations (5.10) and (5.11) — we need to make the cooperation evaluation task as deterministic as possible. Therefore, this Section is based on $t = 1$, and on using the min-max normalisation scaling of the interactions to probability values. In other words, the cooperation evaluation task — hence the whole CM component — will perform its computations each time an occurrence of the other-other game is observed ($t = 1$); it will receive in input a sequence of $O_P = \{o_i, o_j\}$ vectors and will either output $V_P = \{v_i, v_j\}$, $v_i, v_j = \{0, 1\}$ or ignore the interaction, in case $v_i = v_j$, see Appendix 13.2.

Figures 6.2(a) and 6.2(b) depict, respectively, the average performance and related standard deviation, across 30 simulation runs, of ten CM components, when $\alpha \in [0.1, 1]$ at regular intervals of 0.1, for \mathcal{S}_{AD} and \mathcal{S}_{BC} society types.

The first, most remarkable finding, is that $\alpha = 1$ is the setting for which the *true* Q converges to maximum values in the quickest time. Moreover, it also constitutes the Q -limit for any other α values considered. The reason why this occurs is because the constant- α update rule, for normalised v_i values and $0 < \alpha \leq 1$, will generate cooperation networks in which the \mathcal{C} values are also normalised (see Appendix 13.4). By using

Table 6.2: The four possible intervals of *in-group* size which alternates the order of the most likely interactions to be observed.

Interval	<i>in-group</i> Size	Probability of Interactions
A	$(0, \frac{1}{3}(n+2)]$	$p(out-out) \geq p(in-out) > p(in-in)$
B	$(\frac{1}{3}(n+2), \frac{1}{2}(n+1)]$	$p(in-out) \geq p(out-out) \geq p(in-in)$
C	$(\frac{1}{2}(n+1), \frac{1}{3}(2n+1)]$	$p(in-out) \geq p(in-in) > p(out-out)$
D	$(\frac{1}{3}(2n+1), n)$	$p(in-in) > p(in-out) \geq p(out-out)$

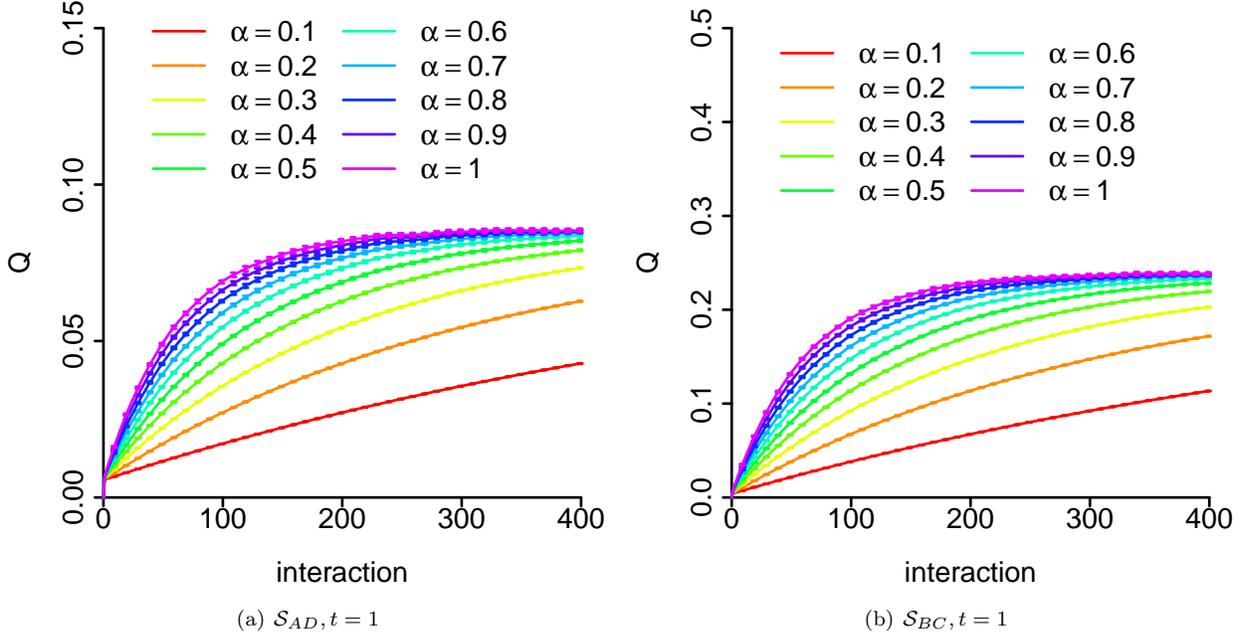


Figure 6.2: The impact that α has on the maximisation of the *true* Q , when the min-max normalisation and constant- α network update rule are used, in presence of *CLAS* societies of size $n = 128$, $m = 2$, and the two partitions of Table 6.2 are considered.

$\alpha = 1$ we would simply update the \mathcal{C} values to their maximum/minimum possible values, whilst for any other $0 < \alpha < 1$ the update would be less extreme.

It seems nevertheless surprising that the $\alpha = 1$ case constitutes the upper limit for the performance of cooperation modelling, considering that the update values are completely ignoring previous estimate values and rather entirely relying on current observations. One possible reason could be that the *CLAS* societies considered are static, meaning that the agent behaviours do not change across the 400 game interactions, and that the agents are fully affected by their group identities, meaning that we will almost certainly see favouritism towards the *in-group* individuals in *in-out* interactions. It is obvious that in case of dynamic societies, for instance in case an individuals decided to leave one group and to join another one, or in case the agents might not be completely affected by their group identities, the updates based on $\alpha = 1$ might lead to lower Q values than those obtained by using a lower α parameter. For these reasons, we will prosecute with our empirical evaluations by considering $\alpha = 1$ and the more “prudent” $\alpha = 0.5$.

The highest *true* Q values obtained for society type \mathcal{S}_{BC} — sample average 0.24 and standard deviation 0.0032, see Figure 6.2(b) — are significantly higher than those obtained for society types \mathcal{S}_{AD} — sample average 0.085 and standard deviation 0.0013, see Figure 6.2(a) — which indicate the clear difficulty that our interaction-based approach has with filtering out *in-in* and *out-out* interactions, since we know that a modularity value above 0.2 is an indication of a very good community structure partition. The question

whether network which score a low *true* modularity can still allow our group modelling framework to unveil the *true* \mathcal{T} partition will be investigated in Chapter 8.

6.5 Preliminary Analysis of γ

We now embark the analysis of the impact that parameter γ , used in TD0 update rule (5.11), has on the cooperation modelling task. We here build up on the findings of the previous Section, that is, we will present the results when $\alpha = 1$ and $\alpha = 0.5$.

Figures 6.3(a) and 6.3(b) depict, respectively, the average performance and related standard deviation, across 30 simulation runs, of ten CM modules, when $\gamma \in [0.1, 1]$ at regular intervals of 0.1, $\alpha = \{0.5, 1\}$, and for society type \mathcal{S}_{AD} . Similarly, Figures 6.3(c) and 6.3(d) depict the corresponding analysis for \mathcal{S}_{BC} society type.

The most remarkable finding is that TD0 improves its performance when $\gamma \rightarrow 0$, that is, when the update rule converges to the constant- α update rule. Three are the reasons which justify this trend (see Appendix 13.5 for more details):

- through TD0 we can no longer assure to maintain normalised cooperation values;
- when $\mathcal{C}_{i,j} < \gamma \mathcal{C}_{j,i}$, the update rule will move the $\mathcal{C}_{i,j}$ values towards the *out-group* estimation (value zero) even if the cooperation value v_i is 1, which is instead an indication of an *in-group* relationship;
- when $\mathcal{C}_{i,j} > 1 + \mathcal{C}_{j,i}$, the update rule will move the $\mathcal{C}_{i,j}$ values towards the *in-group* estimation — which does not have an upper bound — even if the cooperation value v_i is 0, which is instead an indication of an *out-group* relationship.

In other words, through the use of cooperation network update rules which, similarly to TD0, aim to take into account reciprocal cooperation values, we run the risk into performing incremental updates which go in the wrong direction with respect to the cooperation values of the interactions. For these reasons, we are no longer going to present the performance of CM when temporal difference is used.

6.6 Cooperation Modelling Analysis

Figures from 6.4 to 6.7 depict the performance of cooperation modelling in case of \mathcal{S}_{AD} society type; Figures from 6.8 to 6.11 depict, instead, the performance of cooperation modelling in case of \mathcal{S}_{BC} society type.

In both cases, the 24 plots are presented on a grid format: the three columns describe the three possible methods for cooperation evaluation which we considered in our dissertation — that is *raw* interactions, *in-group*-ness probability via min-max normalisation, and interactions classified into *in-group* and *out-group* labels/values — whilst the rows represent eight possible lengths for t , hence $|D| = 400/t$, which constitutes the number of cooperation network updates. Moreover, each plot depicts the performance of two update rules considered in our dissertation, that is standard average and constant- α , with $\alpha = \{0.5, 1\}$. Finally, the 72 plots depict the average performance and standard deviation of the CM configurations over 30 experimental runs based on the same society type and size. For $t > 1$, in case a_P plays the other-other game with the same agent a_i more than once, yet by maintaining the same proposer-receiver roles, prior to be sent for cooperation evaluation, all the offers o_i , proposed by a_P to a_i , will be averaged out.

A core finding is that constant- α update rule, for $\alpha \geq 0.5$, appears to be the best cooperation update rule, both with respect to the modularity values scored and the speed of convergence to the *true* Q values. The rationale behind this has to be necessarily found on the incremental factor — constant vs. variable — which dictates how much the update rule takes into account current interactions. This means that not only standard average might not be a viable method for cooperation network updates in dynamic societies, but also, and possibly more surprisingly, even for static societies of agents which, although behaving stochastically, tend to favour their *in-group* members rather than the *out-group* ones.

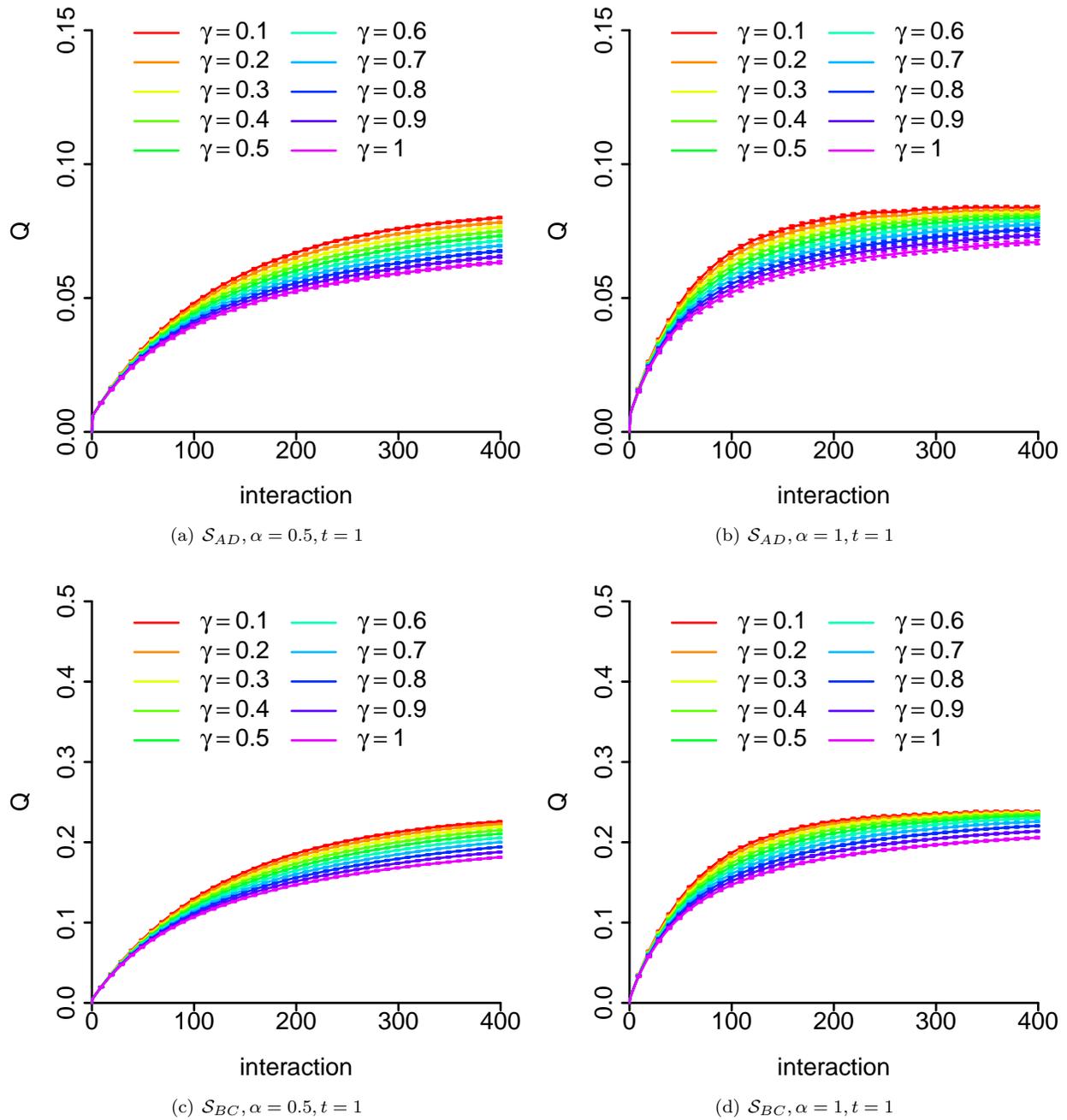


Figure 6.3: The impact that γ has on the maximisation of Q , when the min-max normalisation and constant- α network update rule are used, with $\alpha = \{0.5, 1\}$, in presence of *CLAS* societies of size $n = 128$, $m = 2$, and the two partitions of Table 6.2 are considered.

Another core finding is that via normalisation/discretisation of the interactions we can aim for the maximisation of the *true* Q values, whilst when *raw* interactions are considered, the modularity values converge almost instantaneously to very small values, $Q = 0.038$ for the \mathcal{S}_{AD} society type (standard deviation is 0.00082), and $Q = 0.0988$ (standard deviation is 0.0034) for the \mathcal{S}_{BC} type. Moreover, we observe that the probabilistic approach for cooperation evaluation is outperformed by the deterministic interaction classifier

approach, when t increases towards its possible maximum value $t = 400$, across the two society types. This results is not surprising, since the discretised values are a more “saturated” way to represent the *in-group* probabilities, hence, the same consideration made for $\alpha = 1$ as opposed to $\alpha = 0.5$ holds: the deterministic classifier constitutes the limit of performance of the probabilistic approach.

When t increases, although the CM component performs less network updates ($|D| = t/400$), the *true* Q value increase, providing however that the *raw* interactions are transformed into normalised cooperation values. This is due to the creation of larger O_P vectors, which allows the module to better leverage on the *one-to-many* approach and hence filter out *in-in* and *out-out* interactions, which cannot be used to identify *in-group* and *out-group* boundaries among the agents, via *TGB* and under the *in/out-group* dichotomy. Moreover, the *one-to-many* approach appears to be robust even against different magnitudes of interactions: recall that the agents play five versions of the other-other game, depending on the size of the endowment e , ranging between 200 and 400 tokens (see Section 4.1).

Finally, analogously to what we observed when analysing the α parameter in Section 6.4, we observe that, independently of the cooperation evaluation technique and the cooperation network update rule used, the modularity values for \mathcal{S}_{AD} society types are lower than the corresponding ones for the \mathcal{S}_{BC} type; this further consolidates the claim that the relationship between *in-group* sizes, and the generic *out-group* sizes, and, as a consequence, the probabilities to observe *in-out* interactions, has a significant impact on the ability to subsequently detect the *true* community structures.

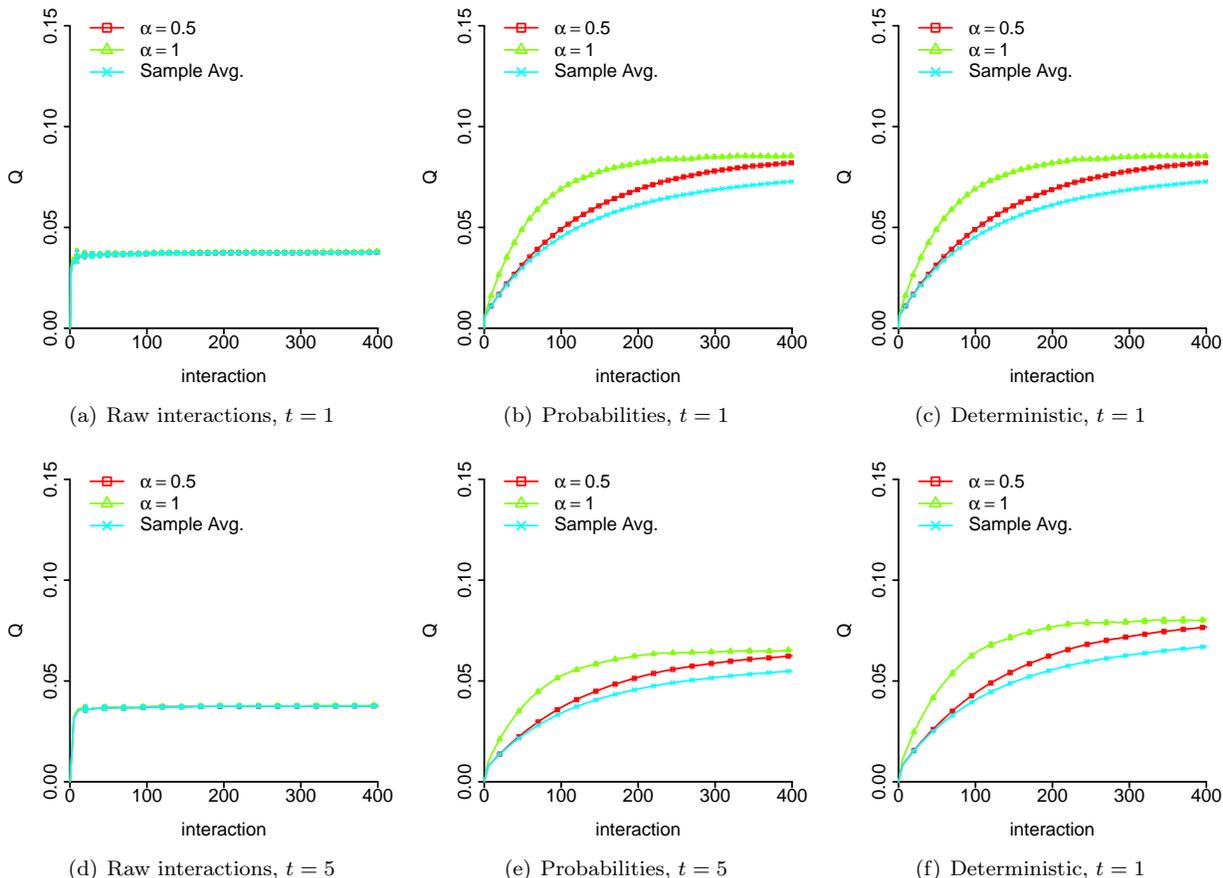


Figure 6.4: Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{AD} society type and depending on two different sizes of t .

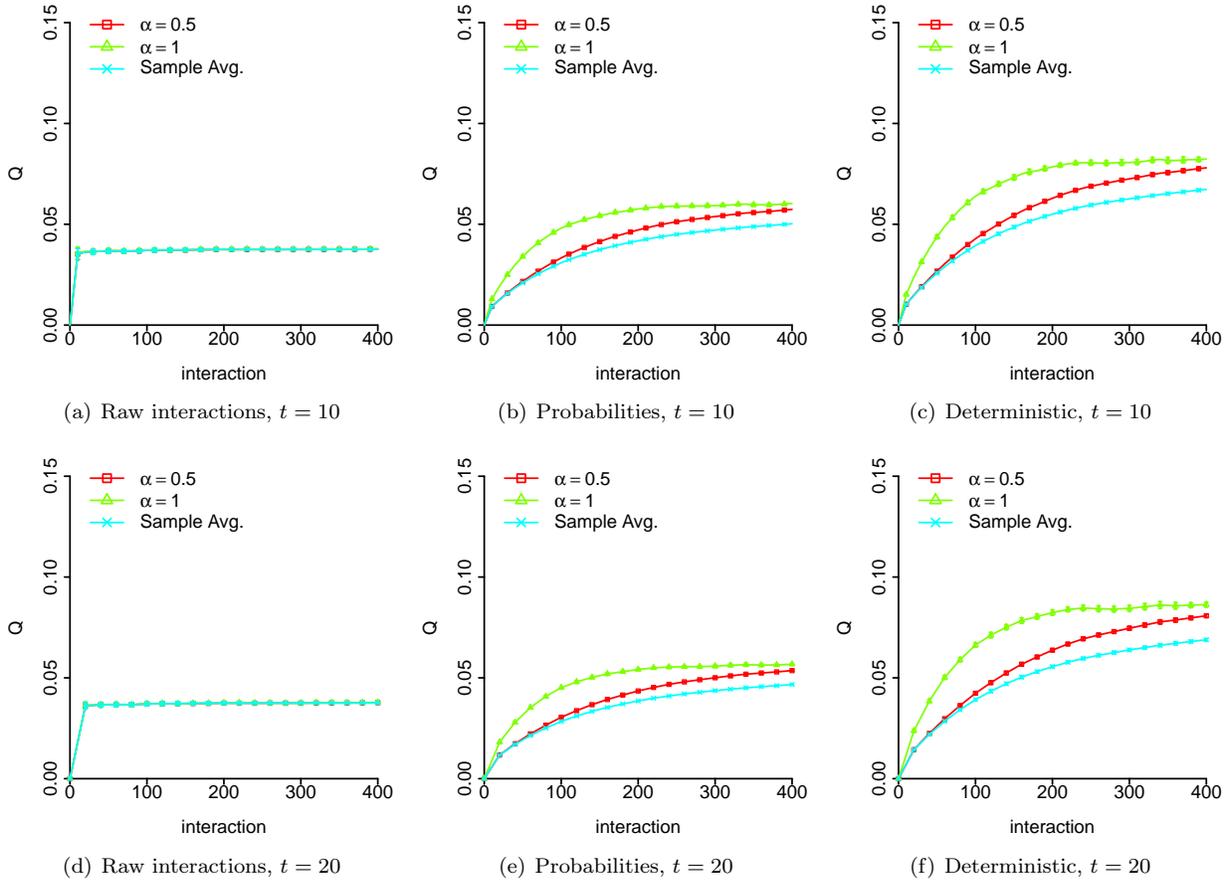


Figure 6.5: Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{AD} society type and depending on two different sizes of t .

6.7 Summary

This Chapter reported the results of the empirical investigations aimed to evaluate the validity of the first module of our interaction-based group modelling framework, namely the cooperation modelling component. The results gathered showed that a cooperation modelling framework which (1) aggregates a relatively small amount of consecutive interactions, e.g. 10 executions of the other-other game out of 400 for each agent; (2) transforms these observations into discretised values representing *in-groupness* and *out-groupness* relationships; and (3) uses them, myopically, for the update of the society's cooperation network via the constant- α update rule ($\alpha = 1$), would lead to cooperation networks in which the modularity score of the *true* group structures is the highest.

The next Chapter will describe, in details, the second component of the group modelling framework, namely the group identity detection module, which is in charge of performing community structure detection, on the inferred cooperation networks, in order to unveil the *true* group structures of the society under investigation.

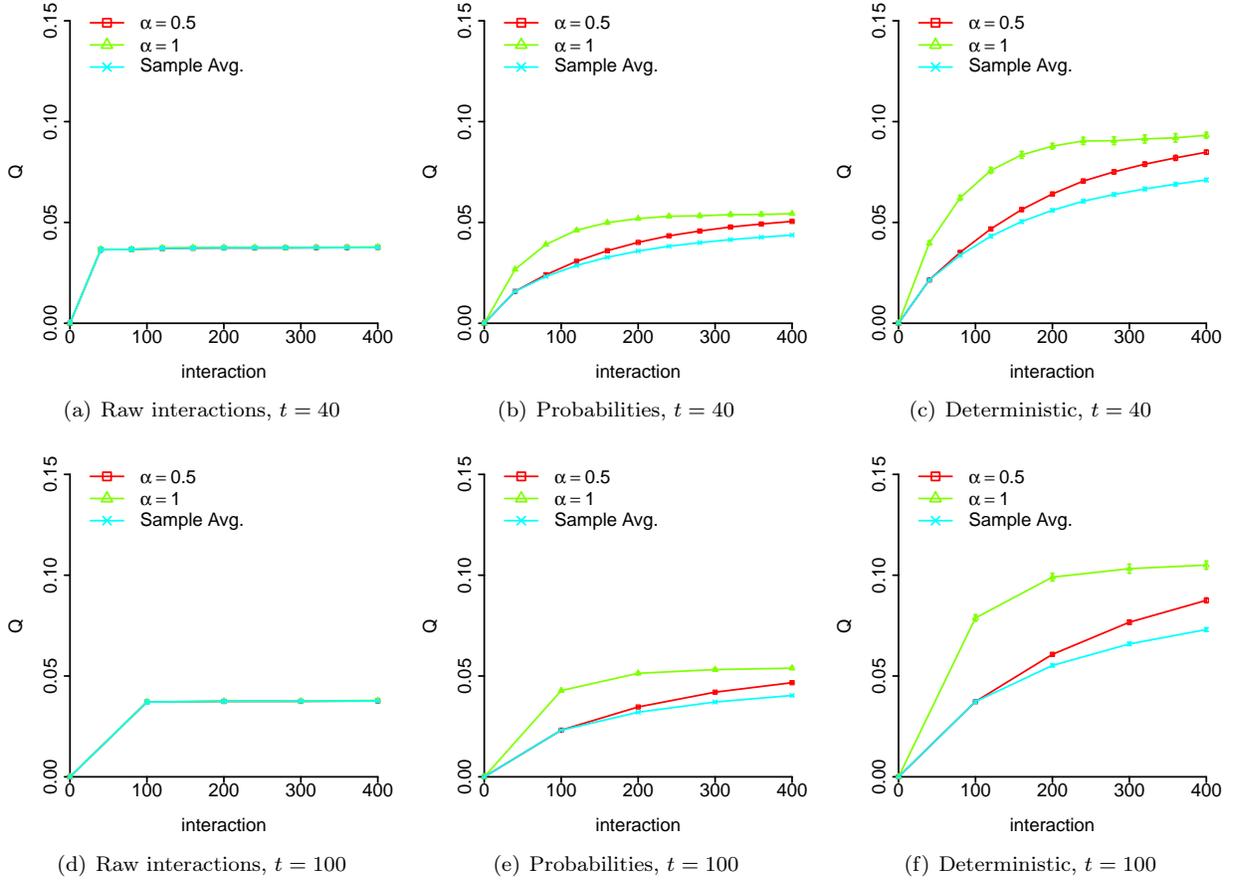


Figure 6.6: Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{AD} society type and depending on two different sizes of t .

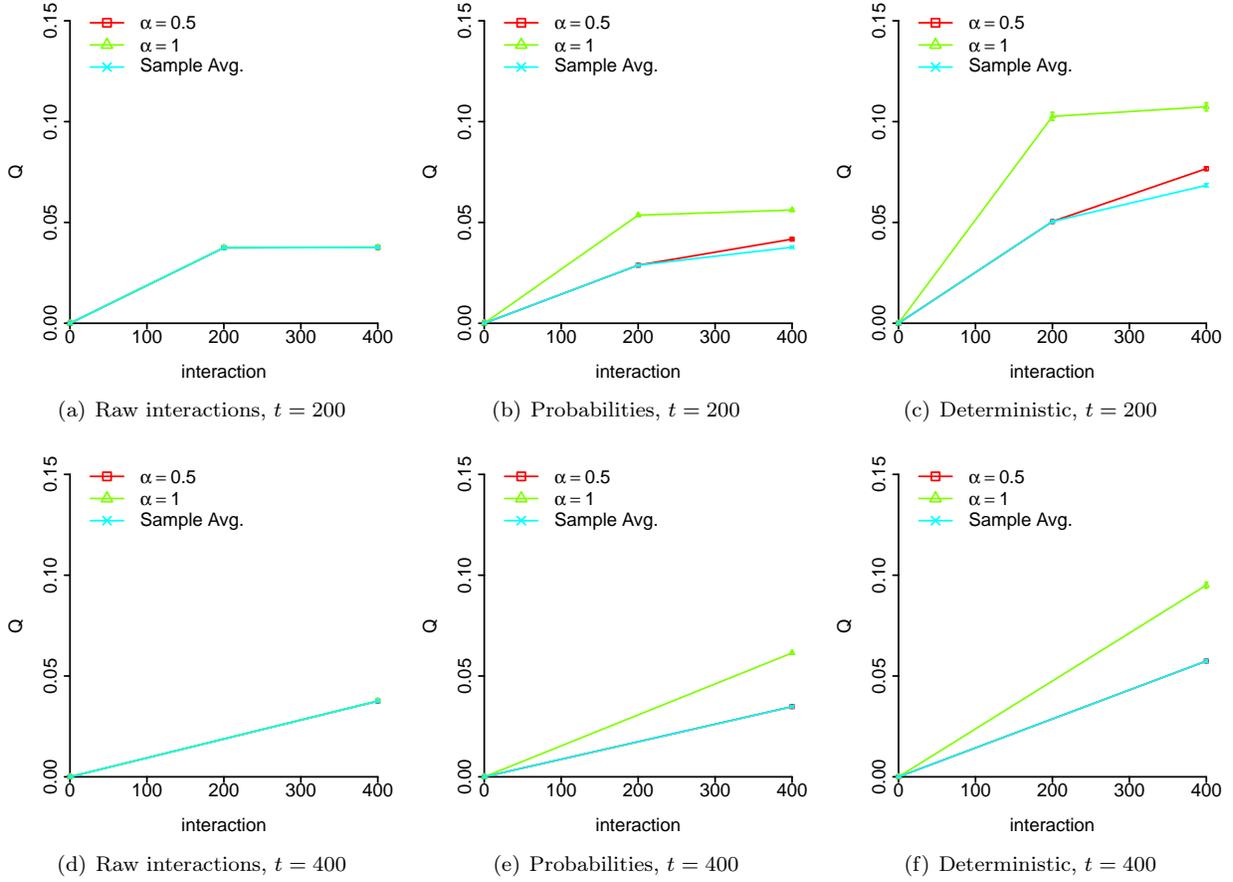


Figure 6.7: Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{AD} society type and depending on two different sizes of t .

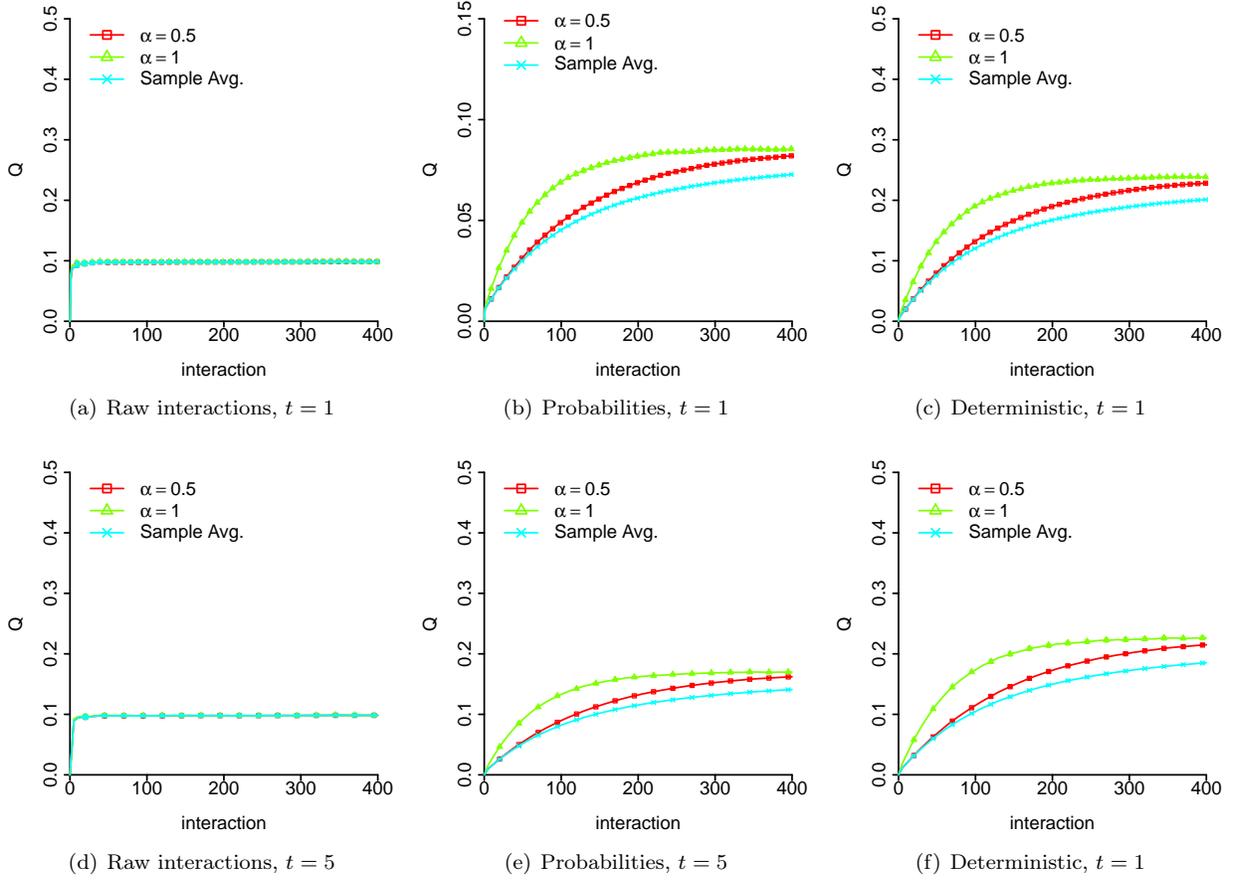


Figure 6.8: Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{BC} society type and depending on two different sizes of t .

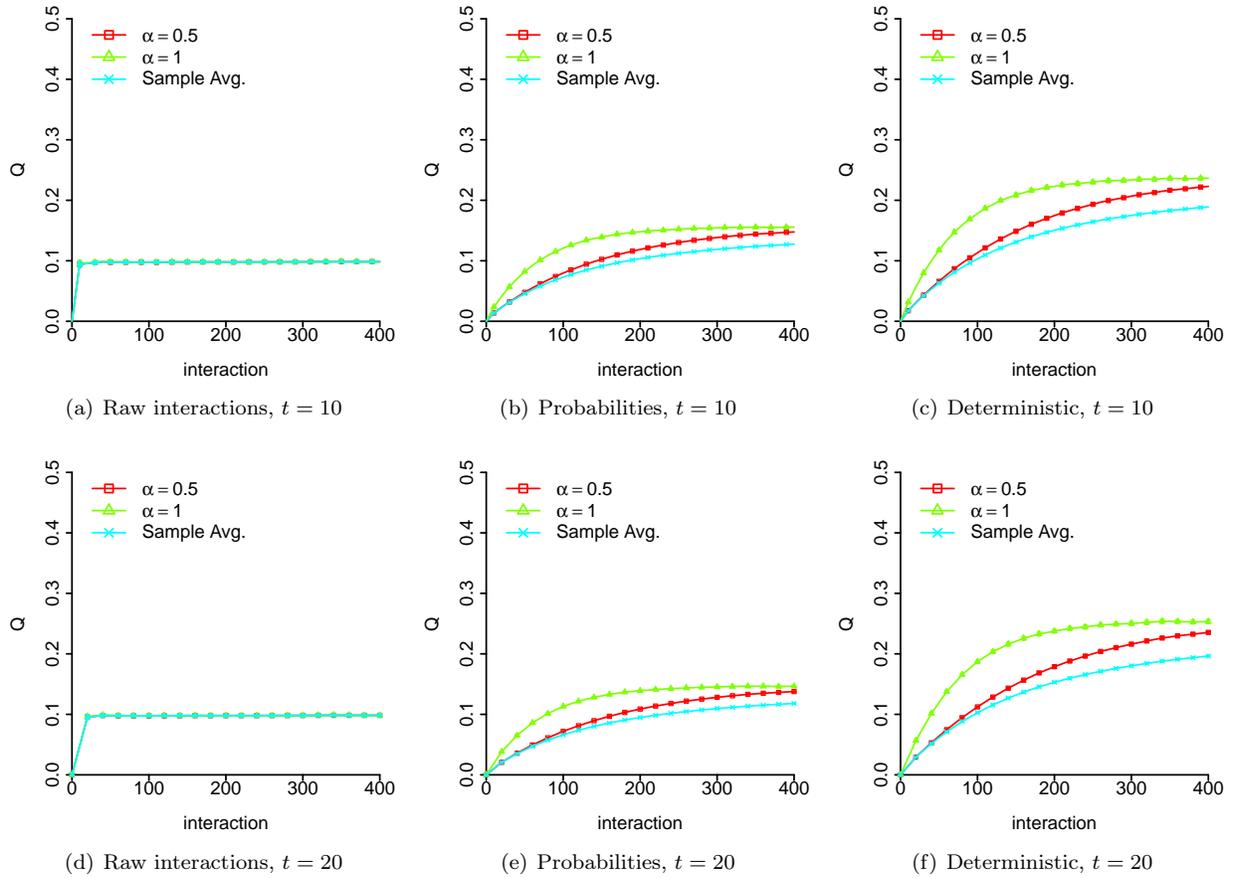


Figure 6.9: Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{BC} society type and depending on two different sizes of t .

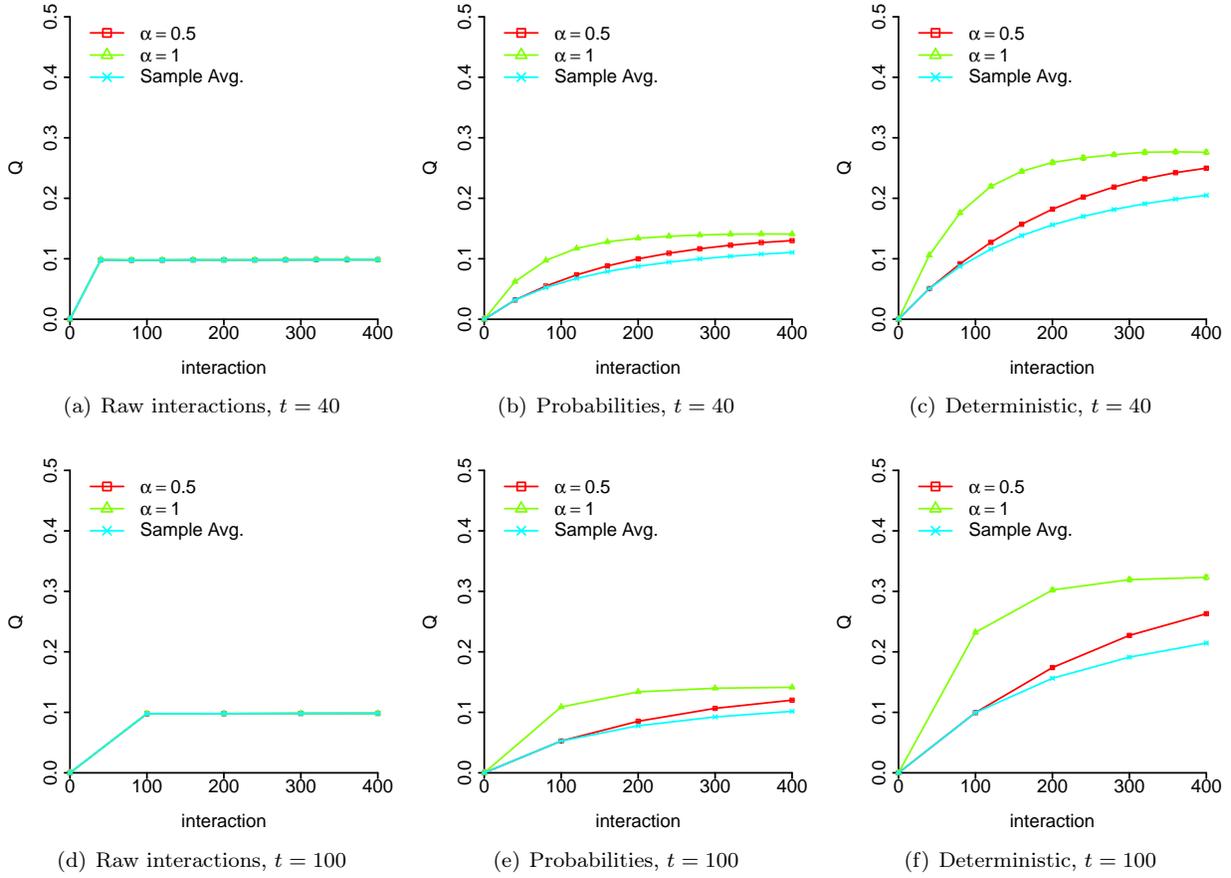


Figure 6.10: Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{BC} society type and depending on two different sizes of t .

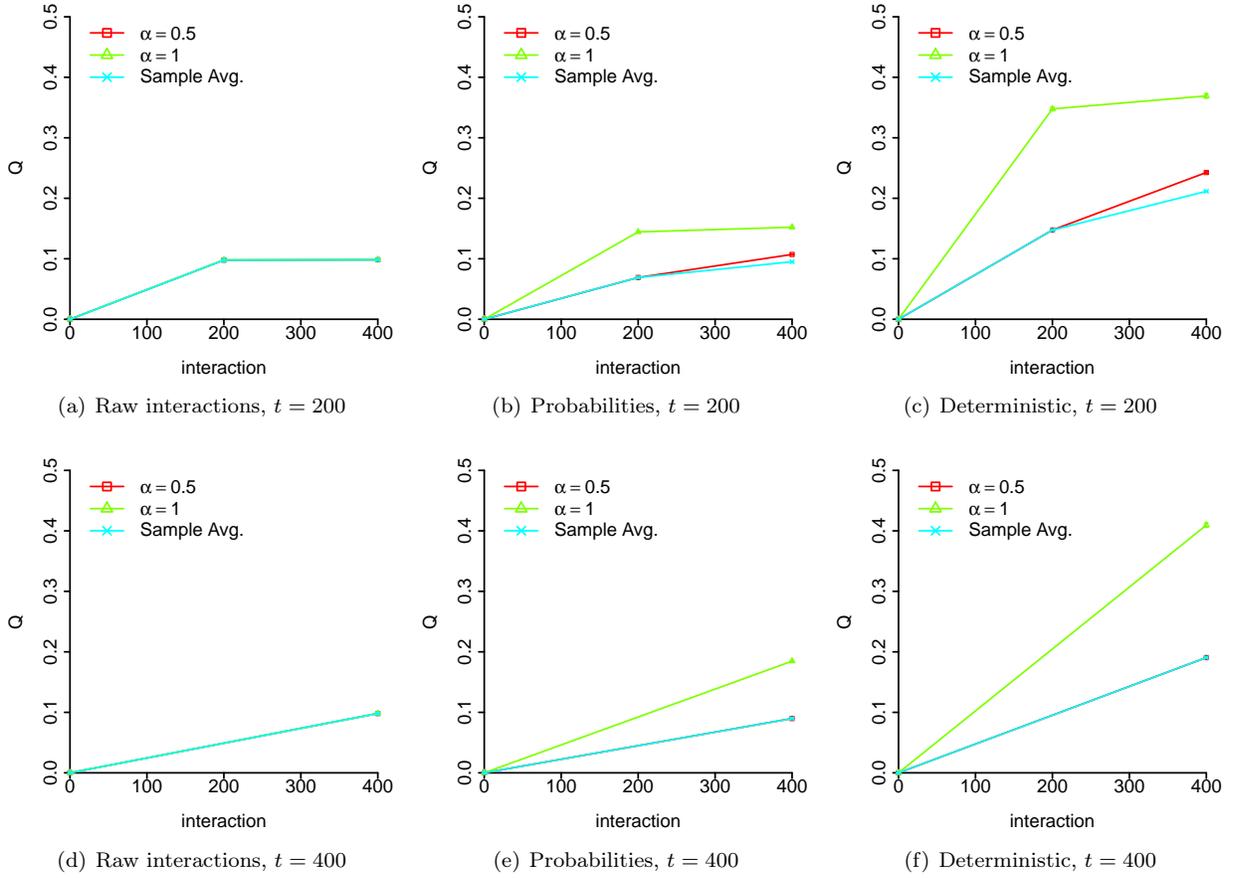


Figure 6.11: Comparison of the performance of constant- α ($\alpha = \{0.5, 1\}$) and standard average cooperation network update rules, combined with the three methods for cooperation evaluation, based on \mathcal{S}_{BC} society type and depending on two different sizes of t .

Chapter 7

Cooperative Group Identity Detection

Group Identity Detection (GID) is the second and last module of our proposed interaction-based group modelling framework, see Figure 1.2. The first module, namely cooperation modelling, has the duty to observe the ongoing interactions among the society, to evaluate their levels of cooperation, and to maintain an up-to-date estimate of the society’s cooperation network. The duty of the GID module is to process the cooperation network in order to partition the society into groups, so that the *within*-group cooperation is maximised, whilst the *between*-group cooperation levels are minimised [39]. The GID’s task is done by performing community structure detection (CSD).

Section 7.1 motivates the approach used for CSD, that is modularity optimisation. Subsequently, Section 7.2 illustrates and motivates the evolutionary computation approach we decided to investigate for CSD. In this dissertation we will focus on three chromosome representations; these will be presented, together with how the evolutionary algorithm is customised to them, in their dedicated Sections 7.4, 7.5 and 7.6. Finally, Section 7.7 introduces a possible further development of the evolutionary computation approach, centred on the use of niches for the evolutionary search of the solution space. Empirical evaluations of the algorithms are instead investigated and presented in Chapter 8.

7.1 Group Identity Detection via Modularity Optimisation

As it has already been reported in Section 3.2, CSD, and network theory in general, are rather young research fields. Moreover, since in this dissertation we deal with many types of networks — ranging from directed and weighted to sparse undirected and unweighted — we run the risk not to grasp, even on a broad level, what the best approaches and techniques could be. Additionally, we aim to use CSD for unveiling *true* group identities from interaction data, a task that, to the best of our knowledge, has not been previously attempted.

It is for these reasons that, in Section 4.3, we decided to adopt, as benchmark social synthetic environments, artificial societies of 128 agents, with four groups of equal size — in accordance with the well-established benchmark synthetic networks used in many other CSD studies [161] — and societies with group sizes following a power law distribution, similarly to the arguments raised by Lancichinetti et al. [124]. These decisions were taken so that it could be easier to bridge our research and related results with those obtained in network theory, focused on community structure detection, based on synthetic networks. Moreover, as we will present later in Chapter 8, we will consider spectral partitioning for modularity maximisation [160, 59] as a baseline performance algorithm. Spectral partitioning is a widely accepted approach for CSD and is generally robust for mid-sized networks; this will make our empirical evaluation robust and even more comparable with others.

In order to further constrain our investigation, we decided to mainly focus on modularity maximisation for CSD. Our decision is motivated by many reasons. Broadly speaking, this method is one of the most consolidated ones among the CSD research. It has already provided promising results in both sparse [59] and complete [88] directed and weighted networks. Semantically, Equation (3.6) seems to fit well the *in-group*

vs. *out-group* dichotomy. Furthermore, Equation (3.6) can be applied, without any adjustments, to any kind of network typologies; this property fits well with the dynamic nature of the cooperation network under investigation.

7.2 Modularity Maximisation via Evolutionary Computation

For what it concerns the algorithms used for modularity optimisation, we decided to focus our research on the application of evolutionary computation. Some motivations, more generic, were already introduced in the previous Sections of this dissertations (see e.g. Sections 1.3, 3.2 and 3.3). Here we highlight just some of the potential advantages of using evolutionary computation for CSD and, more generically, for the interaction-based group modelling:

- evolutionary computation has already provided promising results in CSD for sparse undirected and unweighted case — see Section 2 — though, to the best of our knowledge, there has not been any other attempt, apart from ours [83, 90, 88], to solve CSD for directed, weighted, and possibly complete networks. The results we obtained in our previous work were highly promising; this encouraged us to further investigate evolutionary CSD in this dissertation;
- evolutionary computation can be designed to be fitness independent with respect to the CSD problem, meaning that the same algorithmic implementation, e.g. based on the maximisation of a quality measure of a network partitioning, can evolve candidate solutions independently whether the fitness function is e.g. modularity, LinkRank, the community score, and so on;
- though related to undirected and unweighted networks, arguments against modularity have been raised [122], see Section 3.2.3. However, it might be possible that part of the limitations in e.g. finding small/large communities is also due to the maximisation algorithm, which might be overly structured, and which does not allow for the detection of irregularities in the community structure partitions. With this respect, evolutionary computation, being a stochastic process, implements a “loose” optimisation strategy. This feature might be able to overcome, or at least reduce, the weaknesses of modularity;
- unlike spectral partitioning, which requires that the adjacency matrix of the cooperation network is symmetric, evolutionary computation can directly rely on e.g. the most generic modularity measure and evolve its candidate solutions on asymmetric adjacency matrices. Moreover, the symmetrisation of the asymmetric adjacency matrix would most likely introduce artefacts, with potential negative consequences with respect to the unveil of the *true* groups;
- Evolutionary computation has proven to be extremely robust with dynamic changes of their search space; this makes it very suitable for CSD in dynamic networks [66].

Moreover, with respect to possible future application:

- Evolutionary computation can also be used as a multi-objective optimisation technique. This means, for instance, that the candidate solutions being evolved could be evaluated across different fitness functions [176];
- It could also be possible to combine the evolutionary search of network-based solutions with those based on node properties, closely related to standard clustering;
- The use of particular evolutionary computation approaches, for instance those based on niching or island models, might allow for the concurrent seek for community structures partitioning multiplex networks.

We are now going to describe, in details, the structure, genetic representation, and recombination procedures, of the Evolutionary Algorithms (EA) considered in this dissertation.

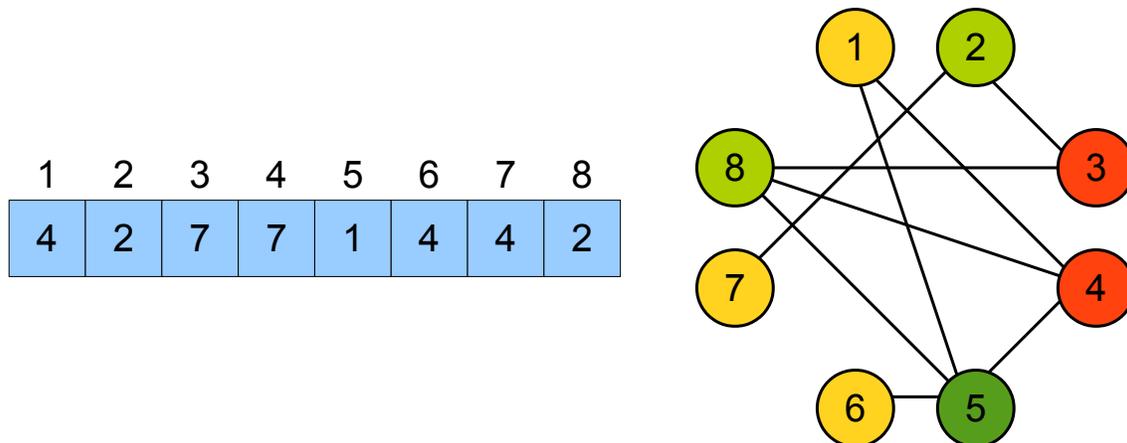


Figure 7.1: An example of a node representation-based chromosome. The reference network is the one of Figure 3.1. The allele values of the chromosomes are represented by colours in the corresponding network.

7.3 The Template Evolutionary Algorithm

All the Evolutionary Algorithms (EAs) considered in this dissertation will perform (non overlapping) community structure detection of a given cooperation network $\mathcal{N} = (\mathcal{S}, \mathcal{C})$. \mathcal{S} is the set of nodes, representing the society of n individuals — one node per individual — and \mathcal{C} is the set of edges/adjacency matrix of the network. The EAs will evolve candidate solutions representing, more or less directly, possible partitions of the network into community structures $H = \{h_1, h_2, \dots, h_n\}$. Intuitively, the chromosome with the highest fitness score, at the end of each EA execution, will represent the inferred community structure partition solution of the GID module for the given cooperation network \mathcal{N} .

The EAs will all share the same parent selection procedure via rank selection, and elitism, see Section 3.3. The main emphasis is given on the use of modularity, defined in Equation (3.6), as fitness function of our EAs. However, part of the empirical evaluation will be also dedicated on the study of LinkRank as an alternative fitness function, see Section 8.5.

We will hereafter investigate three different chromosome representations; these will necessarily have an impact on the initialisation procedure and their related recombination mechanisms. In addition to the use of EAs based on a single population, we will also embark the research aimed to the use a niching approach, which requires that the chromosomes must satisfy some pre-defined constraints (see Section 7.7). However, each single niche would still be based on the template EA customised for the three chromosome representation.

7.4 Node Representation

This representation, commonly used for standard clustering problems [100], is possibly the most intuitive one. It is also generally called *string* or *object* representation [55]; we will however refer to it as *node* representation, in order to better situate it, at least syntactically, with the other two representations considered in this study.

A chromosome has length n , and each gene represents a node of network \mathcal{N} . The allele values of the i -th gene represents the group identity h_i of node v_i .

The EA adopting the node representation will evolve the community identities of the network nodes, meaning also that the chromosomes will be directly used for the calculation of their fitness score. Figure 7.1 depicts an example of a chromosome adopting node representation and the corresponding partitioning, into community structures, of the network originally presented in Figure 3.1. As it can be hinted, the node representation can lead to network partitions in which the nodes are not connected with each other, meaning that the evolutionary search does not leverage on the underlying network topology.

Algorithm 7 Node Representation - Chromosome Mutation

```
1: sample  $r \in [0, 1)$  uniformly
2: while  $r < p_{mutation}$  do
3:   sample  $i \in [1, n]$  uniformly
4:   sample  $k \in [1, n]$  uniformly
5:    $h[i] \leftarrow k$ 
6:   sample  $r \in [0, 1)$  uniformly
7: end while
```

7.4.1 Chromosome Initialisation

Each gene of each chromosome is initialised by assigning an allele value sampled uniformly within the $[1, n]$ interval, that is, the EA starts its computation by assuming the existence of an arbitrary number of groups, from 1 to n . We are aware that there exists a plethora of possible initialisation procedures which would allow the EA to reduce its search space, see e.g. Tasgin et al. [204]. However, since network \mathcal{N} is highly dense, especially during the early stages of the group modelling framework, we decided not to adopt any ad hoc initialisations for this chromosome representation, as not to bias the search unnecessarily.

7.4.2 Crossover Operator

We decided to rely on uniform crossover, since the nodes of the network are assumed not to have any spatial relationships, meaning thus that genes with far apart indices could still belong to the same community structure. Once two parents are selected, they will reproduce and generate two offsprings with probability p_{cross} , see Section 3.3.4.

7.4.3 Mutation Operator

The mutation operation will select a gene uniformly within the $[1, n]$ interval and change its allele value, with another one, uniformly sampled within the $[1, n]$ interval. This operation is iterated a variable number of times, based on a threshold probability p_{mut} . Algorithm 7 presents the pseudocode for the *node*-based mutation operator.

7.5 Group Representation

This chromosome representation is inspired by the work of Falkenauer [57, 58, 55, 56]. Moreover, to the best of our knowledge, there has not been any attempt to use this chromosome representation to solve CSD problems. Each chromosome is composed of two parts: a *node* and a *group* part. The node part, intuitively, is similar to the node representation previously introduced: it has length n genes, corresponding to the nodes of the network, and the allele values represent the group identity of the nodes. The group part, on the other hand, holds the list of unique group identity symbols of the node part. The peculiarity of this representation resides on the recombination operators, which are now made at the group part, with consequent repercussions on the node counterpart. Intuitively, the evolutionary process based on this representation would deal with chromosomes of variable length. Figure 7.2 depicts an example of a chromosome with group representation.

The motivations which led Falkenauer to introduce this representation, and the corresponding Group Genetic Algorithm (GGA) [55], came from the understanding that through the sole node representation there is a risk to generate offsprings that would locate very far away from their parents. An example in support of this claim can be seen in Figure 7.3.

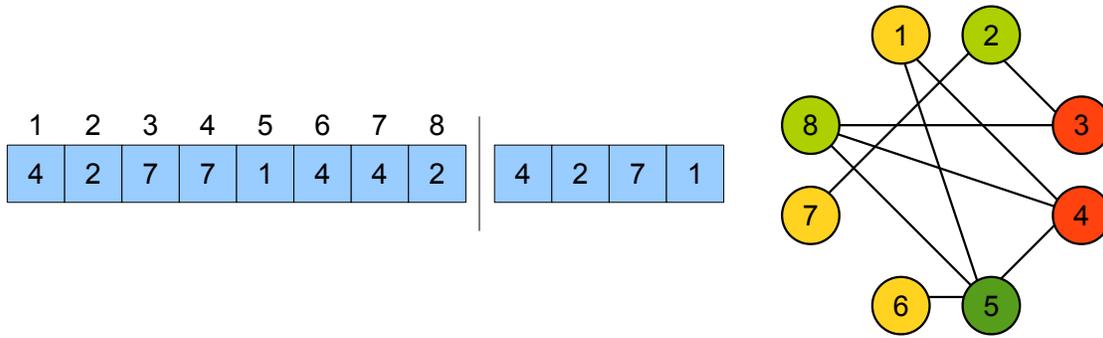


Figure 7.2: An example of a chromosome adopting the group representation. The reference network is the one of Figure 3.1. The allele values of the object (left) part of the chromosome correspond to the group identities of the nodes, and are represented by colours in the corresponding network. The (right) group part of the chromosome is used for the recombination operators.

7.5.1 Chromosome Initialisation

The initialisation procedure is the same as the one used for the node representation, with the additional phase in charge of extracting the unique group identity labels at the node part and their storage at the group part.

7.5.2 Crossover Operator

The crossover operator is applied at the group part and has repercussions on the node part. Once two parents are selected, they will reproduce and generate two offsprings with probability p_{cross} . The operation follows Falkenauer’s original proposed pattern [55]:

1. select at random two crossing sites, delimiting the crossing sections, in each of the two parents;
2. inject the contents of the crossing section of the first parent at the first crossing site of the second parent. Recall that the crossover works at the group part of the chromosome, so this means injecting some of the groups from the first parent into the second;
3. eliminate all symbols now occurring twice from the groups they were members of in the second parent, so that the “old” membership of these symbols gives way to the membership specified by the “new” injected groups. Consequently, some of the “old” groups coming from the second parent are altered: they do not contain all the same symbols anymore, since some of those had to be eliminated;
4. if necessary, adapt the resulting groups, according to the hard constraints and the cost function to optimise. At this stage, local problem-dependent heuristics can be applied;
5. apply the points 2 through 4 to the two parents with their roles reversed in order to generate the second offspring.

A graphical representation of the crossover operation we have implemented is depicted in Figure 7.4.

7.5.3 Mutation Operator

Similarly to the crossover operator, the mutation operator is applied on the group part and has repercussions on the node part. Falkenauer suggested to implement the mutation operator depending on the grouping problem on hand [55]. Hence, the group-based mutation operations we implemented are those which can resemble typical group dynamics, and were inspired by the relevant work of Palla et al. [170]. They are schematically depicted in Figure 7.5.

Algorithm 8 Group Growth Mutation Operator

```
1: sample allele value  $k$  uniformly within the group part of the chromosome
2: sample  $r$  uniformly within  $[0, 1)$ 
3: while  $r < p_{mut}$  do
4:   sample gene  $i$  uniformly within  $[1, n]$ 
5:    $h_i \leftarrow k$ 
6:   sample  $r$  uniformly within  $[0, 1)$ 
7: end while
```

Algorithm 9 Group Contraction Mutation Operator

```
1: sample allele value  $h$  uniformly within the group part of the chromosome
2: for all gene  $i \in [1, n]$  and  $h_i = h$  do
3:   sample  $r$  uniformly within  $[0, 1)$ 
4:   if  $r < p_{mut}$  then
5:      $k \leftarrow h$ 
6:     while  $k = h$  do
7:       sample allele value  $k$  uniformly within the group part of the chromosome
8:     end while
9:      $h_i \leftarrow k$ 
10:  end if
11: end for
```

For each new offspring generated via crossover, one group mutation operation is uniformly chosen and performed. Each of the six mutation operations, instead, are presented in pseudocode in Algorithms 8 to 13.

7.5.4 Group Inversion Operator

This form of mutation operator, originally proposed by Falkenauer [55], is once again applied to the group part of the chromosome, though this time it does not have any repercussions on the node part. The inversion operator — which is applied in our EAs to each newly generated offspring — serves to shorten good schemata in order to facilitate their transmission from parents to offspring, thus ensuring an increased rate of sampling of the above-average ones [55, 99]. All this operation does is to sample a sequence of consequent group identity symbols in the group part and flip them. Figure 7.6 depicts a simple example of the inversion operation.

7.6 Edge Representation

The third and last chromosome representation considered is the *locus-based* [172], which is very popular among the evolutionary CSD literature, see Section 2.5.2. In a nutshell, the EAs adopting the locus-based representation evolve node edges rather than group identities. For this reason, we will refer to it as the *edge* representation, in order to better situate it together with the node and group representations.

