Using a Solution Archive to Enhance Metaheuristics for the Rooted Delay-Constrained Minimum Spanning Tree Problem

Mario Ruthmair, Andreas Hubmer, and Günther R. Raidl

Institute of Computer Graphics and Algorithms
Vienna University of Technology, Vienna, Austria
{ruthmair|raidl}@ads.tuwien.ac.at, andreas.hubmer@tuwien.ac.at

1 Introduction

When designing a communication network with a central server broadcasting information to all the participants of the network, some applications, such as video conferences, require a limitation of the maximal delay from the server to each client. Beside this delay-constraint minimizing the cost of establishing the network is in most cases an important design criterion. This network design problem can be modeled as an $\mathcal{NP}$-hard combinatorial optimization problem called *rooted delay-constrained minimum spanning tree (RDCMST) problem.* The objective is to find a minimum cost spanning tree of a given graph with the additional constraint that the sum of delays along the paths from a specified root node to any other node must not exceed a given delay-bound.

More formally, we are given an undirected graph $G = (V, E)$ with a set V of n nodes, a set E of m edges, a cost function $c : E \rightarrow \mathbb{R}_0^+$, a delay function $d : E \rightarrow \mathbb{R}^+$, a fixed root node $s \in V$, and a delay-bound $B > 0$. An optimal solution to the RDCMST problem is a spanning tree $T = (V, E'),\ E' \subseteq E$, with minimum cost $c(T) = \sum_{e \in E'} c(e)$, satisfying the constraints $\sum_{e \in P(s,v)} d(e) \leq B,\ \forall v \in V$, where $P(s, v)$ denotes the unique path from root s to node v.

Exact approaches to the RDCMST problem have been examined by Gouveia et al. in [1] based on the concept of constrained shortest paths utilized in column generation, Lagrangian relaxation methods and a flow-based reformulation of the problem on layered acyclic graphs. All these methods can only solve small instances with significantly less than 100 nodes to proven optimality in reasonable time when considering complete graphs. A constructive heuristic approach based on Prim's algorithm to find a minimum spanning tree is described by Salama et al. in [6]. In [4] we present a more de-centralized approach by applying the basic concept of Kruskal's minimum spanning tree algorithm to the RDCMST problem. Two metaheuristics based on GRASP and variable neighborhood descent (VND) improve the constructed solution. In [5] we reuse this VND as the local search component in a general variable neighborhood search (GVNS) and a $\mathcal{MAX} - \mathcal{MIN}$ ant system (MMAS). The MMAS mostly obtains the best results in the performed tests.

2 The Solution Archive

One of the basic problems of local search and population-based heuristics is the potentially repeated examination of already visited solutions. To tackle the problem of revisits a complete solution archive can be built efficiently storing solutions and making it possible to derive new unvisited solutions as replacements of detected duplicates. First, promising experiments with solution archives to enhance standard genetic algorithms for binary benchmark problems are presented in [3]. Here we introduce a specialized solution archive for the RDCMST problem.

A feasible solution to the RDCMST problem can also be interpreted as a directed spanning tree (outgoing arborescence) rooted at the source node s. Therefore each node except the root has a unique predecessor corresponding to the origin of the incoming arc. Our solution archive uses a *trie* data structure, which is mostly known from the domain of (language) dictionaries, where a huge number of words has to be stored in a very compact way. In our trie, each node contains an array of $n-1$ references to nodes at the next level, and at each level a dedicated node's predecessor in a given solution decides which pointer to follow. Therefore, a single solution is uniquely represented by $n-1$ trie nodes. In this way, the trie has maximum height $O(n)$, and an insertion operation and a check whether or not a solution is already contained can always be done in time $O(n)$ independently of the number of stored solutions.

Some special adaptions are applied to the basic trie data structure in order to reduce the used space while at the same not increasing access time too much. More specifically, the preprocessed input graph usually is not complete but typically sparse, so the number of array elements of a trie node can be reduced to the actual degree of the corresponding node. To maintain constant access time to an array element a global mapping between adjacent nodes and array indices is stored. Furthermore we can possibly reduce space by allocating the elements of a trie node dynamically, e.g. by a linked list, naturally at the expense of access time. Last but not least, fully explored subtrees are pruned and replaced by an appropriate marker.

The essential aspect which makes our archive approach different to more common simple solution caching strategies as e.g. described in [2] is the provision of a function that derives for each duplicate a typically similar but definitely not yet visited solution. This operation can also be seen as a kind of "intelligent" mutation. Here, one has to consider that only a subset of possible predecessor assignments correspond to a feasible RDCMST solution satisfying both the tree structure and the delay constraints. In general finding an unvisited predecessor assignment in the archive takes $\mathcal{O}(n)$ time which is extended by a feasibility check. In a first step we only consider solutions differing in one edge. There are $\mathcal{O}(n^2)$ such candidates and each one can be tested for feasibility in constant time when using appropriate pre-calculated information. If no new feasible solution can be found the search is successively extended to more distant solutions.

An interesting, although more theoretical side effect of the extension of a metaheuristic by our archive is that the metaheuristic in principle becomes a complete, exact optimization approach with bounded runtime: In each iteration, (at least) one new solution is evaluated, and by the archive it is also efficiently possible to detect when the whole search space has been covered and the search can be terminated.

3 Integration in Metaheuristics

An important question is where to integrate the archive in a (meta-)heuristic process and which metaheuristics can benefit from such an extension at all. At some points the solution diversity may be very high lowering the probability of a revisit, e.g. after shaking the solution randomly. Therefore, the archive just grows very large possibly consuming too much space. At other points revisits typically occur more frequently, e.g. after applying local improvement methods, but due to the structure of the metaheuristic it cannot benefit much from consulting the archive. Detailed analyses of variants of archive-extended GRASP, ant colony optimization, and evolutionary algorithms are currently ongoing work. Generally speaking, the solution archive must be used with caution but has the potential to speed up a metaheuristic significantly.

References

1. Gouveia, L., Paias, A., Sharma, D.: Modeling and Solving the Rooted Distance-Constrained Minimum Spanning Tree Problem. Computers and Operations Research 35(2), 600–613 (2008)
2. Kratica, J.: Improving Performances of the Genetic Algorithm by Caching. Computers and Artificial Intelligence 18(3), 271–283 (1999)
3. Raidl, G.R., Hu, B.: Enhancing Genetic Algorithms by a Trie-Based Complete Solution Archive. In: Cowling, P., Merz, P. (eds.) Evolutionary Computation in Combinatorial Optimisation – EvoCOP 2010. LNCS, vol. 6022, pp. 239–251. Springer (2010)
4. Ruthmair, M., Raidl, G.R.: A Kruskal-Based Heuristic for the Rooted Delay-Constrained Minimum Spanning Tree Problem. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2009. LNCS, vol. 5717, pp. 713–720. Springer (2009)
5. Ruthmair, M., Raidl, G.R.: Variable Neighborhood Search and Ant Colony Optimization for the Rooted Delay-Constrained Minimum Spanning Tree Problem. In: et al., R.S. (ed.) PPSN XI, Part II. LNCS, vol. 6239, pp. 391–400. Springer (2010)
6. Salama, H.F., Reeves, D.S., Viniotis, Y.: The Delay-Constrained Minimum Spanning Tree Problem. In: Blum, C., Roli, A., Sampels, M. (eds.) Proceedings of the 2nd IEEE Symposium on Computers and Communications. pp. 699–703 (1997)