Layered Graph Models and Exact Algorithms for the Generalized Hop-Constrained Minimum Spanning Tree Problem

Markus Leitner*
Department of Statistics and Operations Research,
University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria
markus.leitner@univie.ac.at

January 12, 2015

Abstract

This article studies the generalized hop-constrained minimum spanning tree problem (GHMSTP) which has applications in backbone network design subject to quality-of-service constraints that restrict the maximum number of intermediate routers along each communication path. Different possibilities to model the GHMSTP as an integer linear program and strengthening valid inequalities are studied. The obtained formulations are compared theoretically, i.e., by means of their linear programming relaxation. In addition, branch-and-cut approaches based on these formulations are developed and compared in a computational study.

Keywords. Integer programming, Branch-and-cut, Generalized network design, Hop-constraints, Layered graph

1 Introduction

Generalized network design problems (GNDPs) with applications in backbone network design have been studied in detail in the recent years (see, e.g., [4, 5, 14, 22] and the references therein) and exact as well as heuristic solution approaches for them have been proposed. Given an undirected graph $G = (V, E)$ with nonnegative edge costs c_e, $\forall e \in E$, GNDPs are characterized by the fact that the node set V is partitioned into a set of K disjoint clusters V_k, $1 \leq k \leq K$, $V = \bigcup_{k=1}^{K} V_k$, $V_k \cap V_l = \emptyset$, $1 \leq k \neq l \leq K$. A feasible solution $S = (P, T)$, $P \subset V$, $T \subset E$, is a subgraph of G which selects precisely one node from each cluster and connects these nodes according to the requirements of the concrete GNDP at hand, e.g., by a spanning tree in case of the generalized minimum spanning tree problem (GMSTP), cf. [5, 7, 21, 23], or a Hamiltonian cycle in the generalized traveling salesman problem (GTSP), cf. [6, 17, 27]. The usual objective is to identify a solution yielding minimum edge costs $\sum_{e \in T} c_e$.

Applications of GNDPs arise for example in the design of backbone networks. Thereby, clusters represent areas (e.g., cities) each of which is connected via a local network and nodes model possibilities to interconnect these local networks. In case, failure tolerance by means of installing redundant connections does not need to be considered this application domain naturally leads to aforementioned GMSTP for which a variety of exact and heuristic solution approaches have been proposed in the recent scientific literature, see, e.g., [5, 14] and the references therein.

In this work, we study a new variant of the GMSTP which additionally incorporates side constraints that allow to limit the communication delay of the resulting network in centralized networks. Several studies have suggested that reducing the number of intermediate routers or edges along a communication path increases availability and reliability of a network as well as the quality of the transmitted signal, cf. [3]. It has also

*Supported by the Austrian Science Fund (FWF) under grant I892-N23.

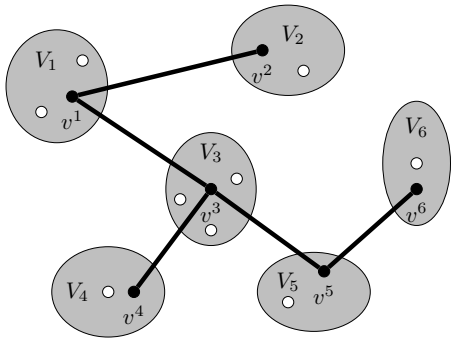


Figure 1: An illustrative instance of the GHMSTP with six clusters and an exemplary feasible solution $S = (P, T)$ with $P = \{v^1, \ldots, v^6\}$ with $v^i \in V_i$, $1 \leq i \leq 6$, for $H \geq 3$. Edges not contained in the solution are not shown.

been observed that such bounds reduce the possibility of failures without the need to consider quite costly alternatives such as node- or edge-disjoint networks.

Therefore, bounds on the total communication delay, are typically ensured via enforcing an upper bound on the number of hops from a predefined distribution node to each other node (in centralized networks) or between any two nodes (i.e., a bound on the diameter of the resulting network) in decentralized networks. Consequently, in particular the hop-constrained minimum spanning tree problem (HMSTP) has been studied exhaustively in the scientific literature, see, e.g., [3, 8, 9, 11, 10, 28]. The current state-of-the-art approach has been proposed by Gouveia et al. [11] who proposed a branch-and-cut algorithm based on a transformation of the HMSTP to a Steiner arborescence problem on an extended, so-called layered graph a concept which will be used intensively in this article.

The *Generalized Hop-Constrained Minimum Spanning Tree Problem (GHMSTP)* captures aforementioned aspects for the design of centralized backbone networks and is formally defined as follows. Given is an undirected graph $G = (V, E)$ with costs $c_e \geq 0$ associated to each edge $e \in E$ whose node set V is partitioned into K mutually disjoint clusters $\emptyset \neq V_k \subset V$, $1 \leq k \leq K$, $\bigcup_{k=1}^{K} V_k = V$, $V_k \cap V_l = \emptyset$, $1 \leq k \neq l \leq K$. Let furthermore without loss of generality V_1 be the *root cluster* and $H \in \mathbb{N}$ be the given hop limit. A feasible solution is a cycle-free, connected subgraph $S = (P, T)$, $P \subset V$, $T \subset E$, of G spanning exactly one node from each cluster (i.e., $P = \{v^1, v^2, \ldots, v^K\}$, $v^k \in V_k$, $1 \leq k \leq K$, that satisfies all hop constraints, i.e., the unique path between the chosen root node $v^1 \in V_1$ and any other chosen node consists of at most H edges. The objective is to identify a feasible solution $S^* = (P^*, T^*)$ yielding minimum costs $\sum_{e \in T^*} c_e$. An exemplary instance together with a feasible solution for $H \geq 3$ is given in Figure 1.

The GHMSTP is NP-hard in general since it contains both the hop-constrained minimum spanning tree problem (see, [2, 8]) as well as the GMSTP as the special case when $|V_i| = 1$, $i = 1, \ldots, K$, and when $H \geq K - 1$, respectively. We note that for $H = 1$ the problem can, however, be solved in polynomial time by the following algorithm: First select a root node $u \in V_1$. Then, for each remaining cluster V_i, $2 \leq i \leq K$, choose a node $v \in V_i$ that can be connected to the root u at minimum cost, i.e., $v = \text{argmin}_{w \in V_i} c_{uw}$ and add the edge $\{u, v\}$ to the partial solution. By iterating over all possible root nodes $u \in V_1$ and adopting the overall cheapest solution, one obtains an optimal solution to the considered instance of the GHMSTP for $H = 1$.