More specifically, each chromosome has length n , and each gene represents a node in the network. Allele value j of gene i will represent the edge (i, j) connecting the two nodes. A subsequent conversion task will extract subnetworks from the chromosomes and then assign the same group identity to the nodes of the same subnetwork. An example of such conversion process is depicted in Figure 7.7.

It is worth noting that the number of edges represented in each chromosome is equal to n , that is, the number of nodes. This means that it is not possible to represent, within a single chromosome, all the edges of the given cooperation network, which can instead be on the order of n^2 . However, this is not necessary, as the purpose of the representation is only to encode which nodes belong to which subnetwork.

Algorithm 10 Group Merge Mutation Operator

```
1: sample allele value  $h_1$  uniformly within the group part of the chromosome
2:  $h_2 \leftarrow h_1$ 
3: while  $h_2 = h_1$  do
4:   sample allele value  $h_2$  uniformly within the group part of the chromosome
5: end while
6: for all gene  $i \in [1, n]$  and  $h_i = h_1$  do
7:    $h_i \leftarrow h_2$ 
8: end for
```

Algorithm 11 Group Split Mutation Operator

```
1: sample allele value  $h$  uniformly within the group part of the chromosome
2:  $k \leftarrow$  new allele value not present in the group part of the chromosome
3:  $r \leftarrow 1$ 
4: for all gene  $i \in [1, n]$  and  $h_i = h$  do
5:   if  $r < p_{\text{mut}}$  then
6:      $h_i \leftarrow k$ 
7:   end if
8:   sample  $r$  uniformly within  $[0, 1)$ 
9: end for
```

7.6.1 Chromosome Initialisation

Unlike for the node and group representations, for the edge representation it is necessary to implement an ad hoc initialisation phase. This would make sure that gene i has allele value j if and only if the corresponding edge (i, j) is present in \mathcal{N} . If this feature was not implemented, then the EA would run the risk to evolve candidate solutions based on false edges [174, 177]. Clearly, the edge representation is the one, among the three considered, which most depends on the underlying network topology.

In our implementation, the initialisation phase would firstly check for the existence of each (i, j) edge encoded in the chromosome. In case the edge is not present in \mathcal{C} , then a new allele value for gene i would be sampled, among its neighbouring nodes, proportionally to their weights. More specifically, given node/gene i and its list of neighbours $Neigh_i$, neighbour $j \in Neigh_i$ will be chosen with probability

$$p_{i,j} = \frac{\mathcal{C}_{i,j}}{\sum_{k \in Neigh_i} \mathcal{C}_{i,k}} \quad (7.1)$$

Figure 7.8 depicts the gene/edge repair procedure, whilst its pseudocode is presented in Algorithm 14.

7.6.2 Crossover Operator

Similarly to the node representation, for the edge representation we will rely on uniform crossover. It is worth pointing out that through this type of recombination, repaired chromosomes will still generate, once crossed over, repaired offsprings [174, 177]. Once selected, two parents will generate the offsprings with probability p_{cross} .

7.6.3 Mutation Operator

The mutation operator is applied to single edges/allele values by means of the same repair procedure used to during chromosome initialisation. The algorithm implementing the mutation operator for the edge representation is presented in Algorithm 16.

Algorithm 12 Group Birth Mutation Operator

```
1:  $k \leftarrow$  new allele value not present in the group part of the chromosome
2: sample gene  $i$  uniformly within  $[1, n]$ 
3:  $h_i \leftarrow k$ 
4: sample  $r$  uniformly within  $[0, 1)$ 
5: while  $r < p_{mut}$  do
6:   sample gene  $i$  uniformly within  $[1, n]$ 
7:    $h_i \leftarrow k$ 
8:   sample  $r$  uniformly within  $[0, 1)$ 
9: end while
```

Algorithm 13 Group Death Mutation Operator

```
1: sample allele value  $h$  uniformly within the group part of the chromosome
2: for all gene  $i \in [1, n]$  and  $h_i = h$  do
3:    $k \leftarrow h$ 
4:   while  $k = h$  do
5:     sample allele value  $k$  uniformly within the group part of the chromosome
6:   end while
7:    $h_i \leftarrow k$ 
8: end for
```

7.7 Niching Approach

Possibly, the main advantage coming from the use of evolutionary computation is given by its stochastic nature, as opposed to e.g. spectral partitioning’s greedy deterministic approach. The stochastic features of evolutionary algorithm might help correctly identify some peculiar group identities which a deterministic greedy approach could overlook.

For instance, it might happen that a node is at the “boundary” of its group, i.e. it has a lot of edges connecting it with nodes belonging to other groups. Due to this fact, and also other links in the network, or even due to a symmetrisation of the adjacency matrix, it could happen that spectral partitioning would classify this “ambiguous” node as belonging to an erroneous group, even after the fine parameter tuning. This misclassification will not be revised throughout the execution of spectral partitioning. On the other hand, through evolutionary computation, there is still a non-zero chance, for that node, to be moved to its *true* group, by means of e.g. the mutation operator.

Evolutionary computation allows for the convergence of the candidate solutions to subregions of the search space, within which local (or possibly global) optima are found. This is driven by many factors, such as the selection process (*pressure* and *noise*) and operator disruption [143]. With respect to the EAs considered in this dissertation, the definition of the fitness function has a key role especially with respect to the latter factor. We know that modularity — at least its definition for undirected and unweighted networks — suffers from the resolution limit, meaning that the evolutionary search will tend to converge towards subregions of the search space which consider either small-sized or large-sized groups. In other words, the EAs considered might suffer from a lack of population diversity, which would allow for the exploration of subregions of the search space which are initially suboptimal, though providing a sufficient evolutionary time they might lead to the exact partitioning of the network into *true* groups. In parts of the next Chapter we investigate the use of niching methods in order to promote population diversity.

The approach we adopt to implement niching methods is centred on the idea to partition the search space, hence the whole set of possible chromosomes, by the number of community structures encoded in the candidate solutions. We aim to maintain this *spatial* approach throughout the whole execution of the evolutionary process, since articulated *true* solutions, for instance composed of many groups of several different sizes,

Algorithm 14 Edge Representation - Chromosome Repair

```
1: for all gene  $i \in [1, n]$  do
2:   if  $C_{i,h_i} = 0$  then
3:     mutate_repair( $i$ )
4:   end if
5: end for
```

Algorithm 15 Edge Representation - Gene i Repair/Mutation Operator (mutate_repair)

```
1:  $N \leftarrow \emptyset$ 
2:  $W \leftarrow \emptyset$ 
3:  $sum \leftarrow 0$ 
4: for all gene  $j \in [1, n]$  do
5:   if  $C_{i,j} > 0$  then
6:      $N \leftarrow N \cup \{j\}$ 
7:      $W \leftarrow N \cup C_{i,j}$ 
8:      $sum \leftarrow sum + C_{i,j}$ 
9:   end if
10: end for
11: for all edge weight  $w \in W$  do
12:    $w \leftarrow w/sum$ 
13: end for
14: sample  $r$  uniformly within  $[0, 1)$ 
15:  $j \leftarrow 0$ 
16: while  $r \geq 0$  do
17:    $j \leftarrow j + 1$ 
18:    $s \leftarrow s - w_j$ 
19: end while
20:  $h_i \leftarrow N_j$ 
```

might require a long computational time prior to their identification. Moreover, by forbidding the niches to *dissolve*, we aim to tackle modularity’s resolution limit. In order to achieve that, we decided to rely on *repairing* procedures — one for each of the three chromosome representations — which would be applied to newly generated offsprings which number of groups would make them belong to other niches. The repairing procedures have the clear goal to maintain a stable population in each niche, making thus the exploration of the search space as uniform as possible. It is worth pointing out that the niching methods implemented do not consider chromosome migrations, unlike we investigated in our previous study [88].

However, if on one side the partitioning of the search space into “*sealed*” niches might allow for the evolution of concurrent local optima solutions, hence increasing the chances to detect articulated community structure partitions, on the other hand we face the problem to dedicate unnecessary long evolutionary time to regions of the search space which are very far from the *true* one. For the problems under investigation in this dissertation, for each network observed, there exists one and only one *true* community structure, hence only one global optima. Furthermore, one of the requirements of our group modelling framework is the ability to work online, which means that most of the computational time, i.e. evolutionary generations, should be mostly dedicated to promising niches, since the stochastic evolutionary process introduces uncertainty on *when* and *if* the *true* solutions are found. In order to solve the *exploration* of the subregion-partitioned search space in order to seek for new promising niches, and the *exploitation* of good ones for the refinement of current local optima, in this dissertation we will consider four simple *niching activation mechanisms*.

The next Sections will present the algorithmic choices we adopted in order to implement the *sealed* niches and their selective activation.

Algorithm 16 Edge Representation - Chromosome Mutation

```
1: sample gene  $i \in [1, n]$  uniformly
2: while  $r < p_{mut}$  do
3:   mutate_repair( $i$ )
4:   sample gene  $i \in [1, n]$  uniformly
5: end while
```

7.7.1 Niching Repair Procedure

In the method we implemented, we consider the existence of n niches, with each niche k being populated by chromosomes which would represent the partitioning of the network in exactly k groups. The niching repair procedure has the duty to make sure that all the chromosomes of niche k would represent network partitioning into exactly k communities, so that the population size is stable, and the niche would not dissolve throughout the evolutionary process. Clearly, the repair procedures depend on the chromosome representation used.

For the node and group representation, the procedure first extracts, for each chromosome of a given niche k , the list of unique labels (this is trivially retrieved in the group representation). The size of this list corresponds to the number of groups inferred by the chromosome. As long as this number is higher than k , the procedure will perform the “group death mutation” operator, which pseudocode is presented in Algorithm 13. On the other hand, as long as the number of inferred groups is lower than k , the procedure will introduce new symbols among the genes of the chromosome which are not a single-node group.

For the edge representation, the procedure first decodes, for each chromosome of a given niche k , the related subnetworks, hence community structures, hence number of inferred groups. As long as this number is higher than k , the procedure will uniformly sample nodes belonging to different subnetworks, and make one of them pointing to the other one, providing however that the corresponding edge exists in \mathcal{N} . On the other hand, as long as the number of inferred groups lower than k , the procedure will introduce one-size groups by (1) uniformly sample a gene i , (2) set its allele value to i , and (3) change all allele values i of genes j into j . It is worth pointing out that this repair procedure introduces self-linking nodes. This implies the introduction of artefacts in the cooperation network — an individual does not interact with himself in the experimental protocol we considered in this dissertation, meaning that $C_{i,i} = 0 \forall v_i \in \mathcal{S}$ and hence $\forall \mathcal{N}$ considered, see Chapter 4.

Furthermore, we are aware that the two niche repair procedures are rather naïve, which might introduce artefacts in the genetic populations. Our original goal was to implement recombination operations, for the three chromosome representation, which would allow for the creation of offsprings with exactly the same number of groups as their parents, though we could not achieve it. The investigation of different niche repair mechanisms, or even the reshaping of the evolutionary process, for instance by means of using two feasible and infeasible populations [113], are left to future work.

7.7.2 Niching Activation Mechanisms

In order to find a tradeoff between exploring the niche/subregions of the search space — in order to search for new promising solutions — and exploiting currently well performing ones — so that the niches’ local optima can be refined, possibly towards the *true* global optima — in this dissertation we focus on four techniques for niche activation, presented in the next Subsections. All of them make their decisions based on a mono-dimensional niche landscape composed of each niche’s highest fitness score recorded at the end of the previous evolutionary generation. Intuitively, since there is only one partition of \mathcal{N} into either one or n groups, the niching activation mechanisms will operate on a landscape of $n - 2$ niches. Further studies centred on the use of additional features, such as historical evolutionary information, genetic population diversity, or even aggregate information of the population fitness scores, are left to future work.

Algorithm 17 Armed Bandit Niching Activation — ϵ greedy [12]

- 1: set $c > 0$ and $0 < d < 1$
 - 2: define the sequence $\epsilon_t \in (0, 1], t = 1, \dots$ by $\epsilon_t = \min \left\{ 1, \frac{cn}{d^2 t} \right\}$
 - 3: **for** $t = 1 \dots$ **do**
 - 4: let k_t be the niche with highest fitness score
 - 5: with probability $1 - \epsilon_t$ evolve niche k_t and with probability ϵ_t evolve a random niche
 - 6: **end for**
-

7.7.3 Reading Head Niching Activation

This algorithm is derived from the promising results obtained from the NicheShift algorithm [88], which we designed to solve the CSD task in complete directed and weighted networks. The reading head algorithm can be considered as a hill climbing algorithm. Once the $n - 2$ niches are initialised, and the niche landscape is retrieved, a *pivot activator* is set to point at niche k , which scores the highest landscape value. At each evolutionary loop, the reading head mechanism will evolve z consecutive niches centred at niche k . When the z niches have evolved, the niche landscape is update with the new fitness scores. Subsequently, the pivot activator is updated in order to point to the current best performing niche.

The rationale behind this deterministic activation mechanism resides on the assumption that, providing enough evolutionary time, the closer the niches are to the *true* one, the higher their highest fitness score would be. The window of size z allows for the landscape of the niches closer to the *true* one to evolve, so that the direction to be taken by the pivot activator — either to the left, right, or none (in case it is already pointing to the *true* niche) — can attempt to solve the exploration vs. exploitation dilemma, though more emphasis is given to the exploitation phase.

7.7.4 Roulette Selection Niching Activation

This niching activation mechanism resembles the homonymous parent selection mechanism used in many evolutionary algorithms (see Section 3.3.3). The normalised fitness landscape of the niches is built as follows:

$$\hat{f}_i = \frac{f_i}{\sum_{k=2}^{n-2} f_k}$$

where f_i is the highest fitness score recorded in niche i . Subsequently, a random number r is sampled uniformly within the $[0, 1)$ interval. The niche to activate is then retrieved. Intuitively, through this activation mechanism, all niches — also those far from the *true* one — have a non-zero chance to evolve, meaning thus that the exploration vs. exploitation dilemma is attempted by favouring exploration.

It is important to note that this activation mechanism, similar to the corresponding parent selection procedure, might excessively suffer from a smoothed niching landscape, that is a landscape in which the fitness scores are nearly equal to each other. The consequence is a nearly uniform distribution of niche activation, which would limit the exploitation for the most promising niche.

7.7.5 n -armed Bandit Niching Activation

This niching activation mechanism is directly inspired from the corresponding n -armed bandit technique studied in reinforcement learning [199, 12], see Section 3.4. We here consider two possible armed bandit techniques: ϵ -greedy and UCB1. Their pseudocode procedures are listed in Algorithms 17 and 18.

The decision to focus on these two algorithms were the fact that, in general, they aim to solve a reinforcement learning problem which is very similar, at least conceptually, to the niching activation problem. More in particular, we relied on the UCB1 algorithm since it attempts to solve the exploration vs. exploitation dilemma in a deterministic way, whilst ϵ -greedy attempts to solve it in a more stochastic fashion.

Algorithm 18 Armed Bandit Niching Activation — UCB1 [12]

- 1: # x is the overall number of evolutionary loops performed
 - 2: # x_k is the number of evolutionary loops performed by niche k
 - 3: evolve niche k that maximises $f_k + \sqrt{\frac{2\ln x}{x_k}}$
-

7.8 Summary

This chapter described, thoroughly, the second and last module of our proposed interaction-based group modelling framework, namely Group Identity Detection (GID). The GID’s task corresponds to process the cooperation network, output of the framework’s first component (cooperation modelling), and identify the group identities of the society. In order to achieve so, the GID module performs the community structure detection task by means of modularity maximisation, being it possibly the most established technique. Moreover, we decided to rely on evolutionary computation for modularity maximisation, due to many reasons, the most important which is its robustness against noisy and dynamic environments, and its promising results in related work on unweighted and undirected networks.

We considered three different chromosome representations, focused on either the *node* or the *edges* of the network, or the *groups* that might exist within the cooperation network. We also specified their relative chromosome initialisation and recombination procedures. Furthermore, we embarked the exploration of a more structured evolutionary approach, based on niching, in which each niche limits the evolutionary search to candidate solutions which would lead to a pre-defined number of community structures/groups. Through niching it might be possible to overcome some apparent drawbacks of single population-based evolutionary algorithms, such as the vastness of the search space. The niching approach could allow for restricting the exploration of the search space — which is mostly composed of suboptimal solutions for the class of problems considered in this dissertation — providing however the existence of selective niche activation mechanisms, which would allow for the evolution of relevant promising niches/subregions of the search space only. In this dissertation we considered four methods for niching activation, ranging from the pure deterministic one, that is the reading head and armed bandit — UCB1 mechanisms, to probabilistic ones, that is roulette selection and armed bandit — ϵ -greedy mechanisms.

The empirical evaluation of the 15 possible evolutionary algorithm configurations are reported in the next Chapter 8.

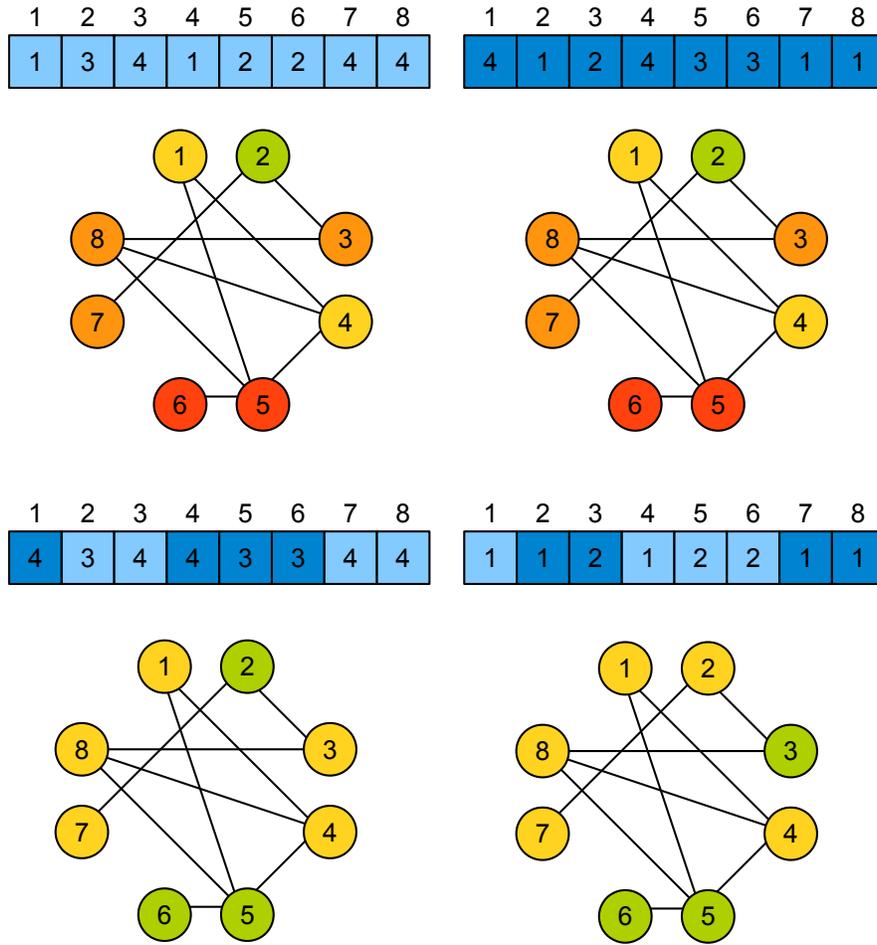


Figure 7.3: An example of uniform crossover operation, performed at the node part, which motivated Falkenauer to introduce the group representation [55]. The two parents, depicted on top of the figure, lead to the same partitioning into four groups. Under a group perspective, the generated offsprings should be clones of the parents. However, through node-based (uniform) crossover there is a risk that the generated offsprings (at the bottom of the figure) would lead to partitioning much different from the original parents, in this case both of them leading to two group structures.

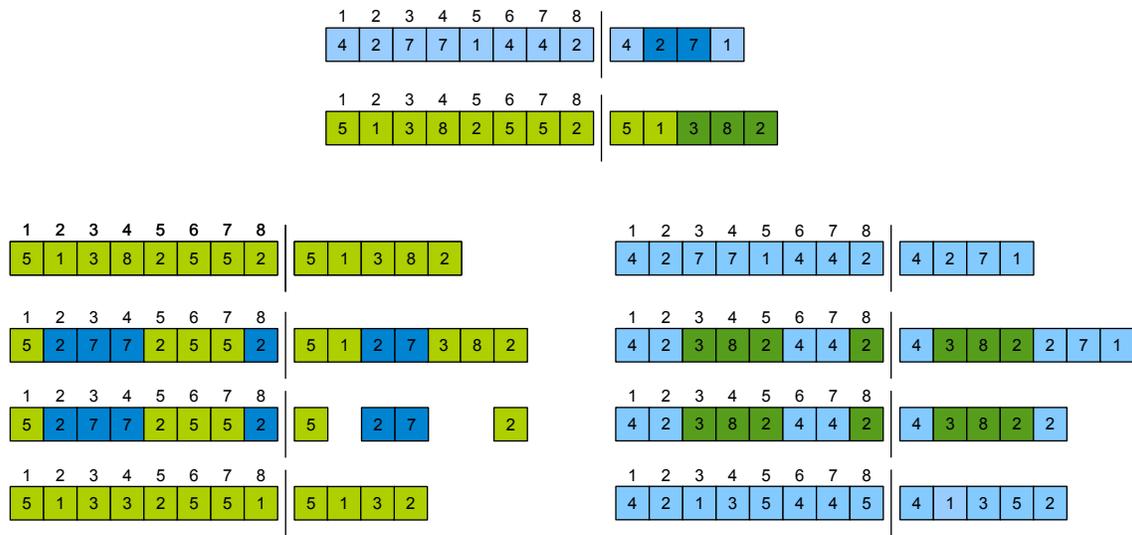


Figure 7.4: An example of a group representation-based crossover operator. The two parent chromosomes are depicted at the top of the figure, the two generated offspring are depicted at the bottom. The intermediate steps of the two crossover operators are depicted in between.

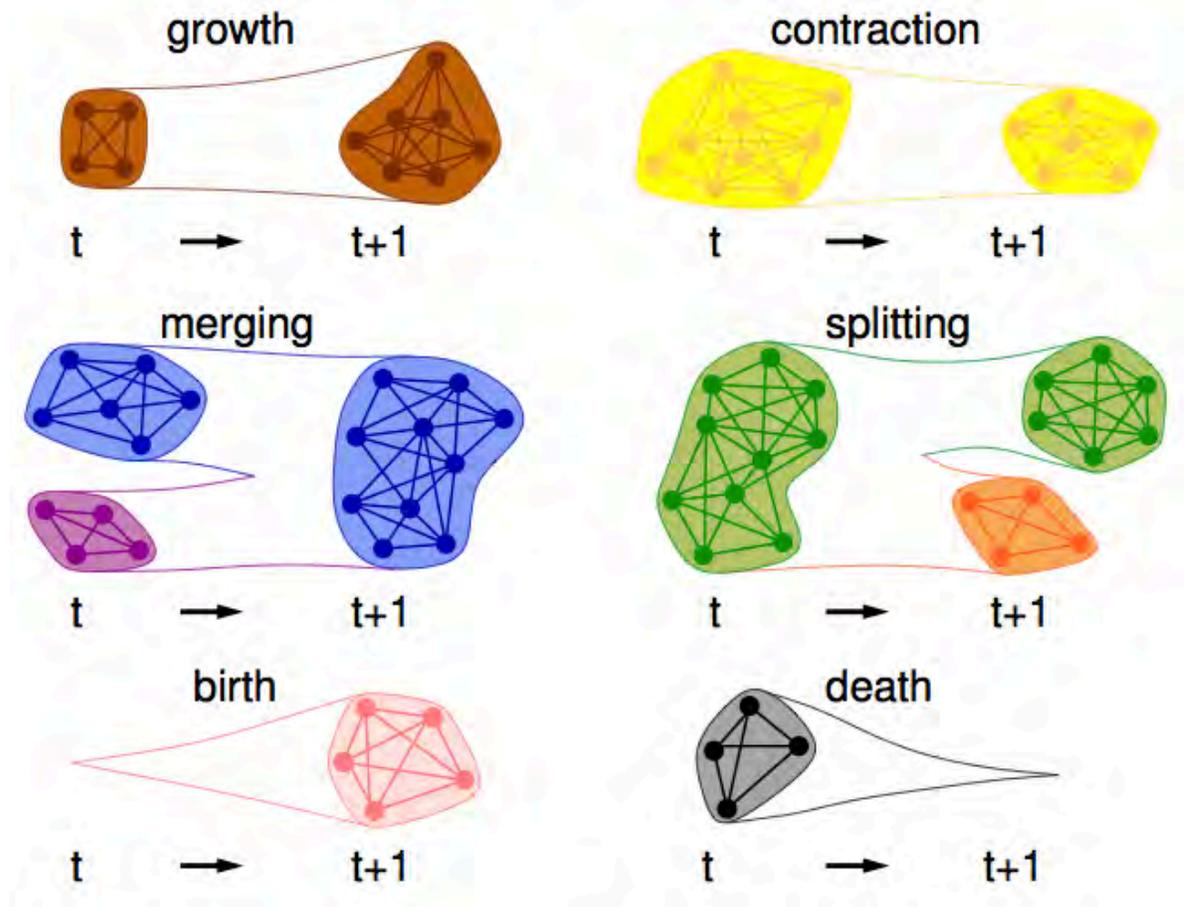


Figure 7.5: The six possible events in group evolution defined by Palla et al. [170] which inspired our group-based mutation operators. Each of the six mutation operations are presented in pseudocode in Algorithms 8 to 13.

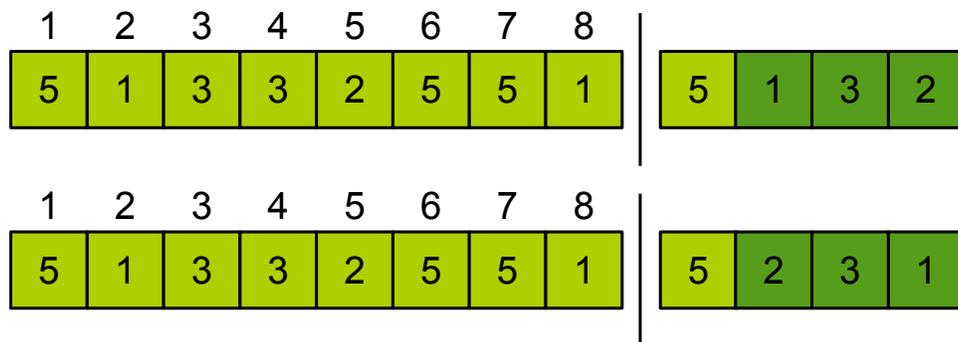


Figure 7.6: An example of inversion operation for group representation. The inversion operator, applied to the dark green labels of the group part, transforms the topmost into the lowermost chromosome, leaves the node part unaltered.

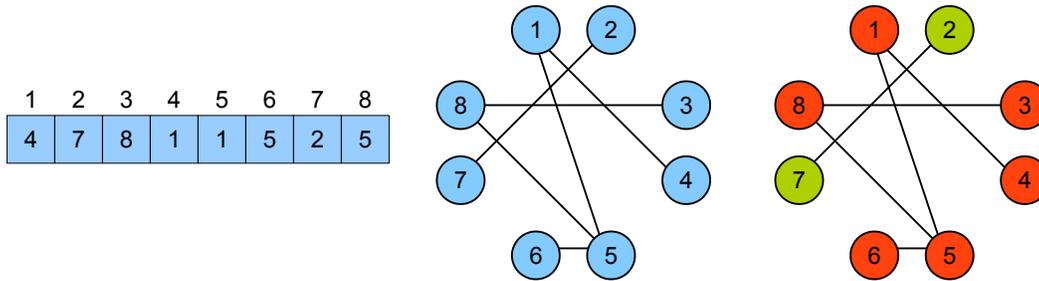


Figure 7.7: An example of an edge representation-based chromosome. The reference network is the one of Figure 3.1. The allele value j of gene i is translated into the edge (i, j) . The subsequent conversion step will detect the chromosome subnetworks and then assign group identities.

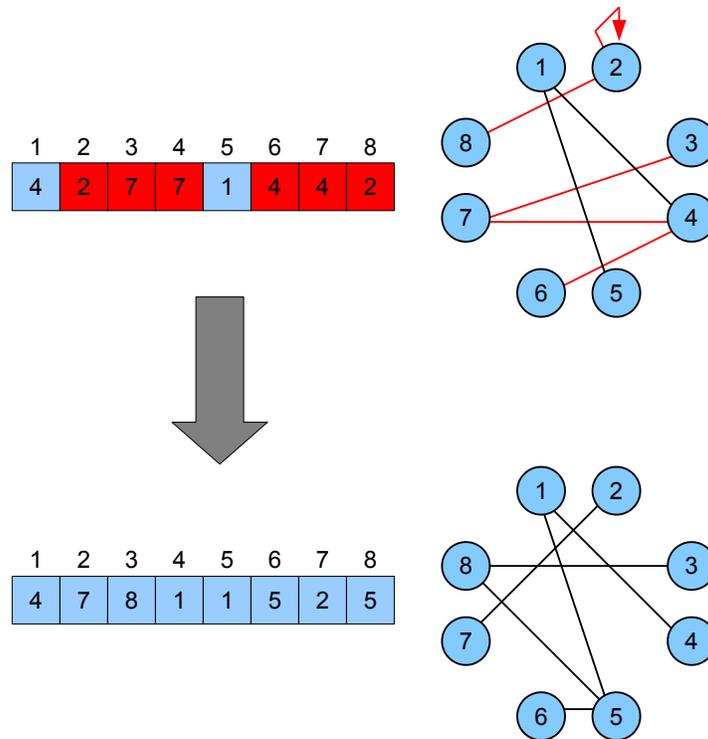


Figure 7.8: An example of chromosome repair in an edge representation-based chromosome. Genes 1 and 5 are the only ones which allele values correspond to edges present in adjacency matrix of Figure 3.1. The other six genes are repaired; their allele values are chosen among their corresponding neighbouring nodes proportionally to their weights. Intuitively, in case of unweighted networks, each neighbouring node is chosen according to uniform distribution.

Chapter 8

Group Identity Detection Empirical Evaluation

This Chapter presents a thorough empirical analysis of the second and last component of our proposed interaction-based group modelling framework, that is, the Group Identity Detection (GID) module. Once again, the duty of the GID module is to process the output of the first group modelling component — that is the representation of the ongoing social dynamics by means of a cooperation network (see Chapter 5) — and to perform Community Structure Detection (CSD), with the aim to maximise *within*-group, and at the same time minimise *between*-group cooperation, so that the existing *true* group structures can be unveiled [39] (see Chapter 7). As a consequence, although the main emphasis of this Chapter is on the GID module, the results presented can also be interpreted as an evidence of the effectiveness of our interaction-based approach.

The Chapter is organised as follows: in Section 8.1 we introduce the two performance measures central to the evaluation of our experiments. Then, in order not to overcomplicate the conceptual flow of this Chapter, especially due to the huge amount of data gathered, we present the most important empirical results as follows:

- first, we investigate the impact the chromosome representation — which constitutes the core of an Evolutionary Algorithm (EA) — has on the performance of the GID framework (Section 8.3);
- subsequently, we deepen the research on the most promising chromosome representation; more specifically, we investigate whether an alternative evolutionary computation approach, based on niching, might improve the inference of the *true* group structures (Section 8.4);
- lastly, we investigate whether we could gain performance advantages by relying on an alternative fitness function — i.e. LinkRank as opposed to modularity (Section 8.5).

All the secondary results can be found in Appendix 14. The core experimental scenario of this investigation is based on the artificial societies derived from the extracted data of the original experiment by Chen and Li (see Section 4.3).

8.1 Performance Measures

Given:

- a (cooperation) network $\mathcal{N} = (\mathcal{S}, \mathcal{C})$, where $\mathcal{S} = \{a_1, a_2, \dots, a_n\}$ is the set of n nodes, and \mathcal{C} is the adjacency matrix/set of edges connecting the nodes in \mathcal{S} ;
- the *true* partitioning of \mathcal{S} into G community structures, and the classification of each node in \mathcal{S} into a unique *true* community structure identity $\mathcal{T} = \{g_1, g_2, \dots, g_n\}$;

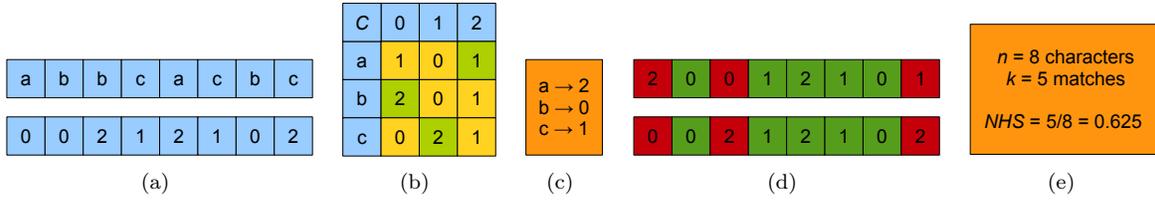


Figure 8.1: An example depicting how the Hungarian algorithm, and consequently NHS , compute the string matching score between two strings. Figure 8.1(a) depicts the two strings that need to be aligned; Figure 8.1(b) is the related confusion matrix; Figure 8.1(c) corresponds to the optimal alphabet assignment computed by the Hungarian algorithm; Figure 8.1(d) depicts the calculation of matches and mismatches after the alphabet assignment; and finally Figure 8.1(e) normalises the alignment score NHS .

- the inferred community structure identities $\mathcal{H} = \{h_1, h_2, \dots, h_n\}$, output of the GID computation, which partition the nodes in \mathcal{S} into H community structures;

the so-called *confusion matrix*

$$C = G \times H$$

is defined. C is composed of G rows and H columns, and each element $C_{i,j}$ corresponds to the number of nodes of community i of partition \mathcal{T} which belong to community j of partition \mathcal{H} . An example of confusion matrix, related to two input strings, is presented in Figure 8.1(b) and 8.1(a) respectively.

We will describe how well how \mathcal{H} approximates \mathcal{T} through two performance measures, namely the *Normalised Mutual Information* and the *Normalised Hungarian Score*, hereafter defined. Both measures rely on the confusion matrix previously defined.

8.1.1 Normalised Mutual Information

The *Normalised Mutual Information* (NMI) is a measure extensively used in network theory, see e.g. [37]. NMI considers information-theoretic concepts and is based on the fact that if two communities are similar to each other, then only a small amount of additional information is needed to infer one community assignment from the other [144, 37]. NMI is calculated as follows:

$$NMI(\mathcal{T}, \mathcal{H}) = \frac{-2 \sum_{i=1}^G \sum_{j=1}^H C_{i,j} \log \left(\frac{n C_{i,j}}{C_{i,*} C_{*,j}} \right)}{\sum_{i=1}^G C_{i,*} \log \left(\frac{C_{i,*}}{n} \right) + \sum_{j=1}^H C_{*,j} \log \left(\frac{C_{*,j}}{n} \right)} \quad (8.1)$$

where $C_{i,*} = \sum_{j=1}^H C_{i,j}$ and, intuitively, $C_{*,j} = \sum_{i=1}^G C_{i,j}$. NMI ranges between zero (complete unmatched) to one (complete match).

8.1.2 Hungarian Algorithm Matching

Despite being possibly the most widely used measure in CSD-based problems, NMI has a main drawback: it is not possible to directly understand how many node assignments are actually mismatched. For this reason, in our previous work [83, 87, 88, 89] we decided to rely on another, more direct measure, obtained by means of the *Hungarian algorithm* [117].

Given two strings \mathcal{T} and \mathcal{H} with two distinct alphabets, the Hungarian algorithm, similarly to the NMI calculations, firstly builds the strings' confusion matrix C . Then, based on C , it seeks for the alphabet assignment — from \mathcal{T} to \mathcal{H} — which maximises the string alignment. The algorithm returns the number of nodes $k \in [0, n]$ of the string alignment. In our investigations we will present the normalised score of the string alignment, NHS , trivially obtained as follows:

Table 8.1: Recap of the CLAS societies with groups of equal size implemented in our empirical evaluation.

m	Group Size Composition	Average Group Size	$p(in - in)$	$p(out - out)$	$p(in - out)$
4	4 * 32	32	0.0596	0.5714	0.369
7	(5 * 18) + (2 * 19)	18.29	0.0185	0.7463	0.2352
11	(4 * 11) + (7 * 12)	11.64	0.007	0.8395	0.1535
17	(8 * 7) + (9 * 8)	7.53	0.0026	0.8998	0.0975

$$NHS(\mathcal{T}, \mathcal{H}) = \frac{k = \text{Hungarian algorithm}(\mathcal{T}, \mathcal{H})}{n} \quad (8.2)$$

Intuitively, NHS ranges between zero (complete unmatched) to one (complete match). Figure 8.1 depicts an example of two strings being aligned via the Hungarian algorithm and the related NHS score.

8.2 Description of the Artificial Society Scenarios

As briefly introduced in Section 4.3, we considered artificial societies derived from the authentic behavioural data registered by the experiment of Chen and Li (CLAS). All the CLAS are based on $n = 128$ agents. The motivation for such parameter is, clearly, to make our work comparable and easily extendible with the related community structure detection work. Moreover, we considered four different number of *true* groups, i.e. $m = \{4, 7, 11, 17\}$. Intuitively, the case $m = 4$ aims to further strengthen the connection and comparison of our results and approaches with the benchmark network initially proposed by Girvan and Newman [77]. The other cases, instead, were introduced in order to generate less trivial, and possibly more difficult to detect, *true* community structures. With this respect, for $m = \{7, 11, 17\}$, we decided to generate two types of group size distributions: groups of equal size, in accordance with Girvan’s and Newman’s benchmark network [77], and power law distributions, generated through the preferential attachment algorithm (see Section 4.3), in accordance with the claims against the *unrealisticness* of the $n = 128$, $m = 4$ equal size scenario [124]. Obviously, in case of $m = \{7, 11, 17\}$, the equal group sizes will differ, at most, of ± 1 agent/node. On the other hand, in case of power law group size distribution, the only constraint we implemented was to have groups of at least two agents, so that *in-out* interaction types could be generated, hence the necessary condition for the application of our interaction-based approach could be satisfied (see Section 5.1). The CLAS scenarios were based on 400 iterations of the *other-other* game. The permutations defining the 128 triplets proposer-receiver1-receiver2 will be generated according to a uniform distribution. Finally, for each of the seven society types, we generated 30 different CLAS implementations. Hence, overall, the empirical investigation of this Chapter is based on $(4 + 3) * 30 = 210$ societies.

Table 8.1 recaps the four society types with equal group size. As it can be understood, an increasing value for m would correspond to a lower probability of observing interactions between both *in-group* and *out-group* receiver agents ($p(in - out)$). As a consequence, increasing values for m would correspond to a higher probability to observe interactions between both *out-group* receiver agents ($p(out - out)$), making thus the identification of the *true in-group* and *out-group* boundaries more difficult (see Section 6.3).

Figure 8.2, instead, depicts the power law group size distributions of the three m types. The frequency and group size values are accumulated across the 30 experimental runs. As it can be easily seen, the case $m = 17$ presents the most striking power law distribution, followed by $m = 11$. On the other hand, for the $m = 7$ case, we observe a more “bell shaped” distribution. The mode for $m = 17$ is groups of 4 agents; the mode for $m = 11$ is groups of 6 agents; the mode for $m = 7$ is groups of 10 agents.

8.2.1 Analysis of the Cooperation Modelling Phase

To extract the cooperation networks \mathcal{N} needed to perform the GID task, based on the considerations made in Chapter 5 and findings presented in Chapter 6, we instantiated the cooperation modelling component based

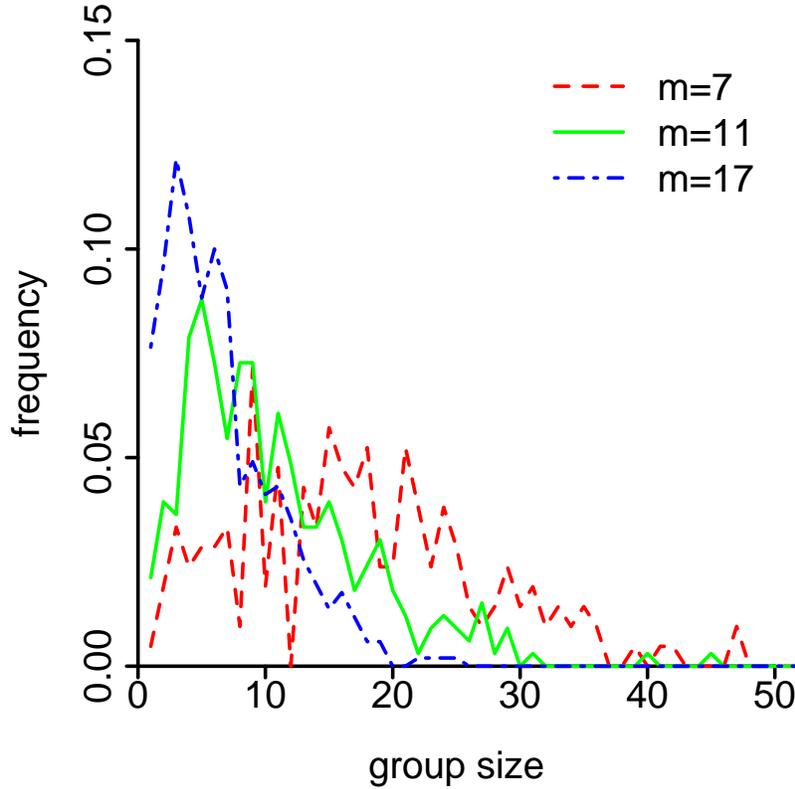


Figure 8.2: The power law group size distribution of the 30 CLAS societies considered for three m values are examined ($m = \{7, 11, 17\}$).

on the following settings:

- in order to aggregate the interaction values into the O_P vectors, needed for the cooperation evaluation task, we relied on the accumulation of $t = 10$ consecutive interactions of each agent a_P .
- for the cooperation evaluation phase, the deterministic interaction classifier was used, see Equation (5.3);
- for the cooperation network update phase, constant- α update rule was used, see Equation (5.10). Moreover, we relied on $\alpha = 1$;
- The initial weights of \mathcal{N} are all set to 0.5, in order to situate the cooperation values halfway between *in-group-ness* and *out-group-ness*.

The settings imply that the GID module will be applied on $400/10 = 40$ cooperation networks; these are incrementally build on top of each other and will evolve, from a fully connected, directed and weighted network, with weights 0.5 — corresponding to full unbiased uncertainty — towards a sparse, symmetric (i.e. undirected) and unweighted, corresponding to the *true* group’s related network.

Figure 8.3 depicts the average modularity score, and related standard deviation, of the 30 implementations of each of 7 society types considered in our empirical evaluation. The modularity adopted in this study, as well as in the remainder of this Chapter, is the most generic one which can be applied to directed and weighted networks — see Equation (3.6). The score is based on the up-to-date cooperation network after the cooperation network update task, and is calculated considering the CLAS *true* group structures, which were generated at the beginning of each simulation.

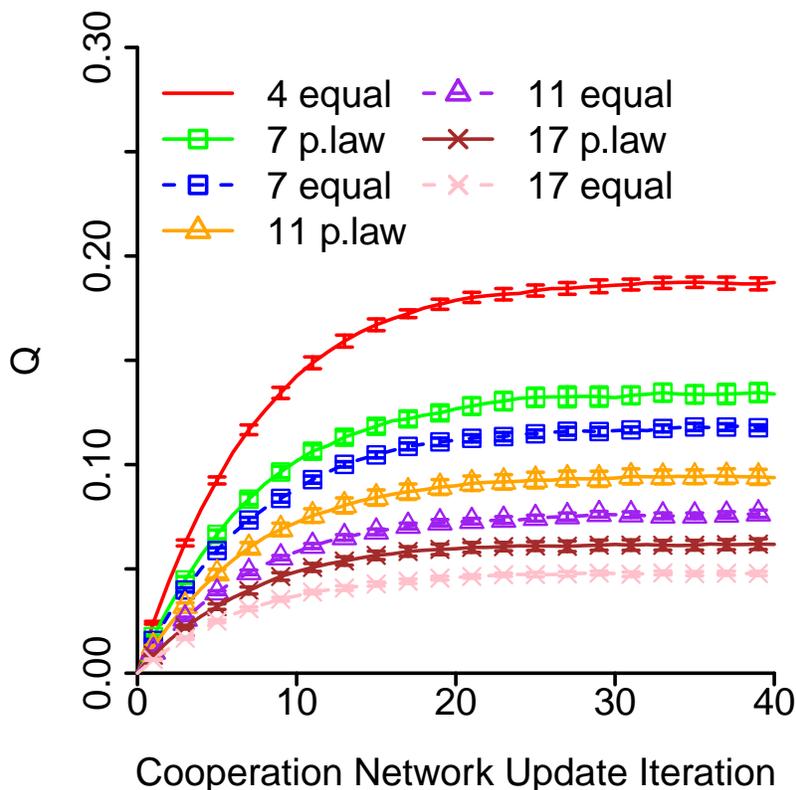


Figure 8.3: The average modularity score, and its standard deviation, of the *true* groups over the ongoing cooperation network, inferred by the CM module, depending on the seven society types considered for the GID empirical evaluation.

Not surprisingly, the case $m = 4$ is the simplest we would have to deal with: its highest average modularity is 0.1874 (standard deviation is 0.00498), which corresponds to a rather good community structure partition¹. Modularity decreases with the increase of m , up to a highest average value of 0.048 for $m = 17$ equal size society scenario (standard deviation is 0.0018); this is clearly due to the increasing noise of the other-other game observations, which would be more likely *out-out* interaction types rather than *in-out* ones.

The standard deviation of the equal size types is generally smaller than the one registered for the corresponding power law counterparts; this is not surprising, since the $p(in - in)$, $p(in - out)$ and $p(out - out)$ vary in case of non-equal sized groups, making thus the societies and their observations more or less noisy, or, alternatively, easier to find large sized communities. Consequently, it is important to point out that the average modularities of power law group distribution types are significantly higher than the corresponding equal sized ones.

8.3 Chromosome Representation Analysis

The main aim of the investigation presented in this Section is to understand how much the chromosome representation affects the performance of the GID module. Moreover, in order to situate it, we will consider, as baseline *NMI/NHS* performance, the one obtained by means of spectral partitioning with fine tuning (SP), introduced in Section 3.2.2.

¹Recall that observed real networks with clear community structures register a modularity score usually ranging between 0.2 and 0.3

Table 8.2: Key properties of the three evolutionary algorithms examined.

Chromosome Representation	Ad hoc Initialisation	Crossover	Mutation	Reference Section
Node	No	Uniform	Node Labelling	7.4
Group	No	Group-based	Group Dynamics	7.5
Edge	Chromosome Repair	Uniform	Neighbouring Node Edges	7.6

Table 8.3: Limit of evolutionary fitness evaluations considered in our GID study based on spectral partitioning’s average (avg) number evaluations and standard deviation (sd).

society type	SP		EA
	avg	sd	
4-equal	15035	321	15000
7-equal	16210	319	16200
11-equal	16061	504	16050
17-equal	16051	537	16050
7-powerlaw	16110	664	16050
11-powerlaw	16033	532	16050
17-powerlaw	16039	527	16050

In accordance with Chapter 7, we here consider three chromosome representations which are used in a common, generic evolutionary algorithm, namely node (N-EA), group (G-EA) and edge (E-EA); their structure is depicted Figure 3.7 and represented in pseudocode in Algorithm 1. The three representations share the same population size of 60 individuals, elitism of 50% of the population, mutation rate $p_{mut} = 0.8$, probability of parent reproduction $p_{cross} = 0.8$, and finally the probabilistic rank parent selection procedure. Moreover, we decided to execute only one iteration of each EA for each of the 210 cooperation networks; the same choice will then be applied to the other EAs presented in the subsequent Sections. Table 8.2 recaps, instead, the properties of each EA used. It is worth noting that E-EA, each time it receives a new cooperation network, it will perform the chromosome repair procedure on its current population.

In order to make a reliable comparison between spectral partitioning and the three EAs, we decided to stop the latter once the number of fitness evaluations — excluding the population initialisation phase — would be as close as possible to the average number of modularity evaluations obtained by spectral partitioning for the same society type. Table 8.3 lists these fitness evaluation limits. At the end of its computation, each EA retrieves the chromosome which scores the highest fitness value. The corresponding network partitioning into community structures — i.e. \mathcal{H} — will be its output. Hence, its related NMI/NHS scores are computed. The analysis of performance of each algorithm considered will take into account both the highest NMI/NHS scores recorded, and the speed these converge to high inference values.

Finally, for all the studies presented hereafter, we decided to detach the analysis of the $m = 4$ equal case from the other six society types for two main reasons: (1) this scenario does not have a corresponding power law distribution case, since we have seen that for decreasing values of m the power law distribution tends to generate groups of more even sizes, see Figure 8.2; (2) in general, the results gathered are significantly better than those related to the other six society types, as it can already be hinted by observing Figure 8.3.

8.3.1 Analysis of $m = 4$ Case

Figure 8.4 depicts (a) the average NMI and standard deviation, (b) the average NHS and standard deviation, and (c) the average inferred number of groups and standard deviation, for $m = 4$ equal size society type, for the three chromosome representation-related EAs and spectral partitioning.

Spectral partitioning appears to be, clearly, the best approach for CSD in this scenario: not only it reaches nearly optimal NMI and NHS scores (avg = 0.9972, sd = 0.0075 and avg = 0.9987, sd = 0.0036 respectively), but it is also the fastest growing algorithm: the first time it scores the maximum value (i.e. 1), which implies

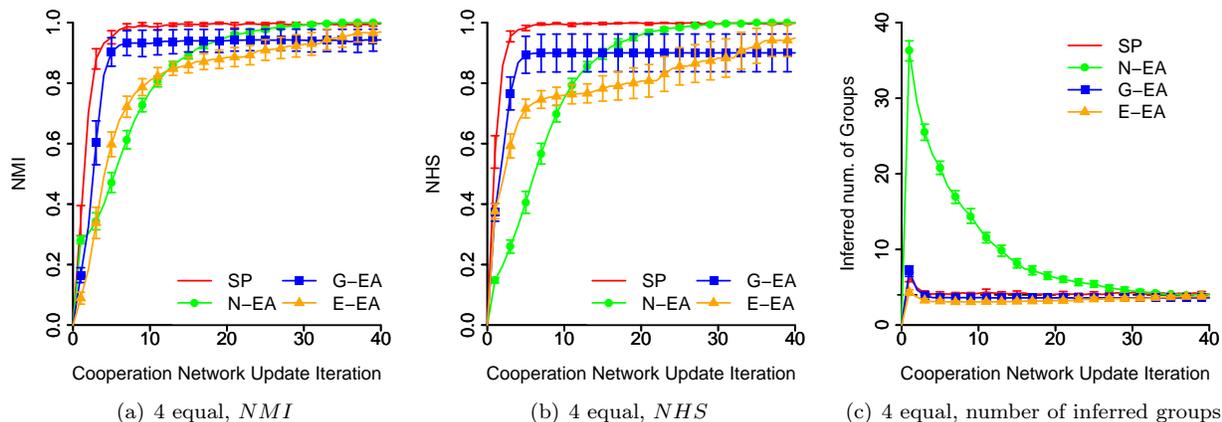


Figure 8.4: Performance (NMI , NHS , and inferred number of groups) of the three chromosome representations compared to spectral partitioning for $m = 4$ equal size group types.

an exact matching between inferred and *true* group identities, is registered just after four network updates (i.e. just after 40 interactions performed by each agent in the society), for 3 out of 30 networks. Moreover, in these three networks, the average ratio of not yet updated edges was 0.5294, standard deviation 0.0007, meaning thus that the networks, at those early stages, were highly noisy.

Among the three chromosome representations, however, we observe such a remarkable result, independently on the performance measure, solely obtained by the node representation, though its speed of convergence to optimal values is much slower than SP. At the end of its computation, N-EA even scores a better average performance than spectral partitioning — avg = 0.9996, sd = 0.0022 for NMI , avg = 0.9997, sd = 0.0014 for NHS — though the difference is not statistically significant under any of the two performance measures (p-value = 0.098, t = 1.6818, for NMI ; p-value = 0.1615, t = 1.4180 for NHS , df = 58). The reason why N-EA has a slower speed of convergence can be seen in Figure 8.4(c): due to its non-ad hoc initialisation process, and to its mutation operator, which both allow the genes to have allele values ranging between [1, 128], N-EA requires many more evolutionary generations to refine its candidate solutions towards those which assume the existence of 4 groups.

Among the three EAs, G-EA is the fastest algorithm to converge to high performance values, though its overall best scores are suboptimal — avg = 0.9428, sd = 0.0712 for NMI , avg = 0.9003, sd = 0.1242 for NHS . Moreover, G-EA has an inference trend much similar to spectral partitioning: it converges rather quickly to the *true* number of groups, and subsequently cannot improve its best performance. In other words, it appears that G-EA cannot benefit from new, more accurate cooperation networks, to improve its candidate solutions.

On the other hand, we observe a mixed trend of E-EA performance: for about the first five-to-ten observed networks its matching improvements is quicker than N-EA, especially for NHS . Then, its improvement speed slows down dramatically, though it appears that the algorithm’s performance trend has not converged yet, unlike N-EA and G-EA. Ultimately, E-EA’s performance settles between N-EA and G-EA — avg = 0.9692, sd = 0.0583 for NMI and avg = 0.9492, sd = 0.1014 for NHS , though the difference in performance with N-EA is statistically significant (p-value = 0.0060, t = 2.8540 for NMI , p-value = 0.0084, t = 2.7275 for NHS , df = 58). Most likely, E-EA could even reach higher scores, providing however that it could let its population evolve beyond the limits defined in Table 8.3. The dependency of E-EA on the underlying cooperation network is observed by considering that even if its estimation of the number of groups quickly converges to the *true* $m = 4$ value (see Figure 8.4(c)), its NMI/NHS scores keep improving — unlike G-EA — in accordance with the increasing *true* modularity scores registered, see Figure 8.3. In other words, with the ongoing network update tasks, the observed cooperation networks get more sparse and accurate in terms of highlighting the *in/out-group* boundaries. As a result, (1) the chromosome repair procedure effectively repositions E-EA’s scope in more suitable subregions of the search space, and (2) the mutation operator will

allow for the selection of less numerous neighbours, most of them being the *true in-group* ones.

To conclude, the standard deviations of both E-EA and G-EA are generally higher than N-EA’s throughout the whole modelling task, and especially towards the end of the observed cooperation networks, for which N-EA’s standard deviation tends to zero. This indicates that E-EA and G-EA might be overly dependent on the underlying network’s noise, that is, they might not be able to improve early erroneous network partitions.

8.3.2 Analysis of $m = \{7, 11, 17\}$ Cases

Figures 8.5 and 8.6 depict, respectively for equal size and power law group size distribution, the average and standard deviation of the *NMI* and *NHS* scores, and inferred number of groups, of spectral partitioning, N-EA, G-EA, and E-EA, for $m = \{7, 11, 17\}$.

Undoubtedly, N-EA appears not only to be the best chromosome representation across all the six scenarios in terms of final *NMI* and *NHS* scores, it also manages to outperform spectral partitioning. However, N-EA still suffers from its slow convergence to optimal values, especially for small m values. On the other hand, for increasing values of m , we observe that N-EA becomes the best approach at any stages of the group identity detection task, see for instance Figures 8.5(g) and 8.5(h). Moreover, by observing the *NMI* and *NHS* trends, N-EA seems not to have reached a plateaued performance yet, see for instance the 11 equal or the 17 power law case. It is likely that the algorithm would even improve its scores, providing, for instance, that it could let its population evolve beyond the limits defined in Table 8.3.

The peculiar trend by which N-EA evolves its solutions — i.e. in an aggregative fashion — is seen across all the six scenarios, similarly to what we observed for the 4 equal case. N-EA infers, on average, the *true* number of groups for both $m = 7$ cases (avg = 6.9667, sd = 0.4138 for equal size, avg = 6.6667, sd = 1.0613 for power law group size distribution) and $m = 11$ equal size case (avg = 11.2333, sd = 1.3565); for the remaining three cases, instead, the inferred number of groups is smaller than the actual one, though it is still the closest among the other algorithms.

The group and edge representations appear not to be suitable for the class of problems we aim to solve. Among the two, G-EA is slightly better performing than E-EA. More specifically, in both $m = 7$ cases, it manages to have an initial performance advantage, though it is then lost throughout the group identity detection task. With respect to the $m = \{11, 17\}$ cases, instead, we observe that the two performance trends are nearly parallel, with G-EA’s recorded scores being above those of E-EA. The suboptimal performance of both G-EA and E-EA is seen especially by looking at the inferred number of group plots, which show their impossibility to adjust their evolutionary trend towards the *true* m values, and instead quickly converge to about four groups, independently on the society case under investigation.

The leading causes for G-EA’s suboptimal performance are possibly due to its highly structured recombination operators. With this respect, preliminary work was conducted in order to examine whether a subset of group-based mutation operations might be more beneficial than others. As result, we could not find any relevant performance patterns which would allow for the discarding some operations and keeping others. For instance, we observed that getting rid of “group birth” and “group death” mutation operators were more beneficial for the $m = 7$ equal case, though it was not the same for $m = 11$ equal case, for which instead the “group growth” and “group contraction” operators seemed not to be beneficial. We also tried to introduce the same mutation operator adopted by the node representation, but we did not observe any substantial performance improvements. Possibly, hence, a lower mutation rate might improve G-EA’s *NMI/NHS* scores — a high mutation probability for the “group growth” mutation operator would cause the extinction of many small-sized groups. Moreover, the fact that the group-based crossover transfers a variable amount of groups at the same time, if on one hand it allows for the generation of less redundant offsprings, on the other hand it might limit the capability of exploring the search space towards higher number of groups, as it can be observed by the rather quick plateauing of G-EA’s performance at about halfway through the cooperation network update phase.

On the other hand, E-EA cannot effectively rely on its chromosome repair procedure to steer the evolutionary search, with consequent repercussions on the effectiveness of its mutation operator. In support of this claim, see the *true* modularity scores depicted in Figure 8.3. These, as opposed to the $m = 4$ case, record much lower *true* Q values, meaning that the cooperation networks are far from being sparse, and that

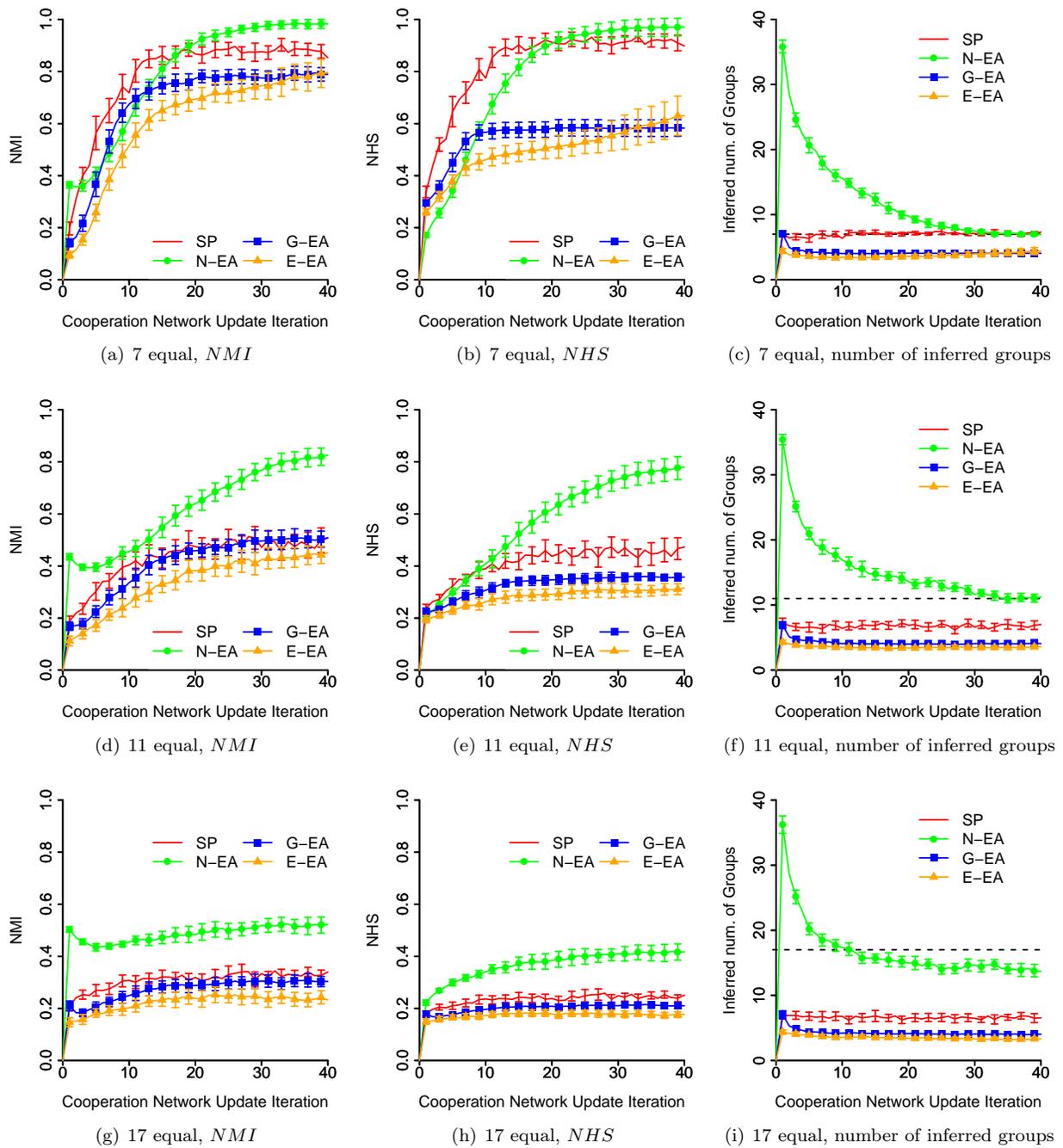


Figure 8.5: Performance (NMI , NHS , and inferred number of groups) of the three chromosome representation-based EAs compared to spectral partitioning for $m = \{7, 11, 17\}$ equal group size distribution.

many of their edges connect nodes belonging to different *true* groups. As a consequence, the chromosome repair procedure cannot correct the candidate solutions by “breaking” subnetworks joined by *true out-group* (i.e. wrong) edges. Moreover, each node of the network has many *out-group* neighbours, meaning that the mutation operator will have a high chance to further “merge” subnetworks corresponding to different *true* groups.

