

Abstract

This dissertation is divided into two parts. Part I concerns algorithms and data structures on trees or involving trees. Here we study three different problems: efficient binary dispatching in object-oriented languages, tree inclusion, and union-find with deletions.

The results in Part II fall within the heading of approximation algorithms. Here we study variants of the k -center problem and hardness of approximation of the dial-a-ride problem.

Binary Dispatching The *dispatching problem* for object oriented languages is the problem of determining the most specialized method to invoke for calls at run-time. This can be a critical component of execution performance. The *unary dispatching problem* is equivalent to the *tree color problem*. The *binary dispatching problem* can be seen as a 2-dimensional generalization of the tree color problem which we call the *bridge color problem*.

We give a linear space data structure for *binary* dispatching that supports dispatching in logarithmic time. Our result is obtained by employing a dynamic to static transformation technique. To solve the bridge color problem we turn it into a dynamic tree color problem, which is then solved persistently.

Tree Inclusion Given two rooted, ordered, and labeled trees P and T the tree inclusion problem is to determine if P can be obtained from T by deleting nodes in T . The tree inclusion problem has recently been recognized as an important query primitive in XML databases. We present a new approach to the tree inclusion problem which leads to a new algorithm that use optimal linear space and has subquadratic running time or even faster when the number of leaves in one of the trees is small. More precisely, we give three algorithms that all uses $O(n_P + n_T)$ space and runs in $O(\frac{n_P n_T}{\log n_T})$, $O(l_P n_T)$, and $O(n_P l_T \log \log n_T)$, respectively. Here n_S and l_S are the number of nodes and leaves in tree S , respectively.

Union-Find with Deletions A classical union-find data structure maintains a collection of disjoint sets under *makeset*, *union* and *find* operations. In the union-find with deletions problem elements of the sets maintained may be deleted. We give a modification of the classical union-find data structure that supports *delete*, as well as *makeset*

and *union*, in *constant* time, while still supporting *find* in $O(\log n)$ worst-case time and $O(\alpha(n))$ amortized time. Here n is the number of elements in the set returned by the *find* operation, and $\alpha(n)$ is a functional inverse of Ackermann's function.

Asymmetry in k -Center Variants Given a complete graph on n vertices with nonnegative (but possibly infinite) edge costs, and a positive integer k , the *k -center problem* is to find a set of k vertices, called *centers*, minimizing the maximum distance to any vertex and from its nearest center. We examine variants of the *asymmetric k -center problem*.

We demonstrate an $O(\log^* n)$ -approximation algorithm for the asymmetric *weighted k -center problem*. Here, the vertices have weights and we are given a total budget for opening centers. In the *p -neighbor* variant each vertex must have p (unweighted) centers nearby: we give an $O(\log^* k)$ -bicriteria algorithm using $2k$ centers, for small p . In *k -center with minimum coverage*, each center is required to serve a minimum of clients. We give an $O(\log^* n)$ -approximation algorithm for this problem. We also show that the following three versions of the asymmetric *k -center problem* are inapproximable: *priority k -center*, *k -supplier*, and *outliers with forbidden centers*.

Finite Capacity Dial-a-Ride Given a collection of objects in a metric space, a specified destination point for each object, and a vehicle with a capacity of at most k objects, the *finite capacity dial-a-ride problem* is to compute a shortest tour for the vehicle in which all objects can be delivered to their destinations while ensuring that the vehicle carries at most k objects at any point in time. In the *preemptive* version of the problem an object may be dropped at intermediate locations and then picked up later by the vehicle and delivered.

We study the hardness of approximation of the preemptive finite capacity dial-a-ride problem. Let N denote the number of nodes in the input graph, i.e., the number of points that are either sources or destinations. We show that the preemptive Finite Capacity Dial-a-Ride problem has no $\min\{O(\log^{1/4-\varepsilon} N), k^{1-\varepsilon}\}$ -approximation algorithm for any constant $\varepsilon > 0$ unless all problems in NP can be solved by randomized algorithms with expected running time $O(n^{\text{polylog} n})$.