Scientific contribution Besides introducing the new problem, the main results of this paper are: a) Integer linear programming (ILP) models (presented in Sections 2 and 3) that successfully combine ideas from related generalized network design problems with those from hop-constrained network design problems; b) A theoretical study comparing all proposed formulations by means of their linear programming relaxation values in Section 4; c) The development and computational comparison of branch-and-cut approaches based

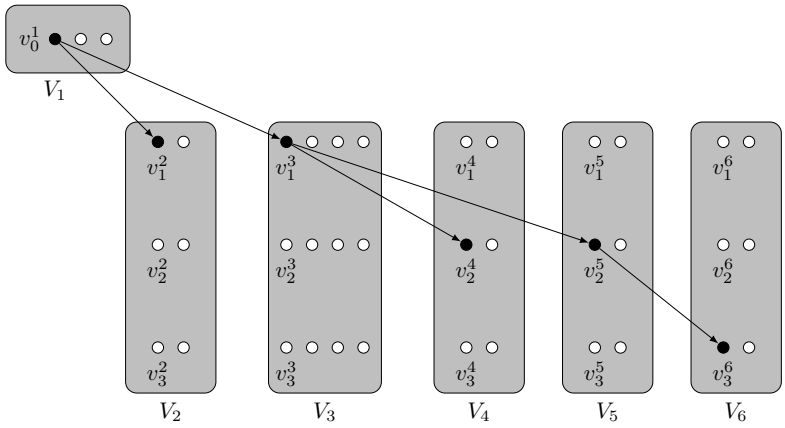


Figure 2: Embedding of the solution given in Figure 1 on layered graph G_{L} for $H = 3$. Arcs not part of the solution are not shown and nodes not selected are not labeled.

on these formulations in Section 5. Our results show that a variant based on the concept of a layered cluster graph (see Section 3) outperforms all other considered algorithms.

Notation In all formulations considered in the following, a solution to the GHMSTP is represented as an outgoing arborescence rooted at a node from V_1 which spans all selected nodes. Therefore, we utilize arc set $A = \{(u, v) \mid \{u, v\} \in E, u \in V_1\} \cup \{(u, v) \mid \{u, v\} \in E, u, v \notin V_1\}$. For each edge $\{u, v\}$ adjacent to a node $u \in V_1$ an arc (u, v) emanating from u is considered while two oppositely directed arcs are added for all remaining edges. Given a set of variables $\mathbf{f}$ defined on all members of set S, and a subset $S' \subseteq S$, we will use notation $f[S'] = \sum_{i \in S'} f_i$. Furthermore, for subsets $U \subset V$ and $W \subset V$ we will denote by $\delta(U, W) = \{(u, w) \in A \mid u \in U, w \in W\}$ the set of arcs from U to W and by $\delta^-(W) = \{(u, v) \in A \mid u \notin W, v \in W\}$ the ingoing cutset for W. Analogous notation will be used to denote ingoing cutsets for subsets of nodes of graphs different from G. Finally, we will use $C = \{1, \dots, K\}$ to denote the set of clusters and $V(w) = \{u \in V_i \mid w \in V_i\}$ to refer to all nodes that are contained in the same cluster than $w \in V$.

2 Layered Graph Models

The formulations introduced in this section make use of layered graph $G_{\mathrm{L}} = (V_{\mathrm{L}}, A_{\mathrm{L}})$ which encodes the distance of nodes and arcs from the chosen root node into its structure. Node set V_{L} contains copies u_0 of all nodes u from the root cluster V_1 at layer zero and copies u_h of all other nodes $u \in V \setminus V_1$ on layers $1 \leq h \leq H$. Arcs (u_h, v_{h+1}) are included in arc set A_{L} whenever an arc (u, v) exists in arc set A. Thus, G_{L} which is formally defined by $V_{\mathrm{L}} = \{u_0 \mid u \in V_1\} \cup \{u_h \mid u \in V \setminus V_1, 1 \leq h \leq H\}$ and $A_{\mathrm{L}} = \{(u_h, v_{h+1}) \mid (u, v) \in A\}$ is acyclic and does not contain paths longer than the given hop limit H. Figure 2 shows the layered graph representation of the solution given in Figure 1 for $H = 3$. We note that, in particular for sparse input graphs and when using an appropriate construction or preprocessing routine, node set V_{L} is typically not complete (i.e., some nodes may not exist at all layers). To simplify notation, we nevertheless assume a complete node set in the following.

Formulation (1)–(8) to which we will refer to as L_x makes use of node decision variables $z_i \in \{0, 1\}$, $\forall i \in V$, indicating whether or not a node is selected as well as arc design variables $x_{uv} \in \{0, 1\}$, $\forall (u, v) \in A$, denoting whether or not arc (u, v) is contained in the directed solution. Furthermore, layered arc design variables $X_{uv}^h \in \{0, 1\}$, $\forall (u_h, j_{h+1}) \in A_{\mathrm{L}}$, are used to define the positions of each used arc in the arborescence, i.e., $X_{uv}^h = 1$ if and only if arc (u, v) is used and node u is at distance h from the root node.

$$\min \quad \sum_{(u,v)\in A} c_{uv}x_{uv} \tag{1}$$

$$\text{s.t.} \quad z[V_i] = 1 \qquad \forall i \in C \tag{2}$$

$$x[\delta^-(u)] = z_u \qquad \forall u \in V \setminus V_1 \tag{3}$$

$$x[\delta^-(W)] \geq z_u \qquad \forall W \subset V \setminus V_1, u \in W \tag{4}$$

$$x_{uv} + x_{vu} \leq z_u \qquad \forall u \in V,\ \forall \{u,v\} \in E \tag{5}$$

$$\sum_{h=0}^{H-1} X^h_{uv} = x_{uv} \qquad (u,v) \in A \tag{6}$$

$$\sum_{(u_{h-1},v_h)\in A_{\mathrm{L}}, u\notin V_i} X^{h-1}_{uv} \geq \sum_{(v_h,w_{h+1})\in A_{\mathrm{L}}, w\in V_i} X^h_{vw} \qquad \forall v_h \in V_{\mathrm{L}}, \forall i \in C, h \geq 1, \delta(\{v\}, V_i) \neq \emptyset \tag{7}$$

$$(\mathbf{x}, \mathbf{z}, \mathbf{X}) \in \{0,1\}^{|A|+|V|+|A_{\mathrm{L}}|} \tag{8}$$

The objective function (1) minimizes the costs of the obtained solution while equations (2) ensure that precisely one node from each cluster will be selected. Indegree constraints (3) link arc and node variables and ensure that precisely one ingoing arc is selected for each chosen node that is not contained in the root cluster. Inequalities (4) are directed connectivity constraints that enforce connectivity of the solution. Together with inequalities (5) – which ensure that the two nodes adjacent to a selected arc are selected as well – they ensure that there exists a path from the selected root node to each other selected node. Observe, that (1)–(5) resembles a model for the GMSTP previously used by Myung et al. [20] as well as by Feremans [4].

Equations (6) link layered arc to arc design variables. Together with (3) they also ensure that for each arc $(u,v) \in A$ at most one of its layered copies (u_h, v_{h+1}) can be used and that the corresponding target node j has to be selected as well. Finally, inequalities (7) make sure that an arc emanating from node v_h, $1 \leq h \leq H$, to a node in cluster i may only be used if an ingoing arc originating at a cluster different from i is used as well. Since G_{L} is acyclic this polynomial number of constraints ensure connectivity of the solution obtained on the layered graph, cf. [26].

Though inequalities (7) are sufficient to ensure connectivity, it is well known from similar layered graph reformulations [11, 25] that one can strengthen such a formulation by additionally considering connectivity cuts (9) defined on G_{L}. We will use L^c_x to refer to the stronger variant of model L_x augmented by these constraints.

$$X[\delta^-(W)] \geq z_i \qquad W \subseteq V_{\mathrm{L}} \setminus \{u_0 : u \in V_1\}, \{i_h, 1 \leq h \leq H\} \subseteq W \tag{9}$$

We also observe that connectivity constraints (4) can be strengthened if all nodes from at least one cluster are contained in set W, yielding

$$x[\delta^-(W)] \geq 1 \qquad W \subseteq V \setminus V_1, \exists k \in C \setminus \{1\} : V_k \subseteq W \tag{10}$$

to which we will refer to as *cluster cuts* while (4) will be called *node cuts*. Their validity can be easily seen since a path from the selected root node to the node selected in cluster k must exist in every feasible solution. The models additionally containing (10) will be called L^+_x and $\mathrm{L}^{c,+}_x$, respectively.