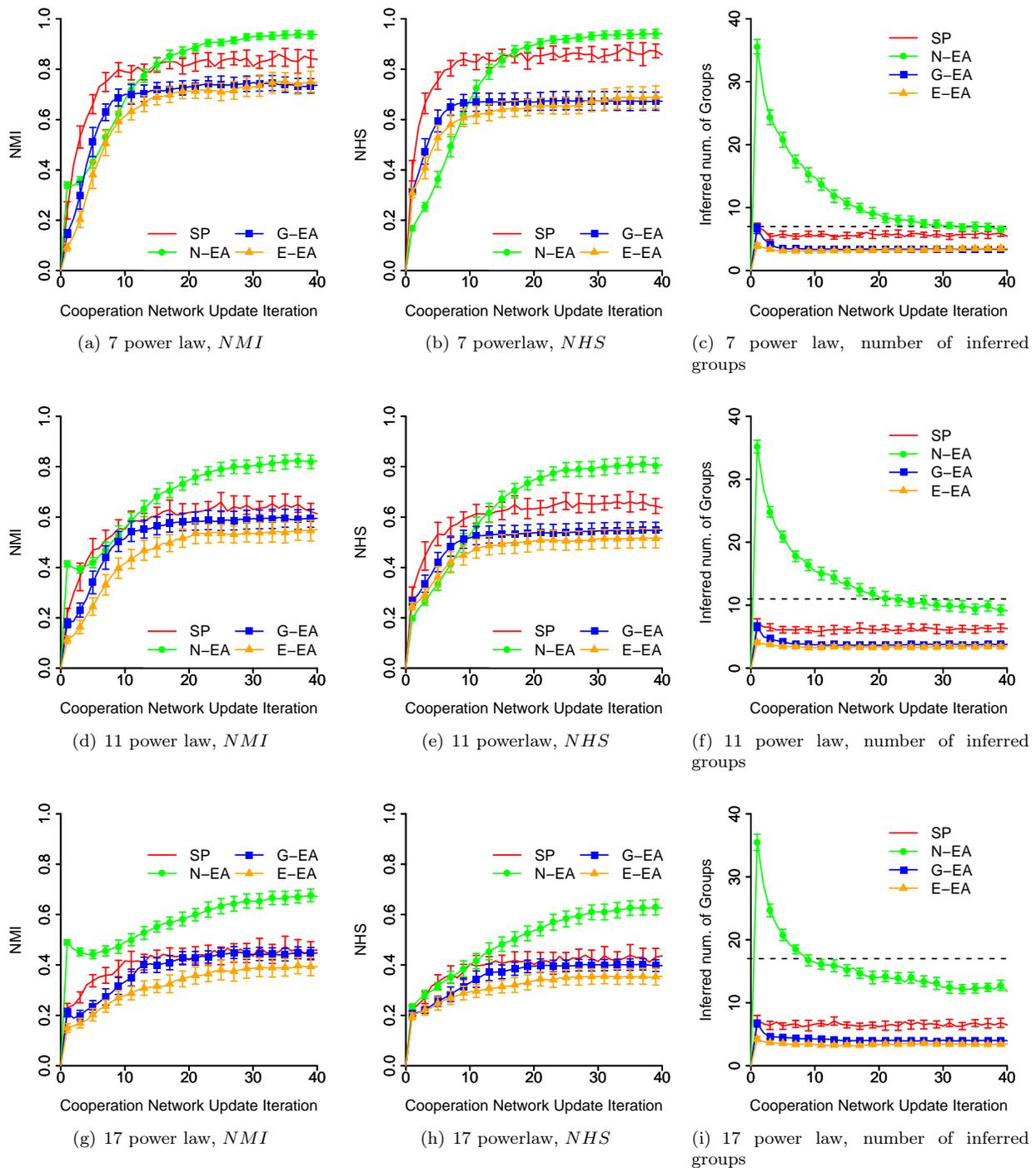


Figure 8.6: Performance (*NMI*, *NHS*, and inferred number of groups) of the three chromosome representation-based EAs compared to spectral partitioning for $m = \{7, 11, 17\}$ power law group size distribution.

Another key finding, though more generic, is the fact that the *NMI/NHS* scores recorded for the power law distribution society types are generally higher than the corresponding one of equal sizes, for $m = 11$ and

$m = 17$, partly for spectral partitioning, and especially for G-EA and E-EA. This is explained by the fact that it is easier to unveil big groups (see Figure 8.3), and especially because a misclassification of a small sized group has less impact on the misclassification of larger groups, like those of equal size, which also have a more regular edge degree distribution. For the $m = 7$ case, instead, we do not observe such a striking difference in performance, and this is possibly due by the group size distributions, which are likely more similar to each other than the $m = \{11, 17\}$ cases. Furthermore, we observe that the *NMI* scores are generally higher than the corresponding *NHS* ones, across all seven society types, and especially for the evolutionary algorithms. A discussion on this finding is provided in Section 10.1.8.

To conclude this Section, due to the results gathered, the next Sections will be solely based on the node-based chromosome representation. Nevertheless, the plots depicting the same experimental scenarios, based on the group and edge representations, can be found in Appendix 14.

8.4 Seeking for Improvements by Means of Niching

The previous Section highlighted the potential of the node representation to unveil the *true* group structures, in terms of highest *NMI* and *NHS* recorded, across the seven society types. However, it is inevitable to observe its slowness of convergence to maximum values, especially when m is small.

Two are the key reasons: (1) the initialisation phase, which allows for the creation of candidate solutions of a variable number of groups, potentially from 1 to n ; (2) the mutation operator, which allows the genes to have allele values ranging between 1 and n . These two factors lead to solutions which, at the beginning of the GID task, are independent on the *true* number of groups of the society. In support of this claim, one may compare the seven subfigures, of Figures 8.4, 8.5, and 8.6, depicting the evolution of the inferred number of groups: they all start from a peak, ranging on average between 35 and 36 groups, and then decrease towards either the exact number of groups — for $m = \{4, 7, 11\}$ equal, and $m = 7$ power law group size distribution — or slightly less — for the remaining society types. Nevertheless, Figure 8.3 showed that there exist difference in *true* modularity scores between the society types, especially between same group size distributions, even at early stages of the GID task. Ideally, a well performing algorithm should be able to spot those differences and quickly settle around the *true* number of groups.

In order to confine the evolutionary search around the *true* m values — with the hope to improve the speed of convergence to optimal values, yet by maintaining or possibly improving the overall maximum *NMI/NHS* scores registered by N-EA — in this Section we investigate the use of *sealed* niches, which are selectively evolved through an activation mechanism, as introduced in Section 7.7. For simplicity, we will hereafter refer to our specific sealed niching approach as niching only. The use of the proposed niching approach introduces two main drawbacks:

- the chromosomes are now forced to exactly represent a certain number of groups, by means of niche repair mechanisms. This might both introduce limitations with respect to the within-niche exploration of the most appropriate candidate solutions, and create artefacts affecting the overall performance scores;
- the niching activation mechanism might not be able to effectively solve the exploration vs. exploitation dilemma of the niching landscape, hence lead to the evolution of niches which assume an erroneous number of groups, possibly too far from the *true* m value.

The next two Subsections aim to investigate the above potential drawbacks.

8.4.1 Analysis of Niching Based on the Exact Knowledge of m

In order to better focus on the niching repair procedure, hence isolate its impact on the overall algorithmic performance from the niching activation mechanism, this Section assumes perfect knowledge about the *true* number of groups. More specifically, we consider a single niche-based evolutionary algorithm with node representation (N-1N) which population partitions \mathcal{N} into exactly m groups.

Analysis of $m = 4$ Case

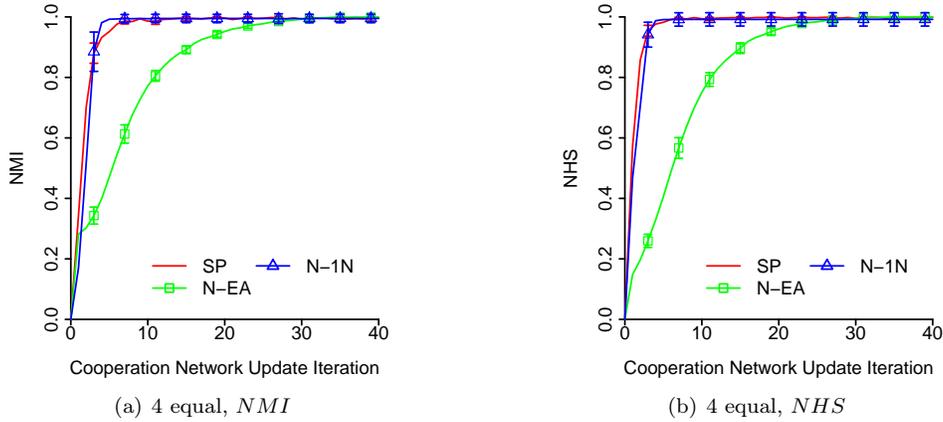


Figure 8.7: Performance (NMI , NHS) of N-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and spectral partitioning SP, for $m = 4$ equal size group types.

Figure 8.7 depicts the average performance and standard deviation of spectral partitioning, N-EA, and N-1N, based on NMI and NHS performance measures.

Remarkably, N-1N nearly perfectly replicates the same performance obtained by means of spectral partitioning. Although the final performance has a lower average performance than N-EA (avg = 0.9949, sd = 0.0273 for NMI ; avg = 0.9919, sd = 0.0435 for NHS), the difference is not statistically significant (p-value = 0.3512, $t = 0.9399$ for NMI ; p-value = 0.3304, $t = 0.9816$ for NHS , $df = 58$). Nevertheless, N-1N manages to score perfect matching after only three network updates, i.e. 30 interactions, for three out of thirty networks. These networks have an average number of non-updated nodes of 0.6208, standard deviation 0.0036.

Analysis of $m = \{7, 11, 17\}$ Cases

Figures 8.8 and 8.9 depict, respectively for the equal size and power law group size distribution, the average performance (NMI and NHS) and standard deviation, of N-EA, N-1N and SP, for $m = \{7, 11, 17\}$.

Not surprisingly, we observe that N-1N converges to high inference scores quicker than N-EA, for the small $m = 7$ case, and partly for $m = 11$, though for the $m = 17$ case — and equal size in particular — we do not observe any substantial improvements. With respect to the inference scores at the end of the GID task, we observe that N-EA outperforms N-1N in all the four scenarios considered and across the two performance measures. The reasons why this occurs in e.g. $m = 17$ equal case, and $m = 11$ power law case, might be due to the noise in the network, which leads to higher scores when a lower number of groups is inferred — see e.g. Figures 8.5(i) and 8.6(i). The performance is nearly matched for the $m = 17$ power law case; this result is rather surprising, especially because for the N-EA case the average number of groups inferred is 11.8, sd = 1.2754. The reason behind this finding might be either (1) N-EA merges on average six small-sized groups, with little impact on its NMI/NHS scores; (2) the niching repair mechanism of N-1N introduces on averages six single-sized groups (artefacts).

However, what we find surprising, is that for both $m = 7$ cases, N-EA outperforms N-1N in the long run, especially considering that the former infers, on average, 6.9667 groups (sd = 0.4069) for equal size case, and 6.6667 groups (sd = 1.0435) for power law case. A possible reason could be that the niching repair mechanism, initially, excessively forces the algorithm to explore a subregion of the search space that is suboptimal; consequently, when the underlying network better manifests the *true* group structures, the algorithm cannot readjust its evolutionary search towards the optimal solution.

Similarly to N-EA, N-1N appears to be better performing for the power law cases than the corresponding equal size ones. Moreover, except for the $m = 7$ cases, N-1N outperforms spectral partitioning, both in terms

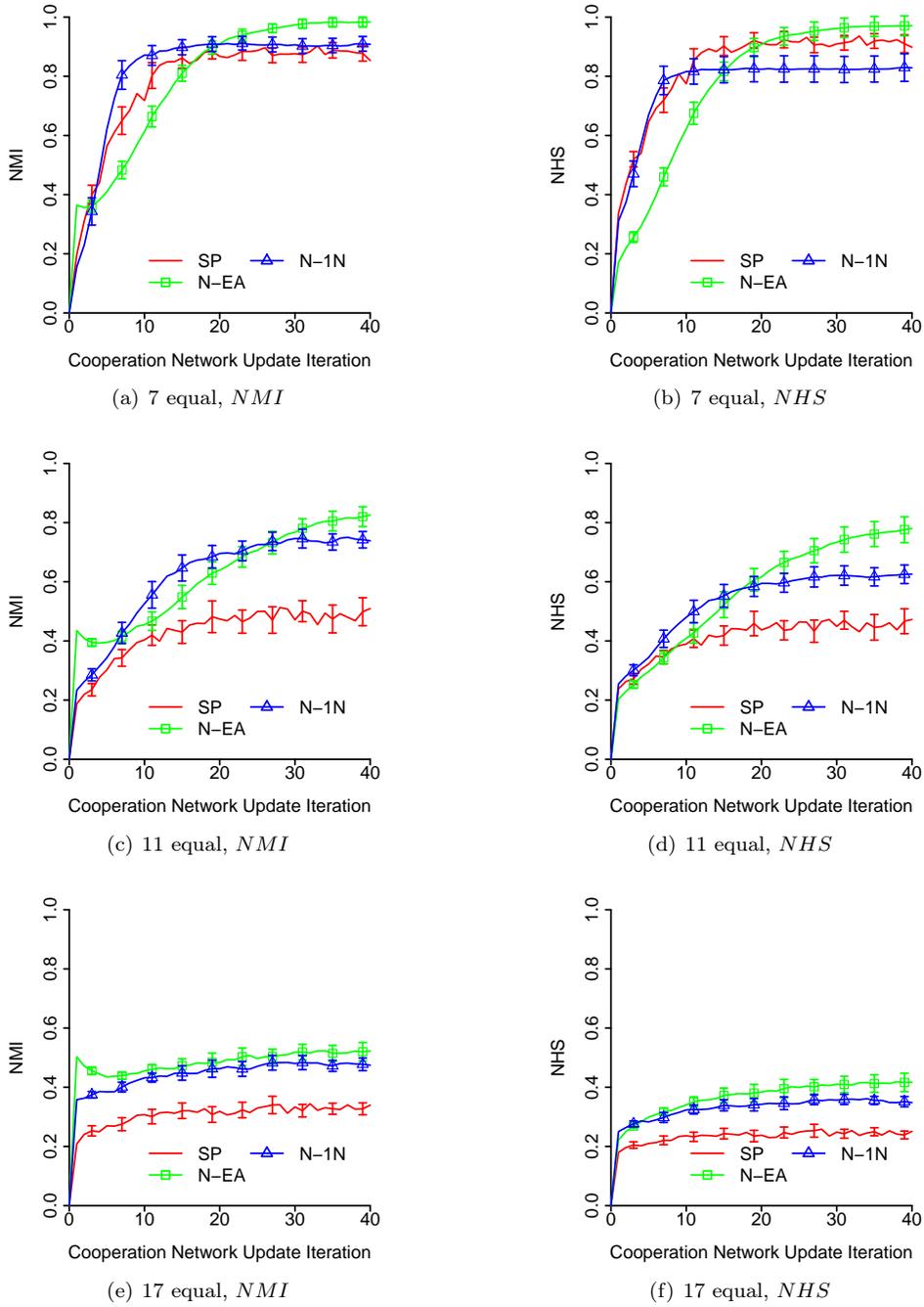


Figure 8.8: Performance (*NMI*, *NHS*) of N-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = \{7, 11, 17\}$ equal size group types.

of overall inference scores and speed of improvement towards maximum *NMI/NHS* values, though for the $m = 11$ power law group size distribution case the improvement is initially slower. Finally, we observe, once again, that for N-1N the *NHS* scores are more rigid than the *NMI*.

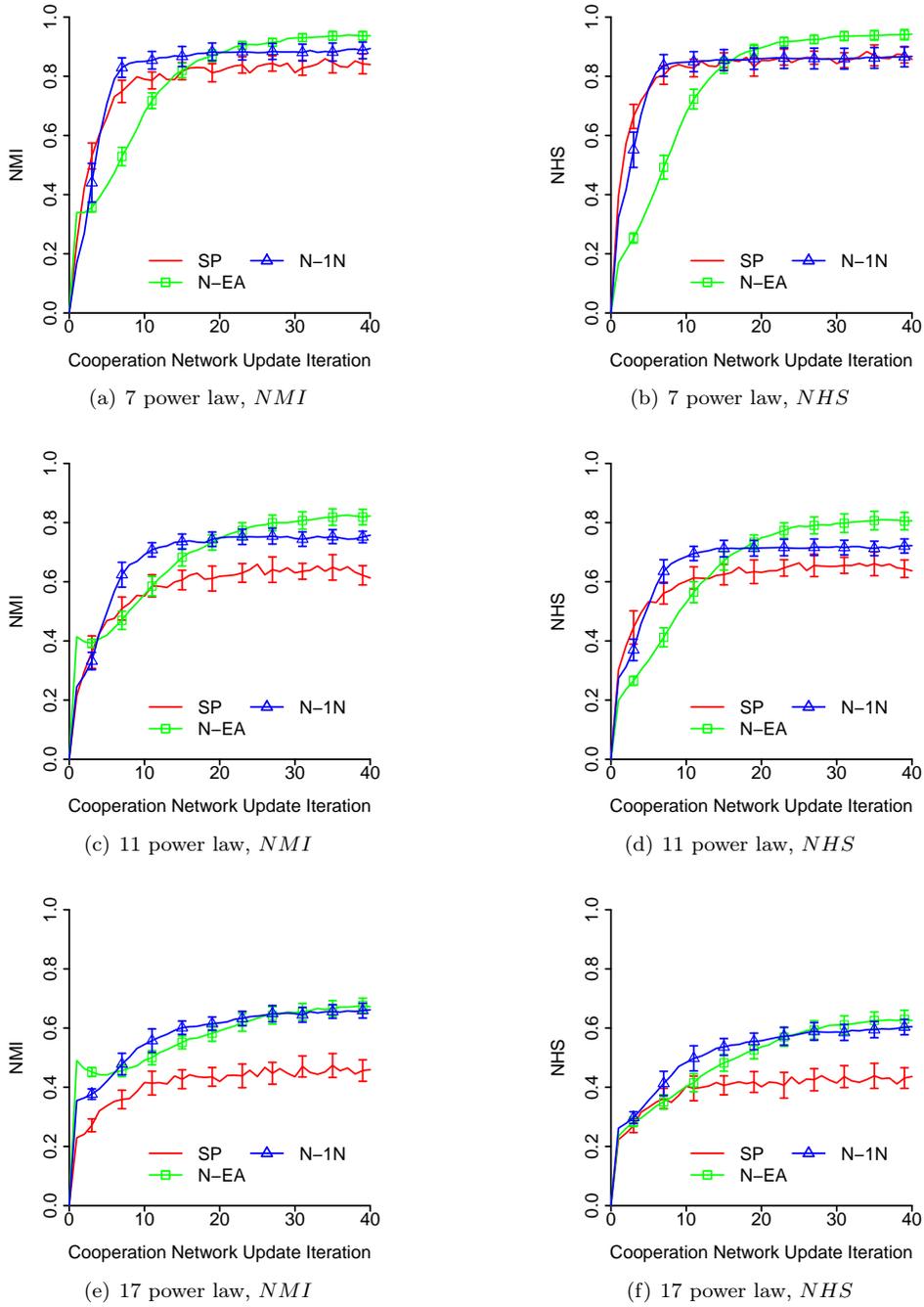


Figure 8.9: Performance (*NMI*, *NHS*) of N-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = \{7, 11, 17\}$ power law group size distribution types.

8.4.2 Analysis of Niching Activation Mechanisms

In a nutshell, the experimental results presented in the previous Subsection showed that the use of niching can provide advantages in terms of convergence to high inference scores, mostly for small m values, though

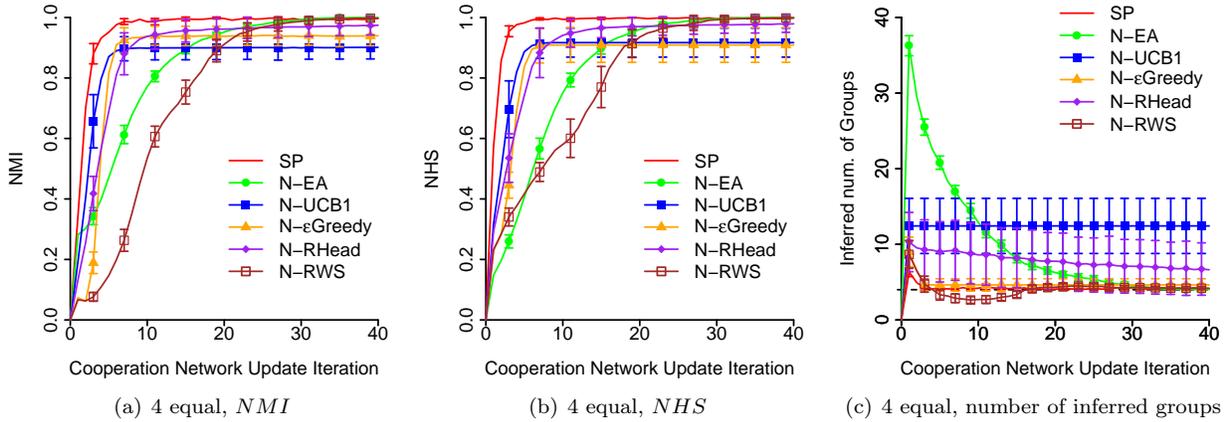


Figure 8.10: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of the four node-based niching activation algorithms, compared to N-EA and SP, for $m = 4$ equal size group types.

its overall performance does not improve what can be achieved by the corresponding single population-based counterpart (see also the results which consider the group and edge representation in Appendix 14).

Since, with the increase of m , N-EA scores the best performance by inferring a lower number of groups, then it might happen that the comparison with another N-1N algorithm type, e.g. one which assumes 12 groups for the $m = 17$ power law case, might provide an inference improvement. However, the previous Section made assumptions over the knowledge of m ; this is not only hardly the case in generic community structure detection problems, but it is also certainly unknown in case of social synthetic environment scenarios with dynamic group structure evolution.

This Subsection aims to conclude the investigation about the use of our niching approach by using n niches, being each niche k composed of a population of chromosomes which partitions \mathcal{N} into exactly k groups. The study conducted and the results hereafter presented are centred on the four niching activation mechanisms described in Subsection 7.7.2, that is roulette wheel selection (N-RWS), reading head (N-RHEAD), and finally n -armed bandit-based UCB1 (N-UCB1) and ϵ -greedy algorithms (N- ϵ GREEDY). It is important to stress the fact that all techniques, except for reading head, activate one niche per time. On the other hand, reading head will evolve $z = 5$ consecutive niches per time. For the ϵ -greedy approach, after a preliminary study, we relied on the following parameters: $c = 0.35$, $d = 0.24$, see Algorithm 17. Finally, the four techniques are let to evolve until the limit of fitness evaluation presented in Table 8.3 has been reached.

Analysis of $m = 4$ Case

Figure 8.10 depicts the average performance (NMI , NHS , and inferred number of groups) and standard deviation of node-based roulette wheel selection, reading head, UCB1 and ϵ -greedy, compared with spectral partitioning and N-EA, for $m = 4$ equal size.

The most striking and somewhat unexpected result is obtained by roulette wheel selection, which manages, at the end of the GID task, to reach an average score of 0.9983 (sd = 0.0042) for NMI and 0.9984 (sd = 0.0031) for NHS , though it is not statistically significantly different from N-EA only for NMI (p-value = 0.1386, $t = 1.5018$; p-value = 0.0407, $t = 2.0933$ for NHS , $df = 58$). Reading head is the second best performing niche-activation algorithm, whilst the two n -armed bandit-based ones manage to quickly converge to a lower score and a higher standard deviation. As expected, roulette wheel selection is the method which converges to optimal values in the slowest way, since it always gives a non-null probability of activation even to very poorly performing niches. On the other hand, the two n -armed bandit based algorithms, and UCB1 in particular, resemble the quick convergence observed by the previous pilot study based on N-1N.

Reading head is the niching activation mechanism which quickly converges to optimal values, though its final performance — which appears not to have plateaued — is lower than the optimal one recorded by both roulette wheel selection and N-EA, that is 0.9743 (sd = 0.0603) for *NMI* and 0.9789 (sd = 0.0542) for *NHS*. With this respect it is also worth noticing its trend in terms of number of inferred groups (see Figure 8.10(c)). Reading head starts by inferring, on average, 10.3 groups (sd = 7.7078), whilst at the end of the GID task its average is 6.6333 (sd = 6.7699). Its very high standard deviation is particularly due to one of its executions, which starts the GID task by inferring 38 groups, and terminates it by inferring 36 groups. Apart from this particularly bad run, we also observed that, when N-RHEAD starts the GID task by assuming a relatively high number of groups — e.g. greater than or equal to fifteen, in 8 runs out of 29 — its convergence towards the *true* $m = 4$ case is hardly ever achieved — just once out of eight runs, when it started by assuming 17 groups. This difficulties in moving through the niche landscape towards the *true* m value are, most likely, dual: (1) the consecutive window of $z = 5$ niches which are evolved at each generation might excessively consume the amount of available fitness evaluation; (2) the niching repair mechanism introduces artefacts that do not allow for the quick finding of new best candidate solutions with a lower number of inferred groups.

By reasoning about the inferred number of groups, we observe that N-UCB1 is the niche activation-based technique which assumes, at the beginning of the GID task, the highest number of groups (avg = 12.4333, sd = 7.1679), though it unfortunately does not manage to converge to the optimal $m = 4$ value. Rather than that, N-UCB1’s trend of Figure 8.10(c) is constant throughout the whole GID task. This means that the niching activation mechanism relies solely on the initialisation of its niche landscape and does not manage to further explore it.

Similarly, N-εGREEDY starts by assuming, on average, 9 groups (sd = 3.847). Then, after only three cooperation networks observed, it converges to an average number of groups of 4.6333 (sd = 1.6017), though only in 6 runs out of 30 it detects the *true* $m = 4$ number of groups. Possibly, different c and d parameter values could have led to a further and better exploration of the niche landscape.

Analysis of $m = \{7, 11, 17\}$ Cases

Figures 8.11 and 8.12 depict, respectively for equal and power law group size distribution types, the average performance (*NMI*, *NHS*, and inferred number of groups) of the four node-based niching activation mechanisms, compared with spectral partitioning and N-EA, for $m = \{7, 11, 17\}$.

Overall, the two best niching activation mechanisms recorded are the deterministic reading head and UCB1. More specifically, reading head appears to have an advantage over UCB1 for $m = \{7, 11\}$ equal size cases, whilst the opposite occurs for $m = 17$ equal size. For the three power law cases, instead, reading head and UCB1 have essentially the same performance scores, though UCB1 is generally faster than reading head in terms of convergence to maximum *NMI/NHS* values.

Reading head starts the GID task by assuming, on average, about ten groups, independently on the society type under investigation. However, unlike for the $m = 4$ case, this niching activation mechanism does not manage to converge towards the *true* number of groups. Moreover, the trend of inferred number of groups is nearly constant, except for the initial spike — see for instance Figures 8.11(i) and 8.12(i). Even more remarkably is the fact that, for $m = 7$ equal size case (see Figures 8.11(a), 8.11(b), and 8.11(c)), reading head infers, at the end of the GID task, 9.7333 groups on average (sd = 7.418), yet its related performance — i.e. avg = 0.9541, sd = 0.0506 for *NMI*, avg = 0.9286, sd = 0.0829 for *NHS* — is rather close to N-EA’s — i.e. avg = 0.9871, sd = 0.0334 for *NMI*, avg = 0.9708, sd = 0.0666 for *NH*. Nevertheless the difference in performance between N-EA and N-RHEAD is statistically significant (p-value = 0.0042, t = 2.9812 for *NMI*, p-value = 0.0338, t = 2.1736 for *NHS*, df = 58); this finding suggests, once again, that the niching repair mechanism might have a generally negative impact on the evolutionary search and its capability to target the *true* number of groups and consequently seeking for the optimal solution.

With respect to UCB1, we observe that, like for the $m = 4$ case, the inferred number of groups trend are constant across all the six society cases. In other words, UCB1 is overly dependent on the initialisation process of the niches. Nevertheless, N-UCB1 becomes the best performing niching activation mechanism for $m = 17$ equal case: it records a final average score of 0.4167 (sd = 0.0714) for *NMI*, and 0.306 (sd = 0.0488) for *NHS*; moreover, its average number of groups inferred is 10.5667 (sd = 5.2894), that is the highest across

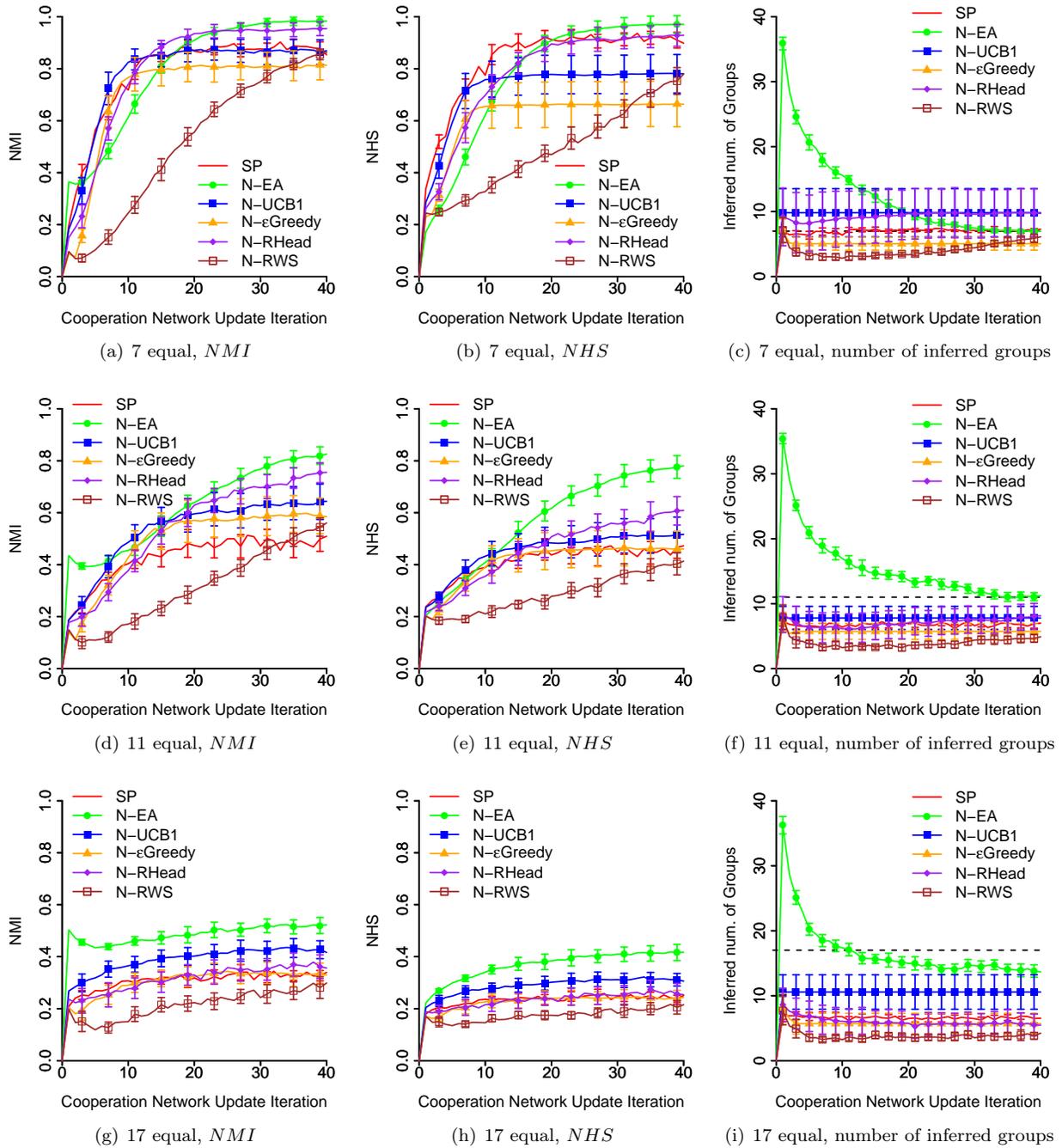


Figure 8.11: Average performance (NMI , NHS and inferred number of groups) and standard deviation of the four node-based niching algorithms compared to SP and N-EA, for $m = \{7, 11, 17\}$ equal size society types.

the four niching activation mechanisms. Furthermore, its difference in performance from reading head — i.e. avg = 0.3563, sd = 0.0877 for NMI , avg = 0.2536, sd = 0.0769 for NHS — is statistically significant (p-value = 0.0049, $t = 2.9253$ for NMI , p-value = 0.0026, $t = 3.1512$ for NHS , $df = 58$).

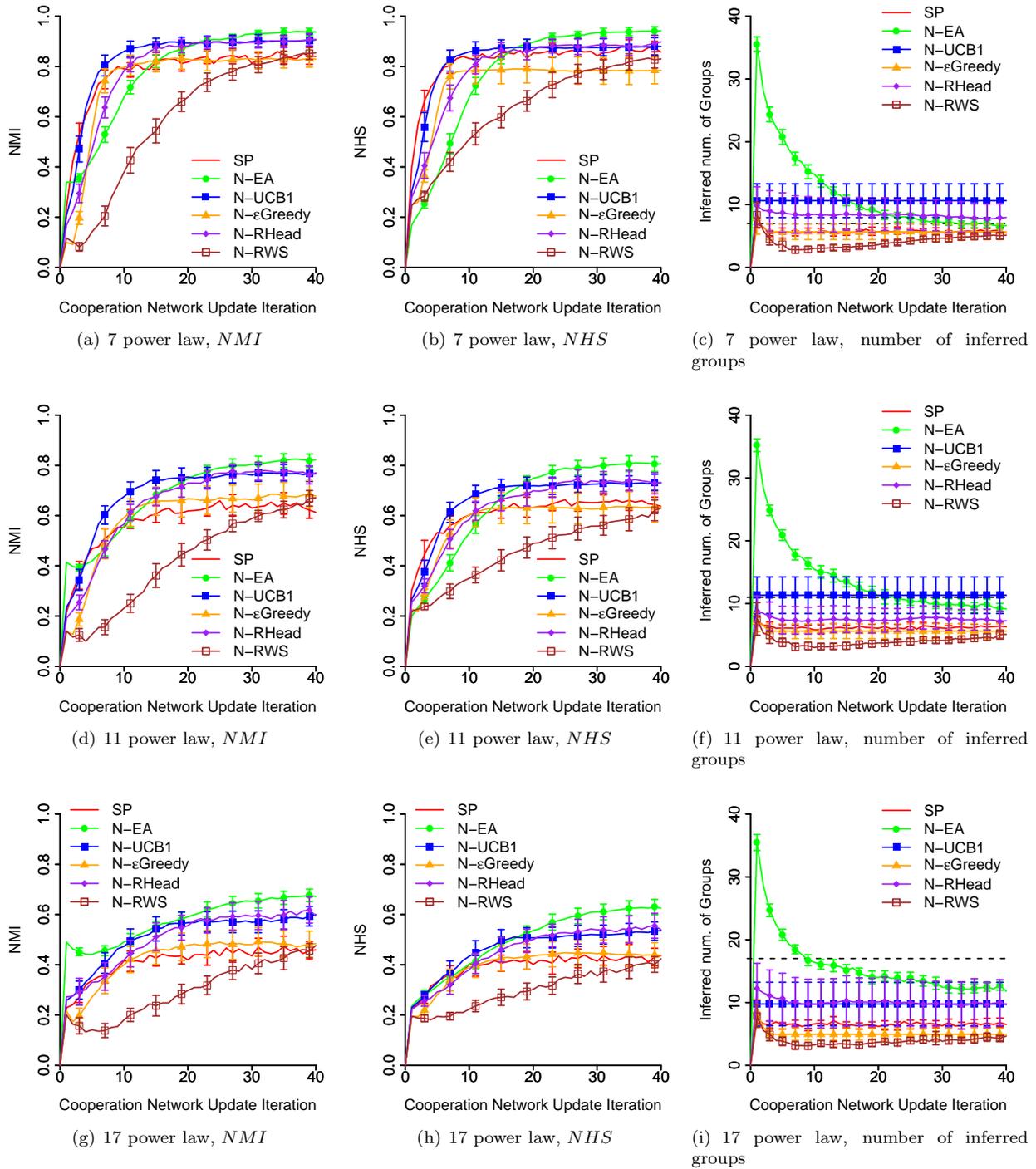


Figure 8.12: Average performance (NMI , NHS and inferred number of groups) and standard deviation of the four node-based niching algorithms compared to SP and N-EA, for $m = \{7, 11, 17\}$ power law group size distribution society types.

Unlike what we have seen for the $m = 4$ case, and as expected, roulette wheel selection now becomes the least performing niching activation mechanism — both in terms of speed of convergence, and highest NMI and NHS values — though its performance has clearly not reached a plateau yet. The reason is straightforward: with the increase of m , the *true* modularity decreases (see Figure 8.3), meaning also that there might be many suboptimal solutions/niches which partition the network into different number of groups. As a consequence, the niche landscapes will be less spiky than the $m = 4$ case counterpart. Hence, roulette wheel will assign higher probabilities to many suboptimal niches, leading thus to more exploration than exploitation. This pattern might be acceptable during the early stages of the GID task, though it should not occur for later stages. Instead, we observe that the performance trends of roulette wheel selection are nearly linear across all scenarios, with the sole exception for the $m = 7$ power law case (see Figures 8.12(a) and 8.12(b)). Finally, we remark its nearly constant (though slightly increasing) trend in terms of inferred number of groups. Intuitively, roulette wheel selection, among all niching activation mechanisms, would certainly benefit from a much higher number of fitness evaluation limit regulating its execution, though its convergence to high performance scores is hardly expected.

The ϵ -greedy activation mechanism, unlike for the $m = 4$ case, manifest in general a lower performance than UCB1. We also observe its rather quick convergence to constantly lower-than-*true* inferred number of groups, and the apparent early plateauing to suboptimal NMI/NHS scores, unlike roulette wheel selection, reading head, and partly UCB1.

To conclude this Subsection, we highlight again the fact that the NHS scores are generally more severe for the all four niche activation approaches than those obtained by means of NMI , especially when $m = \{11, 17\}$.

8.5 Analysis of the Importance of the Fitness Function: Comparison with LinkRank

Section 8.3 (and Appendix 14) resulted to the following key findings:

- the node-based chromosome representation leads to the highest NMI/NHS scores across the seven society types;
- the edge-based chromosome representation appears to excessively suffer the noise in the observed cooperation networks;
- the group-based chromosome representation appears to rely on overly structured recombination operators.

These key findings were analysed by keeping the emphasis on the implementation of the evolutionary algorithms. However, these evolve their candidate solutions based on modularity maximisation, which has been subject of criticism, though related to its undirected and unweighted case, see Section 3.2.3. Among all, it has been proven that modularity is affected by a resolution limit, see Section 3.2.3. This phenomena — which is arguably already observed in the inferred number of groups plots — is much more evident when the discussion deviates from the NMI/NHS scores, and rather centres itself on the mere task of modularity maximisation. More specifically, consider Figures 8.13 and 8.14. These depict, respectively for the equal and power law group size distribution society types, the average ratio (and standard deviation) between the modularity score of the inferred solutions $Q(\mathcal{H})$ and the *true* modularity score $Q(\mathcal{T})$, for N-EA, G-EA, E-EA and spectral partitioning (SP).

Apart from the $m = 4$ case, and especially in the four $m = \{11, 17\}$ cases, we observe that SP, G-EA and E-EA detect community structure partitions which modularity score is higher than the *true* one. Moreover, for the $m = 17$ equal size case (see Figure 8.13(d)), their ratio is always above one. This clearly indicates that the three approaches, which are more structured than N-EA, have the tendency to start by inferring a low number of groups — which increases the *within*-community, and decreases the *between*-community edge degrees, hence increases $Q(\mathcal{H})$ — and not deviate from it — since the further partitioning would reduce the

within-edge and increase the *between*-edge degrees, hence decrease $Q(\mathcal{H})$. In other words, SP, G-EA, and E-EA are more successful than N-EA to maximise modularity, though this does not correspond to an increase of the NMI/NHS scores, which constitute the target performance of the group modelling task.

Another remarkable finding is the fact, despite N-EA manages to get an average modularity ratio of 0.9923 (sd = 0.024) for $m = 11$ power law, and 1.0171 (sd = 0.0413) for $m = 17$ power law (see Figures 8.14(b) and 8.14(c) respectively), its related final NMI/NHS scores were highly suboptimal (see Figures 8.6(d), 8.6(e), 8.6(g), and 8.6(h)). Moreover, in these two society types, the inferred number of groups were lower than the *true* ones (see Figures 8.6(f), 8.6(i)). This seems to indicate that modularity — more than N-EA — has difficulty into disambiguating, hence evaluate differently, community structure partitions which differ by few small-sized communities.

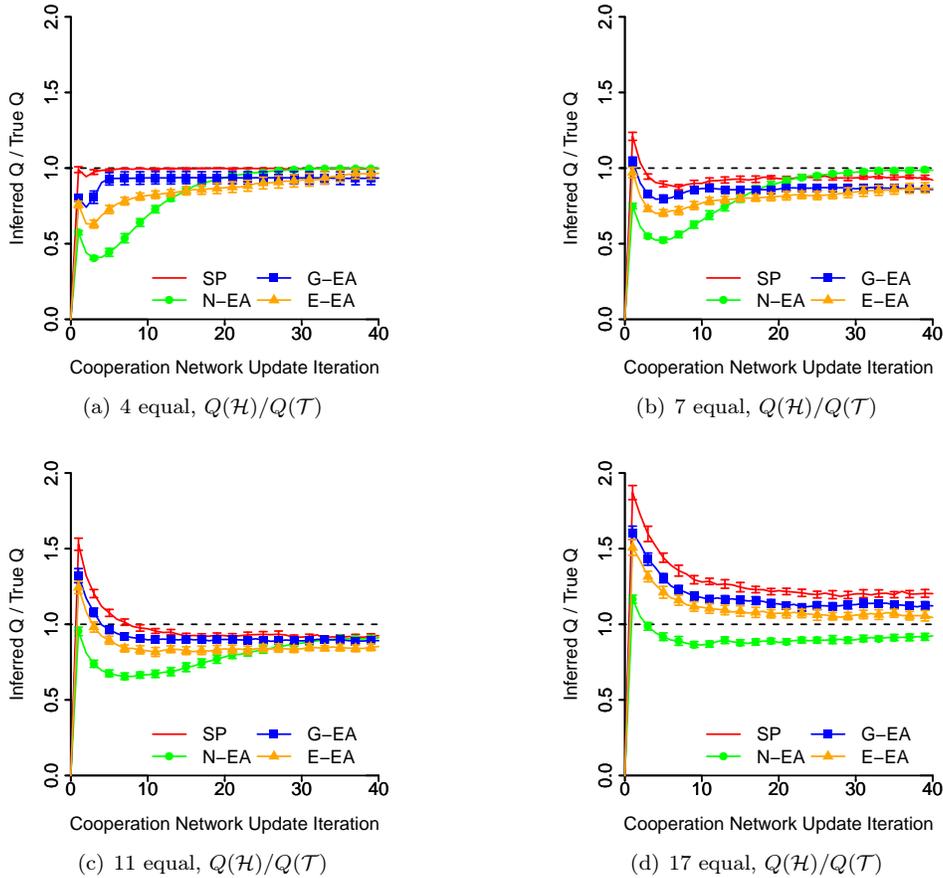


Figure 8.13: Average ratio and standard deviation between the inferred modularity $Q(\mathcal{H})$, and the *true* modularity $Q(\mathcal{T})$, of N-EA, G-EA, E-EA and SP, for $m = \{4, 7, 11, 17\}$ equal size group distribution.

Motivated by these findings, the remainder of this Section aims to examine whether an alternative fitness function — which is still positively correlated to the quality of network partitioning into community structures — could lead better a quicker inference of the *true* groups, especially relatively large m values.

Although in Section 3.2.3 we presented two alternative objective functions for CSD, in the following Subsections we will only focus on LinkRank, defined in Equation (3.10), for two main reasons: (1) it is based on another conceptual principle, that is Markov chains, rather than expected values of node degree, and (2) its intended aim to evaluate the community structure partitions based on the network’s *flow* might provide performance advantages especially during the early stages of the GID task. Moreover, we decided to disregard the resolution-based modularity due to the very function of parameter λ , which fine tuning will

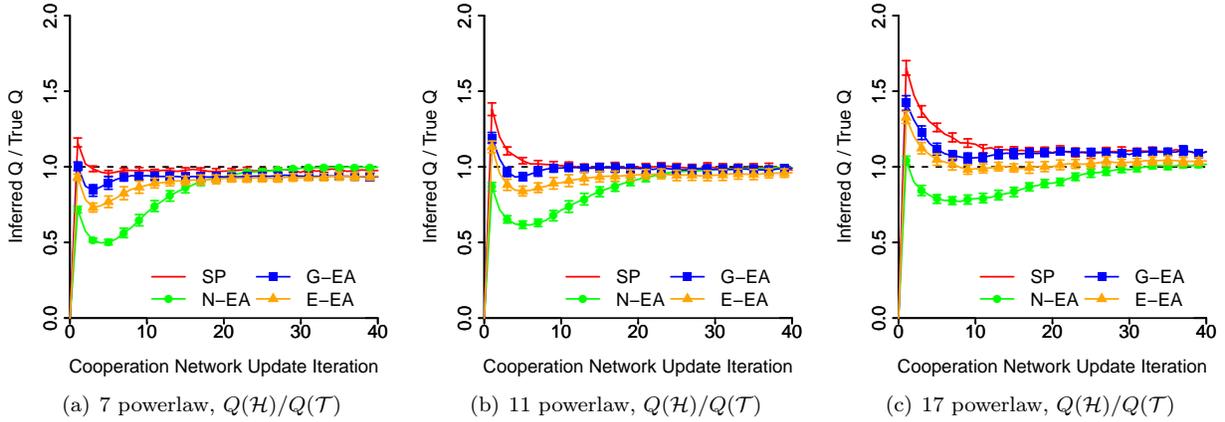


Figure 8.14: Average ratio and standard deviation between the inferred modularity $Q(\mathcal{H})$, and the *true* modularity $Q(\mathcal{T})$, of N-EA, G-EA, E-EA and SP, for $m = \{7, 11, 17\}$ power law group size distribution.

steer the optimisation process towards either small or large m values. On the other hand, we will rely on the teleportation probability $\alpha = 0.85$ initially defined by Kim et al. [112]. Further discussion about these and other possible fitness functions can be found in Chapter 10, and especially in Sections 10.1.6 and 10.1.8.

Finally, we will first focus on the impact that LinkRank has on the three, single population-based algorithms N-EA, G-EA and E-EA. Subsequently, and in accordance with what was done in Section 8.4, we will restrict the scope of our investigation to the chromosome representation which best performs with LinkRank, and analyse the impact of this fitness function with respect to the most promising niching activation mechanism identified in the previous Section 8.4.2.

8.5.1 Analysis of the Chromosome Representation

Figures 8.15, 8.16, and 8.17 depict, respectively for $m = 4$, $m = \{7, 11, 17\}$ equal size, and $m = \{7, 11, 17\}$ power law group size distribution, the average performance (NMI , NHS and inferred number of groups) and related standard deviation, of the node-based, group-based, and edge-based evolutionary algorithms, depending whether they rely on modularity (N-EA-Q, G-EA-Q, and E-EA-Q respectively) or LinkRank (N-EA-LR, G-EA-LR, and E-EA-LR respectively).

The most straightforward finding is represented by the fact that LinkRank does not improve the performances of none of the three EAs considered, neither by considering the NMI/NHS scores, nor by focusing on the speed of convergence towards maximum values. Furthermore, we observe that N-EA is the algorithm which performance is the least affected by the changes in its fitness function. At the other extreme, instead, we have E-EA, which difference in performance is much more striking. Finally, G-EA falls in between the two difference patterns.

8.5.2 Analysis of the Niching Activation Mechanism

As a consequence of the findings presented in the previous Section, in this Section we will deepen the investigation of the node-based chromosome representation only. Moreover, also due to the results presented in Section 8.4.2, this Section will consider, as opposed to N-EA, the performance of reading head, since this niching activation mechanism has provided overall better — arguably non-plateaued — NMI/NHS scores. Nevertheless, similar plots related to the other niching activation mechanisms, and also the group and edge-based chromosome representation algorithms can be found in Appendix 14.

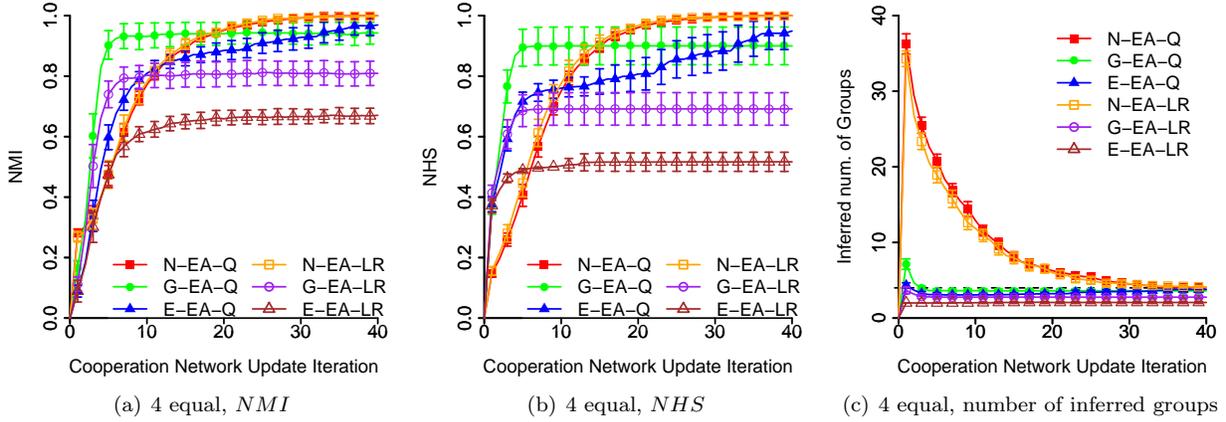


Figure 8.15: Comparison of the average performance and standard deviation of the node, group, and edge-based chromosome representation, single population evolutionary algorithms, depending on whether modularity (N-EA-Q, G-EA-Q, and E-EA-Q respectively) or LinkRank (N-EA-LR, G-EA-LR, and E-EA-LR respectively) is used as objective function, for $m = 4$ equal group size distribution.

Analysis of $m = 4$ Case

Figure 8.18 depicts the average performance (NMI , NHS , and inferred number of groups) and standard deviation of the node-based single population, and the node-based reading head algorithms, depending on whether modularity or LinkRank is used as fitness function (N-EA-Q, N-EA-LR and N-RHEAD-Q, N-RHEAD-LR respectively), for $m = 4$ equal size.

The most observable key result is that reading head-LinkRank improves the speed of convergence of both its NMI and NHS scores, though its final performance, which on average is 0.9971 (sd = 0.0062) for NMI and 0.9982 (sd = 0.0039) for NHS , does not outperform N-EA-modularity, which remains the best performing algorithm. Nevertheless, the difference in performance between N-EA-Q and N-RHEAD-LR is statistically significant only for NMI (p-value = 0.0418, t = 2.0814) — p-value = 0.0521, t = 1.9827 for NHS , df = 58). The differences of evolution between modularity and LinkRank on reading head is clearly seen in Figure 8.18(c): with LinkRank, now, reading head tends to initially assume a small number of groups (avg = 2.3, sd = 0.4582), then to increase it up to the average peak (avg = 4.5667, sd = 0.8035), and then converge to the final average 4.2333, sd = 0.4955. Moreover, we can observe that the related standard deviation is much smaller than the corresponding one registered when modularity is used, meaning hence that LinkRank tends to be less affected by the initialisation phase of the niches, and at the same time it appears to generally start from a low inferred number of groups.

Similar striking effects of the use of LinkRank, instead, were not observed when N-EA is used. The algorithm based on LinkRank is slightly faster than the modularity-based counterpart in terms of convergence to high NMI/NHS scores, though not statistically significant, and at the end of the GID task N-EA-modularity still records the average highest inference score (N-EA-LinkRank scores avg = 0.9988, sd = 0.0037 for NMI and avg = 0.9992, sd = 0.0023 for NHS). However, the difference in performance at the end of the GID task is not statistically significant (p-value = 0.3129, t = 1.0179 for NMI , p-value = 0.3133, t = 1.0171 for NHS , df = 58).

Analysis of $m = \{7, 11, 17\}$ Cases

Figures 8.19 and 8.20 depict, respectively for equal and power law group size distribution types, the average performance (NMI , NHS , and inferred number of groups) and standard deviation of N-EA-Q, N-EA-LR, N-RHEAD-Q, and N-RHEAD-LR, for $m = \{7, 11, 17\}$.

In any of the covered scenarios, as opposed to the $m = 4$ case, LinkRank does not manage to improve the

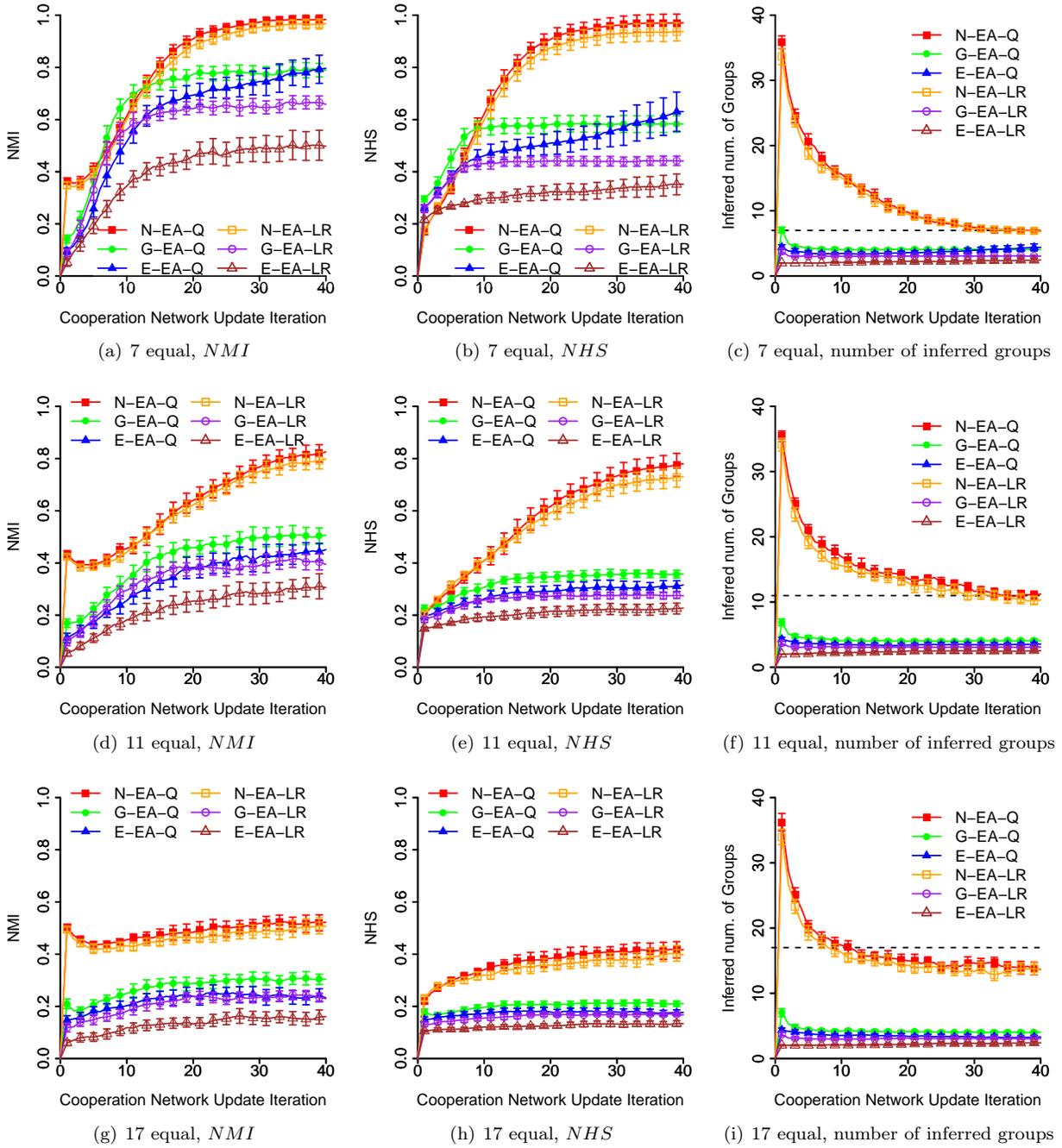


Figure 8.16: Comparison of the average performance and standard deviation of the node, group, and edge-based chromosome representation, single population evolutionary algorithms, depending on whether modularity (N-EA-Q, G-EA-Q, and E-EA-Q respectively) or LinkRank (N-EA-LR, G-EA-LR, and E-EA-LR respectively) is used as objective function, for $m = \{7, 11, 17\}$ equal group size distribution.

results obtained by means of modularity. Moreover, the difference in performance increases together with the increase of m , especially for reading head. The reason why this occurs is the confirmation of what we observed in the $m = 4$ case: through LinkRank the algorithms tend to always infer a lower number of groups

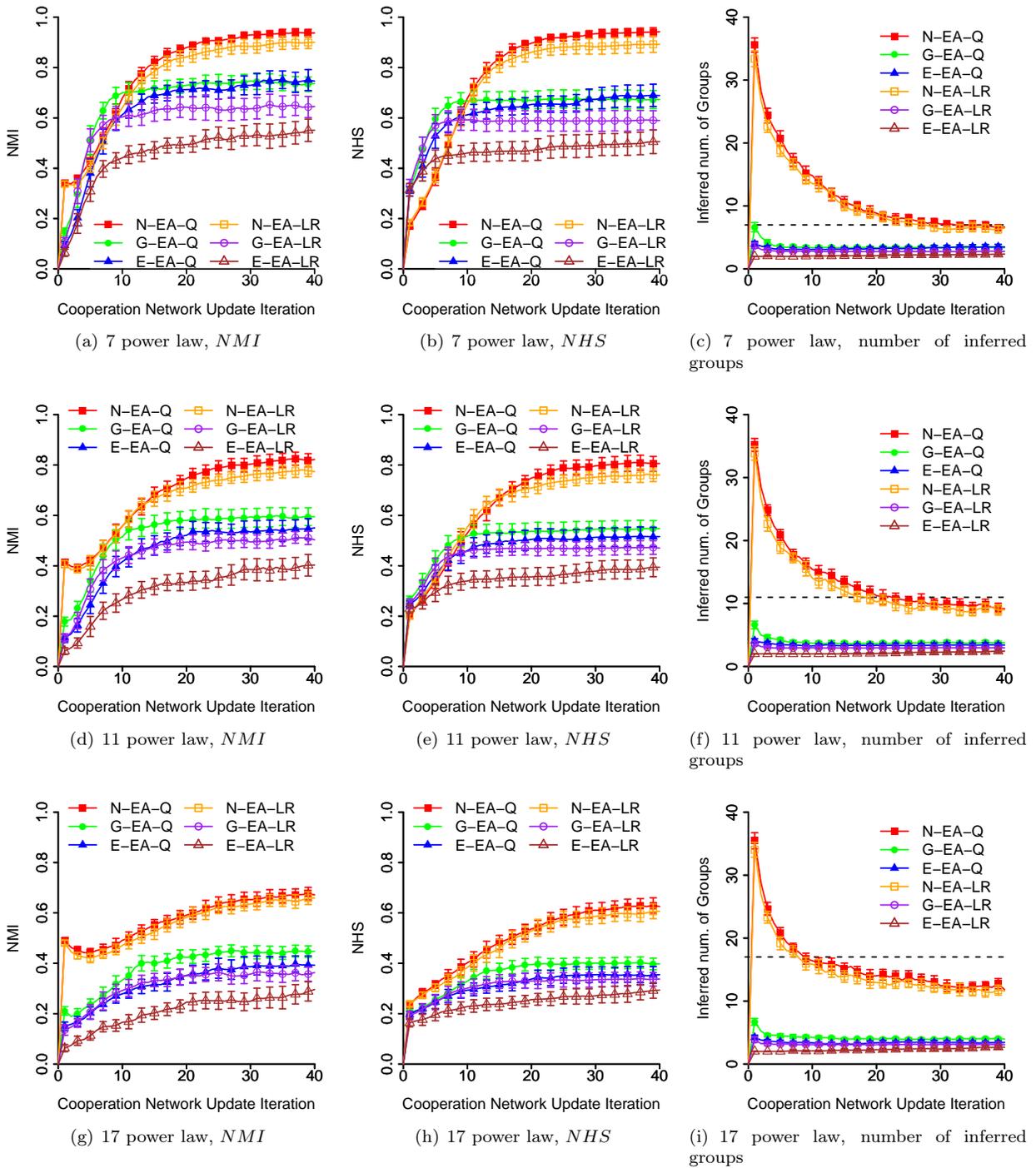


Figure 8.17: Comparison of the average performance and standard deviation of the node, group, and edge-based chromosome representation, single population evolutionary algorithms, depending on whether modularity (N-EA-Q, G-EA-Q, and E-EA-Q respectively) or LinkRank (N-EA-LR, G-EA-LR, and E-EA-LR respectively) is used as objective function, for $m = \{7, 11, 17\}$ power law group size distribution.

