Cluster-Based (Meta-)Heuristics for the Euclidean Bounded Diameter Minimum Spanning Tree Problem

Martin Gruber and Günther R. Raidl

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

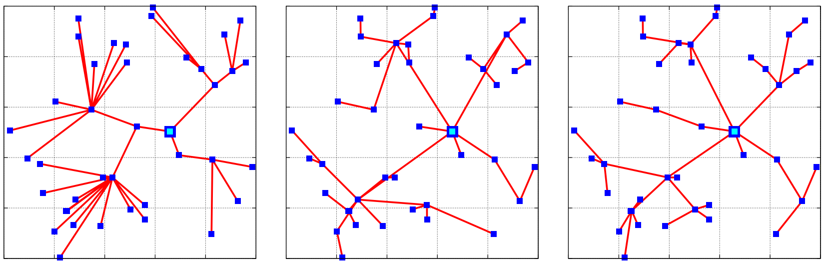
1 Introduction

The *bounded diameter minimum spanning tree* (BDMST) problem is a combinatorial optimization problem appearing in applications such as wire-based communication network design when quality of service is of concern, in ad-hoc wireless networks, and also in the areas of data compression and distributed mutual exclusion algorithms.

The goal is to identify a tree of minimum costs connecting all nodes of a network where the number of links between any two nodes is restricted by a constant D, the diameter. More formally, we are given an undirected connected graph $G = (V, E)$ with node set V and edge set E and associated costs $c_e \geq 0$, $\forall e \in E$. We seek a spanning tree $T = (V, E_T)$ with edge set $E_T \subseteq E$ whose diameter does not exceed $D \geq 2$, and whose total costs $\sum_{e \in E_T} c_e$ are minimal. This task can also be seen as choosing the *center* (one single node if D is even or an edge in the odd diameter case) and building a height-restricted tree where the unique path from this center to any node of the tree consists of no more than $H = \lfloor \frac{D}{2} \rfloor$ edges. This problem is known to be NP-hard for $4 \leq D < |V| - 1$ [1].

To solve this problem to proven optimality there exist various integer linear programming (ILP) approaches like hop-indexed multi-commodity network flow models [2, 3] or a Branch&Cut formulation based on a more compact model but strengthened by a special class of cutting planes [4]. They all have in common that they are only applicable to relatively small instances, i.e. significantly less than 100 nodes when dealing with complete graphs. For larger instances, metaheuristics have been developed, including evolutionary algorithms [5, 6], a variable neighborhood search, and an ant colony optimization [7] which is currently the leading metaheuristic to obtain high quality solutions.

In contrast to the large variety of metaheuristic approaches the number of simple and fast construction heuristics that can also be applied to very large instances is limited. They are primarily based on Prim's MST algorithm [8] and grow a height restricted tree from a chosen center, for example the center based tree construction (CBTC) [9]. This works reasonably well on instances with random edge costs, but on Euclidean instances this leads to a backbone (the edges near the center) of relatively short edges where the majority of the nodes



(a) CBTC (8.284). (b) RTC (5.725). (c) Optimum (5.195).

Fig. 1. A diameter constrained tree $D = 6$ constructed using (a) the CBTC heuristic, compared to (b) RTC (best solution from 100 runs) and (c) the optimal solution (complete, Euclidean graph with 40 nodes distributed randomly in the unit square, the corresponding objective values are given in parenthesis).

has to be connected to the backbone via relatively long edges, see the example in Fig. 1(a). On the contrary, the optimal solution for this instance shown in Fig. 1(c) demonstrates that the backbone should consist of a few longer edges to span the whole area so the large number of remaining nodes can be connected as leaves by much cheaper edges. In a pure greedy construction heuristic this observation is difficult to realize. In the randomized tree construction approach (RTC, Fig. 1(b)) not the cheapest but a random node is connected to the tree in the construction phase using the shortest available edge. Thus at least the possibility to include longer edges into the backbone at the beginning of the algorithm is increased. For Euclidean instances RTC is so far the best choice to quickly create a first reasonable solution as basis for exact or metaheuristic approaches.

The aim of this work is to introduce a new construction heuristic for the Euclidean BDMST problem, especially suited for very large instances. It is based on a hierarchical clustering that guides the algorithm to find a good backbone.

2 Cluster-Based Heuristics

The cluster-based construction heuristic can be divided into three steps: Creating a hierarchical clustering (dendrogram) based on the Euclidean distances, deriving a cluster representation (CR) of the BDMST from this dendrogram, and finding for each cluster in the CR a good (sub-) center node.

For the first step complete linkage agglomerative cluster [10] is used, i.e. two clusters A and B are merged when $\max\{c_{a,b} : a \in A, b \in B\}$ is minimal over all pairs of clusters. The agglomeration starts with each node being a separate cluster, and stops when all nodes are merged within one single cluster, the root of the dendrogram.