A similar lifting can be applied to connectivity cuts (9) on G_{L} if the copies of all nodes of at least one cluster are contained in set W yielding *layered cluster cuts* (11). In the following, $\mathrm{L}^{c,++}_x$ will be used to refer to model $\mathrm{L}^{c,+}_x$ additionally containing these constraints.

$$X[\delta^-(W)] \geq 1 \qquad W \subset V_{\mathrm{L}} \setminus \{u_0 : u \in V_1\}, \quad \exists k \in C \setminus \{1\} : \{u_h : u \in V_k, 1 \leq h \leq H\} \subset W \tag{11}$$

For the GMSTP, Pop [22] considered a reduced graph modeling potential inter-cluster connections to which we will refer to as *cluster graph* in the following. Given an instance to a GNDP with graph $G = (V, E)$,

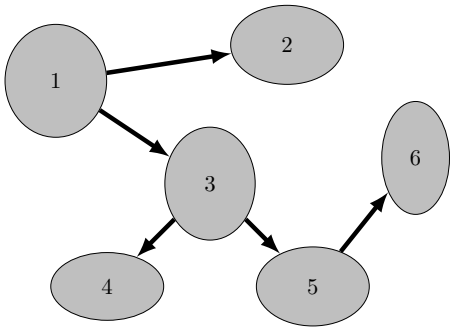


Figure 3: Cluster graph G_{C} corresponding to the instance and the solution given in Figure 1. Arcs not contained in the solution are not shown.

node clusters V_i, $1 \leq k \leq K$, and the associated arc set A, the corresponding *cluster graph* $G_{\mathrm{C}} = (V_{\mathrm{C}}, A_{\mathrm{C}})$ is a directed graph containing one node for each cluster in G, i.e., $V_{\mathrm{C}} = \{1, 2, \ldots, K\}$. *Cluster arcs* (i, j) between *cluster nodes* $i, j \in V_{\mathrm{C}}$ exist if there exists at least one arc in A connecting cluster i to cluster j, i.e., $A_{\mathrm{C}} = \{(i, j) \mid (u, v) \in A, u \in V_i, v \in V_j\}$, see Figure 3 for an example.

Pop [22] used a compact formulation for describing a spanning arborescence on this cluster graph together with a polynomial number of linking constraints to select nodes and arcs from the original graph to derive a valid formulation for the GMSTP. Our next model to which we will refer to as L_y is based on these ideas but uses a cutset based formulation to model the arborescence on the cluster graph. Thus, our formulation needs significantly less variables ($|V| + |A| + |A_{\mathrm{C}}|$ rather than $|V| + |A| + K \cdot |A_{\mathrm{C}}|$) but comes at the price of an exponential number of (efficiently separable) constraints. Besides the variables introduced above we will also use variables $y_{ij} \in \{0, 1\}$, $\forall (i, j) \in A_{\mathrm{C}}$, that indicate whether or not a (directed) connection between two clusters is used or not.

$$\min \quad \sum_{(u,v)\in A} c_{uv} x_{uv} \tag{12}$$

$$\text{s.t.} \quad z[V_i] = 1 \qquad \forall i \in C \tag{13}$$

$$y[\delta^-(i)] = 1 \qquad i \in V_{\mathrm{C}} \setminus \{1\} \tag{14}$$

$$y[\delta^-(W)] \geq 1 \qquad W \subseteq V_C \setminus \{1\},\ W \neq \emptyset \tag{15}$$

$$x[\delta(V_i, V_j)] = y_{ij} \qquad \forall (i, j) \in A_{\mathrm{C}} \tag{16}$$

$$x[\delta^-(u)] = z_u \qquad \forall u \in V \setminus V_1 \tag{17}$$

$$x_{uv} + x_{vu} \leq z_u \qquad \forall u \in V,\ \forall \{u, v\} \in E \tag{18}$$

$$\sum_{(u,v)\in A | u\in V_i, j\in V_j} \sum_{h=0}^{H-1} X_{uv}^h = y_{ij} \qquad \forall (i, j) \in A_{\mathrm{C}} \tag{19}$$

$$\sum_{(u_{h-1},v_h)\in A_{\mathrm{L}}, u\notin V_i} X_{uv}^{h-1} \geq \sum_{(v_h,w_{h+1})\in A_{\mathrm{L}}, w\in V_i} X_{vw}^h \qquad \forall v_h \in V_{\mathrm{L}}, \forall i \in C, h \geq 1, \delta(\{v\}, V_i) \neq \emptyset \tag{20}$$

$$(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{X}) \in \{0, 1\}^{|A|+|A_{\mathrm{C}}|+|V|+|A_{\mathrm{L}}|} \tag{21}$$

Equations (14) are indegree constraints on the cluster graph ensuring that each cluster except the root cluster has exactly one incoming cluster arc. Inequalities (15) are the directed cutset constraints ensuring connectivity of the global (i.e., cluster-wise) structure of the solution and constraints (16) are the linking constraints between cluster arc and standard arc variables. To reduce the number of constraints we use constraints (19) linking layered arc variables to cluster arc variables rather than inequalities (6) which

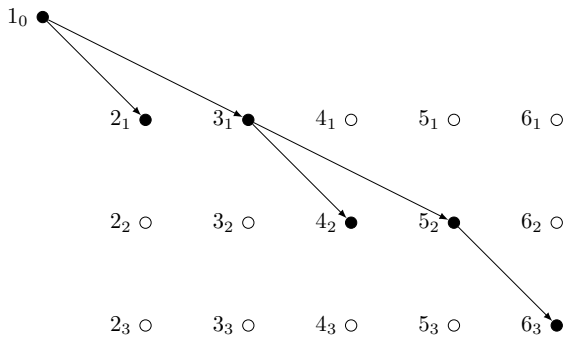


Figure 4: A layered cluster graph G_{CL} corresponding to the instance given in Figure 1 for $H = 3$. Arcs not corresponding to those contained in the exemplary solution from Figure 1 are not shown.

would be another (disaggregated) option to obtain a correct model. A valid formulation for the GHMSTP is obtained together with constraints (13), (17), (18), and (20) which have been discussed above for formulation L_x. Variant L_y^c is obtained by adding layered cuts (9) while $\mathrm{L}_y^{c,+}$ contains layered cuts (9) and layered cluster cuts (11). We note, that while we introduce $|A_{\mathrm{C}}|$ additional variables (compared to L_x), the number of cutset constraints decreases significantly and it remains to analyze the theoretical and computational consequences in Sections 4 and 5, respectively.

3 Layered Cluster Graph Models

While layered graph formulations similar to the ones introduced in the previous section have proven to be quite efficient in solving small and medium instances of different network design problem (see, e.g., [11, 12, 13, 19]) their main drawback is the quite large number of variables involved making them impractical for large scale instances.