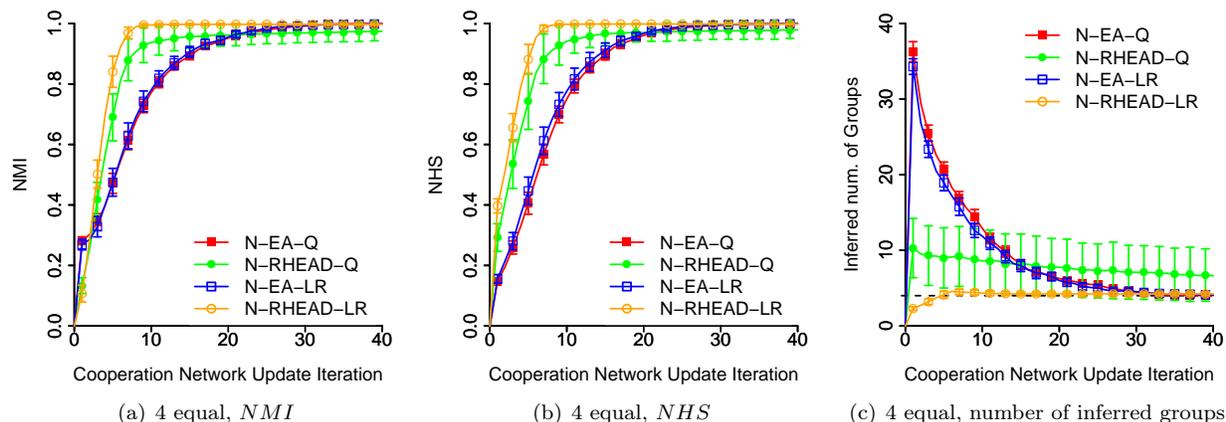


Figure 8.18: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N-EA and node-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function.

than what we would have by means of modularity. It is hence evident that higher values of m are hard to be reached by reading head, which evolves $z = 5$ consecutive niches for each generation, meaning hence that its evolutionary limit is five times smaller than N-EA’s. On the other hand, for N-EA, it appears as if the early convergence to a lower number of groups, which is a more preferred trend than what we observe when modularity is used, does not imply an improved performance; the $m = 11$ cases are worth noticing.

8.6 Summary

This Chapter provided a thorough experimental investigation of the last module of our proposed interaction-based group modelling framework, that is the Group Identity Detection (GID) module. The results gathered also constitute a solid evidence of the effectiveness of the whole group modelling framework, since the cooperation networks parsed by the GID module are computed by the framework’s first module, that is the cooperation modelling component.

The empirical results gathered, based on iterations of seven different society types, evidence the fact that the simplest node-based chromosome representation outperforms the other two, more articulated group and edge-based representations. Moreover, we observed that the most canonical evolutionary algorithm based on a single population leads to the best inference performances in the long run, though its speed of convergence is rather slow for scenarios with small number of groups.

As an alternative method to explore the evolutionary search space, we attempted the use of *sealed* niches — each niche k being populated by chromosomes which are forced to partition a given network into exactly k communities — combined with four different activation mechanisms — two deterministic and two stochastic — which selectively evolve niches based on the overall niche landscape.

The results obtained showed that the proposed niching approach can provide a remarkable advantage in terms of speed of convergence to high scores, though its final inference performance is suboptimal.

Finally, in this investigation we considered two different fitness functions to be maximised, which both represent the quality of the community structure partitions, namely modularity and LinkRank, which however rely on different semantic interpretations of communities. Moreover, the modularity measure we used is a generic measure which can be applied to any network topology; LinkRank, on the other hand is a measure explicitly designed for directed network. The results showed that, overall, modularity is the best target objective for the social synthetic environments we covered in our study.

This Chapter presented only the key results by following a conceptual research path; integrative secondary results can be found in Appendix 14. The next Chapter will apply the best configurations of the group

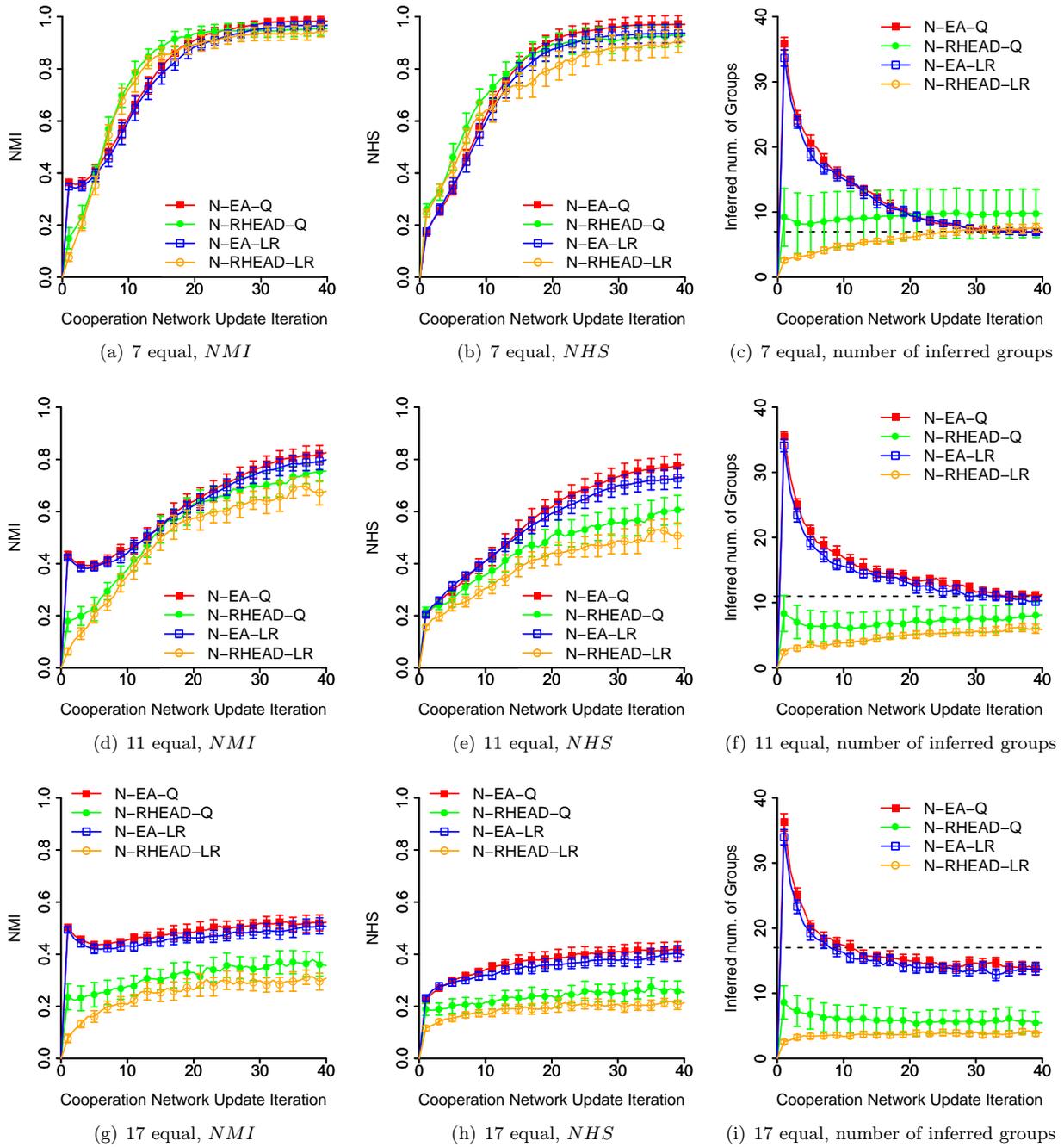


Figure 8.19: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N-EA and node-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = \{7, 11, 17\}$ equal size types.

modelling framework, retrieved by the empirical evaluation of this Chapter and Chapter 6, to the authentic behavioural data extracted from the experiment by Chen and Li (see Section 4.2).

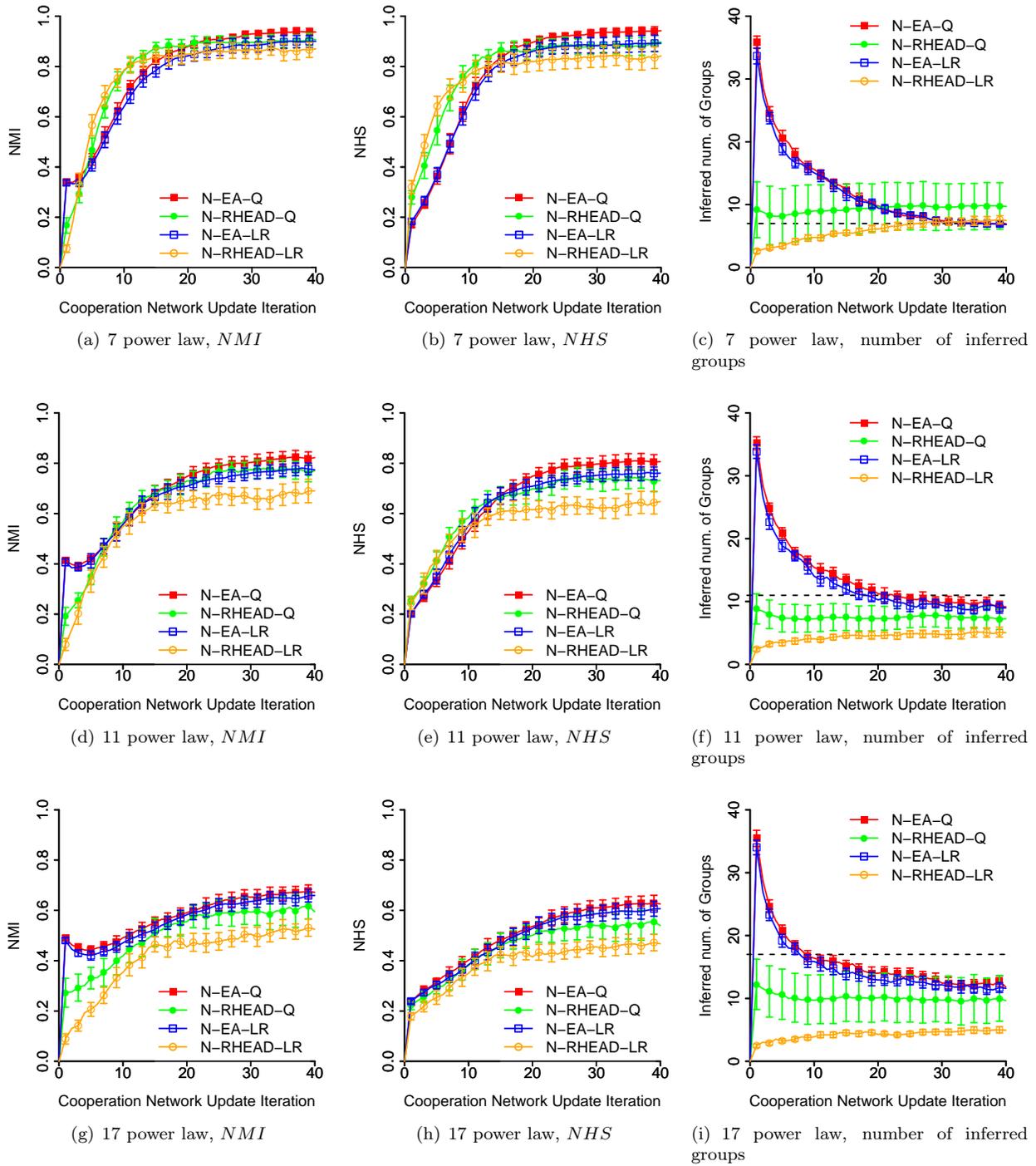


Figure 8.20: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N-EA and node-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = \{7, 11, 17\}$ power law group size distribution types.

Chapter 9

Interaction-based Group Modelling Applied to the Experiment by Chen and Li

This Chapter presents the empirical application of our interaction-based group modelling framework to the extracted actual behavioural data of the experiment by Chen and Li, which was introduced in Section 4.2.

9.1 Generation of the Interaction Data

In order to generate the interactions, we relied on the actual behavioural data, based on the *other-other* game (see Subsection 4.1.3), of the five treatments of the authentic experiment, for a total of 25 experimental runs.

Recall the experiment by Chen and Li described in Section 4.1, and the *other-other* game in particular. The participants were not put in direct interactions with each other; the sequence of participant ID, which actually defines the provider-receiver1-receiver2 (P, R_1, R_2) triplet, was generated only for the ultimate calculation of the participant payoffs. Instead, they reported their game decisions by means of questionnaires — one entry for each *in-in*, *out-out*, and *in-out* interaction types across five different token amounts — for a total of $3 * 5 = 15$ entries. In order to investigate the validity of the one-to-many approach, we decided to generate the sequence of participant IDs for each round of interactions, similarly to what we did for the related artificial society-based experiments, and to retrieve the actual reported behavioural data once the triplet had been defined. Moreover, considering that one survey entry is potentially applied to any of the other 15 participants of the experiment, we decided to generate $(3 * 5) * 15 = 225$ interactions. Algorithm 19 describes, in pseudocode, the loop of interactions of the 25 CL simulations considered. Note that the case in which the proposer P 's group identity is neither equal to R_1 's nor to R_2 's is omitted, since in the authentic experiments only two group structures were considered. Intuitively, “*date*” corresponds to the date-code which identifies the specific experimental run. The details of the recorded authentic behaviours per date-code can be found in Appendix 12.

Across the 25 experimental runs, the average size of group “1” is 9.08, $sd = 1.7059$, whilst the average size of group “2” is 6.8, $sd = 1.8028$. 397 entries for the *other-other* game were recorded, for a total of 5955 splits for R_1 and R_2 , corresponding to 1985 entries per interaction type. Table 9.1 lists the portions of recorded interactions based on the relationship between R_1 and R_2 . To conclude this Section, we remark the fact that in the most even case of $n = 16$, $m = 2$ equal size groups, $p(in - in) = 0.2178$, $p(out - out) = 0.2844$, and finally $p(in - out) = 0.4978$ — see Section 6.3.

Table 9.1: Distribution of preference towards either R_1 and R_2 across the 25 runs of the CL simulation.

Interaction Type	$R_1 = R_2$		$R_1 < R_2$		$R_1 > R_2$	
	Count	%	Count	%	Count	%
<i>in-in</i>	1653	83.27	110	5.54	222	11.18
<i>out-out</i>	1456	73.35	269	13.55	260	13.1
<i>in-out</i>	610	30.73	138	6.95	1237	62.32

Algorithm 19 CL simulation — interaction loop

```

1: #  $\mathcal{T}$  = array of  $n$  elements containing the true group identities
2: for  $i \in [1, 225]$  do
3:   for  $j \in [0, 4]$  do
4:      $e \leftarrow 200 + 50j$ 
5:      $perm \leftarrow$  sequence of  $n$  participantIDs uniformly generated
6:      $P \leftarrow perm[k]$ 
7:      $R_1 \leftarrow perm[(k + 1) \bmod n]$ 
8:      $R_2 \leftarrow perm[(k + 2) \bmod n]$ 
9:      $scenario \leftarrow \emptyset$ 
10:    if  $\mathcal{T}[R_1] = \mathcal{T}[R_2]$  then
11:      if  $\mathcal{T}[P] = \mathcal{T}[R_1]$  then
12:         $s \leftarrow in-in$ 
13:      else
14:         $s \leftarrow out-out$ 
15:      end if
16:    else
17:       $s \leftarrow in-out$ 
18:    end if
19:    extract the survey entry for  $P$ , round with endowment  $e$ , and scenario  $s$ .
20:    if  $s = in-out$  &  $\mathcal{T}[P] \neq \mathcal{T}[R_1]$  then
21:      swap the entry items, so that the first one is allocated to  $R_2$ 
22:    end if
23:  end for
24: end for

```

9.2 Cooperation Modelling Analysis

Given the findings of Chapter 6, we implemented the Cooperation Modelling (CM) component based on the discretisation of the offers for cooperation evaluation (see Equation (5.3)), whilst for the cooperation network update task we rely on the constant- α update rule (see Equation (5.10)), with $\alpha = 1$. Figure 9.1 depicts the average modularity score, and related standard deviation, calculated on the up-to-date cooperation networks, based on the *true* group identities of the 25 experimental runs. The Figure depicts the performance of CM based on seven different temporal sizes used for the accumulation (t) of the offers into the O_p vectors.

Similarly to the findings of Chapter 6, we observe that, except for the $t = 1$ case, larger temporal intervals would lead to generally higher modularity values. The fact that $t = 1$ does not follow this pattern is most likely due to the fact that the recorded entries — as aggregatively listed in Table 9.1 — manifest clear group behaviours: most of the *in-out* entries are of the type $R_1 > R_2$, and the vast majority of all other entries, across all three scenarios, are of the type $R_1 = R_2$, which are ignored by the minmax normalisation process of cooperation evaluation defined by Equation (5.2). In other words, the recorded behaviours are much less noisy than those generated by the derived artificial societies, hence a more “trusty” CM module, i.e. one with $t = 1$, might lead to higher and quicker inferences than a more “prudent” one, e.g. $t = 5$.

Independently on the t setup, furthermore, the CM module manages to quickly build cooperation networks

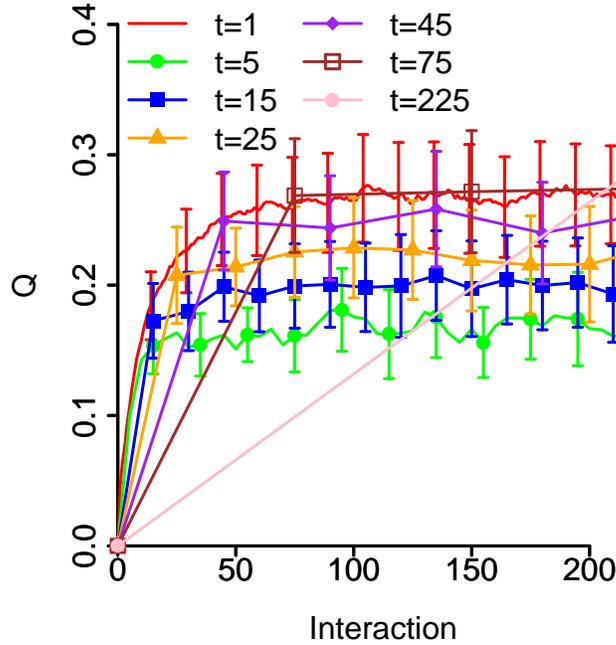


Figure 9.1: average *True* modularity score and related standard deviation, of the 25 experimental runs of the CL simulation, depending on different temporal sizes of accumulation of the interactions.

which *true* modularity is at least above 0.15. When the accumulation is at least of 25 consecutive interactions, moreover, the CM task leads to *true* modularity scores which, on average, are between 0.2 and 0.3. We remind, once again, that many real-life networks with clearly visible community structures have a modularity score which falls within this boundary.

9.3 Group Identity Detection Analysis

Table 9.2: N-EA setups investigated for the GID tasks of the Chen and Li experiment’s

Name	Population Size	Elite Rate	Fitness Evaluation Limit
N-EA-280	8	0.5	280
N-EA-2800	8	0.5	2800
N-EA	60	0.5	16200

Based on the findings of the previous section, we decided to perform the Group Identity Detection (GID) task on the networks obtained in the $t = \{1, 5\}$ case, that is, the settings of most online cooperation network update which lead to the best and worst *true* modularity scores.

The procedure used in this empirical evaluation was similar to the one used for the investigation presented in Chapter 8. First, we executed spectral partitioning on the 255 (for $t = 1$) and 51 (for $t = 5$) inferred networks for each of the 25 experimental runs. The average number of evaluations were 258.18 (sd = 29.53) for $t = 1$ and 281.34 (sd = 24.518) for $t = 5$. At this point, we decided to investigate three setups for N-EA, which are reported in Table 9.3. The three setups share the same mutation rate ($p_{mut} = 0.8$), parent reproduction probability ($p_{cross} = 0.8$) and, obviously, rank parent selection.

Intuitively, the setup for N-EA-280 was decided in order to obtain as many evolutionary generations as possible, yet by staying within the average number of evaluations of spectral partitioning. As we will show

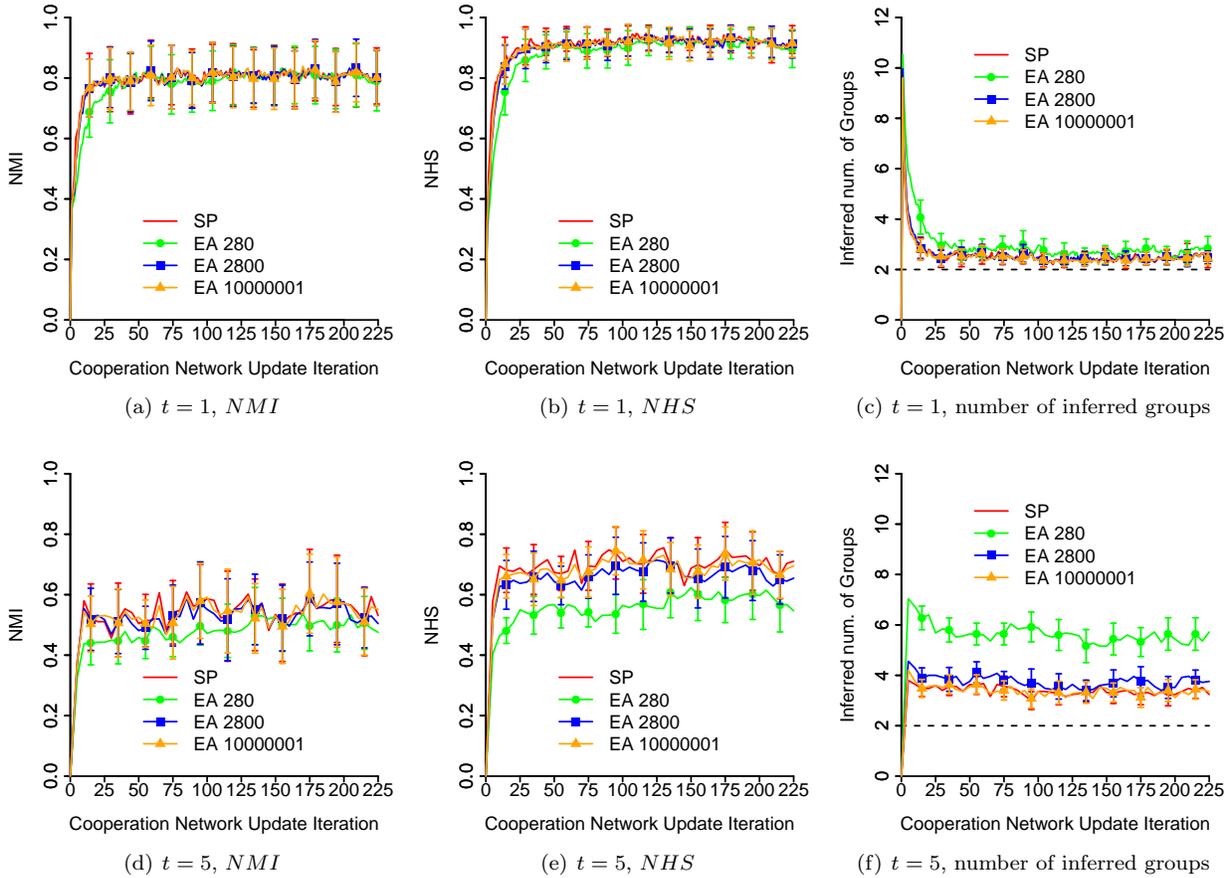


Figure 9.2: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of SP, N-EA-280 and N-EA-2800, based on the networks obtained from a CM phase with $t = \{1, 5\}$.

later, due to the results gathered for N-EA-280, the investigation of N-EA-258 — which would have aligned to the number of evaluations performed by spectral partitioning for $t = 1$ — became pointless. N-EA-2800 was introduced in order to observe if the algorithm would reach or improve spectral partitioning’s performance, given more evolutionary time. Finally, the original N-EA, used and analysed throughout Chapter 8, was also considered, so that its generalisability could be tested, and also to understand whether a longer evolutionary time would improve the inference of the *true* group structures of the experiments.

Figure 9.2 depicts the average performance (NMI , NHS , inferred number of groups), and related standard deviation, of SP, N-EA-280, N-EA-2800 and N-EA, for the networks inferred across the 25 runs of the CL simulations, $t = \{1, 5\}$.

As expected, the $t = 1$ case leads to the overall higher inference scores, both in terms of NMI and NHS , for all the four algorithms considered. Another expected result is the fact that all the algorithms, except for N-EA-280, produce the same inferential performance. That was the reason which discouraged us to pursue the N-EA investigation with a lower number of fitness evaluation limit, i.e. N-EA-258. On the other hand, we observe no significant difference in performance between N-EA-2800 and the original, longer lasting N-EA, though the latter appears to be slightly better performing, see Figure 9.2(e) as an example.

By looking at the inferred number of groups, i.e. Figures 9.2(c) and 9.2(f), we observe that, on average, none of the algorithms considered manage to infer $m = 2$ groups. For $t = 1$, the minimum average number of groups detected is 2.44 for spectral partitioning and N-EA (sd = 0.5831 and 0.5066 respectively), whilst the maximum is 2.92, sd = 0.9539, for N-EA-280. For the $t = 5$ case, the deviation from the *true* number of groups

is even more striking. The minimum average number of groups detected is 3.24 for spectral partitioning, $sd = 0.7234$, whilst the maximum is 5.72 for N-EA-280, $sd = 1.2754$.

9.4 Summary

This Chapter applied the best configurations of the group modelling framework, retrieved from the results of the empirical evaluations of Chapter 6 and 8 to the authentic behavioural data extracted by the experiment by Chen and Li, see Chapter 4. The promising results gathered, especially in case of frequent cooperation evaluation, cooperation network update, and group identity detection, further strengthen the validity of our interaction based group modelling approach.

In the next Chapter discussions about limitations of our study, possible improvement and future work are presented.

Chapter 10

Discussion

The empirical evaluations centred on the cooperation modelling component (see Chapter 6), those centred on the group identity detection component (see Chapter 8), and those based on the performance of the overall group modelling framework (Chapter 9), allow us to embark a thorough discussion about the validity of our interaction-based approach to group modelling, and highlight possible future improvements. The former are presented in Section (Section 10.1), whilst the latter are found in Section 10.2.

10.1 Recap, Limitations and Opportunities

A recap of the proposed methodology, together with its limitations, are summarised in this Section. Suggestions and idea for overcoming such drawbacks are also discussed. These provide the ground for future investigations.

The next Sections will analyse, each component and approaches implemented in our proposed group modelling framework. Moreover, in this Section we will also embark a more generic discussion related to the community structure detection task, with the aim to contribute to the broader network theory research field.

10.1.1 The *One-to-many* Approach to Interaction Modelling

In order to retrieve valuable information regarding the interactions occurring within a social synthetic environment — so that it would be possible to evaluate their levels of cooperation, hence partition the society into cooperative groups — the proposed group modelling framework relies on the so-called *one-to-many* approach and the related *Temporal Group Based (TGB)* metric of fairness of resource distribution [86].

In a nutshell, the *one-to-many* approach aims to gather different interactions each individual has with many others — not necessarily at the same time — so that the subsequent evaluations of e.g. cooperation can be broadened, from the single proposer individual-receiver individual perspective, up to the largest performer individual-society perspective, in accordance with the principles of complex systems. The *one-to-many* approach is at the core of *TGB*, which can be used to measure the levels of fairness of resource distribution an individual manifests towards many receiver individuals, who are assumed to belong to two distinct groups. Although *TGB* assumes perfect knowledge over the group identities of the receiver individuals, its principles can be still applied for our interaction-based group modelling approach, providing that interactions between a proposer individual and individuals belonging to the same group (*in-group*) and other groups (*out-group*) are observed altogether (*in-out*). Clearly, the observation of the *in-out* interaction types is the necessary condition for the application of our interaction-based group modelling framework. Hence, the definition of a temporal window t , aiming to gather *in-out* interaction types, constitutes a central aspect of the validity of our approach and modelling framework.

Through the investigations conducted, we observed that, the bigger t is, the better the existing *true* group structures can be identified in the society’s cooperation network (see Sections 6.6 and 9.2). This finding is somewhat not surprising, since the full observability of the ongoing interactions — which are based

on either authentic behavioural data (CL simulations, see Section 4.2) or derived artificial societies (CLAS, see Section 4.3), whom none of them change their group identities — would help with reducing the noise in the observed behaviours — which noise is due either to stochastic fluctuations in the CLAS scenarios or to “counter-intuitive” behavioural reports in the CL simulations. Nevertheless, through the results presented especially in Sections 8.3.1 and 9.3, we observed that a relatively small size for t with respect to the whole simulation task — $t = 10$ out of 400 for the CLAS scenarios; $t = 1$ out of 225 for the CL simulations — would still lead to very high scores in terms of unveiling the *true* group structures, even at early stages of the whole group modelling task and societal simulation. This not only means that the whole interaction-based group modelling approach can be successful, but also that it can provide very accurate correlations between e.g. group dynamics and temporal events derived from the interactions.

Another strong point in favour of the *one-to-many* approach is its apparent robustness against interactions which differ in magnitude, such as the five different tokens of the *other-other* allocation game considered in our study (see Section 4.1.3). This is achieved due to two reasons: (1) the group behaviours observed are generally easy to identify, especially for the CL simulations (see Table 9.1), which also partly justify why $t = 1$ is the temporal setting which leads to clear *in-group* vs. *out-group* boundaries (see e.g. Section 9.2); (2) the accumulation of interactions procedure we adopted allows for the averaging out of the offers made, hence the smoothing out of the different magnitudes of the offers. With this respect, it is not surprising that the $t = 5$ case leads to the worst identification of *in-group* vs. *out-group* boundaries.

The main objection against the application of *TGB*’s principles — which solely rely on *in-out* interaction types — might come from the consequent impossibility to process interaction types in which the receiver individuals are either both *in-group* (*in-in*) or are both *out-group* (*out-out*). Although in the CL simulation experiments, through $t = 1$, this drawback is partly reduced — the reported behaviours for *in-in* and *out-out* interactions mostly correspond to equal splits of the tokens, which subsequently allows the cooperation evaluation task to ignore these interactions (see Section 10.1.2) — this is clearly not achieved in the CLAS scenarios, since the agent behaviours are stochastic. Moreover, in the CLAS scenarios considered, we observed that the inference performance of the whole group modelling framework decreases with the increase of the probability of observing *out-out* interaction types. To the best of our knowledge, however, we have not found any existing work on group behaviours which go beyond the *in-group* vs. *out-group* dichotomy.

Although the studies presented in this dissertation are solely based on one type of interactions, that is the *other-other* game, the group modelling task via the *one-to-many* approach already provided promising results in our previous studies on the ultimatum game [91, 83, 87]. However, in those experiments, the endowment each proposer had to split with a receiver was maintained the same across all the interactions, unlike the scenarios considered in the experiments presented in the dissertation.

Another limitation present in this dissertation is its lack of comparison with alternative methods of interaction analysis. Although this investigation was in part conducted in our previous work [91, 87], we are aware that further future work should be conducted towards this direction. Nevertheless, it is worth pointing out that, in order to have a clearer picture of the whole *emergent* properties of a *complex system*, it is necessary to go beyond the single-unit perspective and rather focus, more globally, at the system as a whole. Arguably, the *one-to-many* approach goes along these principles.

10.1.2 Cooperation Modelling

Given the currently gathered interactions, the first module of our proposed interaction-based group modelling framework, namely the Cooperation Modelling component (CM), is in charge of (1) evaluating the different levels of cooperation going from the individuals performing the interactions to those receiving them, and (2) update the so-called cooperation network $\mathcal{N} = (\mathcal{S}, \mathcal{C})$ — where $\mathcal{S} = \{a_1, a_2, \dots, a_n\}$ is the set of n nodes, each node corresponding to an individual of the society, and \mathcal{C} is the set of links/adjacency matrix, which specifies the cooperative connections between nodes/individuals — so that *in-group* and *out-group* boundaries could be eventually detected. In other words, the CM module aims to build a complex dynamic network of cooperation of the underlying social synthetic environment.

Computationally, the CM module modifies the edges of \mathcal{N} , and the networks generated range from being fully connected, directed and weighted — i.e. the initial unbiased condition we defined — to being sparse,

undirected and unweighted. The ultimate goal of CM is to represent the *true* cooperation network of the society, in which undirected and unweighted edges exist only between individuals/nodes belonging to the same group.

The duties of the CM task can be divided into two pipelined phases: the first one, called cooperation evaluation, transforms the sequence of observed interactions of each individual, into *cooperation* values; the subsequent cooperation network update task leverages on the cooperation values to update \mathcal{N} . For the former task we considered two transformation functions, for the latter task we considered three update rules.

Through the results obtained, we observed that (1) the conversion of the observed interactions into discretised labels/values *in-group*/1 and *out-group*/0, together with (2) the use of constant- α update rule ($\alpha = 1$) for cooperation network update, can lead to the quickest and most accurate \mathcal{N} networks. Moreover, the discretisation of the observed interactions is achieved by the min-max normalisation function, and subsequent use of discretisation threshold $\tau = 0.5$, see Section 5.2. The accuracy of \mathcal{N} to represent the *true* group identities was evaluated by calculating the network’s *modularity* score [161] based on the *true* group structures. The higher the registered modularity score, the better \mathcal{N} manifested the *true* group structures.

The evaluation of cooperation via min-max normalisation has a clear advantage against the straightforward use of the *raw* observed interactions, see for instance Figures 6.6 and 6.7. Moreover, we attempted to leverage on reciprocal altruism, i.e. *collaboration* [127, 205, 32] via the TD0 update rule [199], though the results obtained were not optimal. The reason why TD0 led to suboptimal scores were dual: (1) the weights of the updated edges in \mathcal{N} could go beyond the $[0, 1]$ interval, and (2) depending on particular configurations of the reciprocal edge weights, the update rule was leading to inverted representations — i.e. *true in-group* cooperative relationships were updated towards zero cooperation and vice versa, see Section 6.5.

On the other hand, the results showed that the constant- α update rule [199] is able to maintain the edge weights between the $[0, 1]$ interval, and that it is not affected by special configuration cases of \mathcal{N} . It is somehow not surprising that this rather simple update rule is effective for our domain of application, given it has been used in many desperate fields, such as in artificial neural networks [95], in ant colony optimisation [42], and in reinforcement learning [199]. Moreover, the fact that it maintains cooperation estimates by considering current observations as more important than past ones, it makes it a suitable update rule even for social synthetic environments in which the group structures are not static, but rather evolve dynamically. The fact that in our experiments we observed that $\alpha = 1$ leads to the best \mathcal{N} networks (see Section 6.4) was most likely due to the rather clear and consistent group behaviours of the individuals of the society. In other, more noisy or stochastic environments, it might be possible that a lower value for α might lead to better results.

The main critique against the use of CM might be its over-dependency on the occurring interactions, which makes the whole component a mere bulky function transforming interactions into cooperation networks. For instance, the module cannot infer unseen cooperative relationships, which might be problematic especially in case the individuals have the power to choose who to interact with, e.g. between *in-group* individuals only. Prior to writing this dissertation, we had attempted to overcome these drawbacks. More specifically, during the research study abroad period, in which the author visited the Computational Behavior Group at University of Southern California¹, it was decided to setup a laboratory experiment in which the participants would play an ad hoc version of the social ultimatum game [110, 24]. Initially, each participant had to label each other participant as either a friend or a stranger. Then, at each iteration of the social ultimatum game, each proposer would face the option to choose a receiver, between two available opponents uniformly sampled among her “friend” and “stranger” set, depending on the three conditions friend-friend, stranger-stranger, and friend-stranger. Unfortunately, the experiment could never be conducted, since at its very execution a bug causing connection crashes was found². Beyond this failed attempt, a possible computational strategy to overcome the cooperation inference of unseen interactions might be achieved by leveraging on e.g. the principles of social balance theory [103, 7]. Moreover, it might be possible that the use of social balance theory-derived inferential mechanisms (see e.g. [130]) might help with repairing wrongly classified interactions, hence links of the cooperation network.

Another weakness of our approach resides on the fact that the CM module does not explicitly represent the temporal structure of the observed interactions, although in principle is able to represent up-to-date

¹<http://cbg.isi.edu/>

²The description of this odd story deserves to be done orally.

cooperation estimates. A possible way to overcome this problem could be the use of a smoothing mechanism, similar to the pheromone evaporation technique existent in ant colony optimisation algorithms [42] which would bring the cooperation values towards value 0.5 — i.e. full uncertainty — which is also the value used to initialise the cooperation network at the beginning of the group modelling task.

10.1.3 Group Identity Detection

Given the current estimates of the society’s cooperation network, the last component of the interaction-based group modelling framework, namely the Group Identity Detection (GID) component, is in charge of performing the so-called *Community Structure Detection* task (CSD), in order to partition nodes/individuals of the network/society so that the *within*-group/community cooperation is maximised and the *between*-group/community cooperation is minimised, in accordance with the social dilemma’s findings on group identity [39]. For CSD we focused on *modularity maximisation* [160], a technique which is widely used and regarded as reliable in the related network theory research field.

The main research was centred on the application of evolutionary computation for CSD. The Evolutionary Algorithms (EAs) examined evolve candidate solutions representing network partitions into community structures. Moreover, we relied on spectral partitioning as baseline algorithm, as it is not only well established, but also greedy and deterministic [162]. Given a rather generic EA structure, we overall investigated the efficacy of three chromosome representations — *node*, *group* and *edge* — two main evolutionary approaches — one based on single population and the other based on *sealed* niches — each niche composed of candidate solutions which are forced to partition the network into a fixed number of community structures — and four possible activation mechanisms which selectively evolve niches — two deterministically and two stochastically. Finally, though the main emphasis was on the use of modularity as the fitness function to be maximised across all the algorithms considered, we also investigated the use of *LinkRank* [112] as an alternative function.

In a nutshell, the main results gathered in Chapter 8 showed that: (1) a single population, node-based chromosome representation evolutionary algorithm (N-EA), produces on average the highest matches, between inferred and *true* group structures, across the seven artificial society structures investigated, though its speed of convergence to high inference values is rather slow; (2) the node-based niching approach might overcome the convergence speed limitation of the single population counterpart, though it has a significant impact on its inference performance; (3) a deterministic niching activation mechanism which evolves the current best performing niche, together with a limited number of contiguous ones, called *reading head* (N-RHEAD), appears to carry the highest potential for improvement; (4) modularity is generally a better fitness function than LinkRank, at least for the class of problems investigated in this dissertation.

The key property of N-EA, which probably leads to its strikingly good performance compared to the other algorithms considered, is that it generally starts by inferring a rather high number of groups, and then it gradually reduces them towards the *true* one. On the other hand, spectral partitioning and the other EAs considered, tend to converge rather quickly, thereby, being largely dependent on the initial noise existent in the cooperation networks.

In general the use of evolutionary computation has provided high robustness with respect to the dynamic changes of the underlying \mathcal{N} network, which does not seem easily achievable by spectral partitioning and similar methods. All the EAs considered evolve the same population throughout the whole GID execution, meaning that the dynamic changes in \mathcal{N} result, at most, in a reordering of the chromosome population, in terms of scored fitness, which has a subsequent impact on both their survival through evolutionary generation and on their probability of being selected for reproduction. On the other hand, spectral partitioning has to symmetrise the adjacency matrix \mathcal{C} , image of \mathcal{N} , each time a cooperation network update task has occurred. This not only involves the reset and recalculation of new solutions and the consequent discard of previous ones, but also the introduction of artefacts in the matrix due to the symmetrisation process.

10.1.4 Chromosome Representation

In order to attempt the search for the optimal solution of a given problem, the very first step of the EA design centres around the definition of its *chromosome representation* [10]. All other aspects of evolutionary

computation, such as the variation operators and fitness evaluations, necessarily depend on the way the chromosomes represent their related candidate solution.

In this study we considered three *string*-based representations. Both of them define, at least, chromosomes as a sequence (string) of n genes, gene i corresponding to node $v_i \in \mathcal{S}$ of \mathcal{N} , with each allele value sampled from an alphabet of symbols. The three representation differ from each other as follows:

- the *node* representation — which is commonly referred to as *object* representation in e.g. evolutionary clustering [100] — considers allele value j of gene i as the community structure identity label of node v_i . The proposed EAs based on this representation evolve the group identity of each single node independently of each other;
- the *group* representation — derived from the work of Falkenauer [55], extends the previous representation by keeping, for each chromosome, the original node part and the list of unique group identity labels (group part). The proposed EAs based on this representation evolve the chromosomes by means of recombination operators centred on the group part, with repercussion on the node part. In other words, the group-based EAs evolve group structures;
- the *edge* representation — which is commonly referred to as *locus based* in e.g. evolutionary CSD [172] — considers allele value j of gene i as the *existing* directed edge $(i, j) \in \mathcal{C}$ connecting node v_i with node v_j . The proposed EAs based on this representation evolve subnetworks of \mathcal{N} , and a subsequent conversion procedure translates these into community structures and assigns community structure identity labels to each gene.

One of the most remarkable findings of our research was that the node representation, which leads to the best performing algorithm in terms of unveiling the *true* group structures at the end of the modelling task (i.e. N-EA), is also one of the most straightforward one, which also possibly holds the most intuitive chromosome representation for CSD and clustering in general. On the other hand, the group and edge representations seemed to excessively suffer the noise in the networks, though it is likely that their performances could be improved, given that they could benefit from longer evolutionary time, and further fine parameter tuning is conducted. Nevertheless, some generic considerations against their application to our dynamic domains could be made.

The group representation was inspired by the original work of Falkenauer [55], who introduced the Grouping Genetic Algorithm (GGA), followed by the application of his work to graph colouring problems [50], which share some properties with CSD problems. GGA, and similarly our group representation EAs, are composed of three evolutionary operators: group-based crossover, group-based mutation, and group inversion.

Our single population, group-based algorithm (G-EA) seems to forbid the search towards a high number of groups, as it can be clearly seen in Figures 8.5 and 8.6, in which the number of inferred groups is seemingly constant independently on the stage of the cooperation network update task — hence on the less random \mathcal{N} networks — and across the number of *true* groups present in the society.

Before conducting the research presented in this dissertation, we first investigated the impact that the six mutation operators implemented, even together with the node-based mutation operator, have on the performance of G-EA. We tested the different algorithm configurations on the $m = \{4, 7, 11, 17\}$ equal group size distribution scenarios. We could not identify any clear pattern which could let us decide which mutation operation would lead to better or worse results. However, prior to certainly consider the group-based mutation operations as a cause of G-EA’s suboptimal performance, we believe that more investigations on the fine tuning of the mutation rate should be conducted. It might be possible that a too high rate — e.g. $p_{mut} = 0.8$ used in our investigations — might lead to the disappearance of small sized groups, e.g. by means of group growth.

We are not entirely aware of the advantages derived by the use of the inversion operator. Similarly to what observed in the mutation operations, we did not observe any significant difference in performance when inversion was toggled. Hence, to remain consistent with Falkenauer’s work, we decided to keep it implemented. We however suspect that the inversion operation, which was introduced in order to promote the inheritance,

by the offsprings, of “good” genes from the parents, might limit the exploration of a high number of groups, and the ability of the algorithm to adjust to the dynamic changes of the underlying network.

Hence, we believe that the leading cause for G-EA’s impossibility to explore a higher number of groups come from the group-based crossover operation, which transfers whole groups between parents and offsprings. We observed that through this operation many group labels disappear, whilst instead, through the uniform crossover operation used by N-EA, more groups are likely to appear. It is worth pointing out that Falkenauer presented his GGA, with group-based crossover operation, as a possible way to overcome uniform crossover’s tendency to generate offsprings representing solutions which could be far from their parents’ [55].

Finally, we are aware that G-EA differs in structure from GGA; a preliminary analysis of performance of the two algorithms, however, did not present any statistically significant difference. To the best of our knowledge, there has not been any application of Falkenauer’s GGA, or any of related group-based EAs, to community structure detection problems. Therefore, prior to conclude that this representation and approach is not viable for CSD, we suggest that more work on this direction is required.

The single population, edge-based algorithm (E-EA), was inspired, among all, by Pizzuti’s GaNet algorithm [174, 177]. In her work, GaNet provided very promising results in the synthetic network benchmark of Girvan and Newman [77]. Similarly to GaNet, we implemented, in our E-EA, the chromosome-repair initialisation procedure and uniform chromosome. The peculiarity of E-EA comes from the mutation operation, which is extended to work with weighted networks: rather than sampling a neighbouring edge uniformly, E-EA samples them proportionally to their weight strength.

We believe that the reasons why E-EA does not achieve an optimal performance are mostly due to the high noise of the networks, which are composed of edges which connect, erroneously, nodes belonging to different groups. This has clearly an impact on E-EA: a high node connectivity reduces the efficacy of the chromosome repair process to steer the evolutionary search and reduce its scope. Moreover, the network’s high node degree affect E-EA’s mutation operator, which has now the tendency to generate big subnetworks, and this was clearly observed in the plots of Figures 8.5 and 8.6 depicting the evolution of the inferred number of groups. Nevertheless, we believe that the chromosome repair process is extremely beneficial in case of locality of networks manifesting a more sparse edge distribution — e.g. in case the individuals mostly interact with a subset of the society — and that the edge-based mutation might be extremely powerful for local-searches of community identity assignments. As a consequence, it is possible that (all) the edge-based EAs would benefit from an alternative cooperation network initialisation, i.e. a zero network.

To conclude, we believe that E-EA is not particularly suitable for the early stages of the GID task, when little is known about the cooperation existing in the society. Nevertheless, E-EA can become potentially advantageous at later stages, when the underlying network becomes more sparse, for instance to refine the inferred solutions obtained by e.g. N-EA. Moreover, by considering e.g. Figures 8.5(a) and 8.5(b), which showed a non plateaued inference trend, it is likely that E-EA might improve its performance providing it could evolve its candidate solution for a higher number of generations.

10.1.5 Niching Approach

Although N-EA appeared to be the most promising chromosome representation, its speed of convergence towards high inference values, especially in case of a low number of groups, i.e. $m = \{4, 7\}$, seem to be discouraging with respect to its application for online group modelling. Similarly to E-EA and G-EA, which showed a rather constant behaviour across the societal scenarios, we observe that N-EA usually begins by inferring a relatively high number of groups — between 35 and 36 on average — independently of the *true* number of groups existing in the society. This appears to be the leading cause of N-EA’s slow convergence to optimal values.

The niching approach was introduced with the assumption that if we were able to maintain different local optima, e.g. those assuming both a low and high number of groups, the evolutionary process better explore and exploit the search space. The niching approach we implemented guarantees that each niche k would be populated by chromosomes which related candidate solution are forced to partitions a network into exactly k groups. This was achieved by ad hoc niche repair algorithms.

The pilot investigation conducted in Section 8.4.1, which assumed knowledge about the *true* m , showed

very promising results in terms of speed of convergence, though their overall inference performance, especially at the end of the group modelling task, was lower than N-EA’s. Nevertheless, a key finding showed that N-EA was tendentiously inferring a number of groups different from the *true* one, which was most likely due to the noise in the network. This further motivated our investigation on the use of n niches, together with an appropriate activation algorithm which could selectively evolve the niches.

We then continued our investigation by considering four simple algorithms for niching activation: (1) *reading head*, which implements a deterministic hill climber and evolves a set z of contiguous niches, the centre-most which, called *pivot activator*, is always pointing to the best performing niche; (2) UCB1, a deterministic algorithm designed to solve the reinforcement learning’s n -armed bandit problem [199]; (3) ϵ -greedy, a stochastic algorithm which, similarly to UCB1, is designed to solve the reinforcement learning’s n -armed bandit problem; (4) roulette wheel selection, a stochastic algorithm which selects the niches to be activated based on their fitness scores proportional to the overall landscape. This activation mechanism is inspired by the homonymous one used for parent selection in evolutionary algorithms.

The study conducted on the niching approach, however, did not allow us to achieve better overall performances than N-EA, though a better speed of convergence was still achieved. The main pattern that we observed was that the niche activation algorithms were still generally assuming a very low number of groups than the *true* one, that is, they could not replicate the effective aggregative pattern obtained by N-EA.

Among the four algorithms investigated we observed that the reading head approach is more successful in presence of a small number of groups of equal size, whilst the UCB1 algorithm is more successful in presence of many groups of equal size; their performance was essentially the same in presence of power law group size distribution. The algorithm based on roulette wheel selection demonstrated an extraordinary good performance only in the $m = 4$ case, whilst for the remaining cases it was always among the worst performing activation techniques. A similar behaviour was observed with the ϵ -greedy algorithm.

Possibly, the most interesting aspect of this finding is that the two best performing algorithms activate the niches deterministically, while the two less well-performing algorithms choose which niche to activate stochastically. This seems to suggest that the four algorithms cannot explore the niche landscape well enough — i.e. the deterministic approaches keep activating the same niches, and the stochastic approaches do not allow suboptimal niches to be activated often enough to improve the current solutions — and that they excessively suffer from the initialisation phase, similarly to E-EA and G-EA. Two possible reasons for the lack of exploration are (1) the strictness of the niche repair process, and (2) the excessively large size of the niche landscape.

With respect to the first assumption, it might be possible that the implemented procedures add unnecessary artefacts in order to generate chromosomes which partition the network in a fixed number of groups. This is most likely observed by analysing the evolutionary trend about the inferred number of groups of UCB1, across the seven society types, which is constant throughout the whole GID task, see e.g. Figure 8.10(c). Possibly, a more structured approach to niche repair, unlike the merely stochastic one implemented in our study, might bring improvement in terms of inference performance.

With respect to the second reason, a possible solution could consider a lower number of niches, in which the chromosomes represent a number of communities within a range rather than a strict number. Alternatively, a multi-niching activation approach, e.g. one composed of x reading head pivots, initially evenly spread through the niche landscape, might allow for a more thorough exploration and exploitation of the niche landscape.

10.1.6 Target Objective

Although marginally explored, the final stage of our empirical research was centred on the impact the fitness function has on evolutionary community structure detection. In accordance with the critiques against modularity (see Section 3.2.3), and more particularly with its resolution limit — which essentially makes either small-sized communities difficult to spot in favour or bigger ones, or vice versa, see Section 3.2.3 — we observed that E-EA and G-EA suboptimal performances in terms of unveiling the *true* group structures did not correspond to their impossibility to find candidate solutions with high modularity scores, see for instance Figure 8.13(d). Moreover, we observed that N-EA often manages to find candidate solutions which modularity score is on average equal to the modularity score of the *true* group identities, despite the fact

that it might infer a lower-than-*true* number of groups, see e.g. Figures 8.14(c) and 8.6(i).

Motivated by these findings we investigated whether an alternative measure of the quality of the network partitioning — i.e. LinkRank [112] — would improve the performance of the all the EAs considered. We are aware of the limitations of this investigation, especially considering that there exist many other fitness functions, e.g. the resolution-based modularity [184] and especially the *community score* [6], which has been extensively used for evolutionary community structure detection. We decided to consider LinkRank due to the different interpretation it gives to the notion of community structures — this time centred on the concept of random walks and *flow* of the network through its edge directionality — as opposed to modularity — which quantifies the goodness of a network partitioning by how much it deviates from a random graph with the same edge degree distribution. Moreover, and arguably, modularity is more similar to its resolution-based version and community score than to LinkRank. The results gathered, however, showed that modularity appears to be a more suitable objective function than LinkRank, at least for the class of problems we here investigated. This difference in performance was observed especially for the edge and group chromosome representation-based algorithms (see e.g. Figure 8.15(a) and 8.15(b)), and by observing the two different evolutionary trends, in terms of number of inferred groups, obtained by the node-based reading head algorithm, see e.g. Figure 8.17(i).

We conclude that the *flow* of the network — central to LinkRank — appears not to be a key feature for the detection of the *true* cooperative group structures considered in our study. These in fact assume reciprocity, high *within*-group and low *between*-group cooperation, features which appear to be more similar to modularity’s node degree distribution key feature. Nevertheless, we do not exclude that a difference fine tuning of parameter α used in its calculation, see Equation (3.9), might provide significant improvements on the unveiling of the *true* group structures.

10.1.7 Interaction-based Group Modelling as a Whole

The Group Modelling (GM) framework has proven, at least in the experiments conducted in this dissertation, to be rather robust against noisy, stochastic, and dynamic social synthetic environments populated by artificial agents. Moreover, the empirical evaluation conducted on the CL simulations (see Chapter 9), provides evidence to state that the conceptual structure of the framework could be potentially successful in unveiling existing *true* group structures in highly interacting, social synthetic environments, populated by human-controlled avatars.

Although it was partly tackled by the application of our GM framework on the CL simulations, the main critique against the above claim is that we heavily relied on artificial societies which, no matter how *believable* their agents might be, they still introduce some forms of artefacts and biases [94]. Moreover, the CL simulations are much simpler than the artificial society counterparts, which poses questions regarding the *validation* of the experimental setup as a whole [94].

We attempted to address the limitations of a research heavily based on artificial societies in a number of ways. First, it is worth noting that the original goal of our GM framework was its application in an adaptive multiplayer serious game for teaching conflict resolution skills [225]. However, for the final game design [107], the role of the GM framework was not prioritised. In fact, the game considers a society of four individuals and no group-based conflicts. Second, as described in Section 10.1.2, we failed to conduct social ultimatum game experiments with the aim to apply our GM framework to (1) more than sixteen human participants, (2) non-induced group/social structures, and (3) different interaction protocol. Finally, we highlighted in Chapter 4 the difficulties we faced when we sought for the most appropriate benchmark social synthetic environment scenarios. For these reasons, one of the most immediate future research paths is the application of our GM framework to social synthetic environments populated by human-controlled avatars. We are confident that, if the users have the power to choose who interact with, the noise in the interactions would be certainly reduced. This, however, might introduce an excessive reduction of *in-out* interactions — i.e. the necessary condition of the application of our interaction-based GM framework based on the principles of the temporal group-based fairness metrics [86]; therefore, the consequent investigation about the ability to perform group modelling based on e.g. *in-in* interaction types would be due.

Another limitation resides on the apparent scaling issues that might make the evolutionary computation

approach unfeasible, for instance, for group modelling of massively multiplayer online games. The three chromosome representations considered have length equal to the number of nodes of the network. As a consequence, for a network of thousands of nodes and many groups — which is reasonable to expect especially in massively multiplayer online games — our EAs might require an intractable amount of time to infer acceptable community structure partitions. On the other hand, a more greedy approach might be preferred, even if it requires e.g. the transformation of \mathcal{N} into a sparse undirected and unweighted network, in change of quicker and *acceptable* community structure partitions. Nevertheless, it is worth stressing that it is most likely that the larger the society is, the more local the interactions are, meaning that an EA with initialisation phase which reduces the search space, like the one implemented by E-EA, might still be a viable approach.

10.1.8 Community Structure Detection

The empirical investigation, conducted especially in Chapter 8, allows us to contribute to the more generic research on Community Structure Detection (CSD).

$n = 128, m = 4$ Equal Group Size Distribution Networks

A key finding appears to be the ease of detection of the *true* community structures in the $n = 128, m = 4$ equal size case. Surprisingly, for this case, even the highly stochastic niching activation approach based on roulette wheel selection (N-RWS) leads to an average inference performance of 0.9975 (sd = 0.005) for the *Normalised Mutual Information* (*NMI*, see Section 8.1.1) and 0.9984 (sd = 0.0032) for the *Normalised Hungarian Score* (*NHS*, see Section 8.1.2), as depicted in Figure 8.10. We also observe that it is the only societal case in which the pilot niche study with knowledge about m is both converging to high inference values quicker than N-EA, and its final performance is not significantly lower than N-EA, see Figure 8.7. Moreover, we register the very good results, both in terms of speed of convergence and final inference performance of the reading head niche activation mechanism combined with LinkRank, see Figures 8.15 and 8.18. In addition, the inferred cooperation networks led to average modularity values of 0.1874 (sd = 0.005), see Figure 8.3. This performance was unmatched by any other society type, which at most reached an average modularity score of 0.1345 (sd = 0.0088), see Figure 8.2.1.

This was most likely due to the fact that the related CLAS societies would generate *in-out* interaction types with the highest probability than the other CLASs see Table 8.1. As a result, the inferred cooperation networks for the $n = 128, m = 4$ equal size group distribution manifest the *true* community structures more accurately than the other cases, and this has consequences on the efficacy of the group identity detection module to unveil the *true* group structures.

Group Size Distribution

Another important finding related to the case scenario structures comes from the comparison of performance in inferring the *true* communities, in presence of equal or power law group size distributions. More specifically, we observe that the inference performance registered in the equal size case is generally lower than the one observed for the power law counterparts, especially with increasing values for m ; this is also explained by the fact that low m values lead to a less striking power law distribution, see Figure 8.2. The reason why this occurs, we believe, is due to the fact that it is *easier* to detect large-sized communities — which might be considered a manifestation of modularity’s resolution limit [122] — and that the cost of the consequent mis-detection of small-sized communities is not as severe as, for instance, mismatching large portions of equal sized groups, which is more likely to occur since these generally present more regularities in the edge distributions.

Performance Measures

The use of two performance measures to evaluate how much the inferred group identities (\mathcal{H}) unveil the *true* ones (\mathcal{T}), that is the normalised mutual information *NMI* and the normalised hungarian algorithm score *NHS* — see Section 8.1 — showed how *NHS* is generally more *rigid* than *NMI*, across all the societal



Figure 10.1: An example of two partition matchings for which NHS returns 0.66667 for both cases, whilst NMI returns 0.5158 in the first example and 0.5794 in the second example.

scenarios and algorithms considered, with NMI becoming seemingly more “generous” with the increase of m . Although both measures depend on a confusion matrix C , which item $C_{i,j}$ holds the number of nodes of community i of partitioning \mathcal{H} which belongs to community j of partitioning \mathcal{T} (see Figure 8.1), NMI relies of the entropic information related to it, that is, it gives an idea of how much it can be inferred about \mathcal{T} , given \mathcal{H} , which is explained by the logarithms and normalisations required for its calculation. On the other hand, NHS is merely solving a matching problem between two strings with different alphabets. In other words, NHS ’s dependency on the number of communities of \mathcal{H} and \mathcal{H} is solely occurring during the construction of the confusion matrix, whilst NMI leverages on them also during the calculation of its scores.

As an example, consider Figure 10.1. It depicts two strings, A and B , with A assuming $|A| = 2$ communities in Example 1 (Figure 10.1(a)) and $|A| = 3$ communities in Example 2 (Figure 10.1(b)), whilst B represents the same partition into $|B| = 3$ communities across the two examples. The NMI score is 0.5158 in the first example and 0.5794 in the second example, suggesting a higher inferential accuracy in the latter case. On the other hand, NHS returns the same score, 0.6667 in both examples. NHS detects two mismatches in both examples, and this is directly retrievable by multiplying its score by $n = 6$. NMI does not directly suggest how “far” A is from B , given its rather complicated formulation.

Although disputable, we would promote NHS as a more generic measure for CSD performance evaluation. This claim becomes even stronger, if research is focused on networks with many *true* communities, since NMI would tend to overestimate (more than NHS) the performance of the algorithms considered. We also believe that the main strength of NHS is on its intuitiveness, as the actual number of matched nodes can be *easily* estimated by observing the given strings.

Modularity, Resolution Limit, and Maximisation Approach

The results gathered showed that the inference performance of all the algorithms considered worsen with the increase of the number of groups, which also means that their sizes tend to be reduced. Moreover, we observed the remarkable lower-than-*true* number of groups detected by e.g. E-EA and G-EA across the $m = \{7, 11, 17\}$ equal and power law group size distributions. In other words, it appears that even the most generic modularity measure — which can be applied to any network typology — might still have a resolution limit, similarly to its related modularity measure defined for undirected and unweighted networks [122].

However, we believe that the causes for the decay in performance are also dependent on the algorithm adopted. In support of this claim, we have observed that N-EA — which is the best algorithm in terms of NMI/NHS performance at the end of the group modelling task — manifested a unique *aggregative* behaviour with respect to the estimation of the number of groups of the given networks. In fact, except for the CL simulation cases covered in Chapter 9, N-EA had the tendency to start inferring a high number of groups — i.e. between 36 and 35 on average — and then slowly refining its estimations, towards the *true* m values, though in some cases N-EA was inferring a lower-than-*true* number of groups (e.g. $m = 17$ power law group size distribution, Figure 8.6(i)). On the other hand, spectral partitioning, the group and edge representations, and all the niche-based evolutionary approaches, did not manifest the same evolutionary behaviour. Instead they converged rather early during the group identity detection task — i.e. when the cooperation networks were highly noisy — to a lower-than-*true* number of groups.

