A Multilevel Heuristic for the Rooted Delay-Constrained Minimum Spanning Tree Problem

Martin Berlakovich, Mario Ruthmair, and Günther R. Raidl

Institute of Computer Graphics and Algorithms
Vienna University of Technology, Vienna, Austria
berlmart@a1.net, {ruthmair|raidl}@ads.tuwien.ac.at

1 Introduction

The rooted delay-constrained minimum spanning tree (RDCMST) problem is an NP-hard [3] combinatorial optimization problem. The objective is to find a minimum cost spanning tree in a given graph with cost and delay values assigned to each edge. Additionally, a delay-bound is given limiting the maximum delay allowed for each path between a specified root vertex and any other vertex in the graph. This problem appears in practice for example when designing a distribution network with a guarantee of timely delivery. Another example would be a centralized broadcasting network where the delay-bound represents a quality of service constraint.

More formally, we are given a graph $G = (V, E)$ with a set V of vertices, a set E of edges, a source vertex $s \in V$ and a delay-bound $B > 0$. Additionally a cost function $c : E \to \mathbb{R}^+$ as well as a delay function $d : E \to \mathbb{R}^+$ assign cost and delay values to the edges. In an optimal spanning tree $T = (V, E'),\ E' \subseteq E$, the costs $c(T) = \sum_{e \in E'} c(e)$ are minimal and the delay constraints $\sum_{e \in P(s,v)} d(e) \leq B,\ \forall v \in V$, are satisfied; $P(s, v)$ denotes the unique path between the source s and vertex v.

Gouveia et al. [1] present approaches based on mixed integer programming and constrained shortest paths to derive strong lower bounds and to solve the problem to proven optimality. However, these methods can only be applied to graphs with significantly less than 100 nodes to obtain optimal solutions within a reasonable amount of time in case of complete graphs.

Two construction heuristics for the RDCMST problem have been presented. In [3] a heuristic based on Prim's algorithm to find a minimum spanning tree is described. A more recent approach is the Kruskal-based heuristic introduced in [2], which uses a merge process similar to Kruskal's minimum spanning tree algorithm while considering the delay-constraints. In addition constructed solutions are improved by variable neighborhood descent.

Both heuristics are based on adding edges to the tree trying to minimize the costs in each step. However, the delay is ignored as long as no constraint violation occurs. This can sometimes lead to relatively poor solutions with a rather low potential for further improvement by local search methods. Therefore

we introduce here a heuristic that uses a new measurement for the suitability of edges. This heuristic is based on the multilevel paradigm [4] firstly creating a hierarchy of approximations of the original problem by recursive coarsening. After an initial solution has been found on the coarsest level it is iteratively refined in each level obtaining a feasible solution for the original problem in the end. This multilevel-based construction heuristic does not primarily attempt to create a low cost solution by itself but a promising starting solution for further improvement by local search.

2 Ranking Score

In the above construction heuristics, the inclusion of an edge with low costs is not necessarily cheap regarding the overall solution. If an edge with low costs but high delay is used it can affect the further construction of the solution negatively. The high delay can force a heuristic to use very expensive edges with low delay in order to not violate the delay constraint. Such decisions sometimes create weak solutions corresponding to poor local optima which even good improvement procedures are not able to overcome.

In an attempt to judge how promising an edge is, the ranking score is introduced. It is more likely that an edge with comparatively low costs and low delay is part of an optimal solution than an edge with very low costs but high delay. The ranking score $score(e)$ describes the relative cost in relation to the delay of an edge $e \in E$ in comparison to other edges:

$$score(e) = \left(1 - \frac{r_e^c - 1}{|E|}\right) \cdot \left(1 - \frac{r_e^d - 1}{|E|}\right) \tag{1}$$

Here, $r_e^c \in \{1, \ldots, |E|\}$ and $r_e^d \in \{1, \ldots, |E|\}$ represent the cost and delay ranks of edge e, which are obtained by sorting the edges according to costs and delays, respectively. The ranking score of a vertex $v \in V$ is the sum of the scores of all incident edges.

3 Ranking-Based Multilevel Heuristic

The ranking-based multilevel heuristic (RBMH) follows a multilevel strategy. In each level a number of vertices, including the source vertex, is selected as so-called supervertices. The remaining vertices are connected directly to these supervertices creating multiple subtrees in each level.

The RBMH uses the ranking scores to determine which vertices and edges are used. Vertices with high ranking scores are supposed to have many or at least high quality incident edges. This makes them more promising to be the root of a subtree in an optimal solution. After choosing a set of these supervertices all remaining vertices are connected to them using the edges with the highest ranking scores satisfying the delay-constraints. The subtrees with supervertices as their roots represent the vertices in the next level. This process is continued

Table 1. Comparison of RBMH and Kruskal-based heuristic, applied on a set of 30 random instances (complete graphs) with 500 and 1000 vertices (B: delay-bound, $\overline{c}$: average final objective values, σ: standard deviations, $t[s]$: runtime in seconds). Tests have been executed on Xeon E5540 processors with about 3 GB RAM per core.

	500 vertices						1000 vertices					
	RBMH + VND			Kruskal + VND			RMBH + VND			Kruskal + VND		
B	$\overline{c}$	σ	$t[s]$	$\overline{c}$	σ	$t[s]$	$\overline{c}$	σ	$t[s]$	$\overline{c}$	σ	$t[s]$
10	4634	225	1.99	**4557**	205	1.45	5290	212	9.33	**5171**	215	7.52
30	**1530**	85	4.42	1554	88	4.37	**1871**	71	23.55	1884	55	20.04
50	**1010**	64	7.99	1042	56	6.22	**1334**	50	33.81	1373	44	32.93
75	**765**	33	10.90	800	37	9.44	**1113**	32	57.75	1146	32	51.42
100	**642**	28	13.64	687	44	12.75	**1038**	12	75.79	1070	32	62.76
150	**547**	11	16.71	587	36	12.25	**1005**	4	74.13	1022	24	57.96
200	**522**	6	13.55	545	27	10.90	**1001**	2	74.58	1008	16	37.65

until only one supervertex, the source vertex, remains corresponding to a feasible solution for the original problem. The RBMH runs in $\mathcal{O}(|E| \log |E| + |V|^2)$ time.

4 Preliminary Results

The RBMH has been tested on large random instances from [2] with complete graphs of 500 and 1000 vertices and the results are compared to those of the Kruskal-based heuristic. In all cases the variable neighborhood descent (VND) of [2] is applied to improve constructed solutions.

The exemplary results listed in Table 1 document that the RBMH combined with VND produces on average better results than the Kruskal-based heuristic using the same improvement with the exception of tight delay-bounds. The RBMH solution usually has higher costs as corresponding solutions of previous construction heuristics but at the same time offers higher potential for later improvement phases. Concerning the runtime the RBMH tests are slower since more time is spent in the improvement phase.

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. 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)
3. 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 – ISCC '97. pp. 699–703 (1997)
4. Walshaw, C.: Multilevel refinement for combinatorial optimisation problems. Annals of Operations Research 131(1), 325–372 (2004)