In order to significantly reduce the number of variables while aiming to keep the main advantage (i.e., a tight linear programming (LP) relaxation due to implicitly ensuring the hop limit) the two ILP models introduced in this section are based on the idea of a *layered cluster graph* in which only global (i.e., inter-cluster connections) are modeled on the layered graph. Formally, such a *layered cluster graph* $G_{\mathrm{CL}} = (V_{\mathrm{CL}}, A_{\mathrm{CL}})$ is defined by its node set $V_{\mathrm{CL}} = \{1_0\} \cup \{i_h \mid 2 \leq i \leq K,\ 1 \leq h \leq H\}$ and its arc set $A_{\mathrm{CL}} = \{(i_h, j_{h+1}) \mid i_h, j_{h+1} \in V_{\mathrm{CL}}, (i,j) \in A_{\mathrm{C}}\}$, see Figure 4.

Our first such formulation to which we will refer to as CL_x uses variables $Y_{ij}^h \in \{0,1\}$, $\forall (i_h, j_{h+1}) \in A_{\mathrm{CL}}$, which express whether or not the path from the root cluster to cluster j consists of precisely $h+1$ arcs and cluster i is the predecessor of cluster j along this path. In addition node design variables $z_u \in \{0,1\}$, $\forall u \in V$, and arc design variables $x_{uv} \in \{0,1\}$, $\forall (u,v) \in A$, with the same meaning as in the previous models are used.

$$\min \quad \sum_{(u,v)\in A} c_{uv} x_{uv} \tag{22}$$

$$\text{s.t.} \quad (2) - (5)$$

$$\sum_{h=1}^{H} Y[\delta^-(i_h)] = 1 \qquad \forall i \in C \setminus \{1\} \tag{23}$$

$$\sum_{h=0}^{H-1} Y_{ij}^h = x[\delta(V_i, V_j)] \qquad \forall (i,j) \in A_{\mathrm{C}} \tag{24}$$

$$\sum_{(i_{h-1},j_h)\in A_{\mathrm{CL}},i\neq w} Y_{ij}^{h-1} \geq Y_{jw}^{h} \qquad \forall (j_h, w_{h+1}) \in A_{\mathrm{CL}},\ h \geq 1 \tag{25}$$

$$(\mathbf{x}, \mathbf{z}, \mathbf{Y}) \in \{0,1\}^{|A|+|V|+|A_{\mathrm{CL}}|} \tag{26}$$

Constraints (2)–(5) ensure that an arborescence spanning exactly one node from each cluster is built, while the remaining constraints are used to model the corresponding arborescence on the layered cluster graph G_{L} and thus guarantee that the hop constraint is met. Equations (23) ensure that for each cluster exactly one ingoing arc on V_{CL} is selected (among all layers) and additionally ensure that one cannot select two layered arcs corresponding to the same original arc. Equations (24) are the linking constraints between layered cluster arc and original arc variables and inequalities (25) are the aforementioned compact connectivity constraints modified to graph G_{CL}.

As for the previous models, one obtains a stronger model CL_x^c by additionally considering directed cutset constraints (27) on G_{CL}.

$$Y[\delta^-(W)] \geq 1 \qquad W \subseteq V_{\mathrm{CL}} \setminus \{1_0\}, W \neq \emptyset \tag{27}$$

Along the same lines as in Section 2, two additional models to which we will refer to as CL_x^+ and $\mathrm{CL}_x^{c,+}$, respectively, are obtained by including cluster cuts (10).

Above formulation can be adapted to simultaneously use the concept of a cluster graph and a layered cluster graph. Besides the variables of CL_x, the resulting model CL_y uses cluster connection variables $y_{ij} \in \{0,1\}$, $\forall (i,j) \in A_{\mathrm{C}}$, with the same meaning as in model L_y.

$$\min \quad \sum_{(u,v)\in A} c_{uv} x_{uv} \tag{28}$$

$$\text{s.t.} \quad (13) - (18)$$

$$\sum_{h=1}^{H} Y[\delta^-(i_h)] = 1 \qquad \forall i \in C \setminus \{1\} \tag{29}$$

$$\sum_{h=0}^{H-1} Y_{ij}^h = y_{ij} \qquad \forall (i,j) \in A_{\mathrm{C}} \tag{30}$$

$$\sum_{(i_{h-1},j_h)\in A_{\mathrm{CL}},i\neq w} Y_{ij}^{h-1} \geq Y_{jw}^{h} \qquad \forall (j_h, w_{h+1}) \in A_{\mathrm{CL}}, h \geq 1 \tag{31}$$

$$(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{Y}) \in \{0,1\}^{|A|+|A_{\mathrm{C}}|+|V|+|A_{\mathrm{CL}}|} \tag{32}$$

Besides using (13)–(18) to ensure the existence of the arborescence spanning exactly one node from each cluster, the only difference to the previous model is given by the fact that linking constraints (30) link layered cluster arc to cluster arc variables. Similar to above, the model additionally including inequalities (27) will be called CL_y^c in the following.

4 Polyhedral Comparison

In this section, we compare the formulations proposed in Sections 2 and 3 and their variants in a theoretical sense, i.e., with respect to their LP relaxation values. Thereby, for formulation M, we denote by $P(M)$ the polyhedron induced by its LP relaxation and by $\mathrm{proj}_{\mathbf{x}^1,\mathbf{x}^2,\ldots,\mathbf{x}^m}(P(M))$ the orthogonal projection of polyhedron $P(M)$ onto the space $(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m)$. The optimal value of the LP relaxation of formulation M is denoted by $v(M) \in \mathbb{R}$. Formulation M_1 is at least as strong than formulation M_2 if $v(M_1) \geq v(M_2)$ and is strictly stronger than M_2 if M_1 is at least as strong as M_2 and there exists at least one instance for

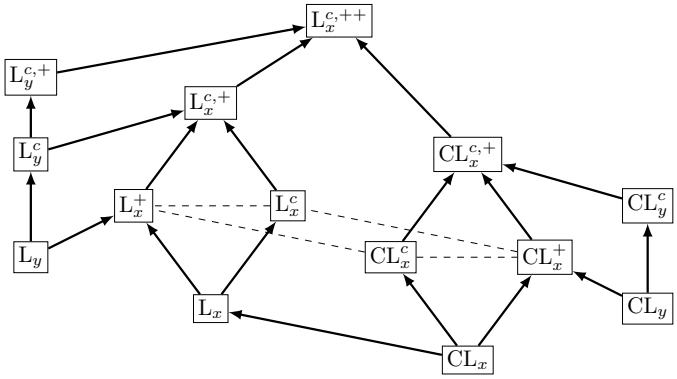


Figure 5: Theoretical relations between the proposed formulations.

which $v(M_1) > v(M_2)$. M_1 and M_2 are incomparable if there exist instances such that $v(M_1) > v(M_2)$ as well as instances with $v(M_2) > v(M_1)$. All obtained results are summarized in Figure 5 where a dotted edge indicates two incomparable formulations and a solid arrow that the formulation at the target is strictly stronger than the one at the source.

We emphasize that besides comparing the proposed formulations our results also show that fractional solutions contained in a polyhedron corresponding to a formulations that makes use of the clustered graph G_{C} or clustered layered graph G_{CL} may contain paths that contain more than H arcs (considering the projection to the $(\mathbf{x}, \mathbf{z})$-space), cf. the proof of Theorems 4.3 and 4. The latter is not true for the formulations not using any of these two graphs.

Theorems 4.1 and 4.2 summarize relations between different formulations for cases when one formulation is contained in the other. Since it is not too difficult to find exemplary solutions (to the corresponding LP relaxations) showing that the added constraints are strengthening (and some of these examples could be easily obtained by slightly modifying or reusing similar examples from the literature) we do not give explicit proofs of these two theorems.