In a nutshell, modularity appears to have the tendency to reward candidate solutions which assume a low number of groups, hence their bigger size; the aggregative nature of N-EA’s computation appears to reduce this drawback. Although the final inference performance is not optimal, we are confident that N-EA, given a reasonably long computation time, would even improve its performance, clearly within the limits dictated by the noise in the network.

Consequently, we observed that a rather structured approach for community structure detection might be not advisable, especially in presence of highly noisy networks. Spectral partitioning seems to have a clear disadvantage in presence of an odd number of groups — as it always attempts to split a given network into two subnetworks — and those with ambiguity in node assignment to community structures, since early steps of spectral partitioning are not revised later on. On the other hand, evolutionary computation, and N-EA in particular, might occasionally find node assignments into communities of very small sizes, or even very articulated communities, given their stochastic nature. Moreover, it appeared that the drawbacks of N-EA in terms of convergence to optimal inference values are reduced with the increase of m .

10.2 Extensibility

To the best of our knowledge, there has not been any prior attempt to unveil existing group structures in social synthetic environments by observing the ongoing interactions and measuring their cooperation levels. Our research has addressed this problem based on a computational framework which relies on a *one-to-many* approach of gathering ongoing interactions, followed by their evaluations in term of cooperation, then construction of the society’s cooperation networks, and finalised by the execution of community structure detection. Nevertheless, the research conducted can be certainly be improved and extended in many different ways.

10.2.1 *One-to-many* Approach

As previously highlighted in Section 10.1.1, further research should be conducted on the *one-to-many* approach. Comparative analysis of performance with another approach, e.g. one solely centred on a single individual perspective, which was partly done in our previous work [91], needs to be extended.

Moreover, this approach for the gathering and analysis of the interactions, and especially *TGB* — which constitutes the necessary condition for the application of our interaction-based group modelling framework — needs to be improved, so that it could provide efficacy in detecting *in-group* and *out-group* boundaries even in presence of only *in-in* or *out-out* interaction types.

With this respect, we believe that the same crowdsourcing approach used to validate *TGB* [86] can possibly help with the definition/validation of metrics centred on other facets of group behaviours. Moreover, it is likely that through crowdsourcing it might be possible to retrieve cultural-based patterns of cooperation and group dynamics. Similarly, crowdsourcing can even help with the design of metrics which are not necessarily tightened to resource management/exchange scenarios, hence allow the application of the group modelling framework to e.g. interacting scenarios based on text/speech exchange, such as online chats, forums, or even computer games mostly based on storytelling.

10.2.2 Cooperation Modelling

The cooperation evaluation task via min-max normalisation of the interactions has two benefits: (1) it allows for the representation of \mathcal{N} the closest possible to undirected and unweighted networks, and (2) it allows for the possible future combination of our indirect interaction-based approach of cooperation evaluation with more direct, user-centred feedback mechanisms [107], such as the “like/dislike” buttons used in many Web 2.0 sites. Moreover, and despite the bad performance registered by the *collaboration*-oriented TD0 update rule, the representation of cooperation into bounded values could be used for the exploration of other aspects of interactions, such as the five conflict management styles [205, 32].

Alternatively, the cooperation modelling component could be enriched, for instance by adding user-centred learning/modelling features, with the consequence of augmenting the cooperation network with node attributes, or even representing social synthetic environments via multiplex networks. In this way, for instance, it could be possible to investigate multi-ethnic societies (node attributes) and related cooperation (edge weights and directions). Similarly, it might even be possible to embed other user modelling techniques, for those centred on user affective modelling [146].

Another different line of investigation might involve models of societal cooperative structures which go beyond the network representation. Alternative *dynamic* forms of clustering, such as those obtained by means of self-organising maps [114], growing gases [72], similar approaches of competitive learning [71], or even membrane systems [173] might lead not only to the unveiling of other forms of group structures, but also provide contributions with respect to the scaling issues our framework might seem to face. We also suggest the investigation of the use of social balance theory [103], for instance, in order to infer *in-group* or *out-group* relationships among individuals which interactions are not observed [130].

To conclude, we intuitively suggest the investigation of alternative normalisation approaches for cooperation evaluation, dynamic thresholding for interaction labelling, and other cooperation network update rules. Similarly we suggest the study of different initialisation methods for the cooperation network’s adjacency matrix, e.g. randomised weights, identity, or zero matrix.

10.2.3 Group Identity Detection

The research conducted further strengthens the validity of evolutionary computation for community structure detection. Nevertheless, the results gathered showed that the algorithms and techniques used can even be improved.

Further research on niching activation mechanisms should be conducted, so that the same *aggregative* approach to infer group identities performed by N-EA — which starts by assuming a high number of groups and then gradually refines it towards *true* values, and which we believe its key advantage in performance — can be replicated. For instance, the study of other techniques commonly used in reinforcement learning, such as an exploration policy based on Boltzmann distribution [199], could be investigated. Further work on identifying alternative techniques to the niching repair procedure — which allows for the implementation of *sealed* niches — or even its removal by means of alternative approaches are also due. With respect to the latter, we suggest the investigation of the use of two population EA (one feasible and one infeasible) such as the FI-2pop genetic algorithm [113]. Alternatively, we suggest the combination of the single population and niching approaches into a unique evolutionary framework. For instance, the two methods could independently evolve, yet exchange candidate solutions, similarly to what is done in the island model evolutionary approaches [215].

With respect to the single-population evolutionary community structure detection, the initialisation phase of N-EA could be improved. For instance, it could regulate the inferred number of group distribution of its candidate solutions, so that the evolutionary search could be limited to relevant regions/spread out more uniformly throughout the search space. Although their results were suboptimal, further research on the group and edge representation should be conducted. Different mutation operators for the group representation could be investigated, so that the search strategy would get less structured. Alternatively, we suggest a more prudent approach for the study of the group-based chromosome representation for community structure detection, by initially focusing solely on static sparse undirected and unweighted networks, and subsequently increasing the complexity level of the problem in hand.

Beyond improving the performance of the algorithms, another line of research could be focused on the enrichment of the Group Identity Detection (GID) component. Similarly to Folino and Pizzuti [66], the evolutionary-based GID module could investigate the presence of community structures based on more than one fitness function, e.g. modularity, LinkRank and community score, in order to further reduce the drawbacks of modularity’s resolution limit. Similarly, multi-objective evolutionary computation could allow for the execution of CSD in multiplex networks, for instance, by evolving solutions which aggregate nodes based on their connectivity (cooperation) and properties (cultural backgrounds).

Another study could be focused on the detection of overlapping communities. This line of research might be effective with respect to highlight dynamic evolutions of the groups, for instance, by detecting focal

individuals leaving one group to join another one.

Finally, the investigation of the robustness of the GID component with respect to very large networks, e.g. thousands if not million of nodes, is due, in order to better grasp the scaling limits of the evolutionary computation approach and eventually identify better ones.

10.2.4 Interaction-based Group Modelling Framework as a Whole

Clearly, the Group Modelling (GM) framework needs to be applied to many other interactive domains, beyond the *other-other* game, covered in this dissertation, or the Ultimatum Game, covered in our previous work [91, 83, 89, 87]. Beyond theoretical games, the framework should be applied to computer-based games, which was intended to be its original domain of application.

Initially, we suggest the design and development of cooperative games, and the execution of experiments, similarly in protocol to the one used by Chen and Li [30]. The research could first be aimed to the identification of the induced *true* groups, then focus on the correct representation of group dynamics, and finally use our GM framework to understand the relationship existent between game events and global pattern evolutions. Then, we could extend the application of our GM framework even in presence of *emergent* groups. Moreover, the online modelling capabilities of our framework might allow its embedding within the Experience-Driven Procedural Content Generation framework (EDPCG) [224], for the creation of multiplayer adaptive games and serious multiplayer games [225, 32, 107]. On this regard, the framework would aim to contribute to the broadening of the research on EDPCG — which has been mostly used for single player games or, at most, multiplayer games though with a single player perspective [23] — to multiplayer collaborative games per se.

Other aspects of group behaviours could be investigated, either separately from or integrated within our current group modelling framework. For instance, the detection of hierarchical group structures or subgroups could contribute to identify within-group leadership roles; the augmentation of the cooperation network into e.g. a multiplex network could help with inferring norms, chains of influence of opinions, and so on. Similarly, the application of the GM framework in scenarios of dynamic and *emergent* group structures, could contribute to the identification of *critical masses* in the societies, and correlations with the underlying environment and interaction protocols.

To conclude, we could even imagine to use the group modelling framework as an internal reasoning mechanism of group-based artificial agents [179]. For instance, each agent might autonomously detect the existence of group structures in the artificial society, identify itself as being part of a particular group, and finally interact with other agents in accordance with the temporal group-based metrics of fairness of resource distribution.

10.3 Summary

This Chapter embarked the discussion of the validity of our group modelling framework — in terms of limitations and potential improvement — based on the empirical results gathered in the previous Chapters 6, 8, and 9.

The next Chapter 11 will sum up and conclude the whole dissertation.

Chapter 11

Conclusions

Being able to detect the presence of groups in societies is an important task, as it can bring insight on how the former affect the latter, and also the environment they both live within. Similarly, it could be possible to better understand, for instance, the leading causes of the evolution of cooperative interactions and social conflicts. A fundamental manifestation of the existence of group structures resides of the fact that individuals, when interacting with members of their own group (*in-group*), tend to be more cooperative with each other than when they interact with members of other groups (*out-group*) [39]. This phenomenon is also referred to as *group identity*, to indicate the tendency of individuals to identify themselves as being part of a group, and behave accordingly [39].

In this dissertation we proposed an interaction-based group modelling framework which builds on the hypothesis that, through the measurement of cooperation of the ongoing interactions, it is possible to retrieve *in-group* and *out-group* boundaries of the society as a whole, hence partition its members into (cooperative) group structures, hence assign group identities to each individual. We rely of the definition of cooperation (or altruism) as the degree to which an individual attempts to satisfy others' concerns [205, 32]. The domain of application of our group modelling framework is, a vast plethora of what we called *social synthetic environments* — e.g. computer-mediated environments populated by human-controlled avatars, or agent-based models of artificial societies — in which the members of these societies interact with each other and manifest group identities. For the empirical evaluations embarked in this dissertation, moreover, we relied on theoretical resource allocation games.

The framework is composed of two pipelined modules. The first module, namely Cooperation Modelling (CM), observes the ongoing interactions, evaluate their cooperation values, and subsequently maintain up-to-date estimates of the society's cooperation network. The second module, namely Group Identity Detection (GID), partitions the network into groups, so that the *within*-group cooperation is maximised, and the *between*-group cooperation is minimised.

The cooperation evaluation task relies on the so-called *one-to-many* approach and the *temporal group-based* metrics of fairness [86], which collectively aim to bring the attention on how differently *one* individual interacts and treats *many* others — even if the interactions do not occur at the same time — in order to assign quantitative measures of cooperation to the interactions. In this dissertation we focused on two approaches for cooperation evaluation: (1) transformation of interactions into probability of *in-groupness*, (2) discretisation of the interactions into the two *in-group* and *out-group* labels. Moreover, during the empirical investigations of the CM module, we also considered the possibility that no cooperation evaluation would occur.

The cooperation network update task uses the evaluated interactions to maintain up-to-date estimations of the society's related complex network of cooperation. In this network, each node corresponds to a member of the society, and each edge corresponds to the ongoing level of cooperation from one individual to another. Hence, the update task modifies the edge weights of the network. In this dissertation we focused on three cooperation network update rules: (1) sample average, (2) constant- α , extensively used in e.g. reinforcement learning [199], ant colony optimisation [42], and artificial neural networks [95], and (3) temporal difference's

TD0, which constitutes one of the most fundamental technique used in reinforcement learning [199].

The partitioning of the network into groups, finally, is achieved by performing the *community structure detection* task extensively studied in network theory [14]. Moreover, we performed community structure detection via *modularity maximisation* [160], which is possibly the most consolidated approach for this task. The main emphasis was given, in this dissertation, on the use of evolutionary computation for community structure detection. More specifically, we considered three different chromosome representations, namely *node*, *group*, and *edge*, two evolutionary algorithm approaches, i.e. based on a single population and niching, four possible mechanisms for niching activation, two deterministic and two stochastic, and finally two possible fitness functions.

In a nutshell, node representation-based evolutionary algorithms search for candidate solutions by evolving each single node’s group identity independently, the group representation-based algorithms search for candidate solutions by evolving whole group structures, whilst the edge representation-based algorithms search for candidate solutions by evolving subnetworks of the given one. The niching method we proposed relies on the compartmentalisation of the search space, so that each niche will maintain a population of chromosomes which lead to network partitioning into a pre-defined number of groups. The niching activation mechanisms are in charge of selectively evolve a subset of the niches based on their current most fit candidate solutions. Finally, the two fitness functions considered were the most generic modularity function, which can be applied to both directed/undirected and/or weighted/unweighted networks [88], and the LinkRank measure, specifically designed for directed networks [112].

We conducted extensive empirical investigations, with benchmark social synthetic environments based on the social dilemma experiment originally conducted by Chen and Li [30], in which they aimed to investigate social preferences in of human participants with induced groups. We relied on their authentic recorded behavioural data to both build a simulation of their interactions, and derive artificial societies of *believable* agents. In all scenarios considered the societies were static, that is, the group identities were induced at the beginning of the experiment and did not change throughout its execution. In total, our experiment were based on 270 artificial societies and 25 authentic behavioural simulations, which were varying in terms of number of individuals, number of groups, and group size distributions.

Through the experimental setups based on artificial societies, and the related gathered results, we observed that a group modelling framework that:

- aggregates current and recent interactions through the *one-to-many* approach;
- transforms the interactions into discretised *in-group* and *out-group* labels;
- updates the society’s cooperation network via an absolutely myopic constant- α update rule ($\alpha = 1$), which only takes into account the most recent interactions;
- uses a single population-based evolutionary algorithm with node-based chromosome representation and modularity as fitness function

would lead to the best performances, in terms of matches between inferred groups and *true* ones, across the vast majority of society types investigated, though it manifested a generally slow convergence to maximum inference values, especially for a low number of groups present in the society. Nevertheless, other configurations of the group modelling framework — such as those based on niching combined with a deterministic niching activation mechanism — provided localised performance improvements in terms of speed of convergence to high performance values, though at a cost of registering an overall suboptimal inference performance.

The best configuration of the group modelling framework was then applied to the authentic behavioural data of the experiment by Chen and Li. The promising results gathered provide even further evidence of our framework efficacy to detect *true* group structures in human-based social synthetic environments. This constitutes as an initial *validation* of our approach for its application to more complex social synthetic environments, e.g. multiplayer computer games.

11.1 Summary of Contributions

The following is the list of main contributions this dissertation gave to the fields of evolution of cooperation, user modelling, network theory and evolutionary computation.

One-to-many approach

- we applied the *one-to-many* approach, for the application of the *temporal group-based* metrics of fairness, to a new interaction protocol, that is the *other-other* game, and overall gathered promising results. Prior to this dissertation, the *one-to-many* approach combined with the group-based fairness metrics was already used for the group modelling task in presence of the ultimatum game as interaction protocol;
- we formalised how the temporal group-based measure of fairness of treatment can be applied to an interaction protocol. This puts the basis of its application to other interaction protocol types;

Evolution of Cooperation

- the temporal group-based metric has been provided efficacy for the investigation of behaviours in social dilemmas, similarly to e.g. inequity aversion [61, 21];
- the group modelling framework provides an indirect approach to investigate behaviours of socially driven individuals. In other words, it is not focused on the definition of a unique behavioural function — e.g. inequity aversion — rather on the overall societal perspective of altruism or cooperation;

Behavioural and Experience Modelling

- to the best of our knowledge, there has not been any other attempt to adopt an indirect, interaction-based approach to group modelling;
- the modelling framework has been successfully used in human-based experiments. Prior to this work, group modelling had been focused solely on artificial societies of *believable* agents. This puts the basis of its application to other, more complex, human-based social synthetic environments;
- our approach allows for the extension of the single-individual perspective to the perspective of the whole society, by building a complex network of ongoing cooperation values;

Evolutionary Computation

- to the author's best knowledge, this dissertation was the first attempt to examine the performance of different evolutionary algorithms for community structure detection, with a high emphasis on different chromosome representations;
- we applied evolutionary computation for community structure detection in dynamic networks which gradually change their typology, from directed and weighted to undirected and unweighted;
- we have further investigated and improved a niching approach for community structure detection;
- to the best of our knowledge, we have applied, for the first time, the group-based chromosome representation [57, 58, 55, 56] for community structure detection;

Network Theory

- the role of evolutionary computation with respect to the community structure detection task has been strengthened, especially for what it regards directed and weighted networks;

- the results gathered evidenced that modularity can effectively work even in dynamic networks which change their typology — from fully connected, directed and weighted, up to sparse undirected and unweighted;
- we have further validated the generalisability of modularity — as opposed to e.g. LinkRank — but only within its resolution limit [122];
- we evidenced that the interpretation modularity gives to the notion of communities fits well with the notion of induced social groups;
- the results highlighted that the established $n = 128$ $m = 4$ equal size benchmark network is a rather simple scenario, in accordance with the ongoing claims against it [124, 167];
- related to the previous claim, we observed that equally sized groups, although arguably *unrealistic*, constitute a more difficult problem than societies with groups following a power law distribution. This contribution is very important especially when synthetic networks are considered as benchmark problems for community structure detection;
- we observed that the normalised mutual information — that is a performance measure extensively used in community structure detection studies — appears to be excessively dependent on number of communities present in the partitions to be evaluated. On the other hand, we highlighted that the use of the Hungarian algorithm [117] — or its normalised matching score (*NHS*) — could be used as a more severe performance measure for community structure detection studies. Moreover, *NHS* is arguably a rather intuitive measure, meaning that it is possible to retrieve the number of mismatched nodes, given its score, in a straightforward way.

Chapter 12

Chen and Li’s Experiment Simulation

This appendix extends Chapter 4 by reporting the behavioural data, of the authentic experiment by Chen and Li [30], which was extracted for the Chen and Li’s Simulation (CL), and which was subsequently used to implement the related artificial society (CLAS). The behavioural data refers to the other-other game only.

From the original empirical data we extracted 25 experimental runs across five treatment sessions, namely *original*, *random within*, *random between others*, *random between same*, and *no chat*, see Table 4.1. The 25 tables specify their treatment name and “date” field, so that it can be easily retrieved in the authentic data. Columns “ID” and “group ID” represent the participant’s ID (a_i) and *true* group identity (g_i). The 15 remaining columns report the allocations the participants made towards the first receiver R_1 , based on the three interaction types *in-in*, *out-out*, and *in-out*, and on five possible endowment amounts e , from 200 up to 400 tokens. The offers made to receiver R_2 can easily be retrieved by subtracting, from e , the reported offer. The offer reported for the *in-out* interaction type refers to the *in-group* receiver R_1 . As it can be seen, Tables 12.1 and 12.4 report 15 and 14 participants respectively. This is due to the fact that, in the authentic experiment, three participants took part into the experiment more than once, hence their behavioural data were discarded by Chen and Li.

Table 12.1: original 050127R4

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
2	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
3	1	100	125	200	150	200	100	150	150	300	200	0	125	25	350	15
4	1	100	125	150	175	200	100	125	150	175	200	200	250	150	175	200
5	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
6	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
7	1	100	125	150	175	200	100	125	150	175	200	150	200	200	300	300
8	1	100	125	150	200	250	75	200	150	300	200	175	225	300	300	325
9	2	100	125	150	175	200	100	125	150	175	200	120	145	170	195	220
10	1	100	150	150	200	200	100	100	150	150	200	100	150	150	200	200
11	1	100	125	150	175	200	100	125	150	175	200	125	150	175	200	225
12	1	100	125	150	200	200	100	125	150	200	200	100	125	150	150	200
13	1	100	150	150	250	200	100	100	150	175	200	150	200	150	175	150
14	1	100	125	150	175	200	100	125	150	175	200	125	175	200	230	275
15	1	100	125	150	170	200	80	100	200	130	280	160	220	270	300	320

Table 12.2: original 050202PR

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	25	175	300	150	300	100	1	99	300	300	175	10	10	250	200
2	1	100	125	150	175	200	100	125	150	175	200	120	150	200	200	250
3	1	100	150	150	175	200	100	100	100	150	250	100	125	200	175	150
4	2	100	125	150	175	200	100	125	150	175	200	125	150	175	200	250
5	2	100	125	150	180	200	100	100	175	200	220	200	250	280	330	368
6	2	100	125	150	175	200	100	125	150	175	200	100	150	200	200	200
7	1	100	125	150	170	200	100	125	150	180	200	150	200	250	250	300
8	1	100	125	150	175	200	100	125	150	175	200	125	150	175	215	250
9	1	100	125	150	175	200	0	0	0	0	200	200	250	300	350	400
10	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
11	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
12	1	100	150	230	210	200	48	125	90	345	80	120	200	50	250	160
13	1	100	150	150	200	200	50	151	109	177	159	200	250	300	350	400
14	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
15	1	100	100	100	175	200	0	249	300	1	350	150	200	299	0	0
16	2	50	150	275	55	200	150	50	295	50	20	150	225	16	175	380

Table 12.3: original 050203QK

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	2	100	140	170	300	200	100	0	3	175	400	145	240	290	200	333
2	2	100	125	150	175	200	110	100	170	100	250	199	249	299	349	399
3	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
4	1	100	125	150	175	200	150	150	250	250	350	100	125	150	175	200
5	1	10	200	150	100	150	10	200	150	100	150	10	200	150	100	150
6	1	100	200	150	250	250	50	150	200	300	300	75	175	160	350	225
7	1	100	125	150	175	200	100	125	150	175	200	101	126	151	176	201
8	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
9	1	100	125	150	175	200	100	125	150	175	200	190	240	290	340	390
10	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
11	2	100	125	150	175	200	100	125	150	175	200	199	249	299	349	399
12	1	110	125	150	300	200	100	1	100	90	260	150	100	250	250	130
13	1	100	125	150	350	200	200	100	150	0	150	0	250	300	350	350
14	1	100	145	150	100	260	75	160	200	175	300	198	214	100	349	200
15	2	100	125	150	175	200	100	125	150	175	200	200	250	300	300	350
16	1	100	125	150	175	200	75	125	150	175	200	125	175	200	250	250

Table 12.4: original 050207OP

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
2	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
3	1	100	125	150	175	200	100	125	150	175	200	150	175	200	200	300
4	2	110	125	150	175	200	1	50	0	250	200	200	248	290	350	400
5	1	100	125	150	175	200	100	125	150	175	200	199	200	200	275	300
6	1	100	125	150	175	200	50	75	50	175	200	150	200	250	175	200
7	1	100	125	150	175	200	100	50	50	175	200	100	249	299	349	399
8	2	50	50	50	50	50	0	250	0	350	0	200	250	300	350	400
9	1	100	130	150	175	200	100	115	125	175	200	100	150	200	185	205
11	2	100	100	150	150	200	100	150	100	200	250	100	125	200	175	150
12	1	100	125	150	175	200	20	50	50	175	200	199	249	100	300	350
13	2	200	250	300	350	400	100	125	150	175	200	200	250	300	350	400
14	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
15	2	100	150	150	175	200	100	125	150	175	200	150	200	250	300	300
16	1	100	125	150	175	200	50	100	175	200	175	150	200	225	250	260

Table 12.5: original 050209Q5

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	175	250	100	175	300	50	0	25	0	200	75	125	300	0	200
2	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
3	2	150	180	215	250	250	80	100	20	200	59	110	169	30	201	220
4	1	100	110	200	200	300	50	200	150	300	250	110	100	100	150	200
5	2	100	125	150	175	200	100	125	150	175	200	175	225	250	300	325
6	2	100	125	150	175	200	100	125	150	175	200	105	130	160	180	220
7	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
8	2	100	125	150	175	200	100	100	100	100	200	150	225	275	349	375
9	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
10	2	100	100	150	175	150	75	125	150	170	200	125	150	250	280	300
11	1	100	125	150	300	200	100	125	150	50	200	100	125	150	200	200
12	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
13	2	100	125	150	175	200	100	125	150	175	200	125	150	175	200	225
14	1	100	125	150	175	200	100	125	150	175	200	100	150	150	200	200
15	2	100	100	150	225	200	50	100	100	175	200	200	200	200	200	200
16	2	100	125	150	175	200	100	125	150	175	200	150	150	200	300	350

Table 12.6: original 050210PP

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
2	1	50	160	150	300	150	70	120	300	200	230	90	150	100	350	40
3	2	125	150	200	250	200	100	125	150	225	200	150	175	150	175	200
4	1	200	250	300	350	400	100	150	150	150	200	0	0	0	0	0
5	2	100	125	150	175	200	50	125	150	200	150	150	200	250	300	350
6	1	100	125	150	175	200	100	125	150	175	200	100	150	150	200	200
7	1	100	125	150	175	200	100	125	150	175	200	85	175	150	150	250
8	1	50	48	176	199	200	123	150	299	222	366	200	250	300	350	400
9	1	100	125	150	175	200	100	125	150	175	200	100	120	150	175	200
10	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
11	2	100	125	150	175	200	100	125	150	175	200	100	150	200	200	250
12	2	100	125	150	175	200	25	200	200	175	250	150	200	175	175	300
13	1	100	125	150	175	300	100	125	150	175	200	100	125	150	175	100
14	1	100	125	150	175	200	150	225	150	175	200	200	200	150	175	200
15	2	150	200	200	300	300	80	180	100	250	200	180	220	260	340	340
16	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200

Table 12.7: original 050211LN

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	2	100	125	150	175	200	123	125	150	175	200	200	250	300	350	400
2	1	100	125	150	175	200	100	125	150	175	200	101	127	153	179	205
3	1	125	175	150	175	150	63	123	200	150	200	200	69	160	175	0
4	1	150	100	150	100	175	75	125	0	300	100	200	0	150	175	250
5	1	100	125	150	175	200	100	125	150	175	200	150	200	250	300	350
6	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
7	2	200	250	300	350	400	100	125	150	175	200	200	250	300	350	400
8	1	100	125	150	225	150	43	150	117	167	200	200	200	250	300	300
9	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
10	1	100	125	150	175	200	100	125	150	175	200	199	249	299	349	350
11	2	100	125	150	175	200	80	200	120	220	250	150	150	175	200	300
12	2	100	125	150	175	200	50	100	200	200	300	200	200	250	300	350
13	2	100	125	150	175	200	100	125	150	175	200	110	150	160	200	225
14	2	0	0	300	0	0	200	250	0	350	400	100	125	150	175	200
15	1	100	125	150	175	200	100	125	150	175	200	150	200	250	300	300
16	1	120	125	150	175	200	100	100	200	150	250	190	225	225	300	380

Table 12.8: original 050216PO

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	100	125	150	175	200	150	250	300	350	350
2	1	100	125	150	150	200	100	125	150	200	200	100	200	100	175	200
3	2	100	125	150	175	200	100	125	150	175	200	150	200	200	250	300
4	2	100	150	140	200	200	100	125	150	175	150	150	200	200	275	350
5	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
6	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
7	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
8	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
9	1	100	125	150	175	200	100	125	150	175	200	150	150	175	200	250
10	2	100	125	150	175	200	50	200	299	325	50	175	200	250	250	300
11	2	100	100	150	175	250	125	200	150	175	200	200	250	250	350	400
12	1	54	150	157	150	250	111	175	22	250	140	99	125	111	176	202
13	2	100	125	150	175	200	50	125	150	175	200	150	200	200	275	300
14	1	100	100	150	175	400	200	50	275	0	200	0	0	50	300	400
15	1	100	125	150	200	200	100	125	150	200	200	200	250	300	350	400
16	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400

Table 12.9: original 050217PL

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	2	100	125	150	175	200	100	125	150	175	200	175	225	275	325	375
2	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
3	2	100	125	150	200	200	120	200	130	150	210	90	150	200	300	100
4	2	100	125	150	150	200	100	130	100	200	100	100	150	200	250	400
5	1	50	100	150	250	200	100	125	200	175	200	100	150	100	150	250
6	2	100	100	150	175	200	100	200	100	170	201	200	250	300	350	400
7	2	100	125	150	175	199	80	150	125	170	175	130	150	180	200	250
8	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
9	2	100	125	150	175	200	100	125	150	175	200	100	150	200	200	200
10	1	100	125	150	175	200	100	125	150	175	200	150	200	250	300	350
11	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
12	1	100	150	150	200	200	100	100	150	150	200	150	200	250	300	300
13	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
14	1	100	125	150	200	200	100	125	150	150	200	190	250	150	300	350
15	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
16	1	100	125	150	175	200	150	125	200	175	200	150	200	250	300	350

Table 12.10: original 050219PM

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	100	125	150	175	200	125	150	175	200	250
2	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
3	1	100	125	150	175	200	100	125	150	175	200	125	150	300	350	400
4	1	100	120	150	300	200	100	120	200	200	200	160	200	120	250	200
5	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
6	2	100	125	150	175	200	100	125	150	175	200	125	150	175	200	200
7	1	100	125	150	175	200	100	125	150	175	200	100	100	150	200	200
8	1	100	125	150	175	200	150	100	200	175	200	199	249	299	349	399
9	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
10	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
11	2	100	125	150	175	200	100	125	150	175	200	105	130	160	190	250
12	2	79	175	150	197	209	54	76	185	184	190	12	190	280	340	394
13	1	100	100	150	175	200	100	100	150	175	200	50	50	50	50	50
14	2	100	125	150	175	200	100	125	150	175	200	150	175	200	250	300
15	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
16	2	100	125	150	175	200	100	125	150	175	200	120	150	200	200	250

Table 12.11: original 050720LQ

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
2	1	100	125	150	175	200	100	125	150	175	200	150	175	225	300	350
3	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
4	1	100	125	150	175	200	100	125	150	175	200	150	175	200	275	300
5	2	100	125	150	175	200	100	125	150	175	200	100	150	175	200	225
6	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
7	1	100	125	150	175	200	101	123	147	170	198	105	121	149	172	199
8	1	100	125	150	175	200	100	125	150	175	200	200	200	250	350	350
9	2	100	125	150	175	200	100	125	150	175	200	150	175	200	225	266
10	1	100	125	150	300	300	150	125	150	50	200	198	249	1	300	399
11	1	120	120	200	180	170	90	90	170	200	200	130	150	175	220	250
12	2	100	100	150	200	201	100	150	200	225	202	100	150	101	175	205
13	1	100	125	150	175	200	100	125	150	175	200	200	125	150	175	200
14	1	115	100	135	300	310	55	85	185	95	65	70	90	15	30	60
15	2	100	120	150	250	200	100	90	200	200	210	199	150	190	210	220
16	2	100	125	150	175	200	100	125	150	175	200	110	130	160	180	210

Table 12.12: original 050720NP

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
2	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
3	1	100	125	200	175	200	100	0	300	150	200	200	250	300	350	400
4	1	100	125	150	175	200	100	125	150	175	200	50	50	100	100	100
5	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
6	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
7	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
8	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
9	1	100	125	150	175	200	100	125	150	175	200	150	175	250	250	300
10	1	100	125	150	150	200	100	125	150	150	200	100	125	150	150	150
11	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
12	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
13	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
14	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
15	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
16	2	100	125	150	175	200	199	249	299	349	399	199	249	299	349	399

Table 12.13: original 050722M3

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	2	100	125	150	200	200	100	125	150	150	200	155	200	280	300	399
2	2	100	125	150	175	200	100	125	150	175	200	150	150	200	250	300
3	1	100	175	150	175	200	100	125	150	175	200	150	125	150	175	200
4	1	100	125	150	175	200	150	170	100	175	200	175	199	299	349	399
5	1	100	150	150	235	175	95	125	45	175	300	175	175	300	325	345
6	1	100	125	150	175	200	100	125	150	175	200	100	150	150	175	200
7	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
8	1	100	125	150	175	200	100	125	150	175	200	150	175	200	225	100
9	1	100	125	150	175	200	100	125	150	175	200	150	175	225	250	275
10	1	100	125	150	350	200	0	0	0	0	400	200	250	300	350	400
11	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
12	2	100	125	150	175	200	100	125	150	175	200	125	125	150	175	200
13	1	50	150	150	175	200	20	200	10	175	200	100	160	50	175	200
14	1	100	125	150	175	200	90	150	1	200	300	120	200	200	300	350
15	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
16	1	50	100	150	200	200	150	200	50	100	200	200	150	250	250	100

Table 12.14: original 050729LO

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
2	1	100	100	100	150	200	100	100	100	100	200	150	200	200	250	300
3	2	100	125	150	175	200	200	250	300	350	400	200	250	300	350	400
4	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
5	2	100	125	150	175	200	100	125	150	175	200	100	150	175	200	225
6	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
7	2	100	100	150	175	200	50	125	200	175	200	200	125	200	200	200
8	1	100	125	150	0	200	100	125	150	300	200	100	125	150	0	200
9	2	200	150	200	50	100	150	50	150	20	350	40	200	50	100	50
10	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
11	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
12	2	100	125	150	175	200	50	125	150	175	200	150	175	150	125	200
13	1	100	125	150	175	200	100	125	150	175	200	100	150	200	200	250
14	2	100	125	150	175	200	100	125	150	175	200	150	150	200	200	250
15	2	100	125	150	175	200	100	125	150	175	200	120	200	200	200	300
16	1	100	125	150	175	200	100	125	150	175	200	150	200	250	300	350

Table 12.15: original 050729NK

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	20	100	200	200	250	150	200	200	300	300
2	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
3	1	100	125	150	175	200	100	125	150	175	200	200	225	300	350	400
4	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
5	2	100	175	150	175	200	150	100	100	175	200	100	150	225	175	200
6	2	100	125	150	175	200	100	125	150	175	200	150	150	200	250	320
7	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
8	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
9	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
10	1	200	250	300	350	400	0	0	0	0	0	100	125	150	175	200
11	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
12	1	100	125	150	175	200	100	50	100	175	200	150	225	200	300	350
13	2	100	150	150	200	200	100	100	150	200	200	100	150	150	200	200
14	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
15	1	0	125	150	200	100	200	125	200	150	200	0	0	0	0	0
16	2	100	125	150	175	200	100	125	150	175	200	150	150	200	250	250

Table 12.16: random within 070522K7

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	100	125	150	175	200	199	250	300	300	400
2	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
3	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
4	2	100	125	150	175	200	100	125	150	175	200	150	200	200	300	300
5	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
6	2	50	100	150	200	200	150	100	150	150	200	100	150	150	150	200
7	1	100	125	150	175	200	100	125	200	200	200	180	250	300	350	400
8	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
9	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
10	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
11	1	100	100	200	200	200	100	250	100	150	200	100	150	150	175	200
12	2	100	125	100	350	250	100	100	200	125	200	175	200	300	300	375
13	1	100	125	150	175	200	200	125	150	300	300	0	0	0	0	0
14	2	100	125	150	175	200	100	125	150	175	200	180	225	250	250	300
15	1	100	150	150	175	200	100	125	200	175	200	100	125	175	200	175
16	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200

Table 12.17: random within 070522MQ

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	0	0	0	300	400	200	250	300	350	400
2	2	100	125	150	175	200	100	125	150	175	200	150	150	200	200	300
3	1	100	125	150	175	200	100	125	0	350	200	200	250	300	350	400
4	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
5	1	100	125	150	175	200	150	125	150	175	200	150	200	200	300	250
6	2	100	125	150	175	200	200	0	150	175	200	200	250	300	350	400
7	1	195	245	295	300	359	0	125	175	196	75	200	250	200	41	200
8	2	100	125	300	175	200	100	125	12	175	200	100	250	300	175	400
9	1	100	125	150	175	200	100	125	150	175	200	200	250	300	325	200
10	2	100	125	150	175	200	20	125	150	175	200	180	200	250	300	300
11	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
12	2	100	125	150	175	200	100	125	150	175	200	180	225	275	325	375
13	1	0	50	100	150	200	0	125	300	200	200	200	250	200	300	400
14	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
15	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
16	2	100	125	150	175	200	100	125	150	175	200	150	200	250	300	350

Table 12.18: random btw other 070525MJ

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	100	125	150	175	200	150	200	250	300	300
2	2	100	125	150	175	200	100	125	150	175	200	200	200	225	275	325
3	1	100	125	150	175	200	100	125	150	175	200	200	200	200	225	250
4	2	80	135	160	190	230	100	125	150	175	200	130	150	180	200	250
5	1	100	125	150	175	200	125	150	200	200	300	150	200	270	300	350
6	2	100	125	150	175	200	100	125	150	175	200	150	200	250	275	300
7	1	100	125	150	175	200	100	125	150	175	200	150	150	200	200	200
8	2	100	125	200	175	250	100	125	150	175	200	125	200	250	275	300
9	1	100	125	150	175	200	100	125	150	175	200	100	249	175	175	200
10	2	116	135	170	190	215	112	130	180	180	220	75	100	140	160	185
11	1	100	125	150	175	200	100	125	100	200	200	200	250	300	350	400
12	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
13	1	100	125	150	175	200	100	125	150	175	200	200	250	300	300	350
14	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
15	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	225
16	2	100	200	13	175	0	20	140	30	20	10	10	1	300	200	200

Table 12.19: random btw other 070526MN

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
2	2	100	125	150	175	200	100	125	150	175	200	125	175	200	275	330
3	1	100	125	150	175	200	1	125	150	175	200	100	249	299	349	399
4	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
5	1	100	125	150	175	200	100	250	150	175	200	100	175	150	175	200
6	2	100	125	150	175	200	100	125	150	175	200	200	125	150	175	200
7	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
8	2	195	220	150	340	200	100	125	150	175	200	190	220	295	350	399
9	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
10	2	100	125	150	175	200	100	125	150	175	200	125	150	200	200	300
11	1	100	125	150	175	200	100	125	150	175	200	150	200	200	250	250
12	2	100	125	150	175	200	100	125	150	175	200	150	150	175	210	225
13	1	100	150	150	250	200	100	125	150	50	200	200	125	300	350	200
14	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
15	1	100	200	150	175	200	200	10	200	325	356	200	250	125	349	395
16	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200

Table 12.20: random between same 070523P7

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	0	175	100	250	200	200	250	300	350	350
2	2	100	125	150	175	200	100	125	150	175	200	150	150	200	200	250
3	1	100	120	150	175	200	100	130	150	175	200	100	120	150	175	200
4	2	100	125	150	175	200	50	175	150	175	200	150	175	200	250	250
5	1	100	125	150	175	200	100	125	150	175	200	200	0	300	300	250
6	2	100	125	150	175	200	150	100	150	175	200	100	125	300	350	400
7	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
8	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
9	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
10	2	100	125	150	175	200	100	125	150	175	200	150	200	275	175	200
11	1	100	125	150	300	200	100	125	150	50	200	200	250	300	275	200
12	2	100	200	150	250	200	1	2	292	300	200	200	250	300	350	400
13	1	100	125	150	175	200	100	125	150	175	200	100	150	150	175	200
14	2	100	150	150	150	200	50	50	150	200	200	200	250	300	350	350
15	1	100	125	150	175	200	100	125	150	175	200	150	175	200	250	300
16	2	100	125	150	175	200	100	125	150	175	200	175	150	175	250	300

Table 12.21: random between same 070524MM

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	100	125	150	175	200	150	175	200	225	300
2	2	100	150	150	200	200	100	100	150	150	200	100	150	150	200	200
3	1	100	150	200	250	250	100	200	50	300	350	150	200	275	325	350
4	2	100	150	150	175	200	100	100	175	200	225	150	180	200	225	300
5	1	100	125	150	175	200	150	175	225	250	350	150	175	225	250	350
6	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
7	1	100	125	150	175	200	100	125	150	175	200	150	200	200	300	300
8	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
9	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
10	2	100	250	0	350	400	2	0	150	175	200	200	250	300	350	400
11	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
12	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
13	1	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
14	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
15	1	100	125	150	175	200	150	200	100	200	250	199	240	295	345	390
16	2	100	100	150	150	200	0	0	0	0	0	150	175	200	250	300

Table 12.22: nochat 070518JH

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	100	125	150	175	200	175	200	250	300	350
2	1	100	200	150	175	200	100	125	100	200	200	100	125	150	175	200
3	2	100	100	150	200	200	100	125	50	150	200	150	200	250	300	350
4	2	100	125	150	175	200	100	125	150	175	200	200	250	300	200	300
5	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
6	1	100	125	150	175	200	100	125	150	175	200	150	175	200	200	250
7	1	100	125	150	175	400	100	125	150	175	200	200	250	300	350	400
8	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
9	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
10	1	100	200	150	300	200	0	50	150	175	200	200	250	300	350	400
11	1	100	125	150	175	200	100	125	150	175	200	130	170	200	230	270
12	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
13	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
14	2	100	125	150	175	200	0	125	150	175	200	200	250	300	350	400
15	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
16	2	100	125	150	175	200	100	125	150	175	200	150	200	225	300	300

Table 12.23: nochat 070518NS

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	200	149	200	200	75	160	5	300	400	175	105	300	325	222
2	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
3	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
4	2	100	125	150	175	200	100	125	150	175	200	150	100	180	190	220
5	1	75	150	150	200	200	100	175	150	150	200	65	100	175	150	150
6	1	100	200	150	200	200	100	200	150	150	200	100	200	150	200	200
7	1	100	125	150	175	200	100	125	100	175	100	100	250	200	350	150
8	2	100	125	150	175	200	100	125	150	175	200	125	150	200	200	200
9	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
10	1	150	200	250	300	300	150	200	250	300	300	150	200	250	300	300
11	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
12	1	100	250	150	50	200	100	150	150	175	200	175	250	300	300	200
13	2	100	125	150	110	225	150	187	145	100	50	200	247	200	250	400
14	2	100	125	150	175	200	100	125	150	175	200	100	125	175	200	250
15	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
16	1	100	250	300	200	200	199	250	300	300	290	199	250	290	50	300

Table 12.24: nochat 070613MJ

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	100	125	150	175	200	5	245	300	345	0	195	245	295	345	395
2	2	100	125	200	175	200	100	125	150	200	100	200	250	300	350	400
3	2	55	125	225	200	300	65	135	300	225	200	100	120	150	175	250
4	2	100	125	120	100	0	100	125	150	175	200	120	150	180	300	300
5	2	100	125	150	175	200	100	125	150	175	200	200	250	300	350	400
6	1	100	125	150	175	200	100	125	150	175	200	150	150	175	200	225
7	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
8	2	100	125	150	175	200	100	125	150	175	200	110	135	160	185	210
9	2	100	125	150	175	200	100	125	150	175	200	100	150	200	200	200
10	1	101	126	151	200	201	199	249	299	349	370	20	40	150	175	100
11	2	100	100	150	150	200	100	100	150	200	200	199	249	299	349	399
12	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
13	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
14	1	100	125	150	175	200	100	125	150	175	200	150	200	250	300	350
15	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
16	1	50	125	150	175	200	100	125	150	300	200	50	125	150	300	220

Table 12.25: nochat 070614MM

ID	group ID	<i>in-in</i>					<i>out-out</i>					<i>in-out</i>				
		200	250	300	350	400	200	250	300	350	400	200	250	300	350	400
1	1	200	200	150	350	0	100	0	0	125	200	200	250	300	350	400
2	1	100	125	100	175	250	100	125	150	175	200	200	249	300	175	200
3	1	100	150	150	200	200	50	125	300	0	200	200	250	300	350	400
4	1	100	125	100	50	200	100	125	200	300	100	100	1	299	50	100
5	2	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
6	1	100	125	150	175	200	100	125	150	175	200	50	75	75	85	100
7	2	100	125	150	175	200	100	100	200	200	150	150	200	250	300	350
8	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
9	2	150	150	200	275	350	50	100	100	75	50	150	225	250	275	325
10	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
11	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
12	2	100	150	100	140	200	75	125	100	300	150	125	125	175	150	250
13	1	90	100	200	50	225	90	100	200	100	250	125	150	50	200	125
14	1	100	125	150	175	200	100	125	150	175	200	100	125	150	175	200
15	2	100	125	150	175	200	100	125	150	175	200	125	150	150	200	200
16	1	120	45	150	300	300	94	180	100	10	200	150	200	50	275	100

Chapter 13

Cooperation Modelling

This appendix integrates the topics covered in Chapters 5 and 6.

13.1 One-to-many Approach and the Other-other Game

This Section extends Section 5.1 by applying *TGB* (5.1) [86], under the *in/out-group* dichotomy, to a single occurrence of the other-other game (see Section 4), that is $t = |D| = 1$.

TGB requires normalised distribution values; the straightforward normalisation $\hat{o}_1 = o_1/e$ and $\hat{o}_2 = o_2/e$ would imply $\hat{o}_1 + \hat{o}_2 = 1$. Under the *in/out-group* dichotomy we can apply *TGB* only in presence of *in-out* interactions. Let us hence assume $a_1, a_p \in A$ and $a_2 \in B$. *TGB* is then reduced to the following equation:

$$TGB = 1 - |\hat{o}_1 - \hat{o}_2| = 1 - |2\hat{o}_1 - 1|$$

TGB is maximised when $\hat{o}_1 = 1/2$ and minimised when either $\hat{o}_1 = 0$ or $\hat{o}_1 = 1$. a_P is assumed to be *in-group* with a_1 , hence we would expect to observe $\hat{o}_1 > \hat{o}_2$. Therefore, in order to aim for the detection of *in-group* and *out-group* boundaries, we would have to focus on receiver individuals for which their $\hat{o} > \frac{1}{2}$.

13.2 Cooperation Evaluation and the Other-other Game

This Section extends Section 5.2 by applying Equation 5.2 to a single execution of the other-other game (see Section 4), that is $t = |D| = 1$ and $O_P = \{o_1, o_2\}$.

The min-max normalisation can be applied only when $o_1 \neq o_2$, otherwise we would obtain $\max(O_P) = \min(O_P)$, hence zero in the denominator of the Equation. When $o_1 \neq o_2$, two are the possible outcomes: either $\hat{o}_1 = 1$ and $\hat{o}_2 = 0$ or vice versa.

In other words, when $t = |D| = 1$, we obtain that Cooperation Evaluation can be applied only in presence of unfair behaviours, which are indications of *in-group* and *out-group* boundaries. Moreover, and possibly more importantly, we obtain that Cooperation Evaluation would necessarily “filter out” the processing of absolute fair behaviours, which are those we would intuitively expect to observe in case of *in-in* interactions (case 3 of Table 5.1, see Section 5.1), and, possibly, even some of the *out-out* interactions (that is cases 1 and 2 of Table 5.1).

On the other hand, with respect to the considerations made in Section 5.2, such a small length for t , which lead to a small O_P , might have an effect on the estimation of the cooperation values. A thorough analysis of the impact that the length of t has on the cooperation modelling task is provided in Chapter 6.

13.3 Cooperation Update via alpha Constant- α

Equation 5.9 is obtained from Equation 5.7 as follows:

$$\begin{aligned}
 \mathcal{C}_{i,j} &\leftarrow \frac{u_{i,j} \mathcal{C}_{i,j} + v_{i,j}}{u_{i,j} + 1} = \\
 &= \frac{u_{i,j} \mathcal{C}_{i,j} + v_{i,j} + \mathcal{C}_{i,j} - \mathcal{C}_{i,j}}{u_{i,j} + 1} = \\
 &= \frac{(u_{i,j} \mathcal{C}_{i,j} + \mathcal{C}_{i,j}) + (v_{i,j} - \mathcal{C}_{i,j})}{u_{i,j} + 1} = \\
 &= \frac{(u_{i,j} + 1)\mathcal{C}_{i,j} + (v_{i,j} - \mathcal{C}_{i,j})}{u_{i,j} + 1} = \\
 &= \mathcal{C}_{i,j} + \frac{1}{u_{i,j} + 1} (v_{i,j} - \mathcal{C}_{i,j})
 \end{aligned}$$

Equation 5.10 can be rewritten as follows:

$$\begin{aligned}
 \mathcal{C}_{i,j} &\leftarrow \mathcal{C}_{i,j} + \alpha (v_{i,j} - \mathcal{C}_{i,j}) = \\
 &= \mathcal{C}_{i,j} + \alpha v_{i,j} - \alpha \mathcal{C}_{i,j} = \\
 &= (1 - \alpha)\mathcal{C}_{i,j} + \alpha v_{i,j}
 \end{aligned}$$

13.4 Constant- α incremental analysis

This Section extends Section 6.4 by explaining why, for $0 < \alpha \leq 1$, and normalised v_i values, the constant- α cooperation network update rule (5.10) will maintain normalised $\mathcal{C}_{i,j}$ values. For the sake of this discussion we will omit the i and j indices.

Let us start from the assumption $\alpha > 0$ (for $\alpha = 0$ we would observe no updates, and for $\alpha < 0$ we would just mirror the \mathcal{C} values to negative values). The increment Δ of the update rule is

$$\Delta = \alpha v - \alpha \mathcal{C}$$

$\Delta = 0$ when $v = \mathcal{C}$, $\Delta < 0$ for $v < \mathcal{C}$, and finally $\Delta > 0$ for $v > \mathcal{C}$.

Let us assume that the initial \mathcal{C} values range between $[0, 1]$. The limit for Δ when $\mathcal{C} \rightarrow 0^+$ is the following:

$$\lim_{\mathcal{C} \rightarrow 0^+} \Delta = \alpha v$$

At the boundaries for v , we will obtain $\Delta_{v=0} = 0$ and $\Delta_{v=1} = \alpha > 0$, hence constant α update rule will never lead to negative \mathcal{C} values.

The limit for Δ when $\mathcal{C} \rightarrow 1^-$ is the following:

$$\lim_{\mathcal{C} \rightarrow 1^-} \Delta = \alpha v - \alpha$$

At the boundaries for v , we will obtain $\Delta_{v=0} = \alpha$ and $\Delta_{v=1} = 0$. This means that if $\alpha \leq 1$ the \mathcal{C} values will never go beyond 1. Therefore, for $0 < \alpha \leq 1$, $v \in [0, 1]$, the \mathcal{C} values will be normalised.

13.5 Temporal Difference incremental analysis

This Section extends Section 6.4 by explaining why, for $0 < \alpha \leq 1$ and normalised v_i values, the Temporal Difference update rule 5.11, for $\gamma > 0$, would not allow for maintaining normalised $\mathcal{C}_{i,j}$ values. It also performs an analysis of the sign of the incremental values in order to highlight under which conditions Temporal Difference will perform wrong updates. In order not to confuse the reader we will omit the i and j indices and use \mathcal{C} to represent $\mathcal{C}_{i,j}$ and \mathcal{D} to represent $\mathcal{C}_{j,i}$.

13.5.1 Temporal Difference Does Not Maintain Normalised \mathcal{C} values

The incremental value for \mathcal{C} is the following:

$$\Delta = \alpha [v - \gamma\mathcal{D} - \mathcal{C}]$$

Let us assume that both \mathcal{C} and \mathcal{D} are initialised with values in the $[0, 1]$ interval. The limit for Δ when $\mathcal{C} \rightarrow 0^+$ is the following:

$$\lim_{\mathcal{C} \rightarrow 0^+} \Delta = \alpha v + \alpha\gamma\mathcal{D}$$

Δ will always be ≥ 0 , since $\alpha, v, \gamma > 0$, and \mathcal{D} is initialised to a value > 0 . Hence, the lower boundary for all the \mathcal{C} values when Temporal Difference is used is zero.

The limit for Δ when $\mathcal{C} \rightarrow 1^-$ is the following:

$$\lim_{\mathcal{C} \rightarrow 1^-} \Delta = \alpha [v + \gamma\mathcal{D} - 1]$$

moreover, when $\mathcal{D} \rightarrow 0^+$ we obtain

$$\lim_{\mathcal{C} \rightarrow 1^-, \mathcal{D} \rightarrow 0^+} \Delta = \alpha [v - 1]$$

whilst, when $\mathcal{D} \rightarrow 1^-$, we obtain

$$\lim_{\mathcal{C} \rightarrow 1^-, \mathcal{D} \rightarrow 1^-} \Delta = \alpha v$$

In case $\mathcal{C} \rightarrow 1^-$ and, at the same time, $\mathcal{D} \rightarrow 0^+$, we observe that $\Delta \leq 0$, since $v \in [0, 1]$, which means that Temporal Difference tends to adjust \mathcal{C} towards \mathcal{D} , in case $v = 0$, or leave it unaltered, hence considering \mathcal{D} a wrong estimate, in case $v = 1$.

Similar is the case for which $\mathcal{C}, \mathcal{D} \rightarrow 1^-$. For $v = 0$, that is an interaction which might suggest that the estimate \mathcal{D} is wrong, Temporal Difference will not change \mathcal{C} ; on the other hand, if the update rule observes $v = 1$, we will obtain $\Delta > 0$, which means that, eventually, the \mathcal{C} value will go beyond one. Hence, Temporal Difference would not allow to have \mathcal{C} values bounded to the right.

13.5.2 Analysis of the Sign of Temporal Difference Incremental

Let us first take a step back and consider $\gamma = 0$, that is, constant- α update rule. Its update increment is

$$\Delta^{c-\alpha} = \alpha [v - \mathcal{C}]$$

When $v = 0$ we obtain

$$\Delta_{v=0}^{c-\alpha} = -\alpha\mathcal{C}$$

which is less or equal than zero for all $\mathcal{C} \in [0, 1]$. When $v = 1$, instead, we obtain

$$\Delta_{v=1}^{c-\alpha} = \alpha - \alpha\mathcal{C}$$

which is greater or equal than zero for all $\mathcal{C} \in [0, 1]$. The signs of $\Delta^{c-\alpha}$ are intuitive: when we observe $v = 0$ we expect to reduce the value of \mathcal{C} , whilst we expect to increase it in the opposite case $v = 1$.

When we analyse Temporal Difference, on the other hand, we observe the following: for $v = 0$ we obtain

$$\Delta_{v=0} = \alpha [\gamma\mathcal{D} - \mathcal{C}]$$

and for $v = 1$ we obtain

$$\Delta_{v=1} = \alpha [1 + \gamma\mathcal{D} - \mathcal{C}]$$

Since $\alpha > 0$, we obtain that $\Delta_{v=0} > 0$ for $\mathcal{D} > 0$ and $\gamma > \mathcal{C}/\mathcal{D}$. In case $v = 1$, instead, we obtain that $\Delta_{v=1} < 0$ for $\mathcal{D} > 0$ and $\gamma < (\mathcal{C} - 1)/\mathcal{D}$. Although γ is a fixed parameter, \mathcal{C} and \mathcal{D} are not. This means that in certain conditions, even if cooperation modelling observes $v = 0$, the update of \mathcal{C} will be positive, that is it will go towards the *in-group* estimation; similarly, for $v = 1$, for some conditions, the update of \mathcal{C} will be negative, that is it will go towards the *out-group* estimation.

Chapter 14

Group Identity Detection Empirical Evaluation — Secondary Results

This appendix integrates Chapter 8 by providing all the secondary results to the empirical evaluation of the Group Identity Detection (GID) module based on the Chen and Li-derived Artificial Societies (CLASs) scenarios.

14.1 Analysis of Niching Based on the Exact Knowledge of m

The results here presented extend Section 8.4 by investigating the benefits that our constrained niching approach might bring to the evolutionary community detection task. More specifically, we compare the single population, group and edge based evolutionary algorithms (G-EA and E-EA respectively) with a single-niche EA which assumes perfect knowledge about the number of *true* groups m of the societies considered (G-1N and E-1N respectively).

Figures 14.1, . . . , 14.7 show promising results by means of the niching approach, in terms of inference results, except for the $m = 4$ case.

The results here presented extend Section 8.4.2 by investigating the combination of our proposed niching method combined with niching activation mechanisms, when the group end edge chromosome representations are used.

More specifically, we consider the UCB1 and ϵ -greedy algorithm used in n -armed bandit problems (G-UCB1, G- ϵ GREEDY, E-UCB1, and E- ϵ GREEDY respectively for the group and edge representations), the reading head algorithm (G-RHEAD and E-RHEAD respectively) and the activation mechanism based on roulette wheel selection (G-RWS and E-RWS respectively).

Similarly to the findings related to the node representation, overall, the group representation's G-RHEAD and G-UCB1 appear to be the best niching activation mechanisms. Moreover, these outperform G-EA for all the society types considered — with the exception for the $m = 4$ case — and with the reading head mechanism being the best performing approach to niche activation. However, the two algorithms have a nearly constant trend in terms of inferred number of groups, across all the society type considered — once again except the $m = 4$ case — and with G-RHEAD manifesting a slightly less constant trend.

For what it concerns the edge representation, instead, E-UCB1 appears to be somewhat the best performing approach, overall across all scenarios. Once again, its trend in terms of inferred number of groups is nearly constant independently on the society type considered. Remarkably, the reading head algorithm (E-RHEAD) manifests the trend, in terms of number of inferred groups, which most replicates the aggregative nature of N-EA, that is, it generally starts assuming a high number of groups, and then it reduces it, though we observe a high initial drop, unseen in N-EA.

However, both G-UCB1, E-UCB1, G-RHEAD and E-RHEAD register a very high standard deviation, suggesting thus that their evolutionary performance and search trends are highly dependent on the network

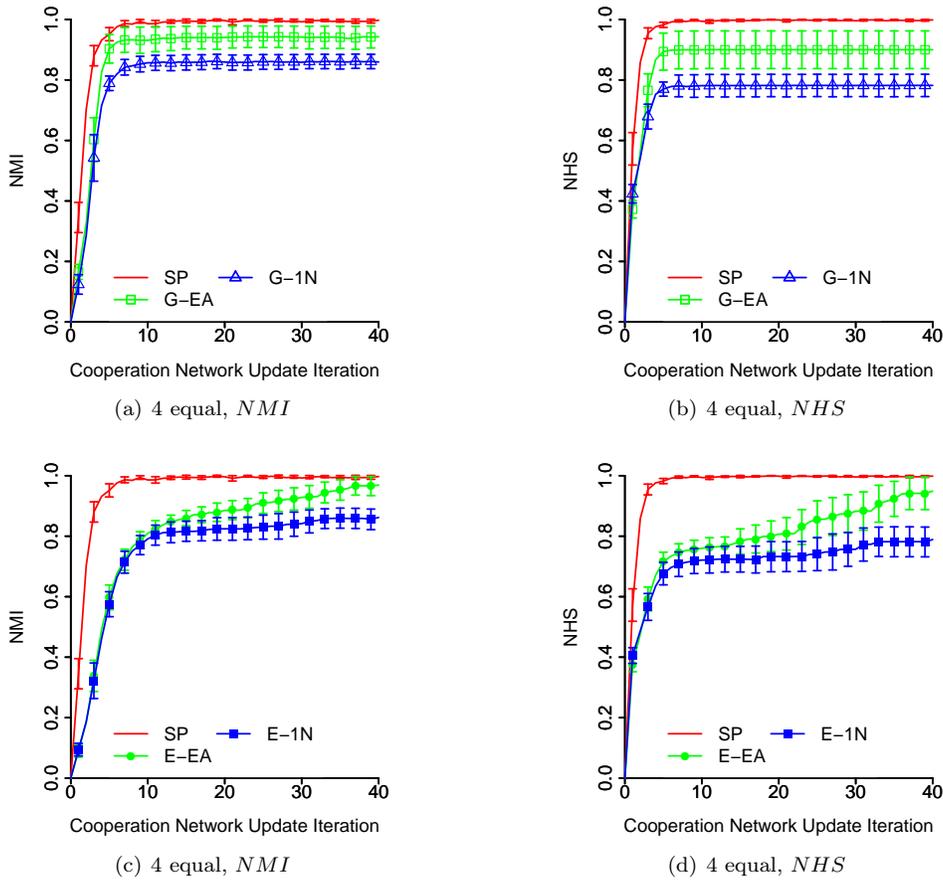


Figure 14.1: Performance (NMI , NHS) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 4$ equal size group types.

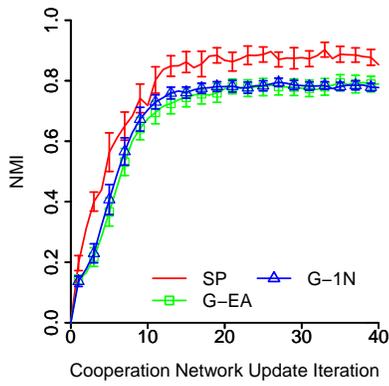
topology under investigation.

14.2 Analysis of the Importance of the Fitness Function: Comparison with LinkRank

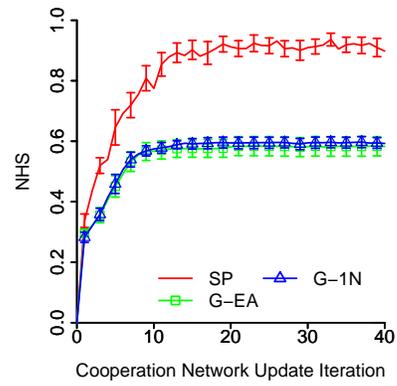
The results here presented extend Section 8.5 by investigating whether an alternative fitness function, specifically LinkRank, can provide performance advantages with respect the community structure detection task.

In all these plots we observe that the use of LinkRank does not outperform modularity.