The dendrogram itself, a binary tree, cannot directly act as cluster representation of the BDMST since in general it will violate the height restriction, see

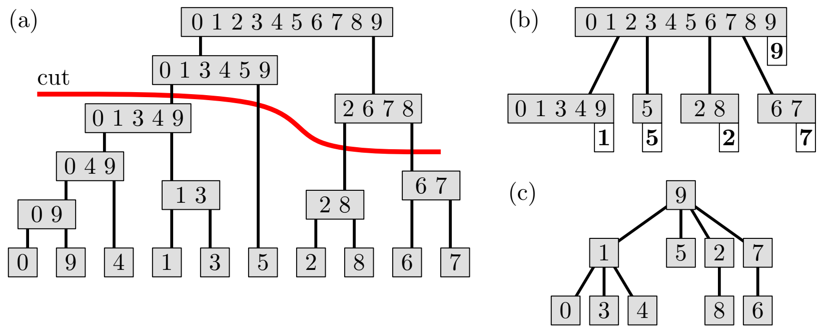


Fig. 2. Dendrogram (a), cluster representation (b) of the BDMST (centers printed in bold), and the resulting diameter constrained tree with $D = 4$ (c).

Fig. 2(a). Therefore, some of the nodes in the dendrogram have to be merged to finally get a tree of height $H - 1$, the CR of the BDMST as illustrated in Fig. 2(b) for a diameter of $D = 4$. For the quality of the resulting tree this merging of dendrogram nodes, i.e. where to *cut* through the dendrogram to reduce the height, is a crucial step. Initial cutting position can easily be calculated but are refined using a greedy randomized adaptive search procedure (GRASP).

Finally, from the CR a BDMST has to be derived by identifying for each (sub-)cluster the best possible center to act as a node of the backbone. This can be done heuristically in a greedy fashion based on rough cost estimations for each cluster, followed by a local improvement step. We also designed a dynamic programming approach to solve this subproblem exactly, however, the selection of possible center nodes for each cluster has to be limited to handle the complexity. A variation of this dynamic program uses correction values to approximate the effect on the tree costs when utilizing a node for the backbone instead of connecting it as a leaf.

Deriving a good tree from a height restricted clustering can also directly be used to locally improve solutions obtained from another (meta-)heuristic since each valid solution also defines an hierarchical clustering of all nodes.

3 Preliminary Results

First results obtained from the cluster construction heuristic on large Euclidean instances with 1000 nodes are promising. Especially when the diameter bound is extremely tight the costs of the computed trees are near the half of the best tree constructed with RTC when using the same time limit, see Table 1 for details. When applying a strong variable neighborhood descend (VND) as proposed in [7], the differences between the construction heuristics flatten, but still the BDMSTs derived from a cluster heuristic solutions are in general of higher quality. On instances with small diameter bounds these trees – computed in a few seconds – can also compete with results from the ACO with computation times of one hour and more.

Table 1. Objective values of CBTC, RTC and the cluster construction heuristic utilizing the enhanced dynamic programming approach (CL), computed on a complete Euclidean instance with 1000 nodes which are distributed randomly in the unit square. The running time t(CL) was used as stopping criterion for CBTC and RTC (without VND). For comparison, the objective values of the ACO presented in [7] after one hour are listed.

D	CBTC	RTC	CL	t(CL)	RTC+VND	CL+VND	ACO
4	333.1778	144.3722	71.4471	2.84	66.0137	65.7060	66.5841
6	319.1373	82.3505	55.8774	2.73	41.6183	41.3944	42.2784
8	298.7454	54.3798	44.2476	2.69	35.2035	34.3103	34.8716
10	271.3976	41.8200	38.0715	3.13	32.1746	31.5035	31.2361
20	205.1811	31.4913	30.2756	2.97	27.2665	26.3760	24.5238

References

1. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (1979)
2. Gouveia, L., Magnanti, T.L.: Network flow models for designing diameter-constrained minimum spanning and Steiner trees. Networks **41**(3) (2003) 159–173
3. Gouveia, L., Magnanti, T.L., Requejo, C.: A 2-path approach for odd-diameter-constrained minimum spanning and Steiner trees. Networks **44**(4) (2004) 254–265
4. Gruber, M., Raidl, G.R.: (Meta-)heuristic separation of jump cuts in a branch&cut approach for the bounded diameter minimum spanning tree problem (2008) submitted to a special issue on Matheuristics of Operations Research/Computer Science Interface Series, Springer.
5. Raidl, G.R., Julstrom, B.A.: Greedy heuristics and an evolutionary algorithm for the bounded-diameter minimum spanning tree problem. In Lamont, G., et al., eds.: Proc. of the ACM Symposium on Applied Computing, ACM Press (2003) 747–752
6. Singh, A., Gupta, A.K.: Improved heuristics for the bounded-diameter minimum spanning tree problem. Soft Computing – A Fusion of Foundations, Methodologies and Applications **11**(10) (2007) 911–921
7. Gruber, M., van Hemert, J., Raidl, G.R.: Neighborhood searches for the bounded diameter minimum spanning tree problem embedded in a VNS, EA, and ACO. In Keijzer, M., et al., eds.: Proc. of the Genetic and Evolutionary Computation Conference 2006. Volume 2. (2006) 1187–1194
8. Prim, R.C.: Shortest connection networks and some generalizations. Bell System Technical Journal **36** (1957) 1389–1401
9. Julstrom, B.A.: Greedy heuristics for the bounded-diameter minimum spanning tree problem. Technical report, St. Cloud State University (2004)
10. Jain, A., Murty, M., P.J.Flynn: Data clustering: a review. ACM Computing Surveys (CSUR) **31** (1999) 264–323