Theorem 4.1. *The following relations hold:*

(i) Formulation $\mathrm{L}_x^{c,++}$ *is strictly stronger than formulation* $\mathrm{L}_x^{c,+}$.

(ii) Formulation $\mathrm{L}_x^{c,+}$ *is strictly stronger than formulation* L_x^{c}.

(iii) Formulation $\mathrm{L}_x^{c,+}$ *is strictly stronger than formulation* L_x^{+}.

(iv) Formulation L_x^{c} *is strictly stronger than formulation* L_x.

(v) Formulation L_x^{+} *is strictly stronger than formulation* L_x.

(vi) Formulation L_y^{c} *is strictly stronger than formulation* L_y.

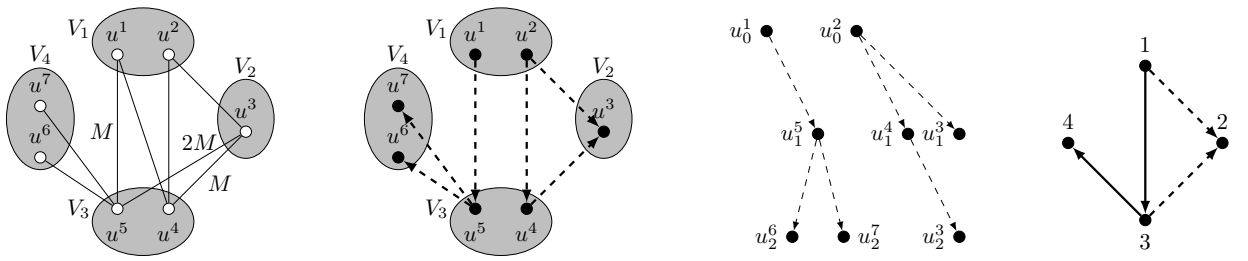
(vi) Formulation $\mathrm{L}_y^{c,+}$ *is strictly stronger than formulation* L_y^{c}.

Theorem 4.2. *The following relations hold:*

(i) Formulation $\mathrm{CL}_x^{c,+}$ *is strictly stronger than formulation* CL_x^{c}.

(ii) Formulation $\mathrm{CL}_x^{c,+}$ *is strictly stronger than formulation* CL_x^{+}.

(iii) Formulation CL_x^{c} *is strictly stronger than formulation* CL_x.



(a) Instance (b) Solution w.r.t. $\mathbf{x}$. (c) Solution w.r.t. $\mathbf{X}$. (d) Solution w.r.t. $\mathbf{y}$.

Figure 6: (a) An exemplary instance with edge costs $c_{u^1u^5} = c_{u^3u^4} = M$, $c_{u^3u^5} = 2M$, and $c_e = 1$ for all other edges. (b) A feasible solution to $\text{proj}_{\mathbf{x}}(P(\text{L}_x^c))$ and $\text{proj}_{\mathbf{x}}(P(\text{L}_y^+))$ of this instance with objective value $M + 2$ for which a feasible assignment of variables $\mathbf{X}$ is given in (c) and a feasible assignment of variables $\mathbf{y}$ in (d). Solid arcs correspond to variable values of 1 and dashed arcs to variable values of 0.5. Observe that this solution is infeasible for $\text{proj}_{\mathbf{x}}(P(\text{L}_x^+))$ since cluster cuts (10) are violated for $W = \{5, 6, 7\}$. Each LP solution satisfying (10) has an objective value greater than $2M$ as it must use edge $\{u^3, u^5\}$.

(iv) Formulation CL_x^+ *is strictly stronger than formulation* CL_x.

(v) Formulation CL_y^c *is strictly stronger than formulation* CL_y.

The next two theorems establish relations between models using cluster graph G_C and there corresponding variants not using this concept.

Theorem 4.3. *The following relations hold:*

(i) Formulation L_x^+ *is strictly stronger than formulation* L_y.

(ii) Formulation $\text{L}_x^{c,+}$ *is strictly stronger than formulation* L_y^c.

(iii) Formulation $\text{L}_x^{c,++}$ *is strictly stronger than formulation* $\text{L}_y^{c,+}$.

Proof. We first show that L_x^+ is at least as strong as L_y, by showing that $P(\text{L}_x^+) \subseteq \text{proj}_{\mathbf{x},\mathbf{X}}(P(\text{L}_y))$. Equation (16), i.e., $y_{ij} := x[\delta(V_i, V_j)]$, $\forall (i,j) \in A_\text{C}$, is used for each solution $(\mathbf{x}, \mathbf{X}) \in P(\text{L}_x^+)$ to construct a feasible assignment of the additional variables $\mathbf{y}$. Thus, we need to show that constraints (14), (15), and (19) are satisfied. We note that validity of (14) and (19) is easily observed by using (16) and the fact that they correspond to (aggregated) variants of (3) and (6). Using (10), we observe that cutset constraints (15) are satisfied as well since

$$y[\delta^-(W)] = x[\delta^-(\{u \in V_j \mid j \in W\})] \geq 1.$$

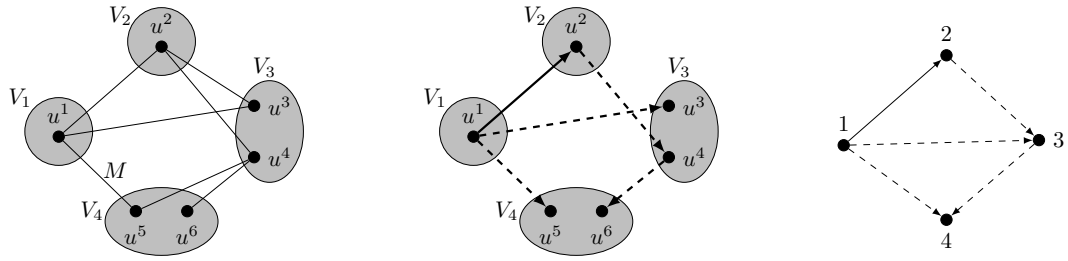
Since the solution given in Figure 6 is feasible for L_y but violates inequalities (10) for the cluster containing nodes u^6 and u^7, we conclude that L_x^+ indeed is strictly stronger than L_y.

Since the same argumentation as well as the example from Figure 6 remains valid when comparing the formulations including connectivity cuts on the layered graph (i.e., $\text{L}_x^{c,+}$ and L_y^c), the second statement of the theorem follows. Similarly, the third statement follows from additionally considering the example given in Figure 7. □

Theorem 4.4. *The following relations hold:*

(i) Formulation CL_x^+ *is strictly stronger than formulation* CL_y.

(ii) Formulation $\text{CL}_x^{c,+}$ *is strictly stronger than formulation* CL_y^c.



(a) Instance (b) Solution w.r.t. $\mathbf{x}$. (c) Solution w.r.t. $\mathbf{y}$.

Figure 7: (a) An exemplary instance with edge costs $c_{u^1u^5} = M$ and $c_e = 1$ for all other edges. (b) A feasible solution to $\text{proj}_x(P(\text{L}_y^{c,+}))$ with objective value $\frac{M+5}{2}$ for $H = 2$ for which the corresponding feasible assignment of variables $\mathbf{y}$ is given in (c) and a feasible assignment of variables $\mathbf{X}$ is easy to derive. Solid arcs indicate variable values of 1 and dashed arcs variable values of 0.5. Observe that this solution is infeasible for $\text{proj}_{\mathbf{x}}(P(\text{L}_x))$ since the path $u^1 - u^2 - u^4 - u^6$ has length 3 and thus, no feasible assignment of variables $\mathbf{X}$ can exist for all models containing L_x. The optimal value of the LP relaxation of formulation L_x must exceed M since each solution must contain arc (u^1, u^5).