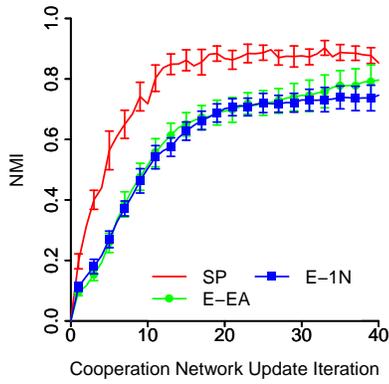
14.2.1 Node/Edge/Group-based UCB1 algorithms



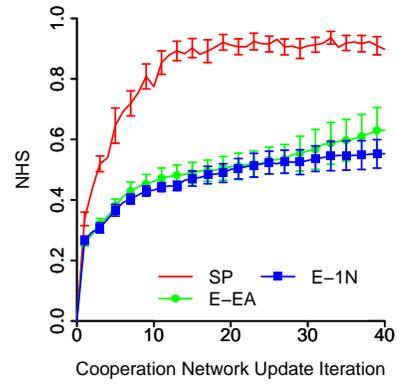
(a) 7 equal, *NMI*



(b) 7 equal, *NHS*

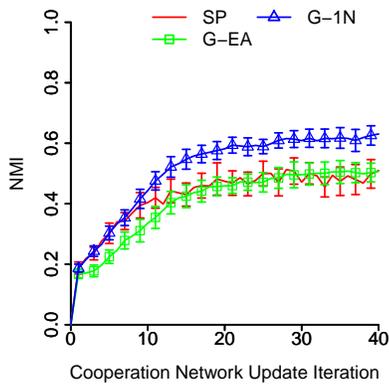


(c) 7 equal, *NMI*

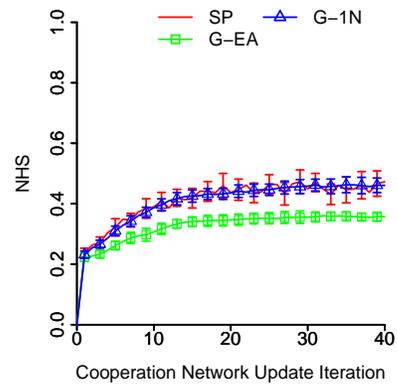


(d) 7 equal, *NHS*

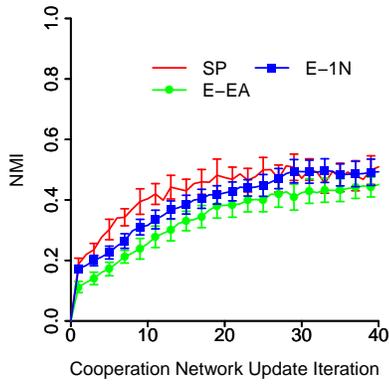
Figure 14.2: Performance (*NMI*, *NHS*) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 7$ equal size group types.



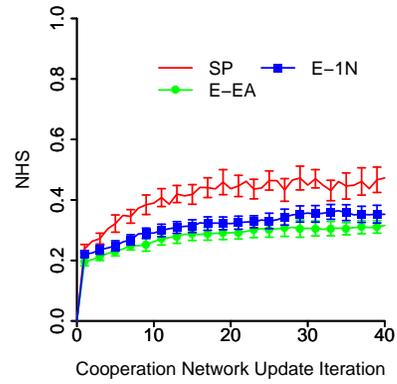
(a) 11 equal, NMI



(b) 11 equal, NHS

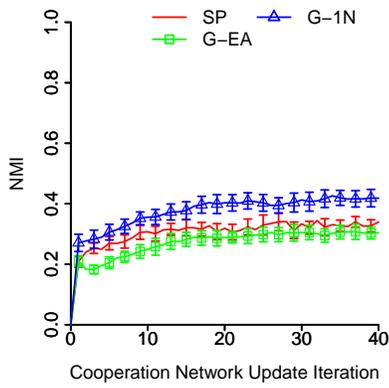


(c) 11 equal, NMI

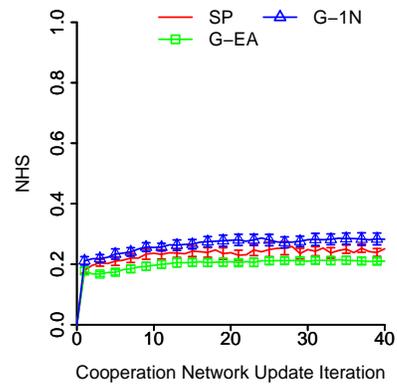


(d) 11 equal, NHS

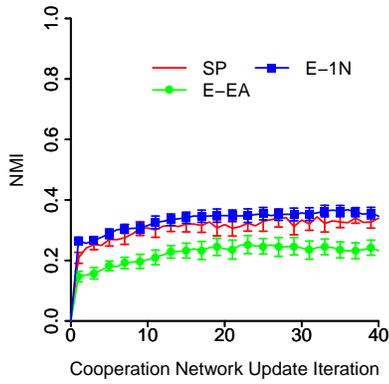
Figure 14.3: Performance (NMI , NHS) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 11$ equal size group types.



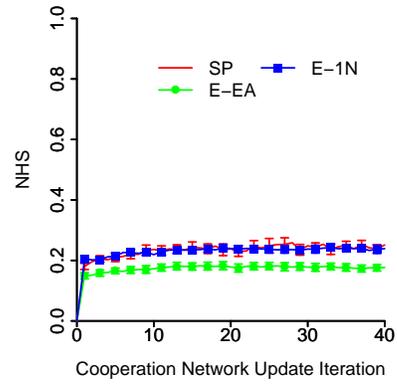
(a) 17 equal, NMI



(b) 17 equal, NHS

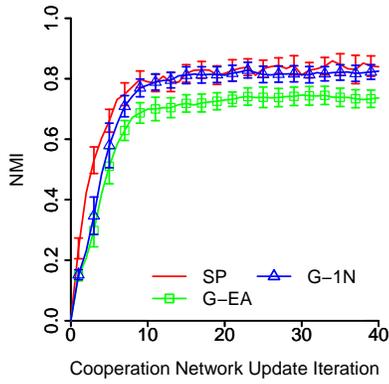


(c) 17 equal, NMI

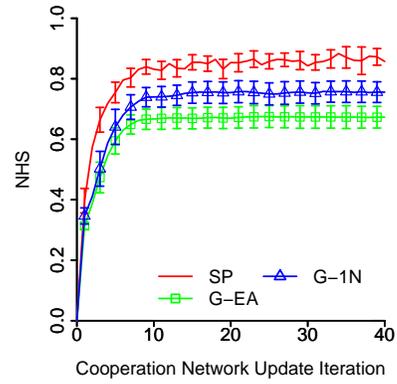


(d) 17 equal, NHS

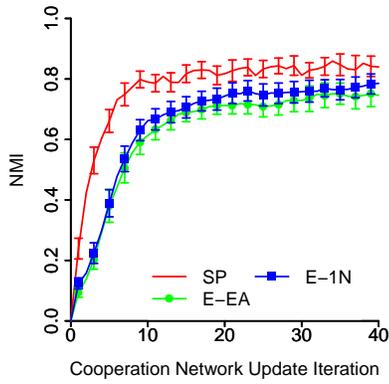
Figure 14.4: Performance (NMI , NHS) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 17$ equal size group types.



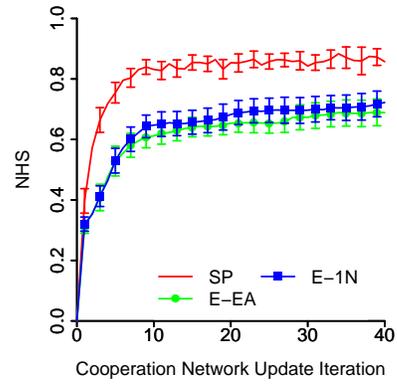
(a) 7 power law, *NMI*



(b) 7 power law, *NHS*

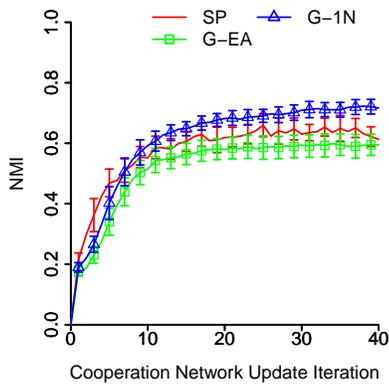


(c) 7 power law, *NMI*

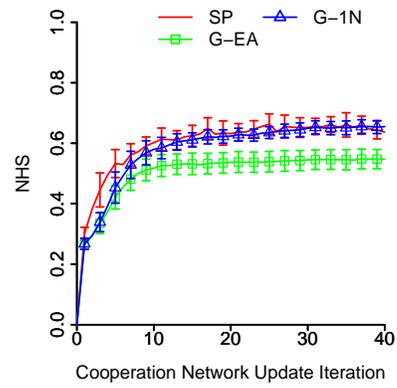


(d) 7 power law, *NHS*

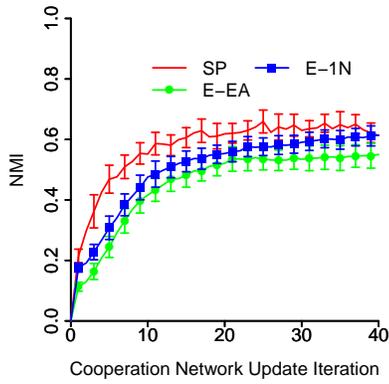
Figure 14.5: Performance (*NMI*, *NHS*) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 7$ power law group size types.



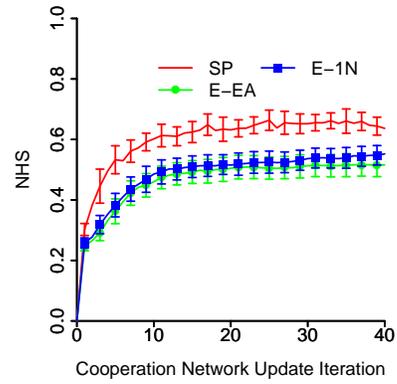
(a) 11 power law, *NMI*



(b) 11 power law, *NHS*

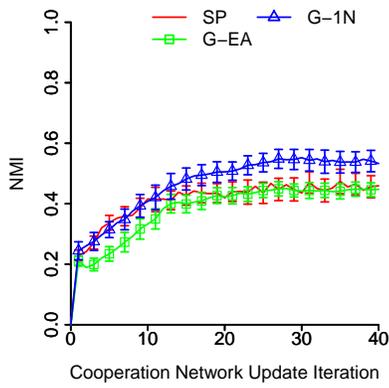


(c) 11 power law, *NMI*

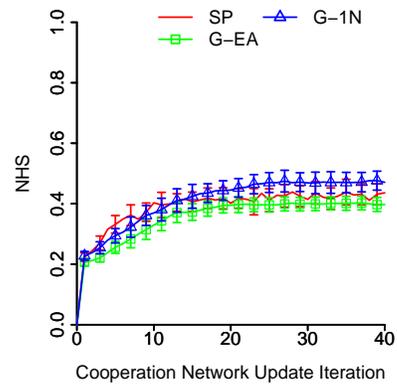


(d) 11 power law, *NHS*

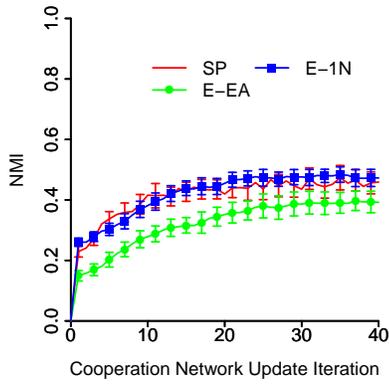
Figure 14.6: Performance (*NMI*, *NHS*) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 11$ power law group size types.



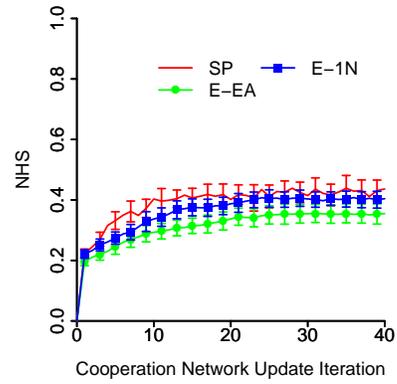
(a) 17 power law, *NMI*



(b) 17 power law, *NHS*



(c) 17 power law, *NMI*



(d) 17 power law, *NHS*

Figure 14.7: Performance (*NMI*, *NHS*) of G-EA and E-EA, as opposed to its single niche with chromosomes representing network partitions into exactly four groups, and SP, for $m = 17$ power law group size types.

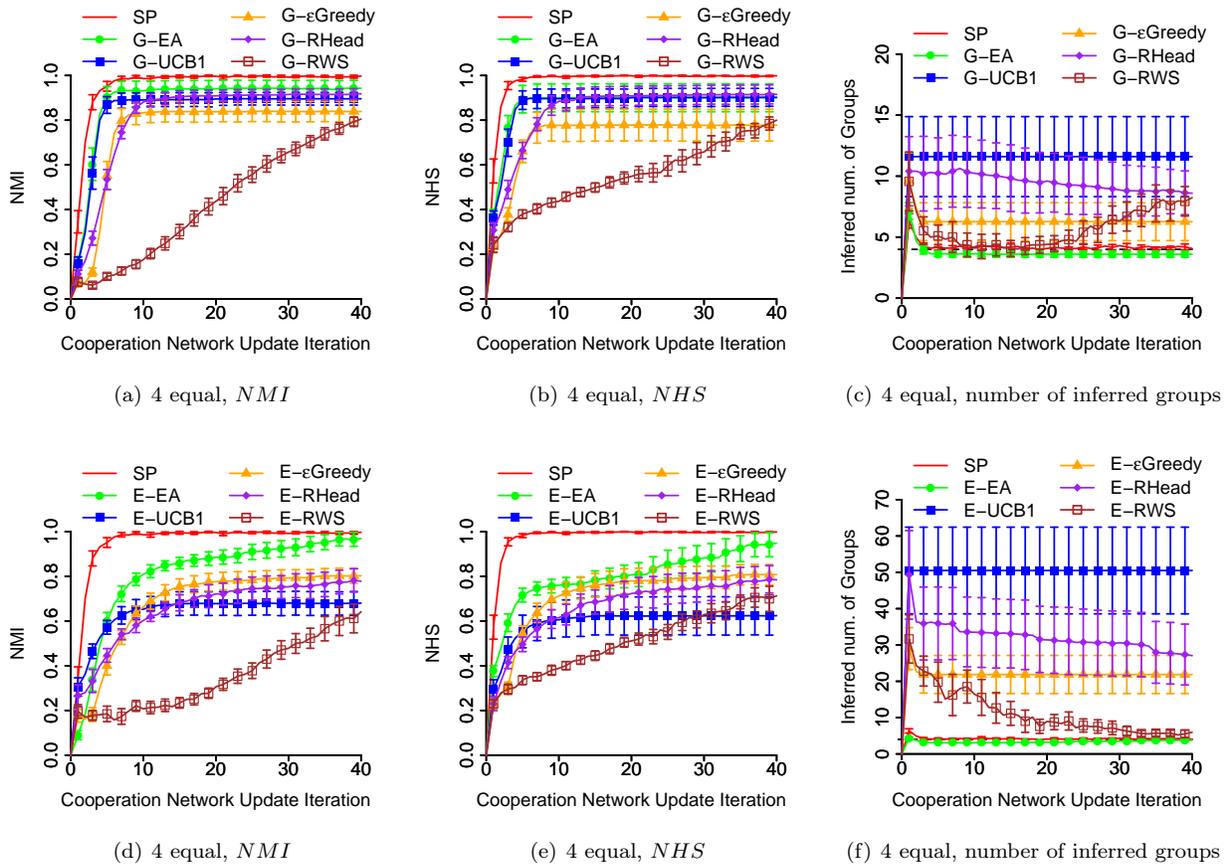


Figure 14.8: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 4$ equal size group types.

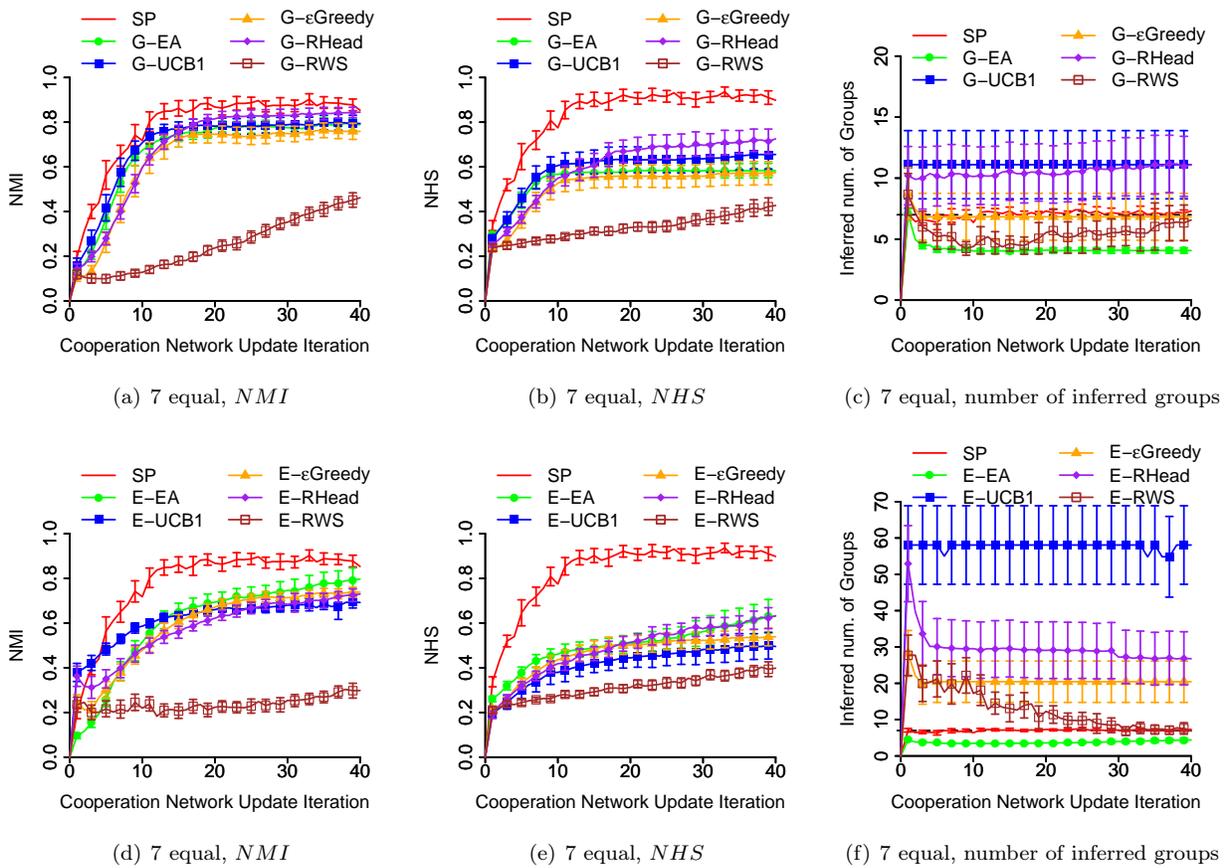


Figure 14.9: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 7$ equal size group types.

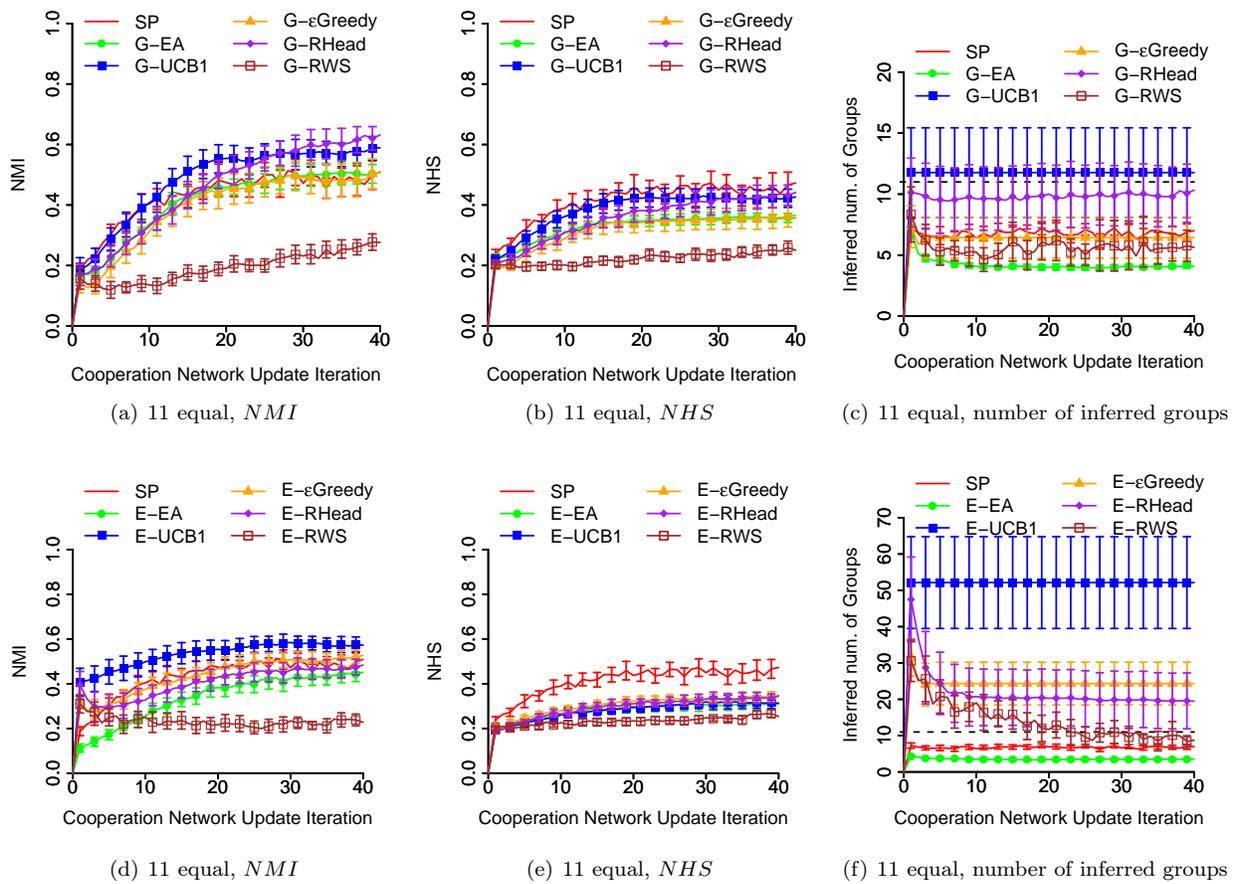


Figure 14.10: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 11$ equal size group types.

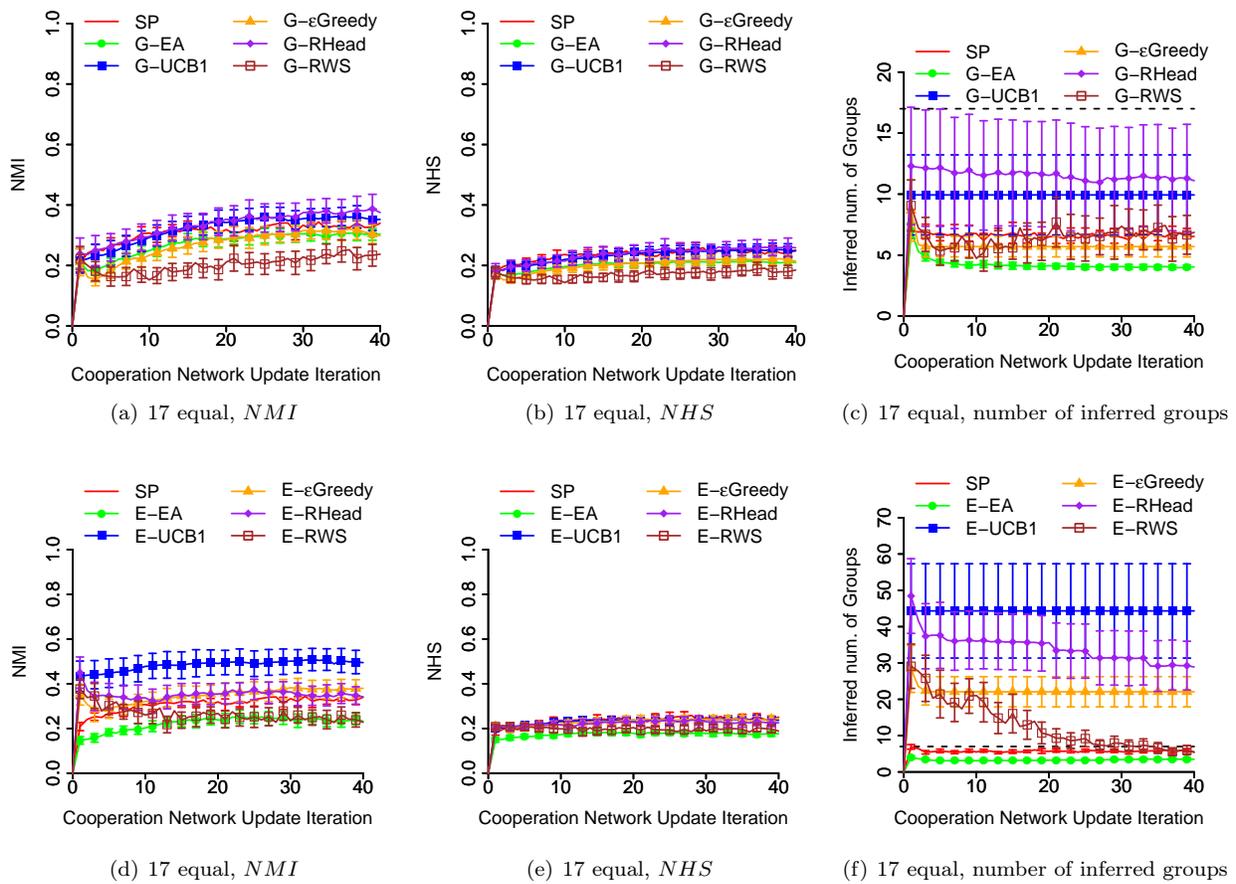
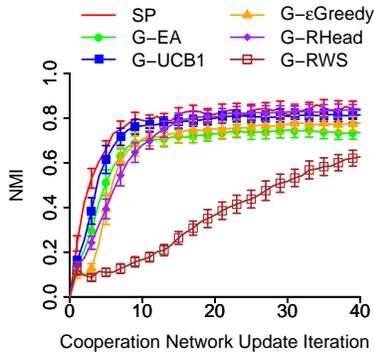
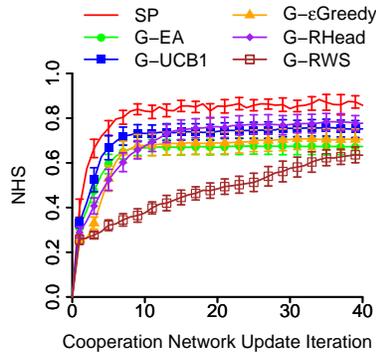


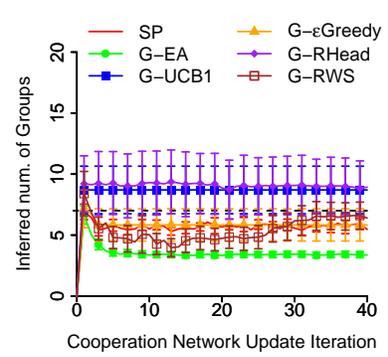
Figure 14.11: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 17$ equal size group types.



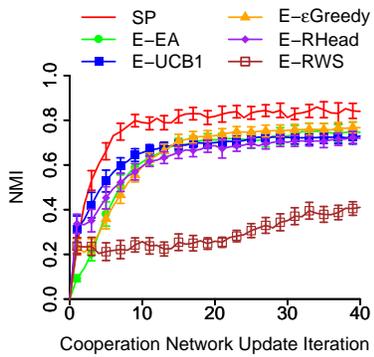
(a) 7 power law, *NMI*



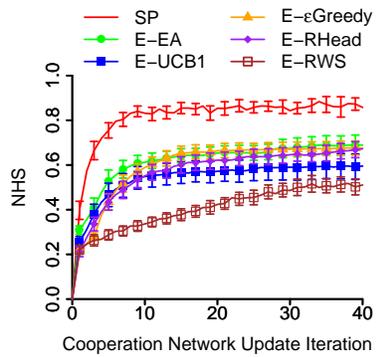
(b) 7 power law, *NHS*



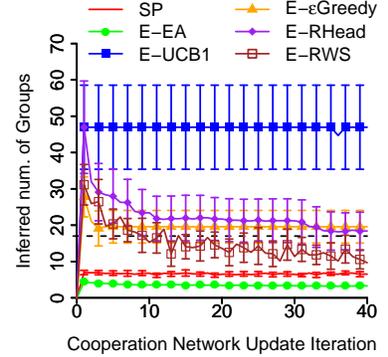
(c) 7 power law, number of inferred groups



(d) 7 power law, *NMI*



(e) 7 power law, *NHS*



(f) 7 power law, number of inferred groups

Figure 14.12: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 7$ power law group size distribution types.

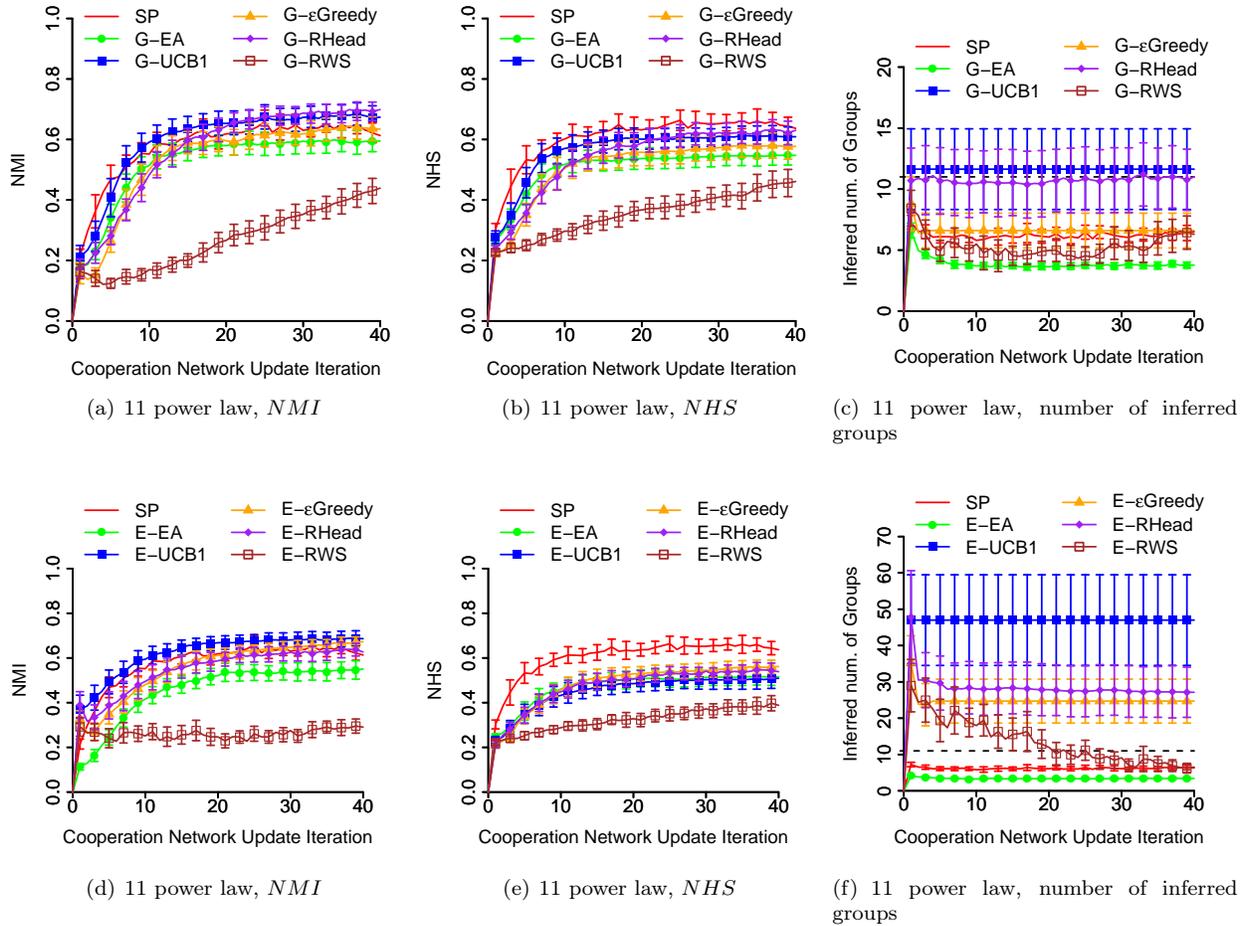
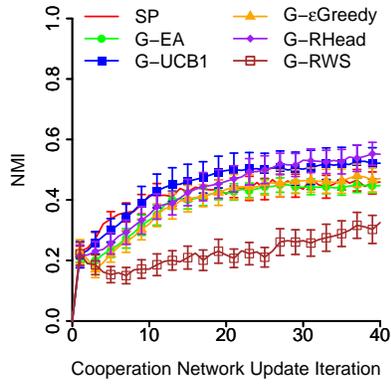
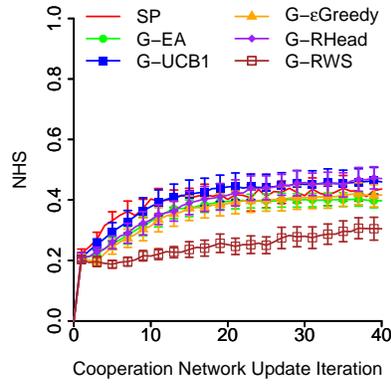


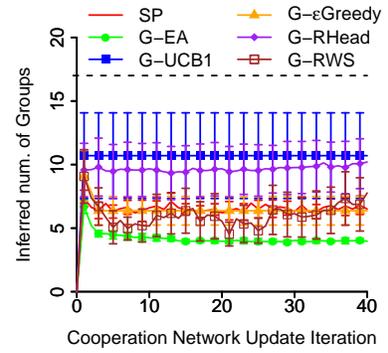
Figure 14.13: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 11$ power law group size distribution types.



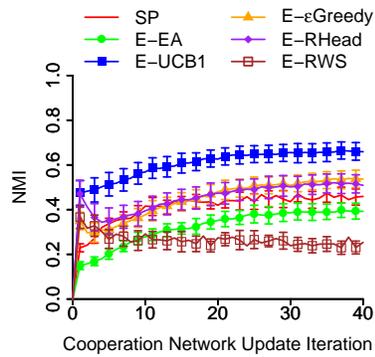
(a) 17 power law, *NMI*



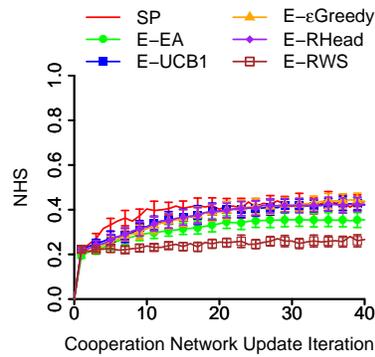
(b) 17 power law, *NHS*



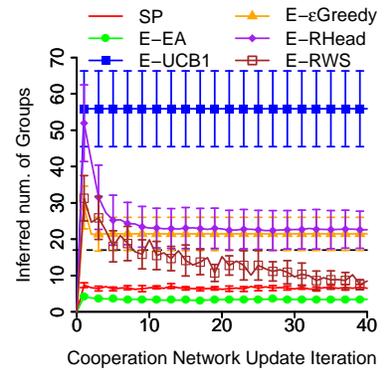
(c) 17 power law, number of inferred groups



(d) 17 power law, *NMI*



(e) 17 power law, *NHS*



(f) 17 power law, number of inferred groups

Figure 14.14: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of the four group/edge-based niching algorithms compared to G/E-EA, for $m = 17$ power law group size distribution types.

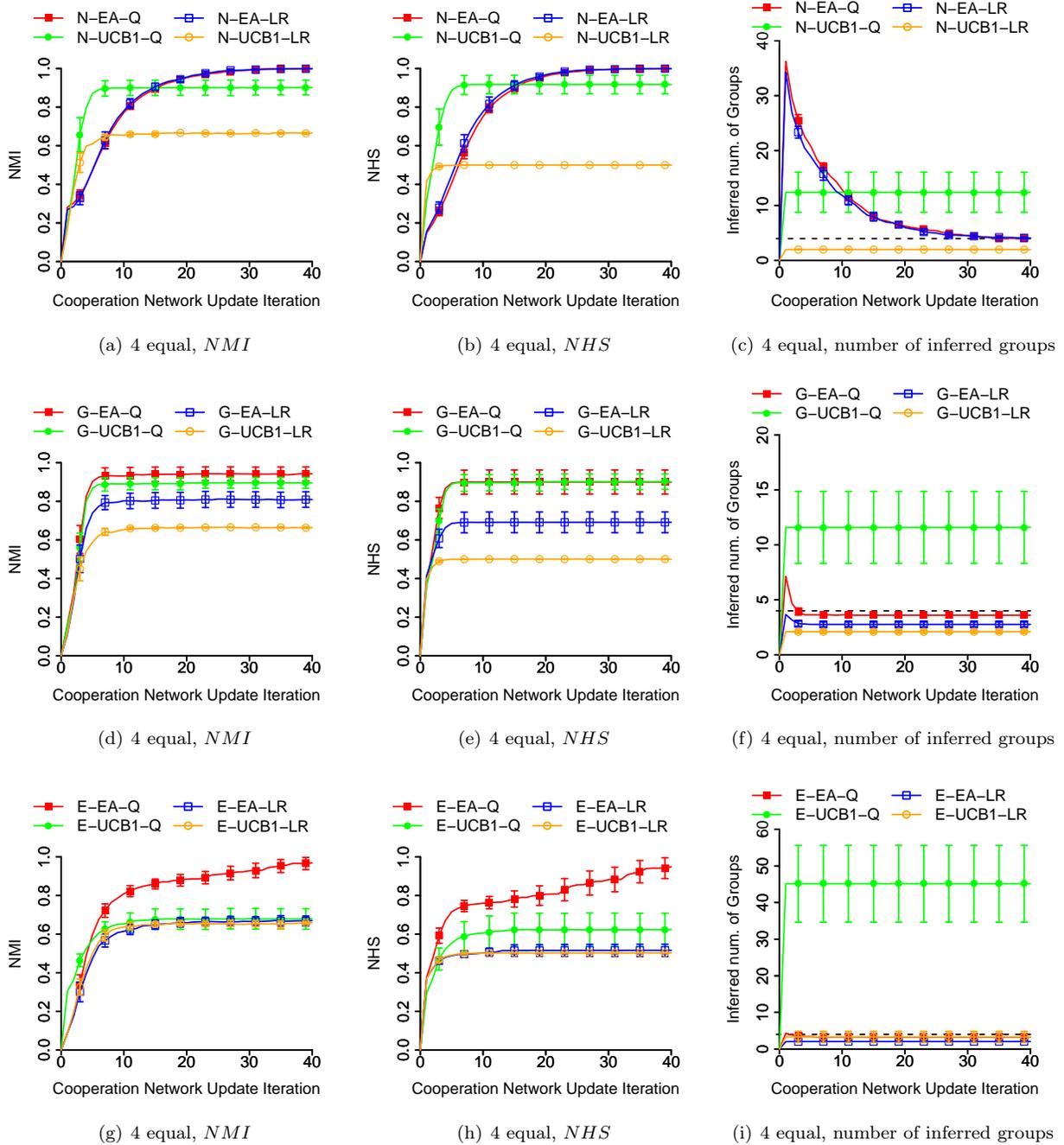


Figure 14.15: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 4$ equal size.

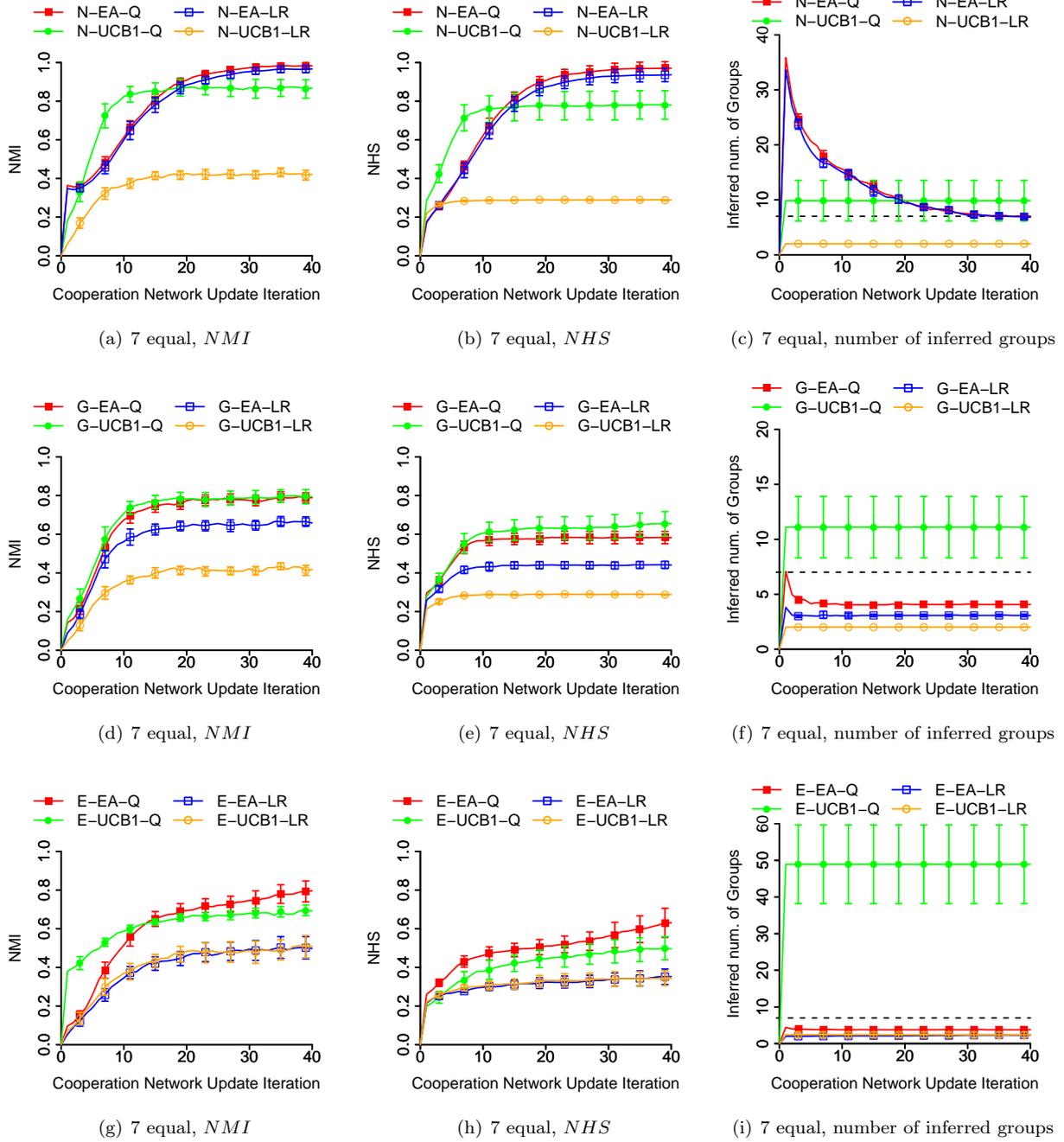
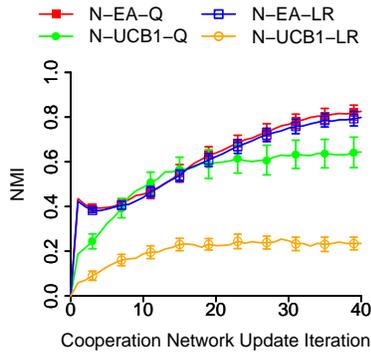
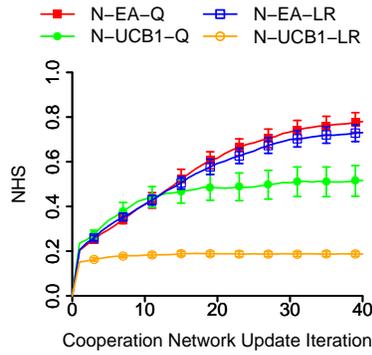


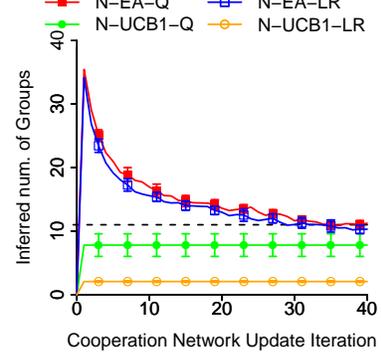
Figure 14.16: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ equal size.



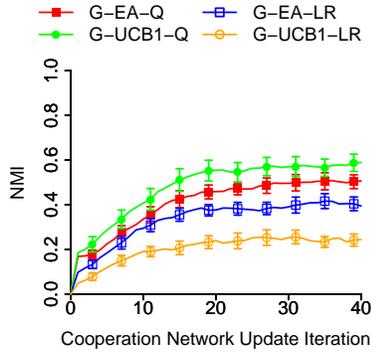
(a) 11 equal, *NMI*



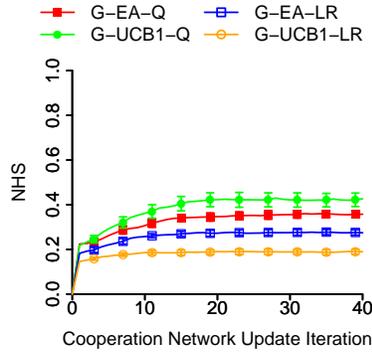
(b) 11 equal, *NHS*



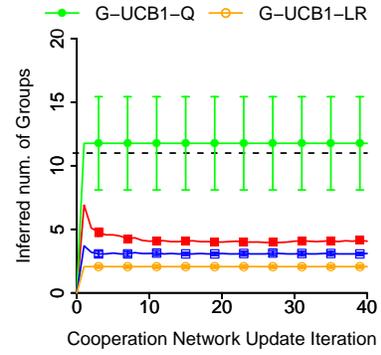
(c) 11 equal, number of inferred groups



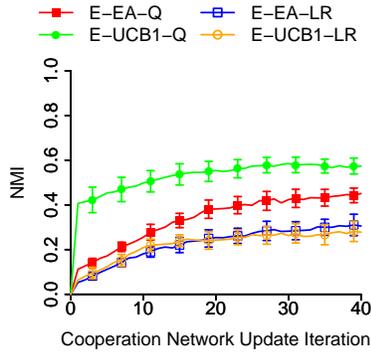
(d) 11 equal, *NMI*



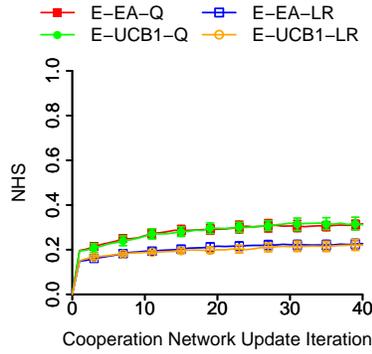
(e) 11 equal, *NHS*



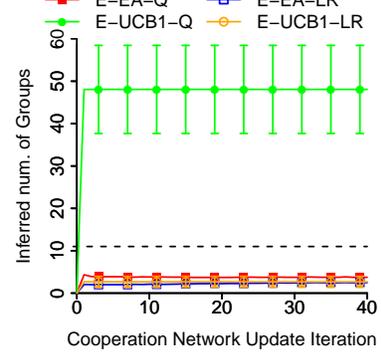
(f) 11 equal, number of inferred groups



(g) 11 equal, *NMI*



(h) 11 equal, *NHS*



(i) 11 equal, number of inferred groups

Figure 14.17: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity *Q* or LinkRank are used as fitness function, for $m = 11$ equal size.

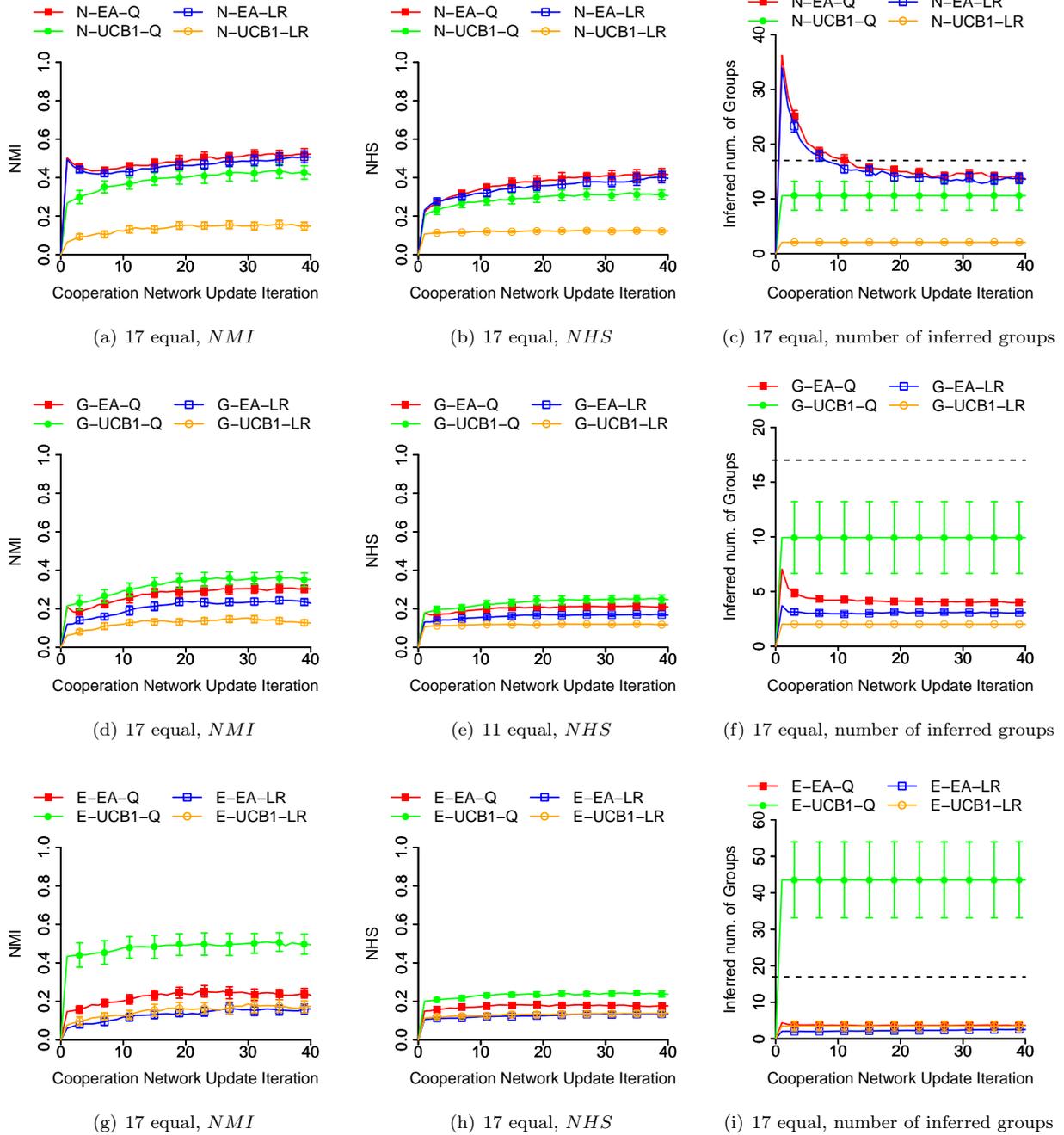
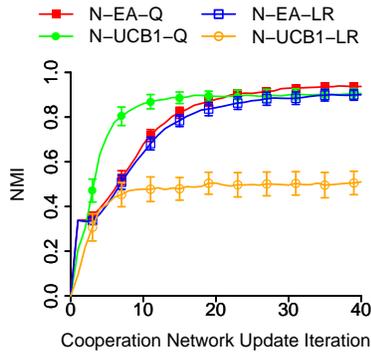
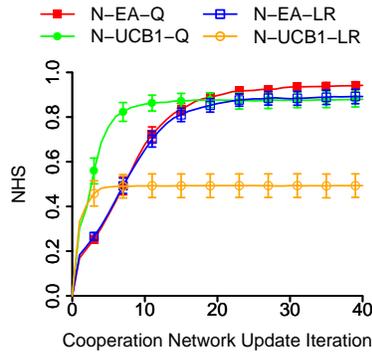


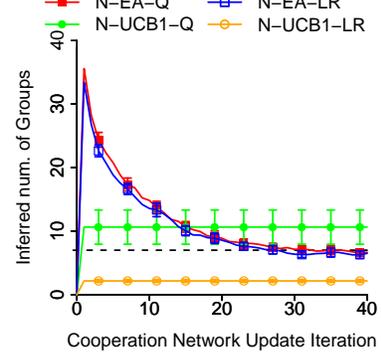
Figure 14.18: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ equal size.



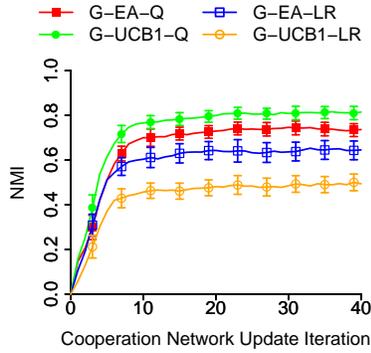
(a) 7 powerlaw, *NMI*



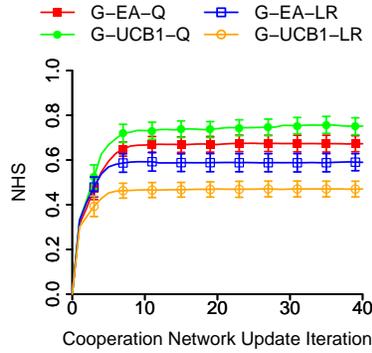
(b) 7 powerlaw, *NHS*



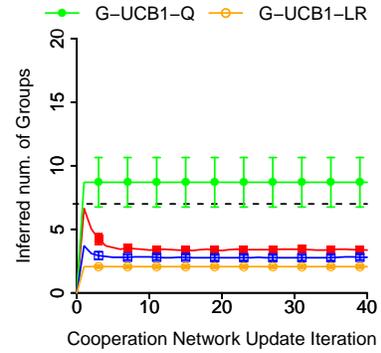
(c) 7 powerlaw, number of inferred groups



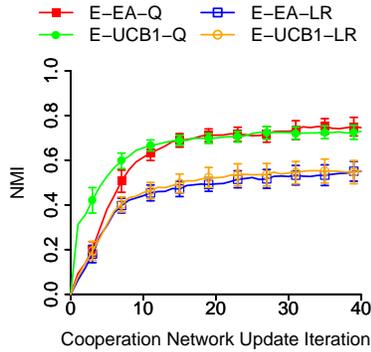
(d) 7 powerlaw, *NMI*



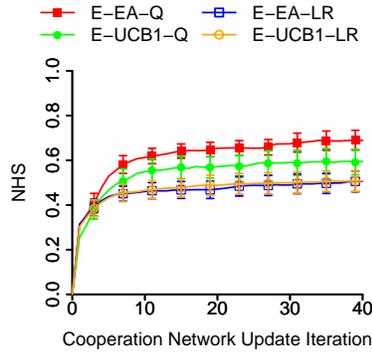
(e) 7 powerlaw, *NHS*



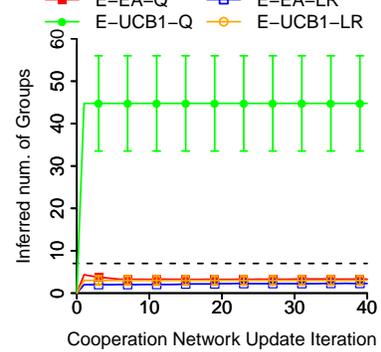
(f) 7 powerlaw, number of inferred groups



(g) 7 powerlaw, *NMI*

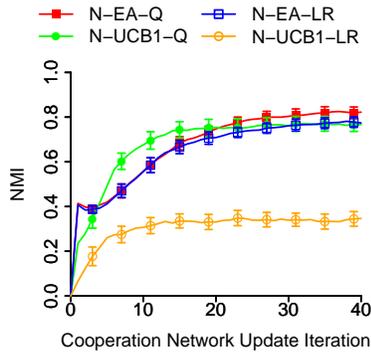


(h) 7 powerlaw, *NHS*

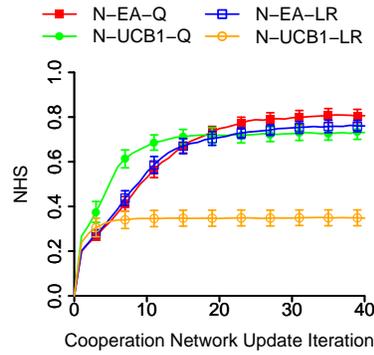


(i) 7 powerlaw, number of inferred groups

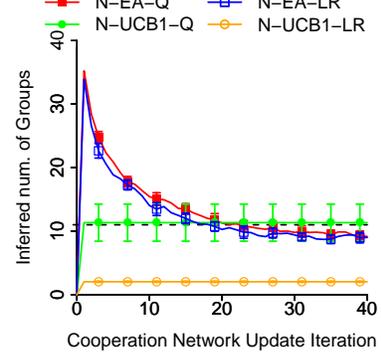
Figure 14.19: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ power law group size distribution.



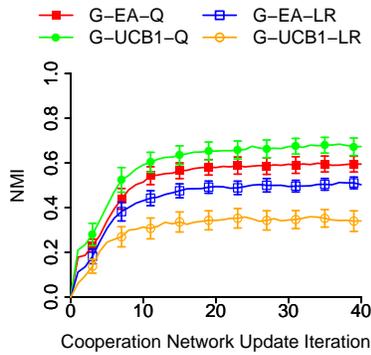
(a) 11 powerlaw, *NMI*



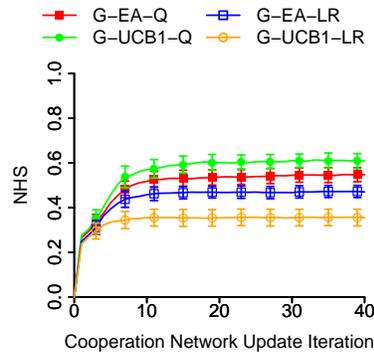
(b) 11 powerlaw, *NHS*



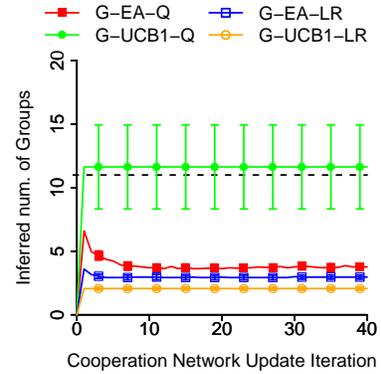
(c) 11 powerlaw, number of inferred groups



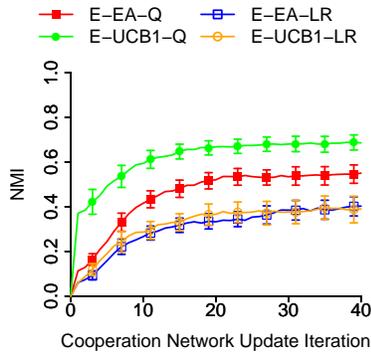
(d) 11 powerlaw, *NMI*



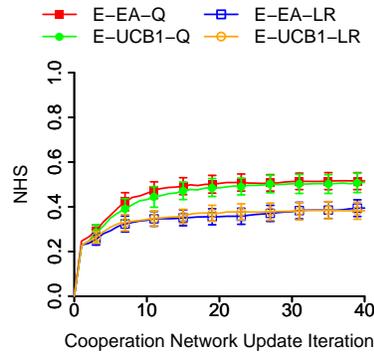
(e) 11 powerlaw, *NHS*



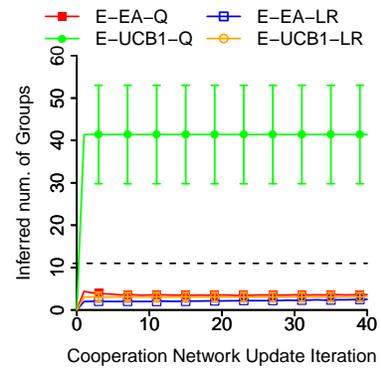
(f) 11 powerlaw, number of inferred groups



(g) 11 powerlaw, *NMI*

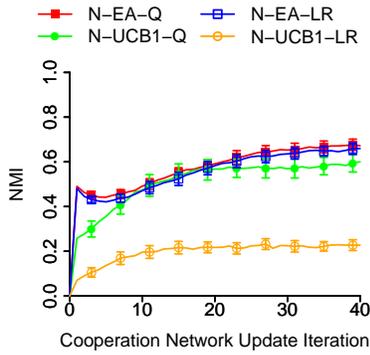


(h) 11 powerlaw, *NHS*

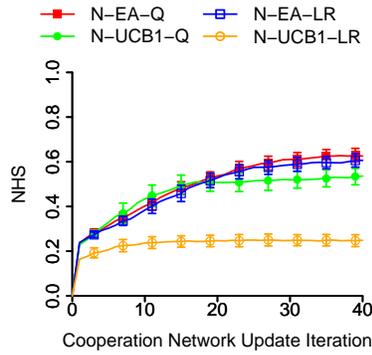


(i) 11 powerlaw, number of inferred groups

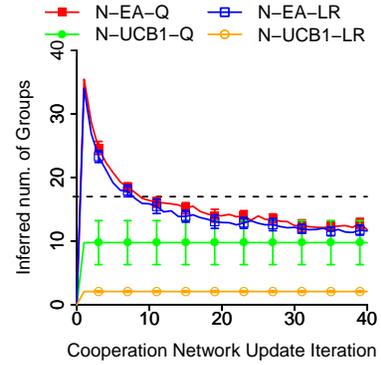
Figure 14.20: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ power law group size distribution.