Proof. Similar as before, we first show that $P(\text{CL}_x^+) \subseteq \text{proj}_{\mathbf{x},\mathbf{z},\mathbf{Y}}(P(\text{CL}_y))$ and $P(\text{CL}_x^{c,+}) \subseteq \text{proj}_{\mathbf{x},\mathbf{z},\mathbf{Y}}(P(\text{CL}_y^c))$, by using the identity $y_{ij} := x[\delta(V_i, V_j)]$. It is immediate that (24) holds if and only if (30) holds as well. Validity of (14), (15), and (19) can be shown by the same calculations and arguments as in the proof of Theorem 4.3.

To see that the formulations are indeed strictly stronger, we observe that the previously discussed solution from Figure 6 indeed is feasible for CL_y and CL_y^c (it is easy to derive a feasible assignment of variables $\mathbf{Y}$) but violates cluster cuts (10) and thus is infeasible for CL_x^+ and $\text{CL}_x^{c,+}$. □

Theorem 4.5. *The following relations hold:*

(i) Formulation L_x *is strictly stronger than formulation* CL_x*.*

(ii) Formulation $\text{L}_x^{c,++}$ *is strictly stronger than formulation* $\text{CL}_x^{c,+}$*.*

Proof. We first show that $\text{proj}_{\mathbf{x},\mathbf{z}}(P(\text{L}_x)) \subseteq \text{proj}_{\mathbf{x},\mathbf{z}}(P(\text{CL}_x))$, i.e., that L_x is at least as strong as CL_x. Given an arbitrary solution $(\mathbf{x}, \mathbf{z}, \mathbf{X}) \in P(\text{L}_x)$, we will construct a solution $(\mathbf{x}, \mathbf{z}, \mathbf{Y}) \in P(\text{CL}_x)$ by using $Y_{ij}^h := \sum_{(u_h, v_{h+1}) \in A_\text{L}, u \in V_i, v \in V_j} X_{uv}^h$.

Observe that constraints (2)–(5) are satisfied as they are contained in both formulations. Constraints (23) hold since

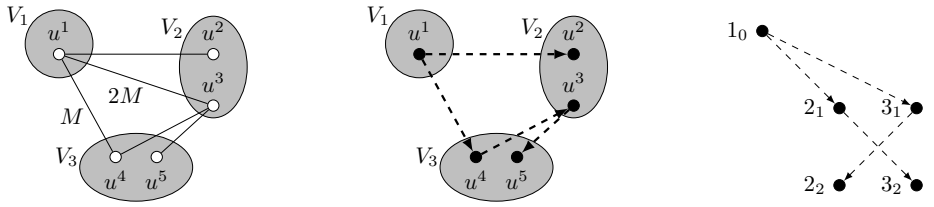
$$\sum_{h=1}^{H} Y[\delta^-(i_h))] = \sum_{h=1}^{H} X^h[\delta^-(V_i)] = \sum_{(u,v) \in A, v \in V_i} \sum_{h=1}^{H} X_{uv}^h = \sum_{(u,v) \in A, v \in V_i} x_{uv} = \sum_{u \in V_i} x[\delta^-(u)] = z(V_i) = 1$$

and (24) since

$$\sum_{h=0}^{H-1} Y_{ij}^h = \sum_{h=0}^{H-1} \sum_{(u,v) \in A : u \in V_i, v \in V_j} X_{uv}^h = \sum_{(u,v) \in A : u \in V_i, v \in V_j} \sum_{h=0}^{H-1} X_{uv}^h = \sum_{(u,v) \in A : u \in V_i, v \in V_j} x_{uv}.$$

Finally, the compact connectivity constraints (25) also hold for each $(j_{h-1}, w_h) \in A_\text{CL}$, as

$$Y_{jw}^h = \sum_{v \in V_j} \sum_{(v_h, k_{h+1}) \in A_\text{L}, k \in V_w} X_{vk}^h \leq \sum_{v \in V_j} \sum_{(u_{h-1}, v_h) \in A_\text{L}, u \neq V_w} X_{uv}^{h-1}$$



(a) Instance (b) Solution w.r.t. $\mathbf{x}$. (c) Solution w.r.t. $\mathbf{Y}$.

Figure 8: (a) An exemplary instance with edge costs $c_{u^1u^4} = M$, $c_{u^1u^3} = 2M$ and $c_e = 1$ for all other edges. (b) A feasible solution to $\text{proj}_x(P(\text{CL}_x^c))$ with objective value $\frac{M+3}{2}$ for $H = 2$ for which the corresponding feasible assignment of variables $\mathbf{Y}$ is given in (c). Dashed arcs indicate variable values of 0.5. Observe that this solution is infeasible for $\text{proj}_{\mathbf{x}}(P(\text{L}_x))$ since compact connectivity constraints (7) are violated along the path $u^4 - u^3 - u^5$. The optimal value of each LP solution to this instance feasible for L_x must exceed M.

$$= \sum_{i \in C \setminus \{w\}} \sum_{(u_{h-1}, v_h) \in A_{\text{L}}, u \in V_i, v \in V_j} X_{uv}^{h-1} = \sum_{(i_{h-1}, j_h) \in A_{\text{CL}}, i \neq w} Y_{ij}^{h-1}$$

To conclude that L_x is strictly stronger than CL_x, we observe that the solution given in Figure 8 is feasible for the LP relaxation of CL_x but infeasible for L_x since it is not possible to find a valid assignment of variables $\mathbf{X}$ due to the path $\{(k, w), (w, l)\}$ where k and l are contained in the same cluster. Thus the first statement of the theorem follows.

Using these results, to see that $\text{L}_x^{c,++}$ is at least as strong as $\text{CL}_x^{c,+}$ it only remains to show that using above transformation each solution to $\text{L}_x^{c,++}$ will also satisfy layered cuts (27). The latter holds for any set $W \subset V_{\text{CL}}$, $1_0 \notin W$, $W \neq \emptyset$, since

$$\begin{aligned} Y[\delta^-(W)] &= \sum_{(i_h, j_{h+1}) \in A_{\text{CL}}, j \in W} Y_{ij}^h = \sum_{(i_h, j_{h+1}) \in A_{\text{CL}}, j \in W} \sum_{(u_h, v_{h+1}) \in A_{\text{L}}, u \in V_i, v \in V_j} X_{uv}^h \\ &= X[\delta^-(\{v \in V_j \mid j \in W\})] \geq 1. \end{aligned}$$

To conclude that $\text{L}_x^{c,++}$ is strictly stronger than $\text{CL}_x^{c,+}$, we consider the solution given in Figure 9 which is feasible for the LP relaxation of $\text{CL}_x^{c,+}$ with a value of $\frac{M+5}{2}$. Since this solution is infeasible for $\text{L}_x^{c,++}$ (indeed already for L_x) and the optimal value of each solution to the LP relaxation of $\text{L}_x^{c,++}$ to this instance is at least M, the theorem follows. □

Theorem 4.6. *The following formulations are incomparable*

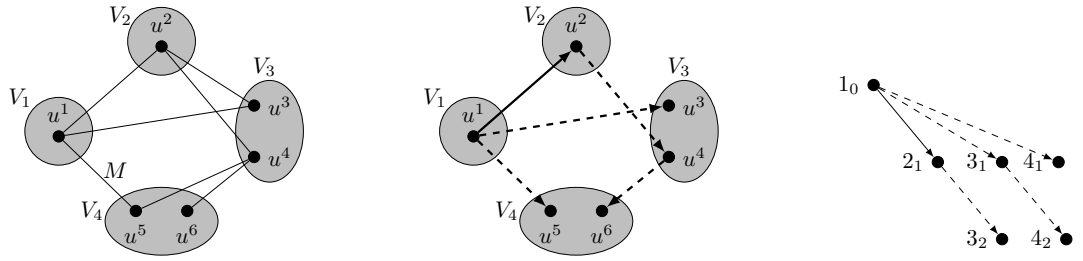
(i) L_x^c *and* L_x^+

(ii) CL_x^c *and* CL_x^+

(iii) L_x^c *and* CL_x^+

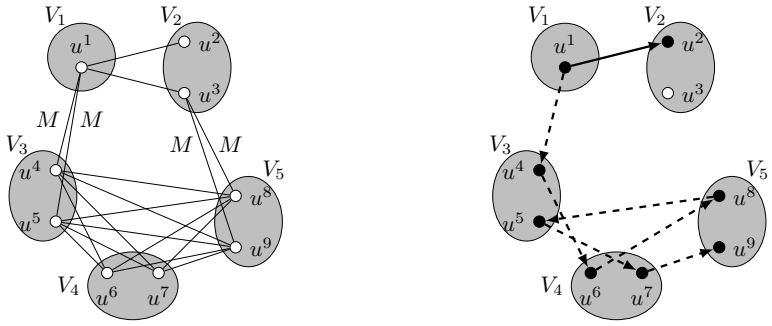
(iv) L_x^+ *and* CL_x^c

Proof. To see that all statements hold, we first observe that the solution given in Figure 6 is feasible for all formulation of the statement not containing cluster cuts (10), i.e., for L_x^c and CL_x^c, but infeasible for the L_x^+ and CL_x^+. Conversely, we note that connectivity cuts on layered graphs known to strengthen formulations including "standard" connectivity cuts for hop- and resource constrained spanning and Steiner tree problems, see, e.g., [25]. For instances with $|V_i| = 1$, $i = 1, \ldots, K$, or when the current LP solution is integral with respect to variables $\mathbf{z}$, these results from the literature imply that there exist instances feasible for L_x^+ and CL_x^+ but infeasible for L_x^c and CL_x^c. □



(a) Instance (b) Solution w.r.t. $\mathbf{x}$. (c) Solution w.r.t. $\mathbf{Y}$.

Figure 9: (a) An exemplary instance with edge costs $c_{u^1u^5} = M$ and $c_e = 1$ for all other edges. (b) A feasible solution to $\text{proj}_x(P(\text{CL}_x^{c++}))$ with objective value $\frac{M+5}{2}$ for $H = 2$ for which the corresponding feasible assignment of variables $\mathbf{Y}$ is given in (c). Solid arcs indicate variable values of 1 and dashed arcs variable values of 0.5. Observe that this solution is infeasible for $\text{proj}_{\mathbf{x}}(P(\text{L}_x))$ since the path $u^1 - u^2 - u^4 - u^6$ has length 3 and thus, no feasible assignment of variables $\mathbf{X}$ can exist. The optimal value of the LP relaxation of formulation L_x must exceed M since each solution must contain arc (u^1, u^5).



(a) Instance (b) Solution w.r.t. $\mathbf{x}$.

Figure 10: (a) An exemplary instance with edge costs $c_{u^1u^4} = c_{u^1u^5} = c_{u^3u^8} = c_{u^3u^9} = M$ and $c_e = 1$ for all other edges. (b) A feasible solution to $\text{proj}_{\mathbf{x}}(P(\text{L}_x^c))$ (and $\text{proj}_{\mathbf{x}}(P(\text{CL}_x^c))$) with objective value $\frac{M+7}{2}$ for $H \geq 6$ and for which a corresponding feasible assignment of variables $\mathbf{X}$ is easy to derive. Solid arcs correspond to variable values of 1 and dashed arcs to variable values of 0.5. Observe that this solution is infeasible for $\text{proj}_{\mathbf{x}}(P(\text{L}_y))$ (and $\text{proj}_{\mathbf{x}}(P(\text{CL}_y))$) since $y_{13} = 0.5$ and thus cutset constraints (15) are violated for $W = \{3, 4, 5\}$. Observe that the optimal value of each LP solution to this instance feasible for L_y must exceed M since all arcs entering set $W = \{3, 4, 5\}$ have costs equal to M.

Table 1: Overview on the models for which branch-and-cut approaches have been tested. Columns show whether or not cluster graph G_{C} or layered cluster graph G_{CL} and which types of cuts are considered.

Group	Name	G_{C}	G_{CL}	Node	Cluster	Lay. Node	Lay. Cluster
L_x^*	$\mathrm{L}^{+}_{x,\bar{n}}$	-	-	-	(10)	-	-
	$\mathrm{L}^{c,+}_{x}$	-	-	(4)	(10)	-	-
	$\mathrm{L}^{c,++}_{x,ln}$	-	-	(4)	(10)	-	(11)
	$\mathrm{L}^{c,++}_{x}$	-	-	(4)	(10)	(9)	(11)
L_y^*	L_{y}	+	-	-	(15)	-	-
	$\mathrm{L}^{c,+}_{y,ln}$	+	-	-	(15)	-	(11)
	$\mathrm{L}^{c,+}_{y}$	+	-	-	(15)	(9)	(11)
CL_x^*	$\mathrm{CL}^{+}_{x,\bar{n}}$	-	+	-	(10)	-	-
	CL^{+}_{x}	-	+	(4)	(10)	-	-
	$\mathrm{CL}^{c,+}_{x}$	-	+	(4)	(10)	-	(27)
CL_y^*	CL_{y}	+	+	-	(15)	-	-
	CL^{c}_{y}	+	+	-	(15)	-	(27)

5 Computational Study

Branch-and-cut algorithms (B&C) for all formulations proposed in Sections 2 and 3 have been developed in C++ using IBM CPLEX 12.6. Standard settings of CPLEX have been used and each experiment has been performed on a single core within a cluster of computers, each consisting of 20 cores (2.3GHz) and 64GB RAM. An absolute time limit of 10 000 CPU-seconds has been applied to each experiment and CPLEX has been configured to use at most 3GB.

5.1 Branch-and-Cut Configuration

We initialized all models by their sets of compact constraints while the various variants of directed cutset contraints are separated by using the maximum-flow algorithm of Cherkassky and Goldberg [1]. Thereby, we add back- and nested-cuts as well as considered creep-flows to generate sparse inequalities, see [18]. To avoid tail-off effects in branch-and-cut nodes, we only separate cuts if they are violated by a value of at least 0.1 in the current LP solution.