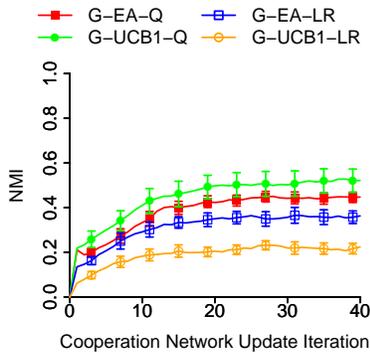
(a) 17 powerlaw, *NMI*



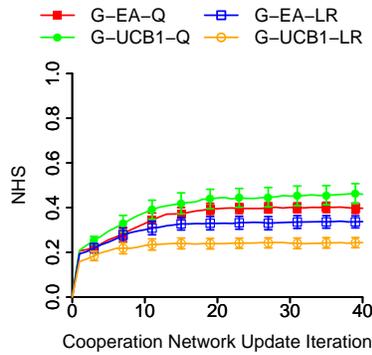
(b) 17 powerlaw, *NHS*



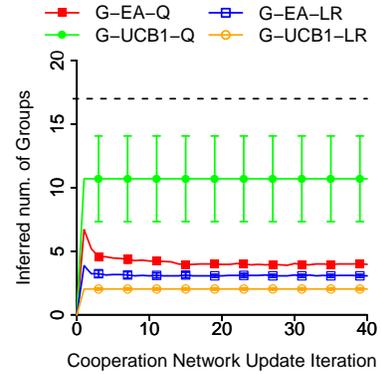
(c) 17 powerlaw, number of inferred groups



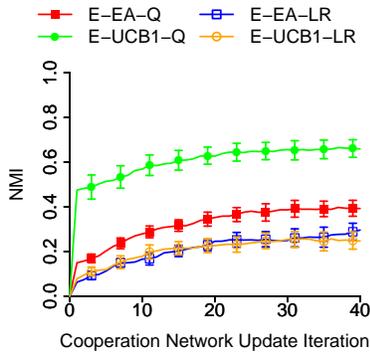
(d) 17 powerlaw, *NMI*



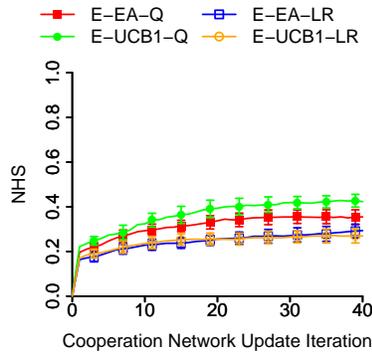
(e) 17 powerlaw, *NHS*



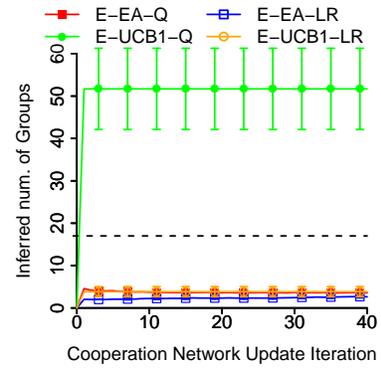
(f) 17 powerlaw, number of inferred groups



(g) 17 powerlaw, *NMI*



(h) 17 powerlaw, *NHS*



(i) 17 powerlaw, number of inferred groups

Figure 14.21: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based UCB1 depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ power law group size distribution.

14.2.2 Node/Edge/Group-based ϵ greedy algorithms

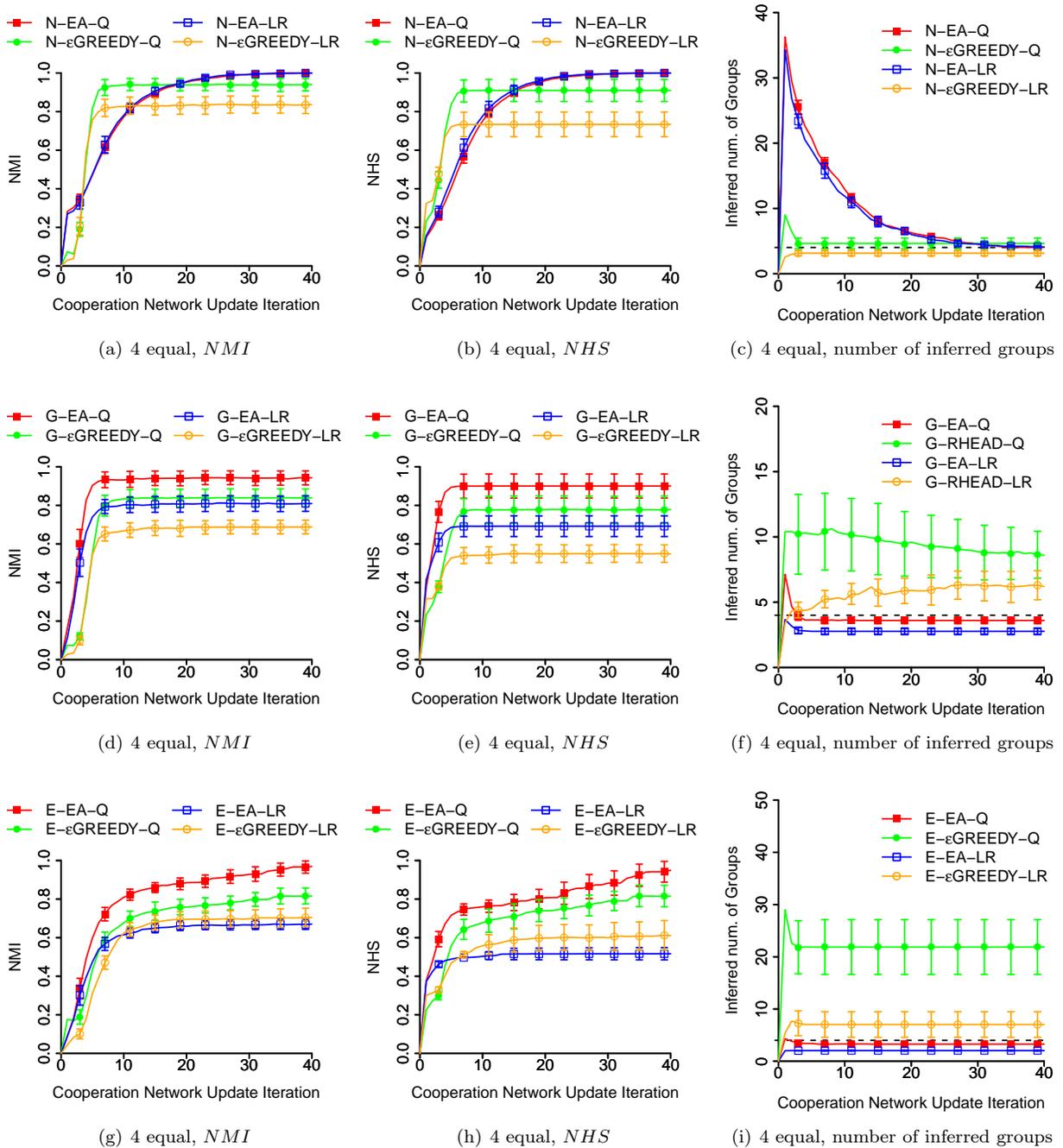


Figure 14.22: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 4$ equal size.

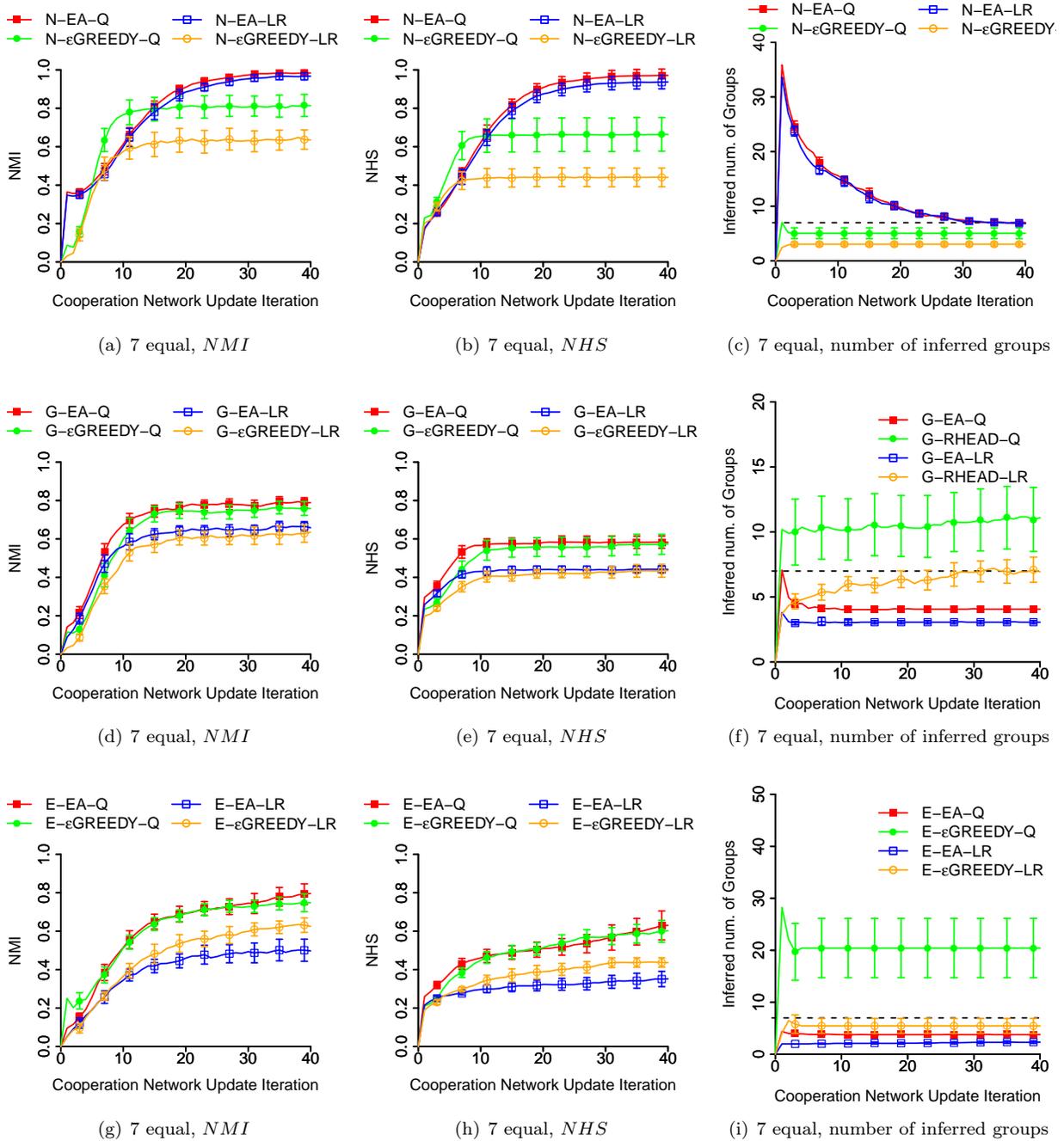


Figure 14.23: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ equal size.

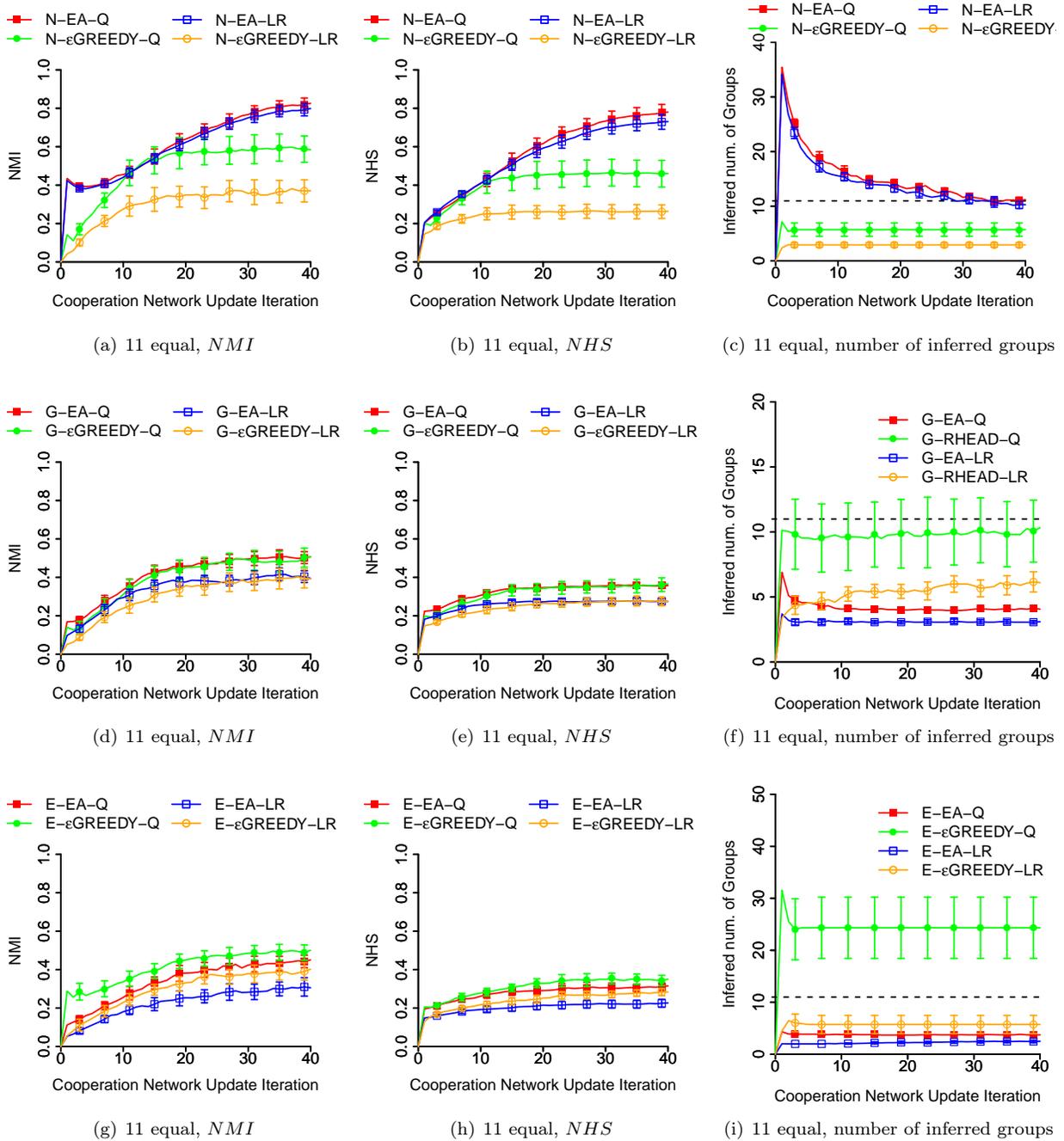


Figure 14.24: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ equal size.

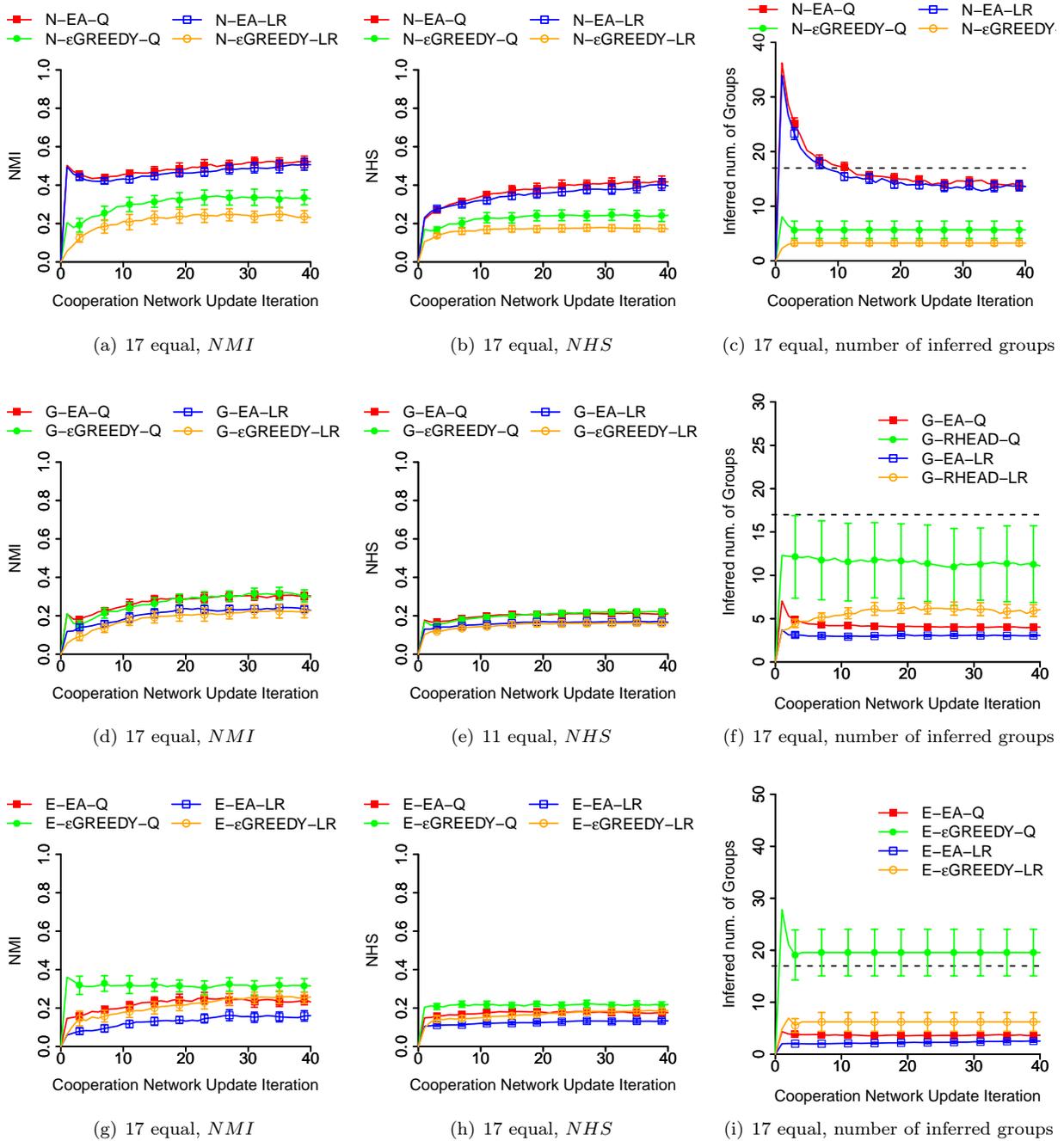


Figure 14.25: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ equal size.

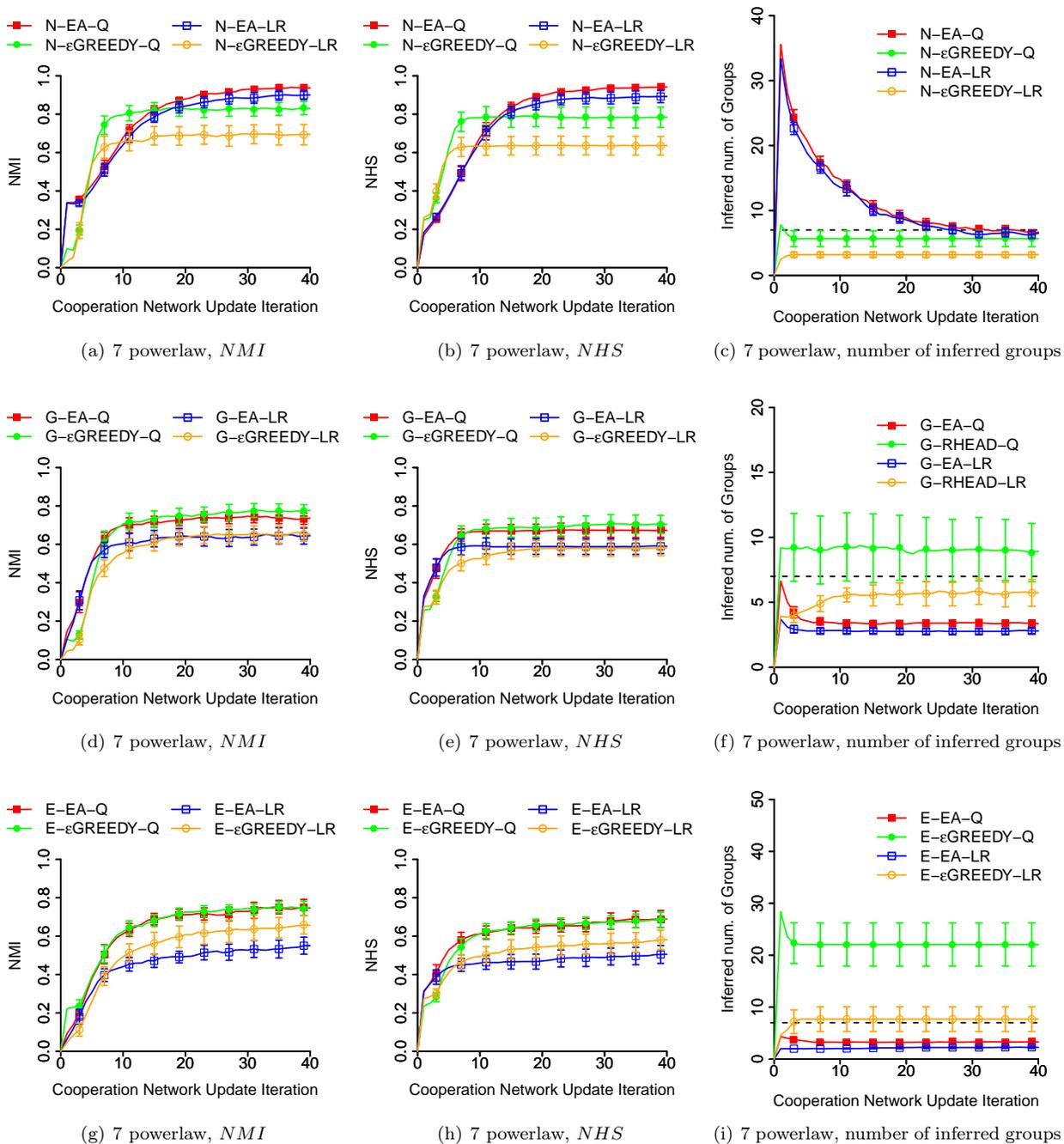


Figure 14.26: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ power law group size distribution.

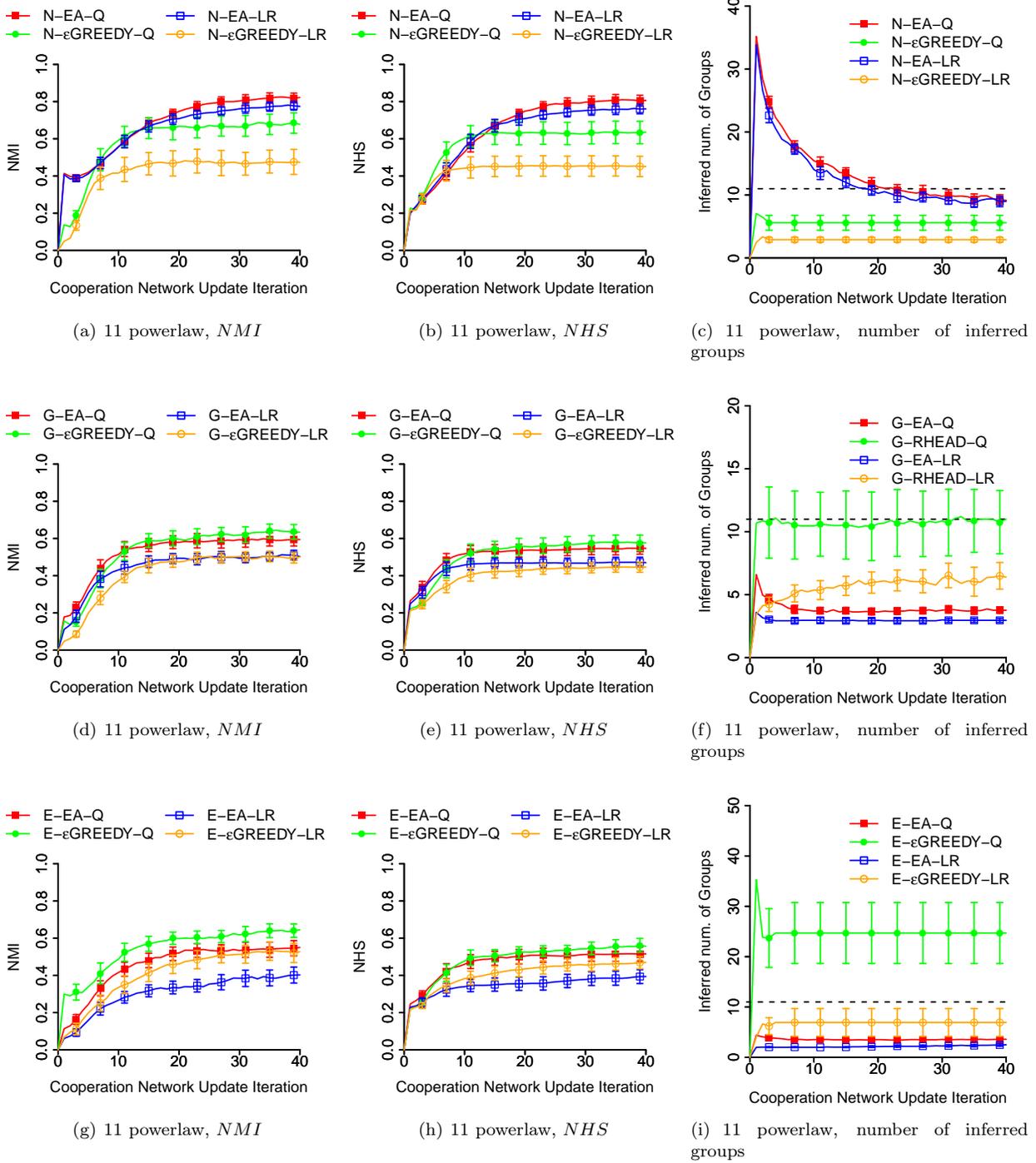


Figure 14.27: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ power law group size distribution.

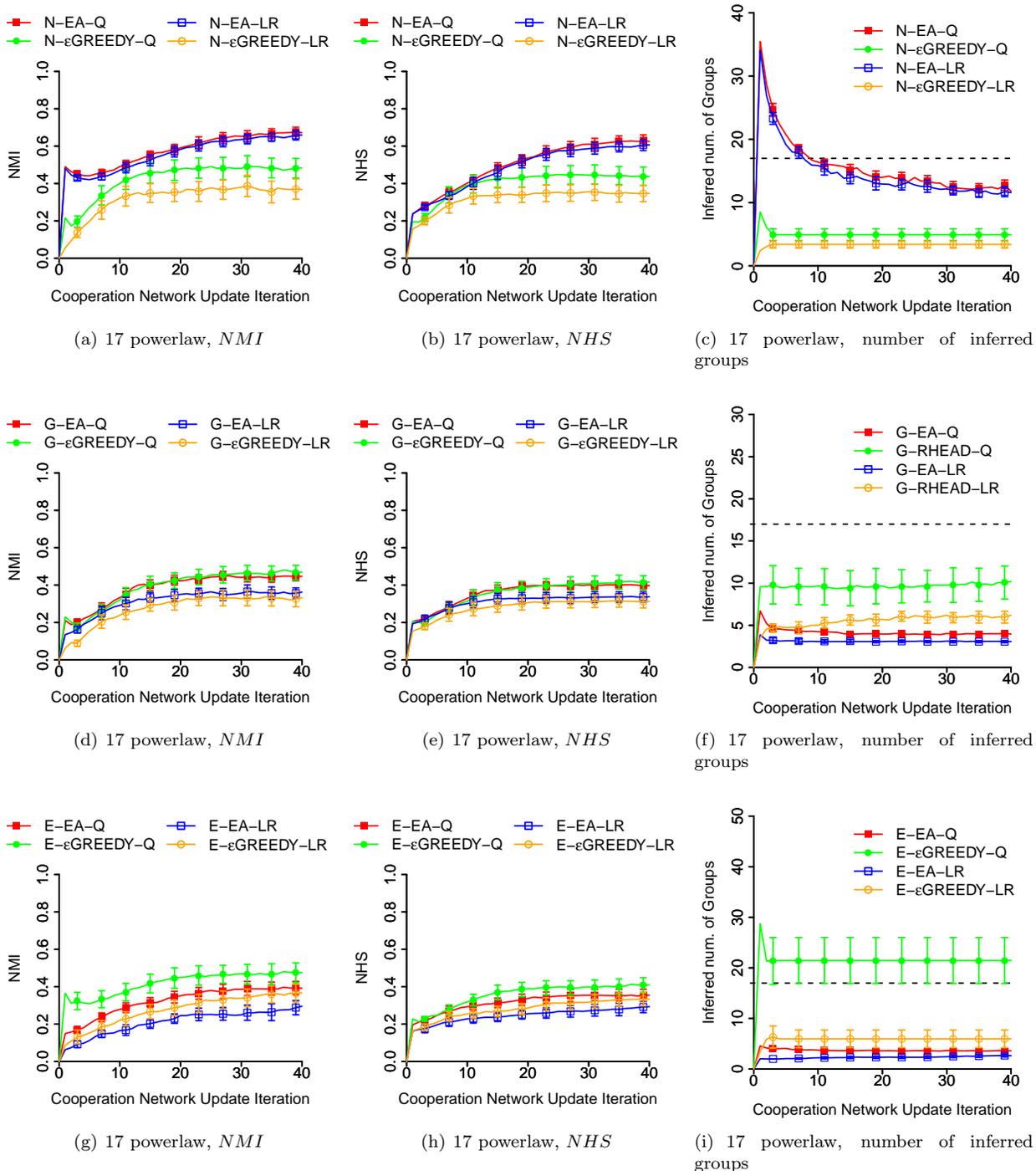


Figure 14.28: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based ϵ -greedy depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ power law group size distribution.

14.2.3 Node/Edge/Group-based Roulette Wheel Selection algorithms

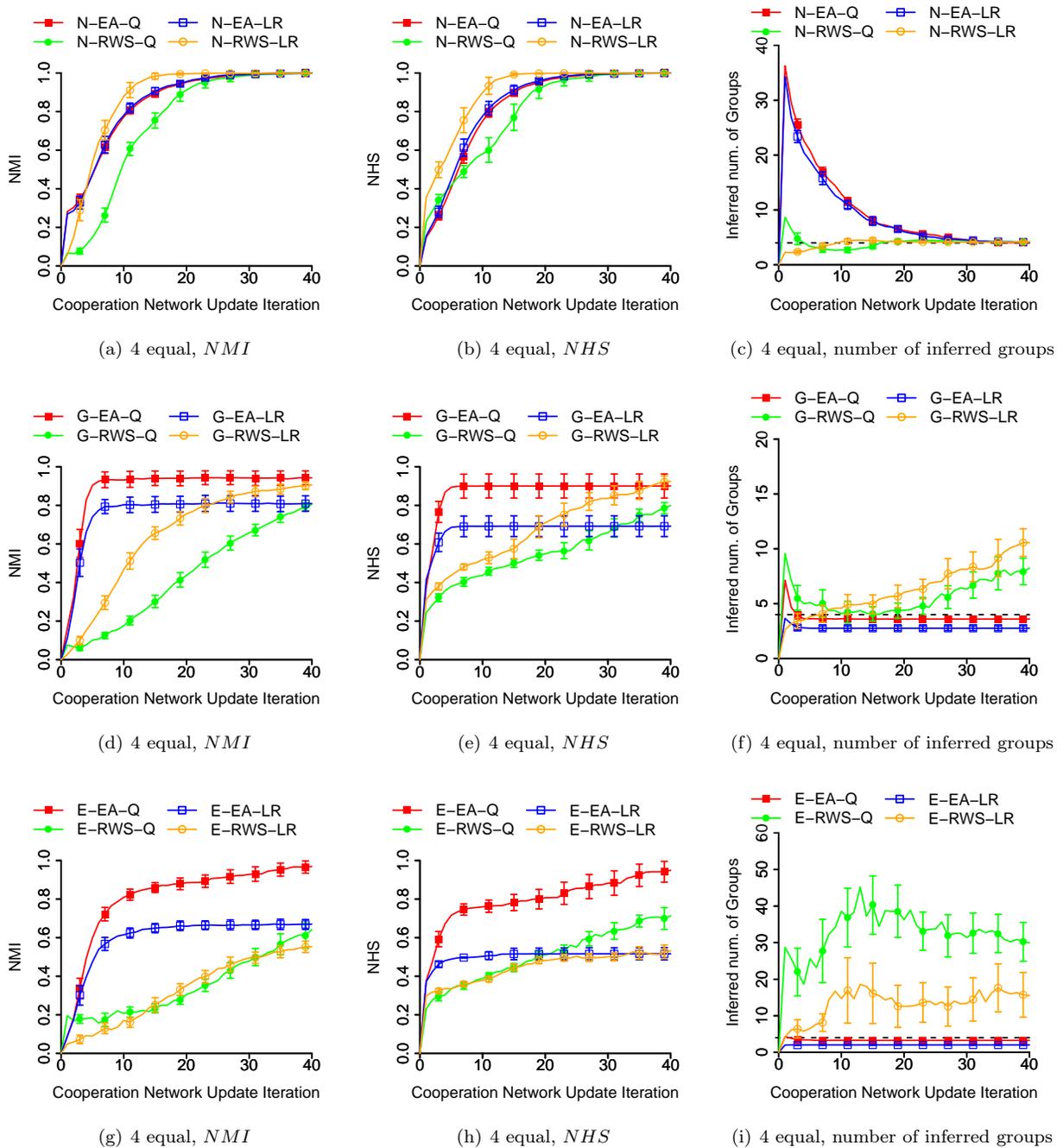
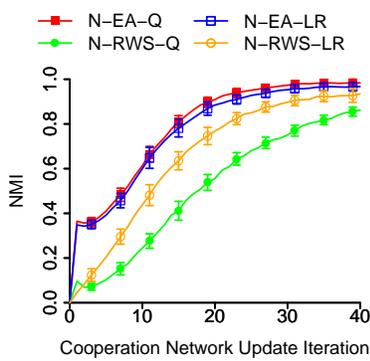
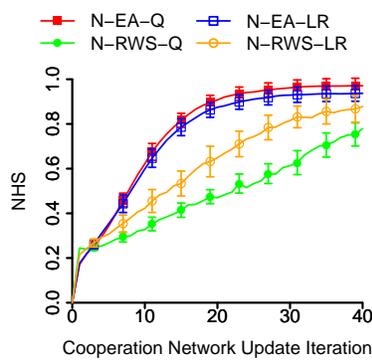


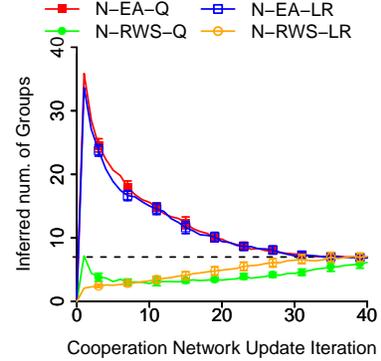
Figure 14.29: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 4$ equal size.



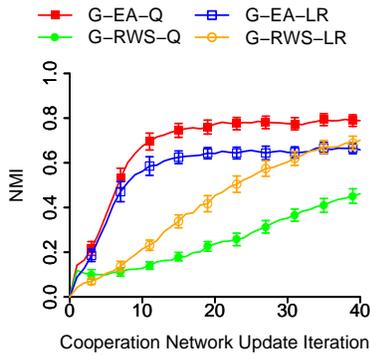
(a) 7 equal, NMI



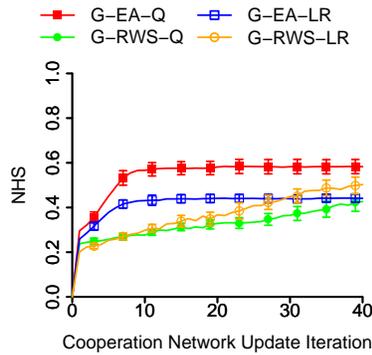
(b) 7 equal, NHS



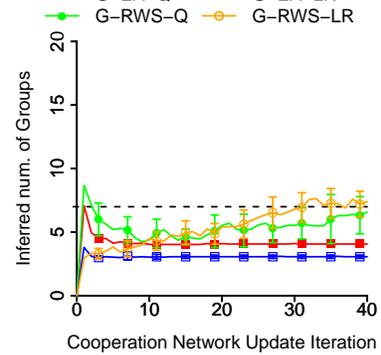
(c) 7 equal, number of inferred groups



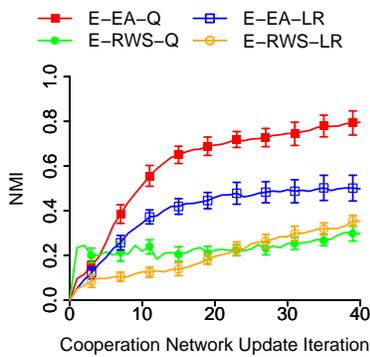
(d) 7 equal, NMI



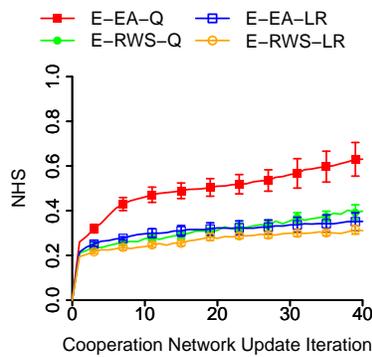
(e) 7 equal, NHS



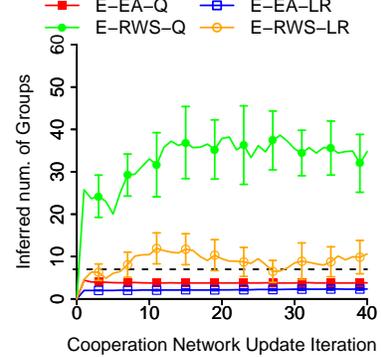
(f) 7 equal, number of inferred groups



(g) 7 equal, NMI

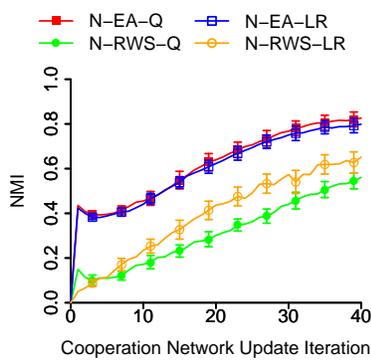


(h) 7 equal, NHS

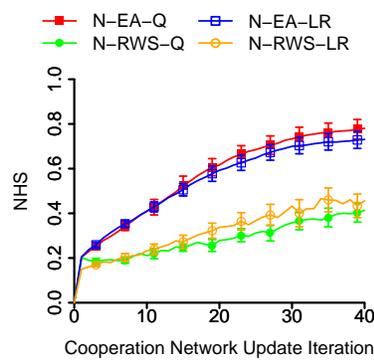


(i) 7 equal, number of inferred groups

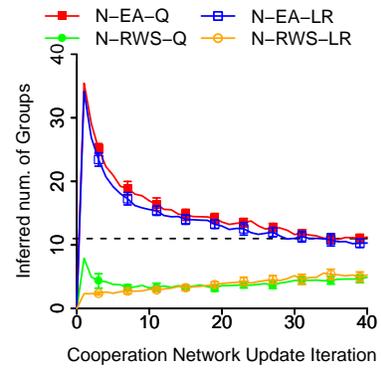
Figure 14.30: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ equal size.



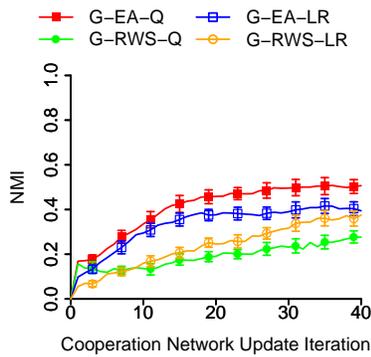
(a) 11 equal, *NMI*



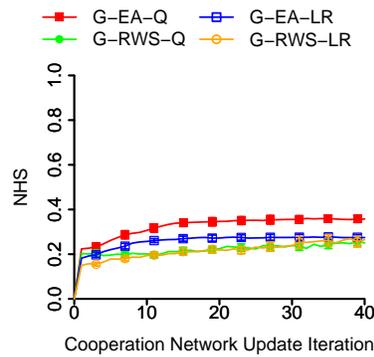
(b) 11 equal, *NHS*



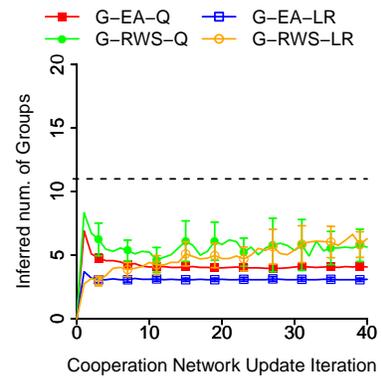
(c) 11 equal, number of inferred groups



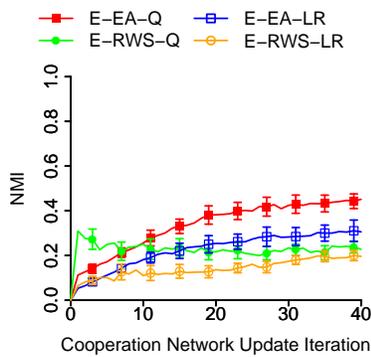
(d) 11 equal, *NMI*



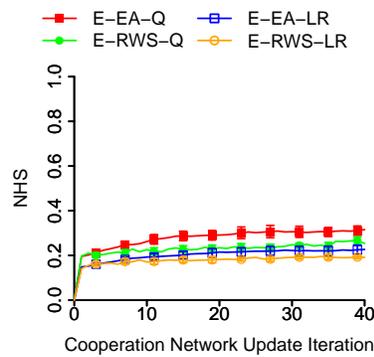
(e) 11 equal, *NHS*



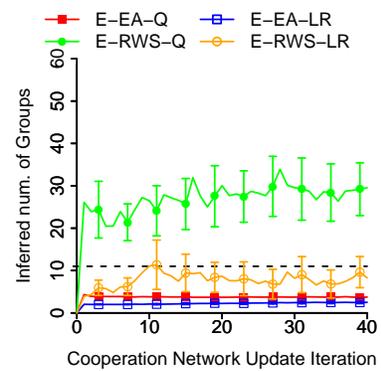
(f) 11 equal, number of inferred groups



(g) 11 equal, *NMI*

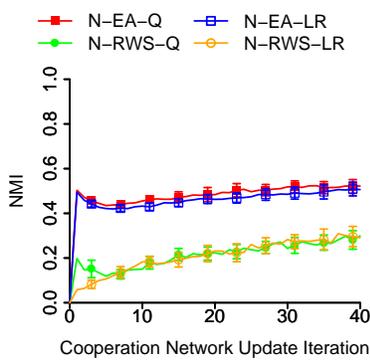


(h) 11 equal, *NHS*

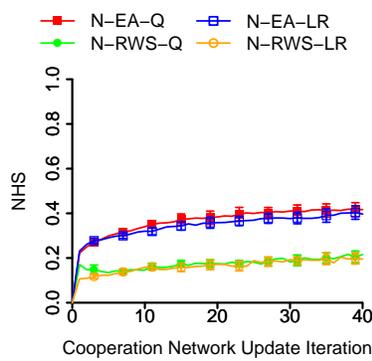


(i) 11 equal, number of inferred groups

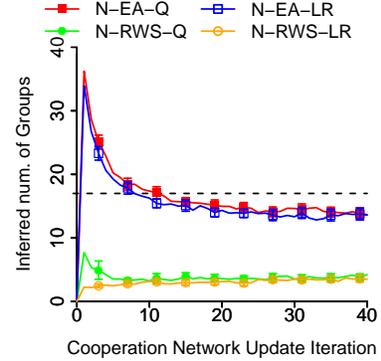
Figure 14.31: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ equal size.



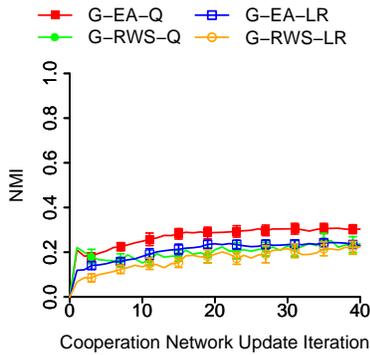
(a) 17 equal, *NMI*



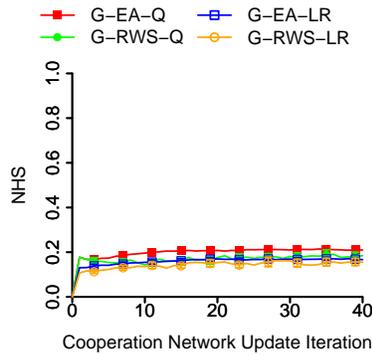
(b) 17 equal, *NHS*



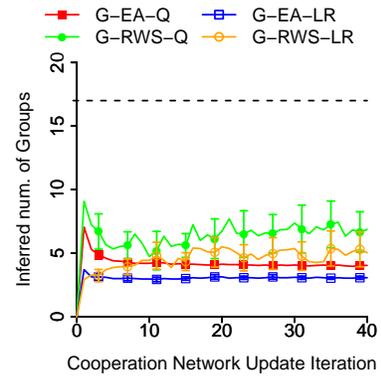
(c) 17 equal, number of inferred groups



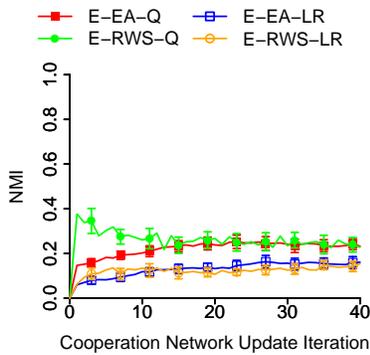
(d) 17 equal, *NMI*



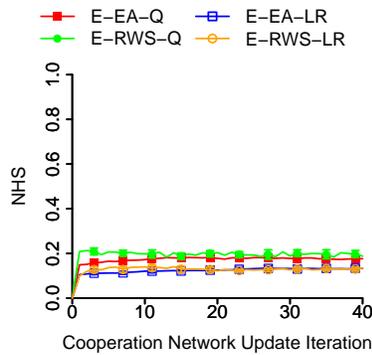
(e) 11 equal, *NHS*



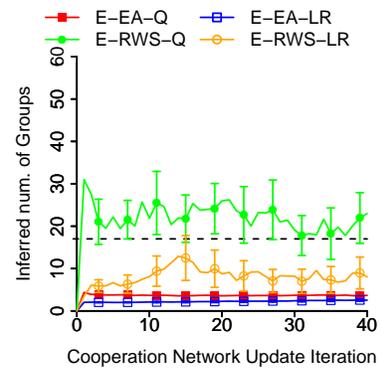
(f) 17 equal, number of inferred groups



(g) 17 equal, *NMI*

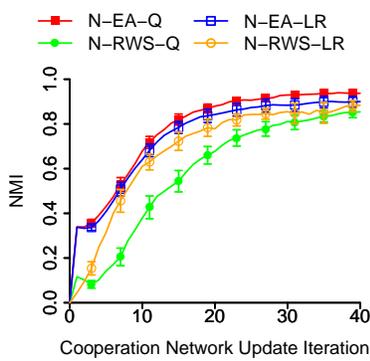


(h) 17 equal, *NHS*

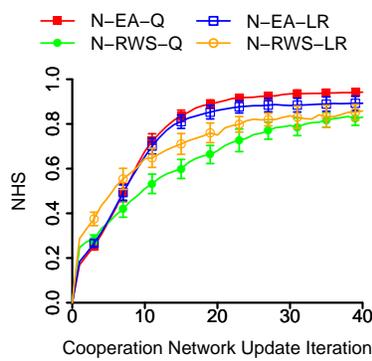


(i) 17 equal, number of inferred groups

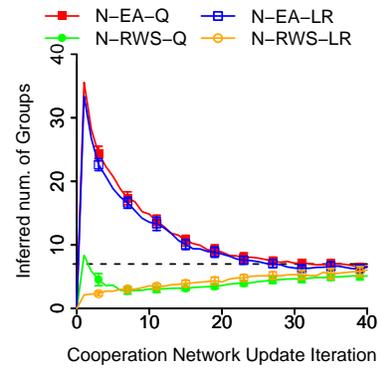
Figure 14.32: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ equal size.



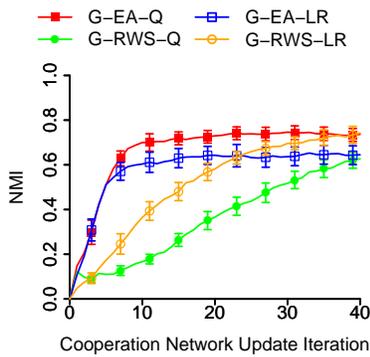
(a) 7 powerlaw, NMI



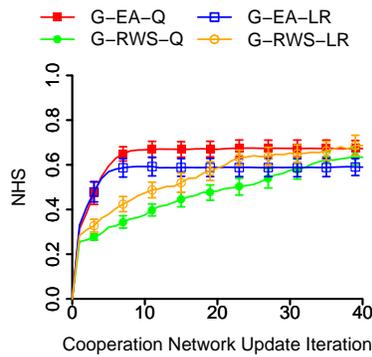
(b) 7 powerlaw, NHS



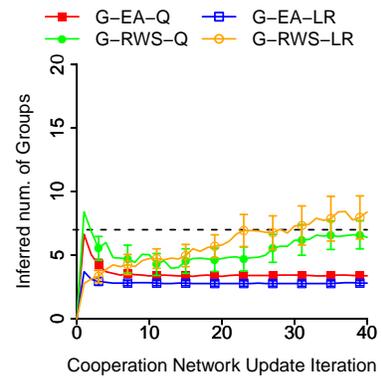
(c) 7 powerlaw, number of inferred groups



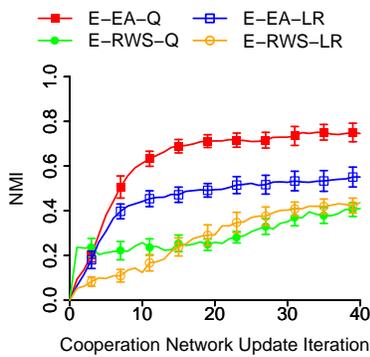
(d) 7 powerlaw, NMI



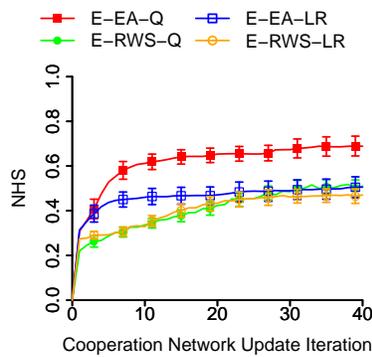
(e) 7 powerlaw, NHS



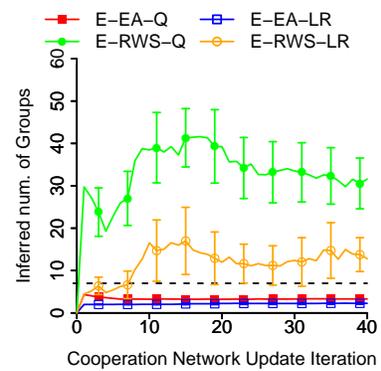
(f) 7 powerlaw, number of inferred groups



(g) 7 powerlaw, NMI

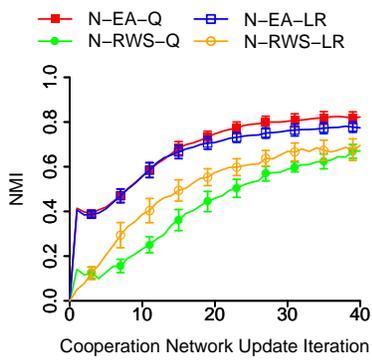


(h) 7 powerlaw, NHS

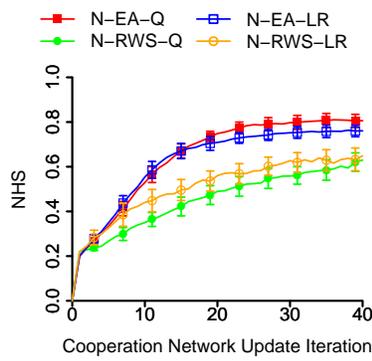


(i) 7 powerlaw, number of inferred groups

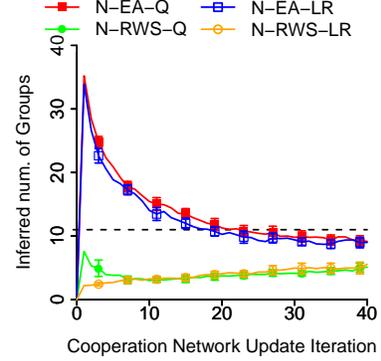
Figure 14.33: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ power law group size distribution.



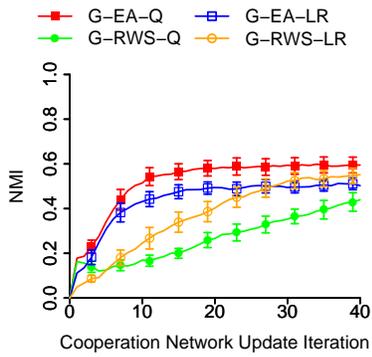
(a) 11 powerlaw, *NMI*



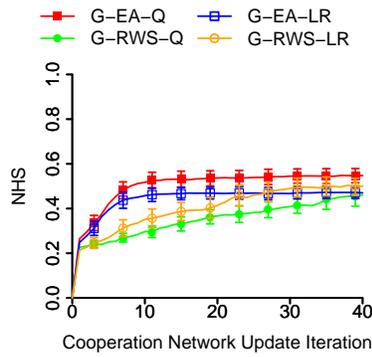
(b) 11 powerlaw, *NHS*



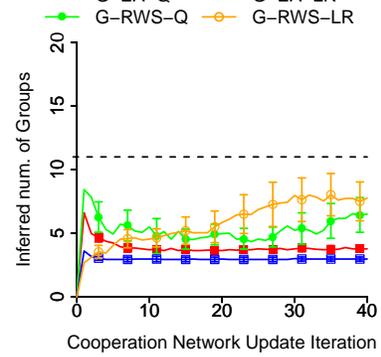
(c) 11 powerlaw, number of inferred groups



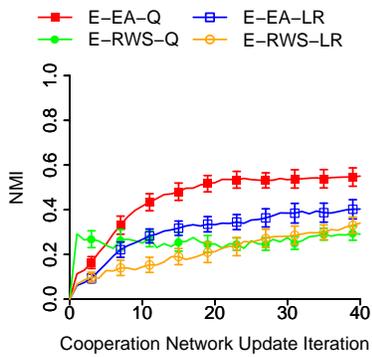
(d) 11 powerlaw, *NMI*



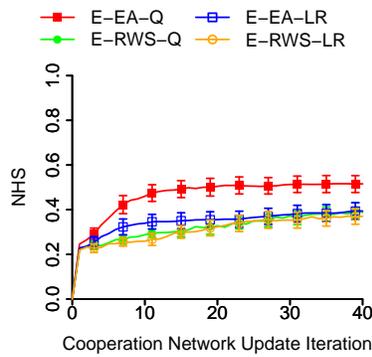
(e) 11 powerlaw, *NHS*



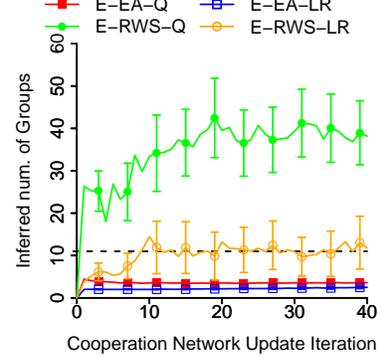
(f) 11 powerlaw, number of inferred groups



(g) 11 powerlaw, *NMI*

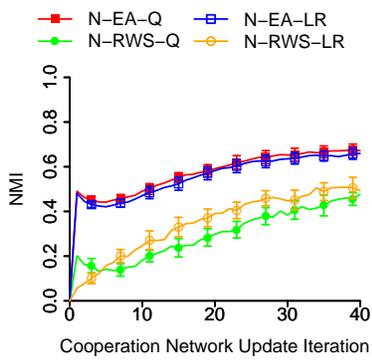


(h) 11 powerlaw, *NHS*

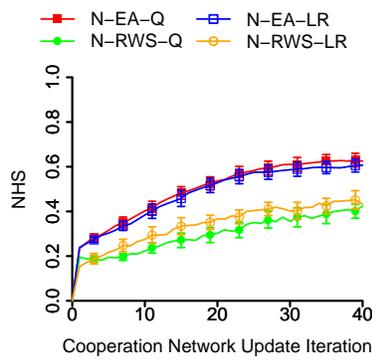


(i) 11 powerlaw, number of inferred groups

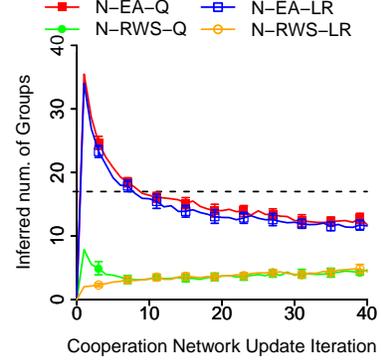
Figure 14.34: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ power law group size distribution.



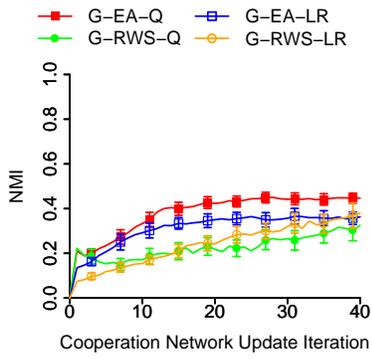
(a) 17 powerlaw, *NMI*



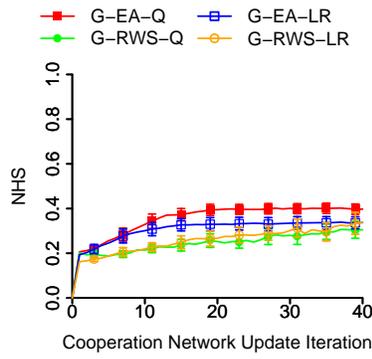
(b) 17 powerlaw, *NHS*



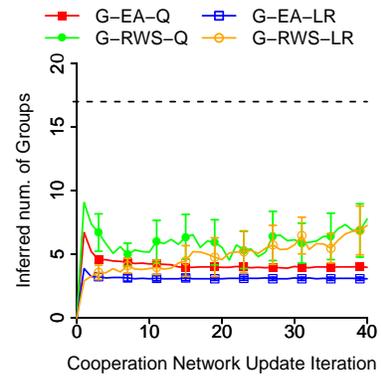
(c) 17 powerlaw, number of inferred groups



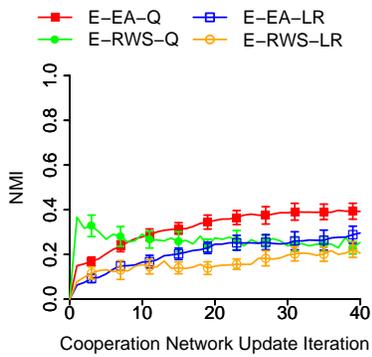
(d) 17 powerlaw, *NMI*



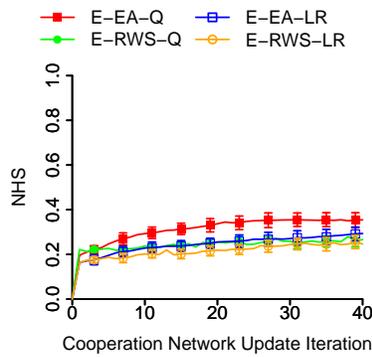
(e) 17 powerlaw, *NHS*



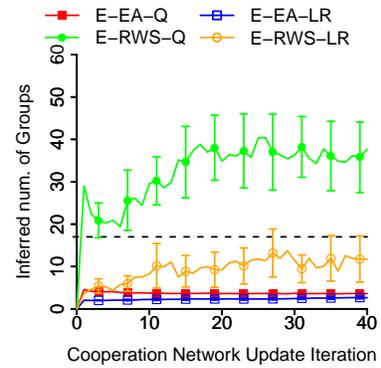
(f) 17 powerlaw, number of inferred groups



(g) 17 powerlaw, *NMI*



(h) 17 powerlaw, *NHS*



(i) 17 powerlaw, number of inferred groups

Figure 14.35: Average performance (*NMI*, *NHS*, and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Roulette Wheel Selection depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ power law group size distribution.

14.2.4 Edge/Group-based Reading Head algorithms

In this Section we have omitted the results of the node representation since it can be found in Section 8.5.2.

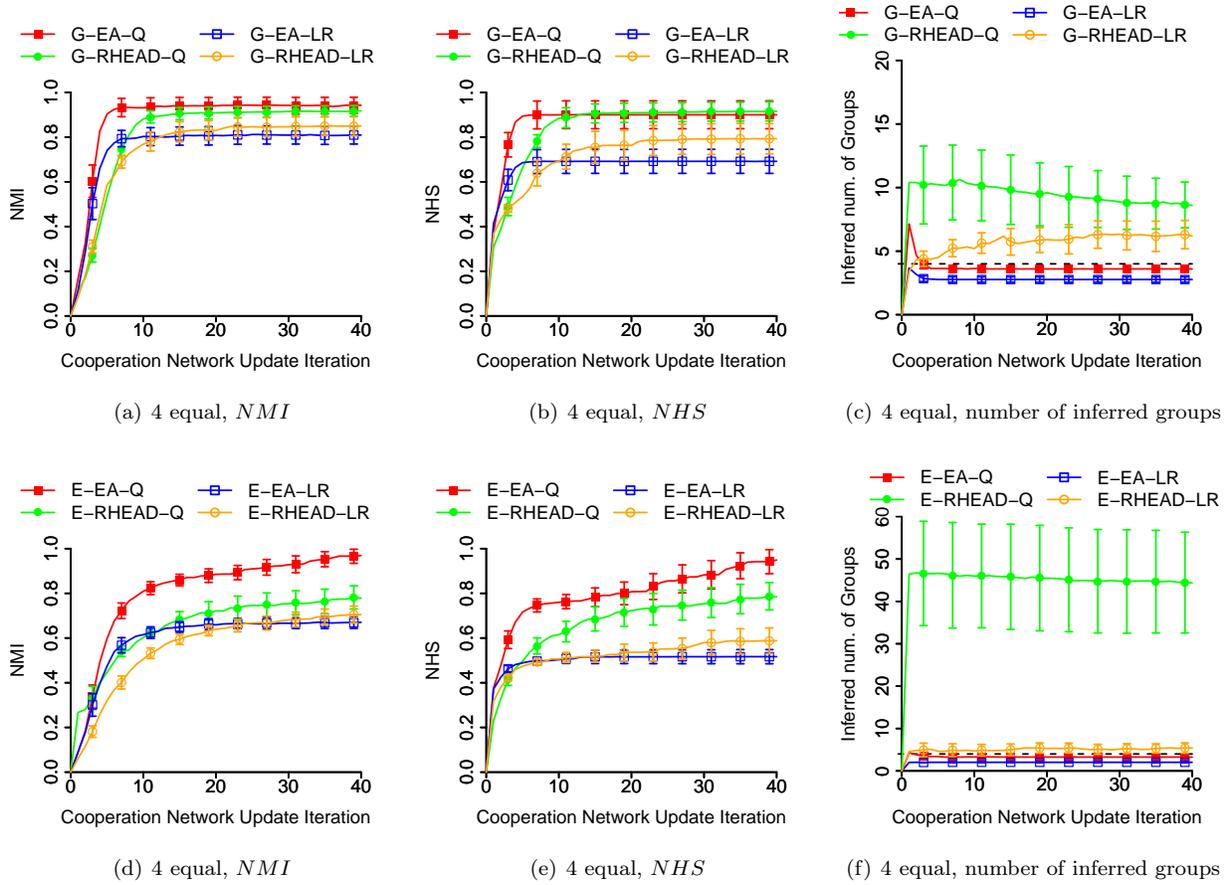


Figure 14.36: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 4$ equal size.

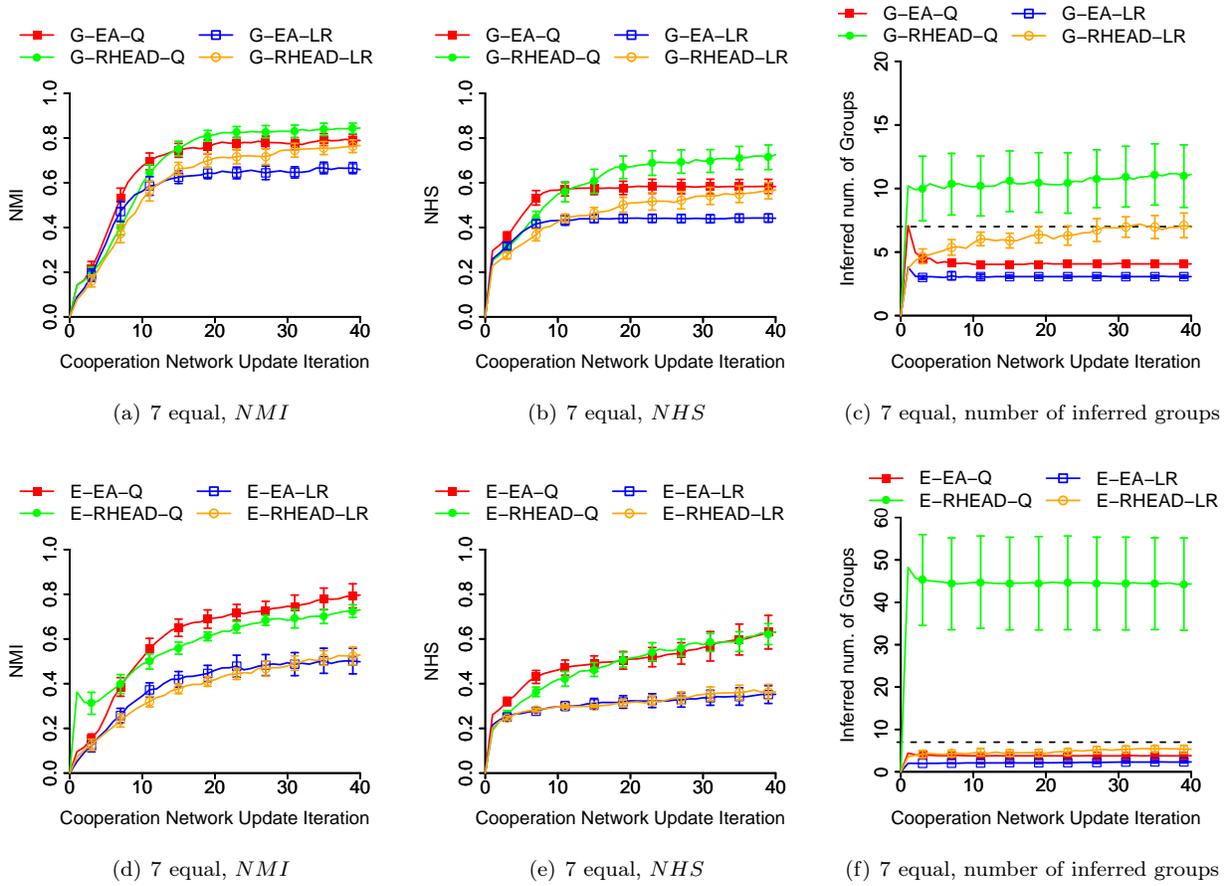


Figure 14.37: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ equal size.

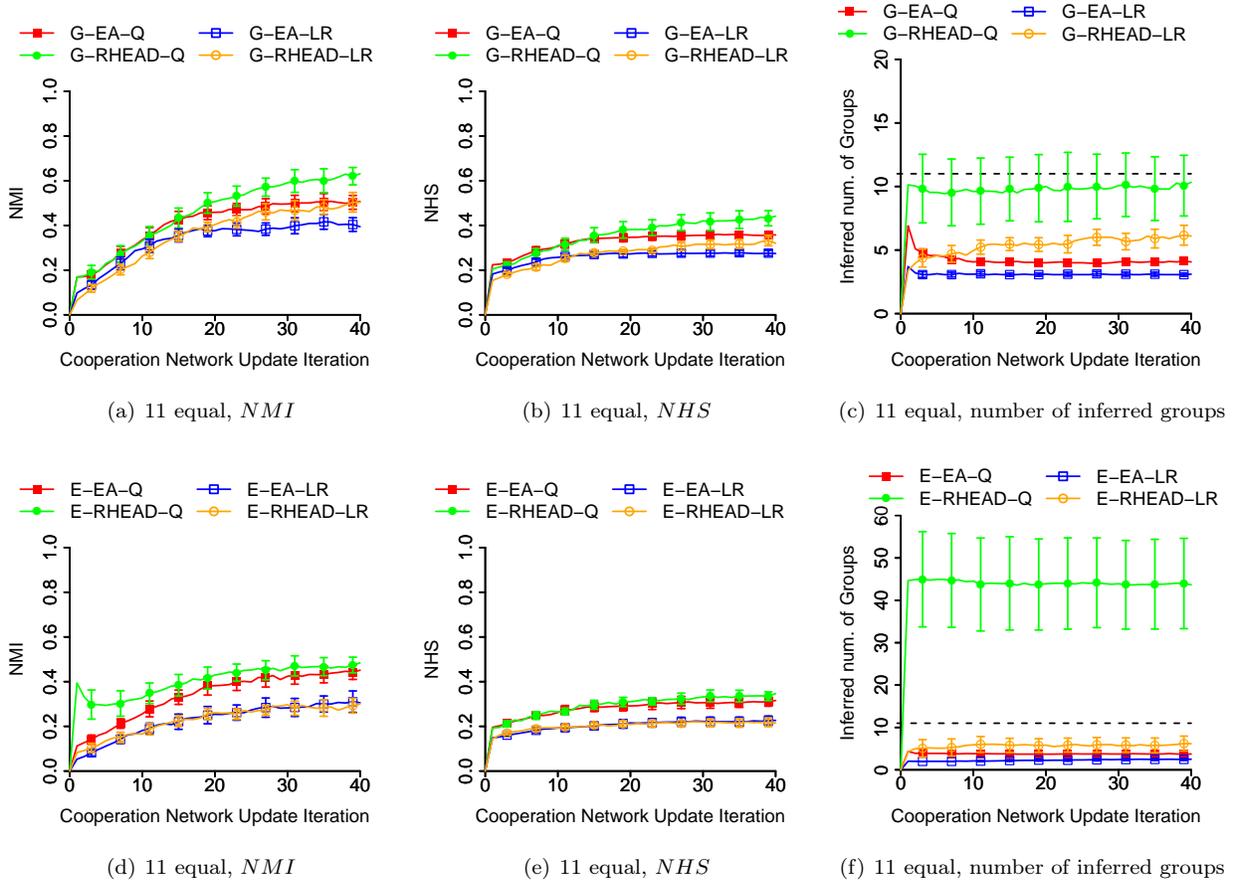


Figure 14.38: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ equal size.

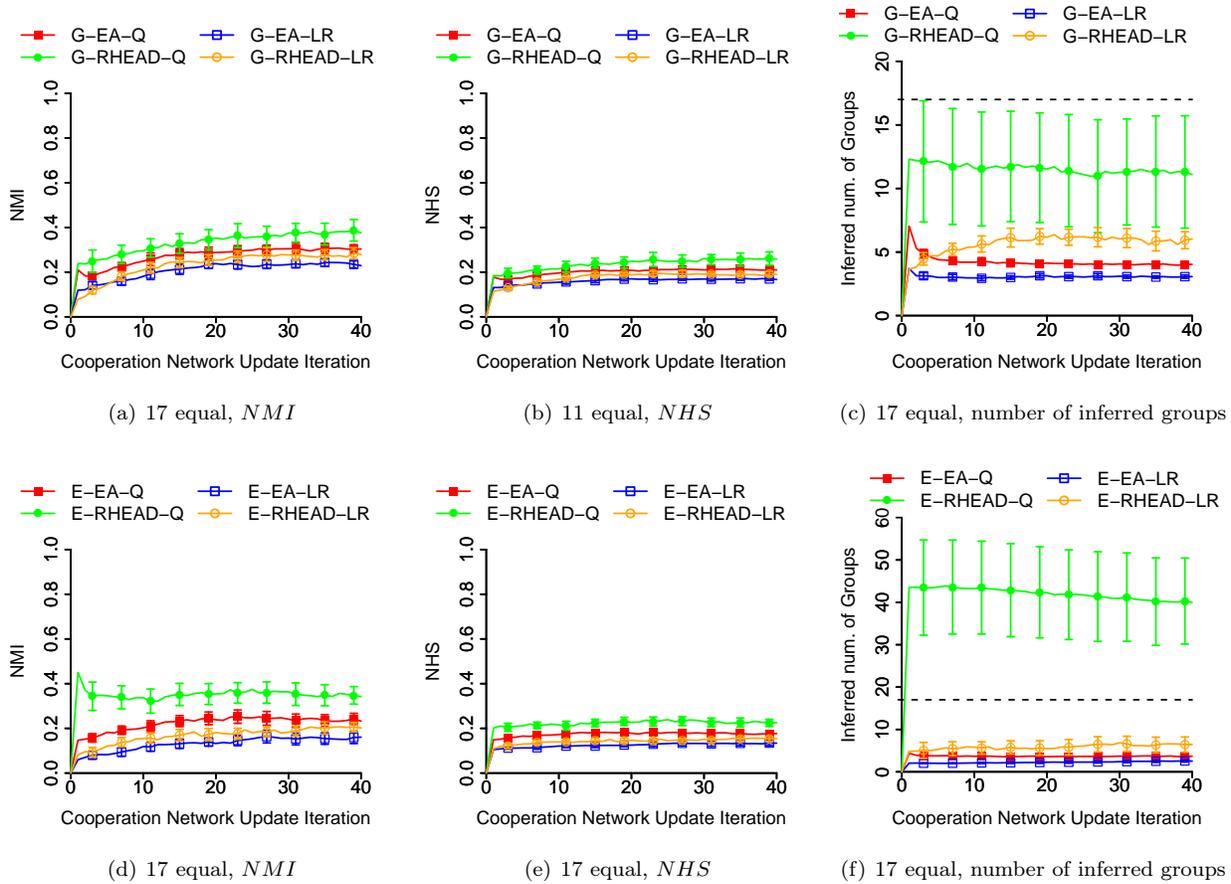


Figure 14.39: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ equal size.

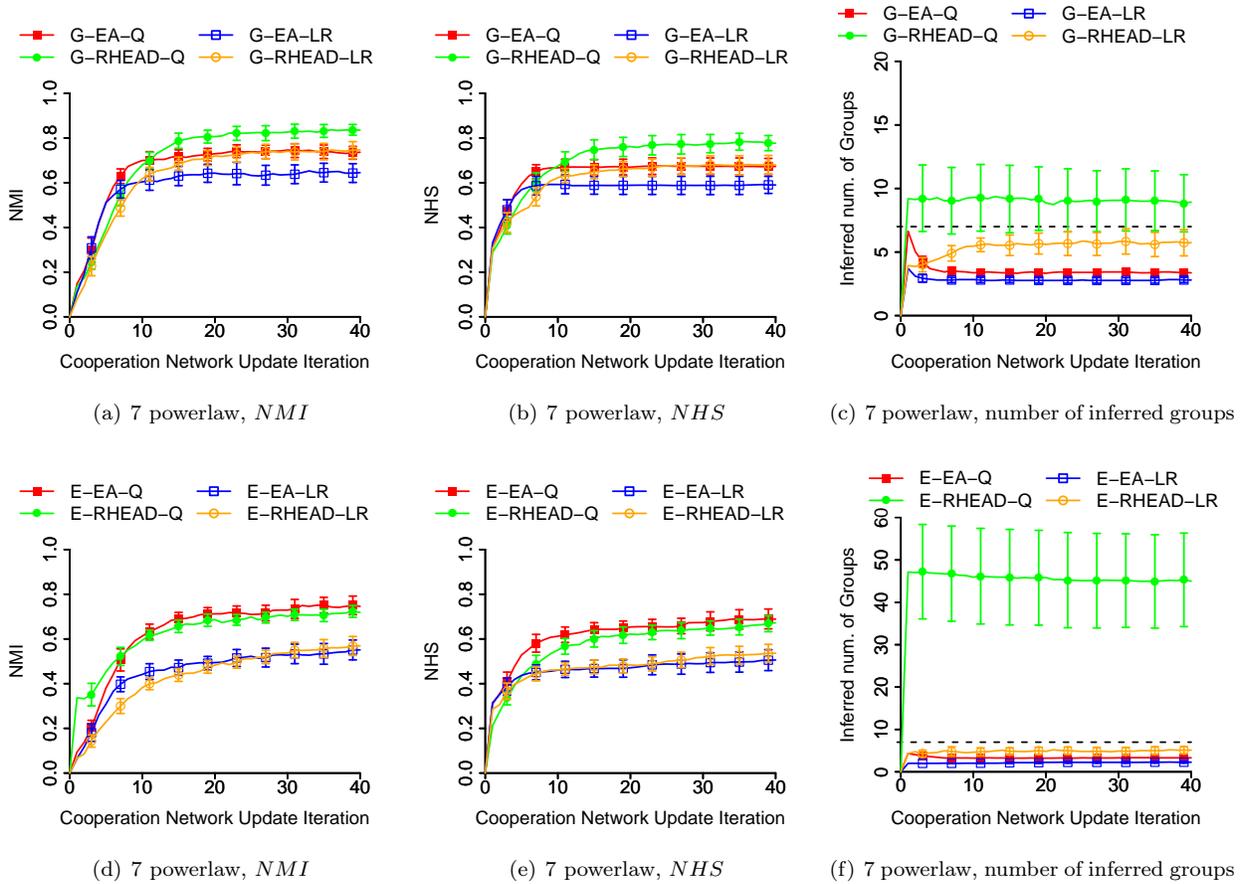


Figure 14.40: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 7$ power law group size distribution.

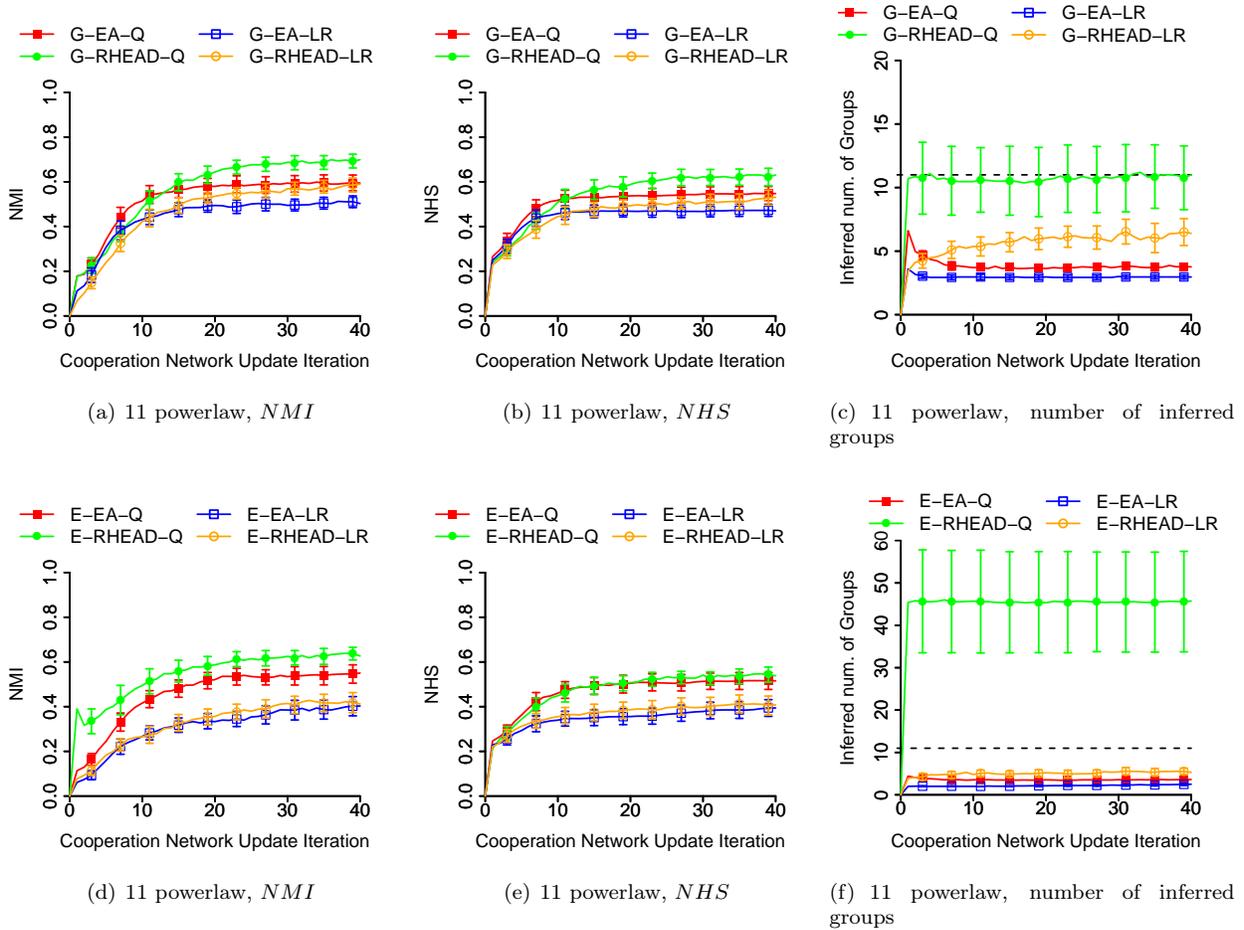


Figure 14.41: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 11$ power law group size distribution.

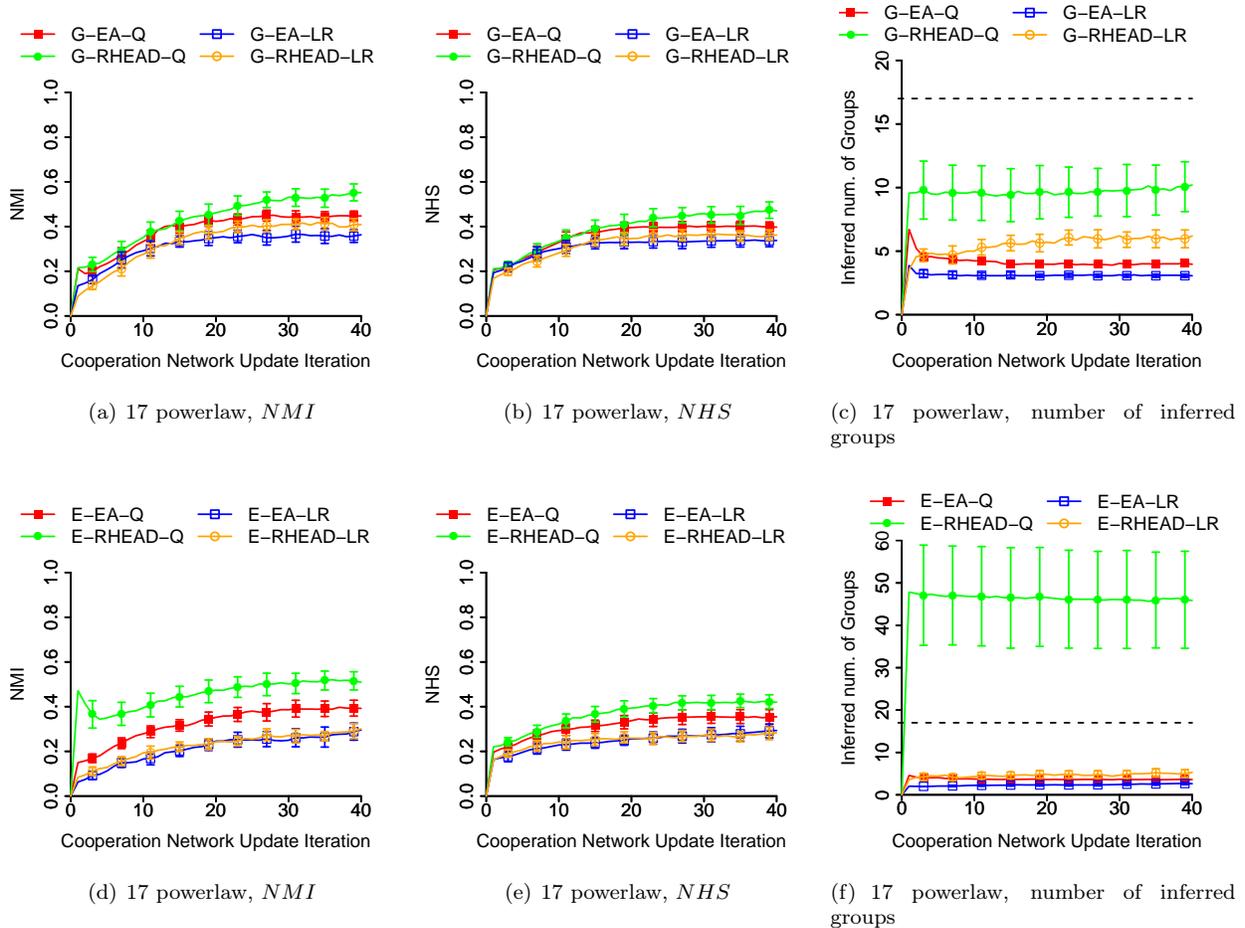


Figure 14.42: Average performance (NMI , NHS , and inferred number of groups) and standard deviation of N/G/E-EA and group/edge-based Reading Head depending on whether modularity Q or LinkRank are used as fitness function, for $m = 17$ power law group size distribution.

Bibliography

- [1] Abbink, K., Brandts, J., Herrmann, B., Orzen, H.: Intergroup conflict and intra-group punishment in an experimental contest game. *The American Economic Review* pp. 420–447 (2010)
- [2] Akerlof, G.A., Kranton, R.E.: Economics and identity. *The Quarterly Journal of Economics* 115(3), 715–753 (2000)
- [3] Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. *Reviews of modern physics* 74(1), 47 (2002)
- [4] Aldecoa, R., Marín, I.: Deciphering network community structure by surprise. *PloS one* 6(9), e24195 (2011)
- [5] Amelio, A., Pizzuti, C.: Community mining in signed networks: a multiobjective approach. In: *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. pp. 95–99. ACM (2013)
- [6] Angiulli, F., Cesario, E., Pizzuti, C.: A greedy search approach to co-clustering sparse binary matrices. In: *Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on*. pp. 363–370. IEEE (2006)
- [7] Antal, T., Krapivsky, P.L., Redner, S.: Social balance on networks: The dynamics of friendship and enmity. *Physica D: Nonlinear Phenomena* 224(1), 130–136 (2006)
- [8] Arenas, A., Duch, J., Fernández, A., Gómez, S.: Size reduction of complex networks preserving modularity. *New Journal of Physics* 9(6), 176 (2007)
- [9] Ashlock, D.: *Evolutionary computation for modeling and optimization*, vol. 103. Springer (2006)
- [10] Ashlock, D., McGuinness, C., Ashlock, W.: Representation in evolutionary computation. In: *WCCI*. pp. 77–97 (2012)
- [11] Asteriadis, S., Tzouveli, P., Karpouzis, K., Kollias, S.: Estimation of behavioral user state based on eye gaze and head pose—application in an e-learning environment. *Multimedia Tools Appl.* 41, 469–493 (February 2009), <http://portal.acm.org/citation.cfm?id=1499070.1499075>
- [12] Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2-3), 235–256 (2002)
- [13] Axelrod, R., Hamilton, W.D.: *The evolution of cooperation* (1981)
- [14] Barabasi, A.: *Network Science*. BarabasiLab (2013)
- [15] Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *science* 286(5439), 509–512 (1999)
- [16] Bernhard, H., Fischbacher, U., Fehr, E.: Parochial altruism in humans. *Nature* 442(7105), 912–915 (2006)

- [17] Berry, J.W., Hendrickson, B., LaViolette, R.A., Phillips, C.A.: Tolerating the community detection resolution limit with edge weighting. *Physical Review E* 83(5), 056119 (2011)
- [18] Billig, M., Tajfel, H.: Social categorization and similarity in intergroup behaviour. *European Journal of Social Psychology* 3(1), 27–52 (1973)
- [19] Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008(10), P10008 (2008)
- [20] Bolton, G.E., Katok, E., Zwick, R.: Dictator game giving: Rules of fairness versus acts of kindness. *International Journal of Game Theory* 27, 269–299 (1998)
- [21] Bolton, G.E., Ockenfels, A.: Erc: A theory of equity, reciprocity, and competition. *American economic review* pp. 166–193 (2000)
- [22] Bornstein, G., Yaniv, I.: Individual and group behavior in the ultimatum game: Are groups more ?rational? players? *Experimental Economics* 1(1), 101–108 (1998)
- [23] Cardamone, L., Yannakakis, G., Togelius, J., Lanzi, P.: Evolving interesting maps for a first person shooter. *Applications of Evolutionary Computation* pp. 63–72 (2011)
- [24] Chang, Y., Maheswaran, R.: The social ultimatum game. In: *Best Demonstration Award. Autonomous Agents and Multi-Agent Systems* (2011)
- [25] Chang, Y.H., Maheswaran, R., Levinboim, T., Rajan, V.: Learning and Evaluating Human-Like NPC Behaviors in Dynamic Games. In: *Proceedings of AIIDE* (2011)
- [26] Charness, G., Rabin, M.: Understanding social preferences with simple tests. *The Quarterly Journal of Economics* 117(3), 817–869 (2002)
- [27] Charness, G., Rigotti, L., Rustichini, A.: Individual Behavior and Group Membership. Available at SSRN 894685 (2006)
- [28] Chen, M.: Addressing social dilemmas and fostering cooperation through computer games. In: *digra* (2005)
- [29] Chen, M., Nguyen, T., Szymanski, B.K.: A new metric for quality of network community structure. *HUMAN* 2(4), pp–226 (2013)
- [30] Chen, Y., Li, S.X.: Group identity and social preferences. *The American Economic Review* pp. 431–457 (2009)
- [31] Cheong, Y.G., Grappiolo, C., Holmgård, C., Berger, F., Khaled, R., Yannakakis, G.N.: Towards Validating Game Scenarios for Teaching Conflict Resolution. In: *Proceedings of the Games for Learning workshop, International Conference on the Foundations of Digital Games* (May 2013)
- [32] Cheong, Y.G., Khaled, R., Grappiolo, C., Campos, J., Martinho, C., Ingram, G.P.D., Paiva, A., Yannakakis, G.N.: A Computational Approach Towards Conflict Resolution for Serious Games. In: *Proceedings of the Sixth International Conference on the Foundations of Digital Games. ACM* (2010)
- [33] Chira, C., Gog, A., Iclanzan, D.: Evolutionary detection of community structures in complex networks: A new fitness function. In: *Evolutionary Computation (CEC), 2012 IEEE Congress on*. pp. 1–8. *IEEE* (2012)
- [34] Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. *Physical review E* 70(6), 066111 (2004)
- [35] Clauset, A., Shalizi, C.R., Newman, M.E.: Power-law distributions in empirical data. *SIAM review* 51(4), 661–703 (2009)

- [36] Cruz, J.D., Bothorel, C., Poulet, F.: Entropy based community detection in augmented social networks. In: Computational Aspects of Social Networks (CASoN), 2011 International Conference on. pp. 163–168. IEEE (2011)
- [37] Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment* 2005(09), P09008 (2005)
- [38] Darwin, C., Bynum, W.F.: *The origin of species by means of natural selection: or, the preservation of favored races in the struggle for life.* AL Burt (2009)
- [39] Dawes, R.M., Messick, D.M.: Social Dilemmas. *International Journal of Psychology* 2(35), 111–116 (2000)
- [40] Dawkins, R.: *The selfish gene.* Oxford university press (2006)
- [41] De Jong, S., Tuyls, K., Verbeeck, K.: Artificial agents learning human fairness. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2. pp. 863–870 (2008)
- [42] Dorigo, M., Di Caro, G.: Ant colony optimization: a new meta-heuristic. In: *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on.* vol. 2. IEEE (1999)
- [43] Du, J., Lai, J., Shi, C.: Multi-objective optimization for overlapping community detection. In: *Advanced Data Mining and Applications*, pp. 489–500. Springer (2013)
- [44] Duan, D., Li, Y., Jin, Y., Lu, Z.: Community mining on dynamic weighted directed graphs. In: *Proceedings of the 1st ACM international workshop on Complex networks meet information & knowledge management.* pp. 11–18. ACM (2009)
- [45] Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. *Physical review E* 72(2), 027104 (2005)
- [46] Ducheneaut, N., Moore, R.J.: Gaining more than experience points: Learning social behavior in multiplayer computer games. In: *Conference proceedings on human factors in computing systems (CHI2004): Extended Abstracts* (2004)
- [47] Ducheneaut, N., Yee, N., Nickell, E., Moore, R.J.: Alone together? exploring the social dynamics of massively multiplayer online games. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems.* pp. 407–416. ACM (2006)
- [48] Eagle, N., Pentland, A.S., Lazer, D.: Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences* 106(36), 15274–15278 (2009)
- [49] Eiben, A.E., Smith, J.E.: *Introduction to evolutionary computing (natural computing series).* Springer (2008)
- [50] Eiben, Á.E., Van Der Hauw, J.K., van Hemert, J.I.: Graph coloring with adaptive evolutionary algorithms. *Journal of Heuristics* 4(1), 25–46 (1998)
- [51] Epstein, J.M., Axtell, R.L.: *Growing Artificial Societies: Social Science from the Bottom Up (Complex Adaptive Systems).* The MIT Press (1996)
- [52] Erdős, P., Rényi, A.: On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci* 5, 17–61 (1960)
- [53] Evans, T., Lambiotte, R.: Line graphs, link partitions, and overlapping communities. *Physical Review E* 80(1), 016105 (2009)

- [54] Falk, A., Fischbacher, U.: A theory of reciprocity. *Games and Economic Behavior* 54(2), 293–315 (2006)
- [55] Falkenauer, E.: A new representation and operators for genetic algorithms applied to grouping problems. *Evolutionary Computation* 2(2), 123–144 (1994)
- [56] Falkenauer, E.: *Genetic algorithms and grouping problems*. John Wiley & Sons, Inc. (1998)
- [57] Falkenauer, E., Bouffouix, S.: A genetic algorithm for job shop. In: *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. pp. 824–829. IEEE (1991)
- [58] Falkenauer, E., Delchambre, A.: A genetic algorithm for bin packing and line balancing. In: *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*. pp. 1186–1192. IEEE (1992)
- [59] Farmer, C.J., Fotheringham, A.S.: Network-based functional regions. *Environment and Planning A* 43(11), 2723–2741 (2011)
- [60] Fehr, E., Fischbacher, U., A, E.F., B, U.F., Lecture, F.H.: Why social preferences matter — the impact of non-selfish motives on competition, cooperation and incentives. *Economic Journal* 112, 1–33 (2002)
- [61] Fehr, E., Schmidt, K.M.: A theory of fairness, competition, and cooperation. *The Quarterly Journal of Economics* 114(3), 817–868 (August 1999)
- [62] Fiedler, M., Haruvy, E., Li, S.X.: Social distance in a virtual world experiment. *Games and Economic Behavior* 72(2), 400–426 (2011)
- [63] Figueiredo, R., Carmo, J., Prada, R.: *Do you trust me or not?*—trust games in agent societies. In: *Proceedings of the 2010 conference on STAIRS 2010: Proceedings of the Fifth Starting AI Researchers’ Symposium*. pp. 101–113. IOS Press (2010)
- [64] Firat, A., Chatterjee, S., Yilmaz, M.: Genetic clustering of social networks using random walks. *Computational Statistics & Data Analysis* 51(12), 6285–6294 (2007)
- [65] Fleurent, C., Ferland, J.A.: Genetic and hybrid algorithms for graph coloring. *Annals of Operations Research* 63(3), 437–461 (1996)
- [66] Folino, F., Pizzuti, C.: An evolutionary multiobjective approach for community discovery in dynamic networks. *IEEE Transactions on Knowledge and Data Engineering* 99(Preliminary), 1 (2013)
- [67] Forsythe, R.: Fairness in Simple Bargaining Experiments. *Games and Economic Behavior* 6(3), 347–369 (1994)
- [68] Fortunato, S., Barthelemy, M.: Resolution limit in community detection. *Proceedings of the National Academy of Sciences* 104(1), 36–41 (2007)
- [69] Freitas, A.A.: A survey of evolutionary algorithms for data mining and knowledge discovery. In: *Advances in evolutionary computing*, pp. 819–845. Springer (2003)
- [70] Friggeri, A., Chelius, G., Fleury, E.: Triangles to capture social cohesion. In: *Privacy, security, risk and trust (passat), 2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (socialcom)*. pp. 258–265. IEEE (2011)
- [71] Fritzsche, B.: *Some competitive learning methods*. Artificial Intelligence Institute, Dresden University of Technology (1997)
- [72] Fritzsche, B., et al.: A growing neural gas network learns topologies. *Advances in neural information processing systems* 7, 625–632 (1995)

- [73] Gal, Y., Grosz, B.J., Kraus, S., Pfeffer, A., Shieber, S.: Colored trails: a formalism for investigating decision-making in strategic environments. In: Proceedings of the 2005 IJCAI workshop on reasoning, representation, and learning in computer games. pp. 25–30 (2005)
- [74] Gal, Y., Pfeffer, A., Marzo, F., Grosz, B.J.: Learning social preferences in games. In: Proceedings of the National Conference on Artificial Intelligence. pp. 226–231. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2004)
- [75] Gao, Z., Jin, N.: Detecting community structure in complex networks based on k-means clustering and data field theory. In: Control and Decision Conference, 2008. CCDC 2008. Chinese. pp. 4411–4416. IEEE (2008)
- [76] Gfeller, D., Chappelier, J.C., De Los Rios, P.: Finding instabilities in the community structure of complex networks. *Physical Review E* 72(5), 056135 (2005)
- [77] Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99(12), 7821–7826 (2002)
- [78] Goette, L., Huffman, D., Meier, S.: The impact of group membership on cooperation and norm enforcement: Evidence using random assignment to real social groups. *The American economic review* 96(2), 212–216 (2006)
- [79] Goette, L., Huffman, D., Meier, S.: The impact of social ties on group interactions: Evidence from minimal groups and randomly assigned real groups. *American Economic Journal: Microeconomics* 4(1), 101–115 (2012)
- [80] Gog, A., Dumitrescu, D., Hirsbrunner, B.: Community Detection in Complex Networks Using Collaborative Evolutionary Algorithms. *Advances in Artificial Life* pp. 886–894 (2007)
- [81] Gómez, S., Jensen, P., Arenas, A.: Analysis of community structure in networks of correlated data. *Physical Review E* 80(1), 016114 (2009)
- [82] Gong, M.G., Zhang, L.J., Ma, J.J., Jiao, L.C.: Community detection in dynamic social networks based on multiobjective immune algorithm. *Journal of Computer Science and Technology* 27(3), 455–467 (2012)
- [83] corrado grappiolo, julian togelius, georgios n yannakakis: artificial evolution for the detection of group identities in complex artificial societies. In: proceedings of SSCI alive 13 (2013)
- [84] Grappiolo, C., Cheong, Y.G., Khaled, R., Yannakakis, G.N.: Modelling Global Pattern Formation for Collaborative Learning Environments. In: Proceedings of the IEEE International Conference on Advanced Learning Technologies (2012)
- [85] Grappiolo, C., Cheong, Y.G., Togelius, J., Khaled, R., Yannakakis, G.N.: Towards Player Adaptivity in a Serious Game for Conflict Resolution. In: Proceedings of the 3rd IEEE International Conference in Games and Virtual Worlds for Serious Applications. pp. 192–198 (2011)
- [86] Grappiolo, C., Martínez, H.P., Yannakakis, G.N.: Validating Generic Metrics of Fairness in Game-based Resource Allocation Scenarios with Crowdsourced Annotations. *Transactions on Computational Intelligence* 13, 176–200 (July 2014)
- [87] Grappiolo, C., Togelius, J., Yannakakis, G.N.: Interaction-based Group Identity Detection via Reinforcement Learning and Artificial Evolution. In: Proceedings of the Evolutionary Computation and Multi-agent Systems and Simulation Workshop, Genetic and Evolutionary Computation Conference (July 2013)
- [88] Grappiolo, C., Togelius, J., Yannakakis, G.N.: Shifting Niches for Community Structure Detection. In: Proceedings of the IEEE Congress on Evolutionary Computation (June 2013)

- [89] Grappiolo, C., Togelius, J., Yannakakis, G.N.: Using Reinforcement Learning and Artificial Evolution for the Detection of Group Identities in Complex Adaptive Artificial Societies. In: Proceedings of the Genetic and Evolutionary Computation Conference (July 2013)
- [90] Grappiolo, C., Togelius, J., Yannakakis, G.N.: Using Reinforcement Learning and Artificial Evolution for the Detection of Group Identities in Complex Adaptive Artificial Societies. In: Proceedings of GECCO (2013)
- [91] Grappiolo, C., Yannakakis, G.N.: Towards Detecting Group Identities in Complex Artificial Societies. In: Proceedings of the Simulation of Adaptive Behaviour Conference (2012)
- [92] Grappiolo, C., Yannakakis, G.N., Asteriadis, S., Karpouzis, K.: Towards Multimodal Player Adaptivity in a Serious Game for Fair Resource Distribution. In: Proceedings of the IEEE International Conference on Multimedia and Expo. [<http://www.youtube.com/watch?v=4EYvawpCC88>] (July 2011)
- [93] Greene, D., Doyle, D., Cunningham, P.: Tracking the evolution of communities in dynamic social networks. In: Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on. pp. 176–183. IEEE (2010)
- [94] Greenwood, G.W., Ashlock, D.: Evolutionary Games and the Study of Cooperation: Why Has So Little Progress Been Made? In: Proceedings of the IEEE World Congress on Computational Intelligence (2012)
- [95] Hagan, M.T., Demuth, H.B., Beale, M.H., et al.: Neural network design. Pws Pub. Boston (1996)
- [96] Hammond, R.A., Axelrod, R.: The Evolution of Ethnocentrism. *Journal of Conflict Resolution* 50(6), 926–936 (Dec 2006)
- [97] He, D., Wang, Z., Yang, B., Zhou, C.: Genetic algorithm with ensemble learning for detecting community structure in complex networks. In: Computer Sciences and Convergence Information Technology, 2009. ICCIT'09. Fourth International Conference on. pp. 702–707. IEEE (2009)
- [98] Heider, F.: The psychology of interpersonal relations. Psychology Press (2013)
- [99] Holland, J.H.: Adaption in natural and artificial systems. The University of Michigan Press (1975)
- [100] Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., De Carvalho, A.P.L.F.: A survey of evolutionary algorithms for clustering. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 39(2), 133–155 (2009)
- [101] Huang, J., Sun, H., Han, J., Deng, H., Sun, Y., Liu, Y.: Shrink: a structural clustering algorithm for detecting hierarchical communities in networks. In: Proceedings of the 19th ACM international conference on Information and knowledge management. pp. 219–228. ACM (2010)
- [102] Huberman, B.A., Glance, N.S.: Evolutionary games and computer simulations. *Proc. Natl. Acad. Sci. U. S. A.* 90(16), 7716–7718 (1993)
- [103] Hummon, N.P., Doreian, P.: Some dynamics of social balance processes: bringing heider back into balance theory. *Social Networks* 25(1), 17–49 (2003)
- [104] Jackson, M.O., Rodriguez-Barraquer, T., Tan, X.: Social capital and social quilts: Network patterns of favor exchange. *The American Economic Review* 102(5), 1857–1897 (2012)
- [105] Janssen, M., Ostrom, E.: Empirically Based, Agent-based Models. *Ecology and Society* 11(2), 37 (2006)
- [106] Kagel, J.H., Kim, C., Moser, D.: Fairness in ultimatum games with asymmetric information and asymmetric payoffs. *Games and Economic Behavior* 13(1), 100–110 (1996)

- [107] Khaled, R., Yannakakis, G.N.: Village voices: An adaptive game for conflict resolution. In: *Foundation of Digital Games Workshop* (2013)
- [108] Khanafiah, D., Situngkir, H.: Social balance theory: revisiting heider's balance theory for many agents. *BFI Working Paper Series* (2004)
- [109] Khatib, F., Cooper, S., Tyka, M.D., Xu, K., Makedon, I., Popović, Z., Baker, D., Players, F.: Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences* 108(47), 18949–18953 (2011)
- [110] Kim, E., Chi, L., Chang, Y.H., Maheswaran, R.: Dynamics of social interactions in a network game. In: *Privacy, security, risk and trust (passat), 2011 IEEE third international conference on and 2011 IEEE third international conference on social computing (socialcom)*. pp. 141–148. *IEEE* (2011)
- [111] Kim, J.H.: The role of identity in intra-and inter-group bargaining in the ultimatum game. *Undergraduate Economic Review* 4(1), 6 (2008)
- [112] Kim, Y., Son, S.W., Jeong, H.: Finding communities in directed networks. *Physical Review E* 81(1), 016103 (2010)
- [113] Kimbrough, S.O., Koehler, G.J., Lu, M., Wood, D.H.: On a feasible–infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch. *European Journal of Operational Research* 190(2), 310–327 (2008)
- [114] Kohonen, T.: *Self-organizing maps*, vol. 30. Springer (2001)
- [115] Komorita, S.S., Parks, C.D.: *Social dilemmas*. Brown & Benchmark (1994)
- [116] Kranton, R., Pease, M., Sanders, S., Huettel, S.: Identity, group conflict, and social preferences. *Working paper* (2012)
- [117] Kuhn, H.W.: The hungarian method for the assignment problem. *Naval research logistics quarterly* 2(1-2), 83–97 (1955)
- [118] Lai, D., Lu, H., Nardini, C.: Finding communities in directed networks by pagerank random walk induced network embedding. *Physica A: Statistical Mechanics and its Applications* 389(12), 2443–2454 (2010)
- [119] Lambiotte, R.: Multi-scale modularity in complex networks. In: *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*. pp. 546–553. *IEEE* (2010)
- [120] Lancichinetti, A., Fortunato, S.: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E* 80(1), 016118 (2009)
- [121] Lancichinetti, A., Fortunato, S.: Community Detection Algorithms: a Comparative Analysis. *Physical Review E* 80(5) (2009)
- [122] Lancichinetti, A., Fortunato, S.: Limits of modularity maximization in community detection. *Physical Review E* 84(6), 066122 (2011)
- [123] Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics* 11(3), 033015 (2009)
- [124] Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Physical Review E* 78(4), 046110 (2008)
- [125] Lancichinetti, A., Radicchi, F., Ramasco, J.J.: Statistical significance of communities in networks. *Physical Review E* 81(4), 046110 (2010)

- [126] Landwehr, P.M.: A collection of economic and social data from glitch, a massively multiplayer online game (2013)
- [127] Lansing, S.J.: Complex Adaptive Systems. *Annual Review of Anthropology* 32, 183–204 (2003)
- [128] Leicht, E.A., Newman, M.E.J.: Community Structure in Directed Networks. *Physical Review Letters* 100(11) (2008)
- [129] Leicht, E.A., Newman, M.E.: Community structure in directed networks. *Physical review letters* 100(11), 118703 (2008)
- [130] Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: *Proceedings of the 19th international conference on World wide web*. pp. 641–650. ACM (2010)
- [131] Leventhal, G.S.: *What should be done with equity theory?* Springer (1980)
- [132] Levine, D.K.: Modeling altruism and spitefulness in experiments. *Review of economic dynamics* 1(3), 593–622 (1998)
- [133] Lewin, K., Cartwright, D.: *Field theory in social science: Selected theoretical papers*. Tavistock London (1952)
- [134] Liebrand, W.B.: A classification of social dilemma games. *Simulation & Gaming* 14(2), 123–138 (1983)
- [135] Lipczak, M., Milios, E.: Agglomerative Genetic Algorithm for Clustering in Social Networks. In: *Proc. of GECCO*. pp. 1243–1250 (2009)
- [136] Liu, J., Zhong, W., Abbass, H.A., Green, D.G.: Separated and overlapping community detection in complex networks using multiobjective evolutionary algorithms. In: *Evolutionary Computation (CEC), 2010 IEEE Congress on*. pp. 1–7. IEEE (2010)
- [137] LIU, T., HU, B.q.: Detecting community in complex networks using cluster analysis. *Complex Systems and Complexity Science* 1, 005 (2007)
- [138] Liu, X., Li, D., Wang, S., Tao, Z.: Effective Algorithm for Detecting Community Structure in Complex Networks Based on GA and Clustering. *Computational Science–ICCS* pp. 657–664 (2007)
- [139] Liu, Y., Liu, Q., Qin, Z.: Community detecting and feature analysis in real directed weighted social networks. *Journal of Networks* 8(6), 1432–1439 (2013)
- [140] Lu, Z., Wen, Y., Cao, G.: Community detection in weighted networks: Algorithms and applications. In: *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. vol. 18, p. 22 (2013)
- [141] Luo, F., Wang, J.Z., Promislow, E.: Exploring local community structures in large networks. *Web Intelligence and Agent Systems* 6(4), 387–400 (2008)
- [142] Macy, M.W., Willer, R.: From factors to actors: Computational sociology and agent-based modeling. *Annual review of sociology* pp. 143–166 (2002)
- [143] Mahfoud, S.W.: Niching methods for genetic algorithms. *Urbana* 51(95001) (1995)
- [144] Malliaros, F.D., Vazirgiannis, M.: Clustering and community detection in directed networks: A survey. *Physics Reports* 533(4), 95–142 (2013)
- [145] Mandala, S.R., Kumara, S.R., Rao, C.R., Albert, R.: Clustering social networks using ant colony optimization. *Operational Research* pp. 1–19 (2013)

- [146] Martínez, H., Yannakakis, G.: Genetic search feature selection for affective modeling: a case study on reported preferences. In: Proceedings of the 3rd international workshop on Affective interaction in natural environments. pp. 15–20. ACM (2010)
- [147] Marzo, F., Grosz, B.J., Pfeffer, A.: Social preferences in Relational Contexts. In: In Fourth Conference in Collective Intentionality (2005)
- [148] Mazur, P., Zmarzłowski, K., Orlowski, A.: Genetic algorithms approach to community detection. *Acta Physica Polonica-Series A General Physics* 117(4), 703 (2010)
- [149] McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in social networks. *Annual review of sociology* pp. 415–444 (2001)
- [150] Metcalf, S., Paich, M.: Spatial dynamics of social network evolution. In: 23rd International Conference of the System Dynamics Society. vol. 51, p. 61801 (2005)
- [151] Meunier, D., Lambiotte, R., Bullmore, E.T.: Modular and hierarchically modular organization of brain networks. *Frontiers in neuroscience* 4 (2010)
- [152] Michie, D.: Trial and error. *Science Survey, Part 2*, 129–145 (1961)
- [153] Morita, H., Servátka, M.: Group identity and relation-specific investment: An experimental investigation. *European Economic Review* (2012)
- [154] Mucha, P.J., Richardson, T., Macon, K., Porter, M.A., Onnela, J.P.: Community structure in time-dependent, multiscale, and multiplex networks. *Science* 328(5980), 876–878 (2010)
- [155] Nash Jr, J.F.: The bargaining problem. *Econometrica: Journal of the Econometric Society* pp. 155–162 (1950)
- [156] Newcomb, T.M.: The acquaintance process. Holt, Rinehart & Winston (1961)
- [157] Newman, M.E.: The structure and function of complex networks. *SIAM review* 45(2), 167–256 (2003)
- [158] Newman, M.E.: Analysis of weighted networks. *Physical Review E* 70(5), 056131 (2004)
- [159] Newman, M.E.: Finding community structure in networks using the eigenvectors of matrices. *Physical review E* 74(3), 036104 (2006)
- [160] Newman, M.E.: Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103(23), 8577–8582 (2006)
- [161] Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. *Physical review E* 69(2), 026113 (2004)
- [162] Newman, M.: Spectral methods for network community detection and graph partitioning. arXiv preprint arXiv:1307.7729 (2013)
- [163] Nicosia, V., Mangioni, G., Carchiolo, V., Malgeri, M.: Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment* 2009(03), P03024 (2009)
- [164] Nowak, M.A.: Evolutionary dynamics: exploring the equations of life. Harvard University Press (2006)
- [165] Nowak, M.A.: Five rules for the evolution of cooperation. *science* 314(5805), 1560–1563 (2006)
- [166] Nowak, M.A., Tarnita, C.E., Antal, T.: Evolutionary Dynamics in Structured Populations. *Philosophical Transactions of the Royal Society* 365(1537), 19–30 (2010)

- [167] Orman, G.K., Labatut, V., Cherifi, H.: Towards realistic artificial benchmark for community detection algorithms evaluation. *International Journal of Web Based Communities* 9(3), 349–370 (2013)
- [168] Osborne, M.J.: *A course in game theory*. Cambridge, Mass.: MIT Press (1994)
- [169] Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. technical report (1999)
- [170] Palla, G., Barabási, A.L., Vicsek, T.: Quantifying social group evolution. *Nature* 446(7136), 664–667 (2007)
- [171] Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435(7043), 814–818 (2005)
- [172] Park, Y., Song, M.: A genetic algorithm for clustering problems. In: *Proceedings of the Third Annual Conference on Genetic Programming*. pp. 568–575 (1998)
- [173] Păun, G.: Computing with membranes. *Journal of Computer and System Sciences* 61(1), 108–143 (2000)
- [174] Pizzuti, C.: Ga-net: A genetic algorithm for community detection in social networks. In: *Parallel Problem Solving from Nature–PPSN X*, pp. 1081–1090. Springer (2008)
- [175] Pizzuti, C.: Overlapped community detection in complex networks. In: *GECCO*. vol. 9, pp. 859–866 (2009)
- [176] Pizzuti, C.: A Multiobjective Genetic Algorithm to Find Communities in Complex Networks. *IEEE Transactions on Evolutionary Computation* 16(3), 418–430 (2012)
- [177] Pizzuti, C.: Mesoscopic analysis of networks with genetic algorithms. *World Wide Web* pp. 1–21 (2012)
- [178] Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: *Computer and Information Sciences-ISCIS 2005*, pp. 284–293. Springer (2005)
- [179] Prada, R., Paiva, A.: Intelligent virtual agents in collaborative scenarios. In: *Intelligent Virtual Agents*. pp. 317–328. Springer (2005)
- [180] Preuss, M.: Improved topological niching for real-valued global optimization. In: *EvoApplications*. pp. 386–395 (2012)
- [181] Preuss, M., Stoean, C., Stoean, R.: Niching foundations: Basin identification on fixed-property generated landscapes. In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*. pp. 837–844. *GECCO '11*, ACM (2011)
- [182] Rabin, M.: Incorporating fairness into game theory and economics. *The American Economic Review* pp. 1281–1302 (1993)
- [183] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America* 101(9), 2658–2663 (2004)
- [184] Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. *Physical Review E* 74(1), 016110 (2006)
- [185] Rocha, J.B., Mascarenhas, S., Prada, R.: Game mechanics for cooperative games. *ZON Digital Games 2008* pp. 72–80 (2008)
- [186] Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* 105(4), 1118–1123 (2008)

- [187] Sareni, B., Krahenbuhl, L.: Fitness sharing and niching methods revisited. *Evolutionary Computation, IEEE Transactions on* 2(3), 97–106 (1998)
- [188] Seif El-Nasr, M., Aghabeigi, B., Milam, D., Erfani, M., Lameman, B., Maygoli, H., Mah, S.: Understanding and evaluating cooperative games. In: *Proceedings of the 28th international conference on Human factors in computing systems*. pp. 253–262. ACM (2010)
- [189] Sen, O., Sen, S.: Effects of social network topology and options on norm emergence. In: *Coordination, Organizations, Institutions and Norms in Agent Systems V*, pp. 211–222. Springer (2010)
- [190] Sheng, W., Swift, S., Zhang, L., Liu, X.: A weighted sum validity function for clustering with a hybrid niching genetic algorithm. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 35(6), 1156–1167 (2005)
- [191] Shi, C., Yan, Z., Cai, Y., Wu, B.: Multi-objective community detection in complex networks. *Applied Soft Computing* 12(2), 850 – 859 (2012), <http://www.sciencedirect.com/science/article/pii/S1568494611003991>
- [192] Shutters, S.T.: Strong reciprocity, social structure, and the evolution of fair allocations in a simulated ultimatum game. *Computational and Mathematical Organization Theory* 15(2), 64–77 (2009)
- [193] Smith, J.M., Price, G.: The logic of animal conflict. *Nature* 246, 15 (1973)
- [194] Sobolevsky, S., Campari, R., Belyi, A., Ratti, C.: A general optimization technique for high quality community detection in complex networks. *arXiv preprint arXiv:1308.3508* (2013)
- [195] Steinhaeuser, K., Chawla, N.V.: Identifying and evaluating community structure in complex networks. *Pattern Recognition Letters* 31(5), 413–421 (2010)
- [196] Streichert, F., Stein, G., Ulmer, H., Zell, A.: A clustering based niching ea for multimodal search spaces. In: *Artificial Evolution*. pp. 293–304. Springer (2004)
- [197] Sutcliffe, A., Wang, D.: Computational modelling of trust and social relationships. *Journal of Artificial Societies and Social Simulation* 15(1), 3 (2012)
- [198] Sutter, M.: Individual behavior and group membership: Comment. *The American Economic Review* 99(5), 2247–2257 (2009)
- [199] Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press (1998)
- [200] Szell, M., Lambiotte, R., Thurner, S.: Multirelational organization of large-scale social networks in an online world. *Proceedings of the National Academy of Sciences* 107(31), 13636–13641 (2010)
- [201] Szell, M., Thurner, S.: Measuring social dynamics in a massive multiplayer online game. *Social Networks* 32(4), 313–329 (2010)
- [202] Tajfel, H.: Experiments in intergroup discrimination. *Scientific American* 223(5), 96–102 (1970)
- [203] Tajfel, H., Turner, J.C.: An integrative theory of intergroup conflict. *The social psychology of intergroup relations* 33, 47 (1979)
- [204] Tasgin, M., Herdagdelen, A., Bingol, H.: Community detection in complex networks using genetic algorithms. *arXiv preprint arXiv:0711.0491* (2007)
- [205] Thomas, K.W.: Conflict and Conflict Management: Reflections and Update. In: *Journal of Organizational Behavior*. vol. 13, pp. 265–274 (1992)

- [206] Tisue, S., Wilensky, U.: Netlogo: A simple environment for modeling complexity. In: in International Conference on Complex Systems. pp. 16–21 (2004)
- [207] Togelius, J., Preuss, M., Beume, N., Wessing, S., Hagelbäck, J., Yannakakis, G.N., Grappiolo, C.: Controllable Procedural Map Generation via Multiobjective Evolution. *Genetic Programming and Evolvable Machines* 14(2), 245–277 (June 2013)
- [208] Travers, J., Milgram, S.: An experimental study of the small world problem. *Sociometry* pp. 425–443 (1969)
- [209] Tremewan, J.: Group identity and coalition formation: Experiments in a three? player divide the dollar game. Tech. rep., CEPREMAP (2010)
- [210] Turchin, P., Currie, T.E., Turner, E.A., Gavrillets, S.: War, space, and the evolution of old world complex societies. *Proceedings of the National Academy of Sciences* 110(41), 16384–16389 (2013)
- [211] Vassie, K., Morlino, G.: Natural and Artificial Systems: Compare, Model or Engineer? In: Proc. of SAB. pp. 1–11 (2012)
- [212] Vlassis, N.: A concise introduction to multiagent systems and distributed artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 1(1), 1–71 (2007)
- [213] Vogiazou, Y., Eisenstadt, M.: Designing Multiplayer Games to Facilitate Emergent Social Behaviours Online. *Interactive Technology and Smart Education* 2, 113–126 (2005)
- [214] Wang, S., Zou, H., Sun, Q., Zhu, X., Yang, F.: Community detection via improved genetic algorithm in complex network. *Information Technology Journal* 11(3), 384–387 (2012)
- [215] Whitley, D., Rana, S., Heckendorn, R.B.: The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology* 7, 33–48 (1999)
- [216] Wilensky, U.: Netlogo cooperation model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Available from: <http://ccl.northwestern.edu/netlogo/models/Cooperation> [Accessed: November 27th 2007] (1998)
- [217] Wilensky, U.: Netlogo ethnocentrism model. Center for connected learning and computer-based modeling, Northwestern University, Evanston (2006)
- [218] Wilensky, U., Reisman, K.: Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—An embodied modeling approach. *Cognition and instruction* 24(2), 171–209 (2006)
- [219] Xianyu, B.: Social preference, incomplete information, and the evolution of ultimatum game in the small world networks: An agent-based approach. *Journal of Artificial Societies and Social Simulation* 13(2), 7 (2010)
- [220] Xiaodong, D., Cunrui, W., Xiangdong, L., Yanping, L.: Web community detection model using particle swarm optimization. In: *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*. IEEE Congress on. pp. 1074–1079. IEEE (2008)
- [221] Yannakakis, G.N., Hallam, J.: Ranking vs. preference: a comparative study of self-reporting. In: *Affective Computing and Intelligent Interaction*, pp. 437–446. Springer (2011)
- [222] Yannakakis, G.N., Hallam, J.: Rating vs. preference: A comparative study of self-reporting. In: Springer (ed.) *Proceedings of the 2011 Affective Computing and Intelligent Interaction Conference*. Springer (2011)

- [223] Yannakakis, G.N., Martínez, H.P., Jhala, A.: Towards Affective Camera Control in Games. *User Modeling and User-Adapted Interaction* 20, 313–340 (2010)
- [224] Yannakakis, G.N., Togelius, J.: Experience-Driven Procedural Content Generation. *IEEE Transactions on Affective Computing* 2, 147–161 (2011)
- [225] Yannakakis, G.N., Togelius, J., Khaled, R., Jhala, A., Karpouzis, K., Paiva, A., Vasalou, A.: Siren: Towards Adaptive Serious Games for Teaching Conflict Resolution. In: *Proceedings European Conference on Games-Based Learning (ECGBL)*. pp. 412–417. Copenhagen (2010)