For formulation $\mathrm{L}_x^{c,++}$, cuts are separated in the following order: i) cluster cuts (10), ii) node cuts (4), iii) layered cluster cuts (11), and iv) layered cuts (9). Thereby, each cut type is only considered if no violated constraints have been added for all preceding variants in the current iteration. An analogous strategy is used for the other variants of model L_x by simply skipping all cut types that are not contained in the corresponding formulation. Similarly, for model $\mathrm{CL}_x^{c,+}$ we first separate cluster cuts (10), before possibly considering node cuts (4) and layered cuts (27). Again, we simply skip not-contained types of cuts for other variants of CL_x. Finally, in models L_y^c and CL_y^c, layered cuts (27) are only added if no violated cuts (15) have been added in the current iteration.

In case all variable values of the current LP solution are integral, the separation routine significantly simplifies due to the presence of compact connectivity constraints on corresponding layered graphs. Hence, for such solutions we only search for violated cluster cuts (10) for all variants of formulations L_x and CL_x and for violated cuts (15) for the variants of L_y and CL_y.

Table 1 summarizes the variants of the different models developed in the previous sections for which corresponding branch-and-cut approaches have been tested. Note that we also test variants considering (layered) cluster cuts but skipping (layered) node cuts. While the opposite was natural for the theoretical development of the models, these variants are more likely to perform better than variants considering (layered) node but not (layered) cluster cuts. In the following, we will for simplicity refer to each branch-and-cut algorithm by the abbreviation of the model it is defined by.

5.2 Benchmark Instances

We evaluate the performance of our approaches on instances from Fischetti et al. [6] originally proposed for the generalized traveling salesman problem but which have been widely used as (basis for) benchmark instances for assessing the performance of algorithmic approaches to various generalized network design problems, see, e.g., [15, 16]. This benchmark set originates from Euclidean TSPlib [24] instances for which two different clustering routines (either *geographical* or *grid* clustering) have been applied. Instances with geographical clustering were created by first selecting $K \cdot \lceil |V|/5 \rceil$ nodes at maximum distance from each other which define the seed nodes of the K clusters. All other nodes were assigned to the cluster containing its individually closest seed node. Instances based on grid clustering have been generated according to the coordinates of nodes in the plane (by dividing the plane into rectangles) and according to an input parameter μ, $\mu \in \{3, 5, 7, 10\}$, which determines the average number of nodes per cluster, see [4, 6] for details. Thus, in total five instances have been created for each original TSPlib instance. To use these instances for the GHMSTP, the first cluster of an instance is assumed to be the root and we tested different values of H for each instance.

5.3 Initial Solution

Preliminary experiments showed that CPLEX fails to find feasible solutions within 10 000 seconds for a few, rare cases. We also observed that given an initial solution of reasonable quality, the built in general purpose heuristics are typically rather successful in finding high-quality (close to optimal) solutions. Thus, we did not implement a primal heuristic that is repeatedly called during the course of the branch-and-cut but construct an initial, feasible solution (which is passed to the solver) as follows: First a node is selected from each cluster randomly. Afterwards, a simple Prim based heuristic is applied which greedily adds the chosen node with cheapest additional costs to the partial tree if it can be reached without violating the hop-constraints. This heuristic is repeated 30 times and the overall best solution is adopted.

6 Computational Results

Tables 2 and 3 give a first overview on the results obtained within our computational study for smaller and larger instances, respectively. Thereby, results are grouped according to the four main classes of models discussed in the previous sections (i.e., whether or not the concept of cluster and / or layered cluster graph is used). Each line of these two tables reports how many instances could be solved by at least one of the models from each particular class ($\#_{\text{solved}}$) and as well as how often a model from each class achieved the best performance ($\#_{\text{best}}$). Thereby, we consider a variant as having the best performance if it could solve that particular instance to proven optimality at least as fast as any other tested variant. In case that instance could not be solved by any variant within the given time limit of 10 000 seconds, all variants for which the obtained optimality gap attains a minimum are considered to perform best. To facilitate a first overview on the influence of the various parameters, each of the two tables depicts those results grouped according all pairs out of the three main criteria (i.e., original instance, clustering method, hop limit) within our benchmark set.

From Tables 2 and 3 we conclude that all proposed model classes allow to solve most of the smaller instances to proven optimality within the given time limit of 10 000 CPU-seconds. Considering the larger instances it is easy to observe that the use of the cluster graph does not pay off computationally. As we have shown in Section 4 the corresponding variants based on L_y and CL_y are dominated by other variants that do not use the cluster graph. Given the computational results, we conclude that the theoretically smaller number of connectivity cuts on the cluster graph and therefore potentially faster solution of the LP relaxations at each node of the branch-and-cut tree does not compensate the weaker bounds and larger number of nodes to consider. We also observe that the relative performance of models based on L_y and CL_y improves (compared to the others) for instances with not too few nodes per cluster (i.e., grid clustering with $\mu \in \{7, 10\}$). The latter might indicate that models utilizing the concept of the cluster graph might pay off on instances with a large number of nodes per cluster.

Table 2: Summarized results for smaller instances grouped by original instance, clustering method and hop limit, respectively. Numbers of solved instances ($\#_{\text{solved}}$) and numbers of cases with best performance ($\#_{\text{best}}$) are reported per model class. L_x^* refers to the variants of L_x (i.e., those not considering the (layered) cluster graph), L_y^* are the variants of L_y (use the cluster graph), models CL_x^* make use of the layered cluster graph, and finally CL_y^* utilize both the cluster graph and the layered cluster graph, see Table 1. Individually best performing variant per model class is considered for each experiment. Best values are marked bold.

			$\#_{\text{solved}}$				$\#_{\text{best}}$			
		#	L_x^*	L_y^*	CL_x^*	CL_y^*	L_x^*	L_y^*	CL_x^*	CL_y^*
Instance	att48	20	**20**	**20**	**20**	**20**	8	0	**12**	0
	eil51	20	**20**	**20**	**20**	**20**	6	0	**16**	0
	eil76	20	**20**	16	**20**	**20**	9	0	**11**	0
	st70	20	**20**	**20**	**20**	**20**	7	0	**13**	0
	gr96	20	**20**	12	**20**	16	7	0	**13**	0
	pr76	20	**20**	17	**20**	19	9	0	**11**	0
	rat99	20	**20**	3	17	5	**12**	0	8	0
Clustering Method	geographical	28	**28**	21	**28**	25	10	0	**18**	0
	grid, $\mu = 3$	28	**28**	15	25	21	**18**	0	11	0
	grid, $\mu = 5$	28	**28**	22	**28**	24	**14**	0	**14**	0
	grid, $\mu = 7$	28	**28**	25	**28**	25	9	0	**19**	0
	grid, $\mu = 10$	28	**28**	25	**28**	25	7	0	**22**	0
Hop Limit (H)	3	35	**35**	32	**35**	34	**33**	0	3	0
	4	35	**35**	26	34	31	**18**	0	17	0
	5	35	**35**	25	34	28	3	0	**32**	0
	6	35	**35**	25	34	27	4	0	**32**	0

Table 3: Summarized results for larger instances grouped by original instance, clustering method and hop limit, respectively. Numbers of solved instances ($\#_{\text{solved}}$) and cases with best performance ($\#_{\text{best}}$) are reported per model class. L_x^* refers to the variants of L_x (i.e., those not considering the (layered) cluster graph), L_y^* are the variants of L_y (use of the cluster graph), models CL_x^* make use of the layered cluster graph, and finally CL_y^* utilize both the cluster graph and the layered cluster graph. Individually best performing variant per model class is considered for each experiment. Best values are marked bold.

		#	$\#_{\text{solved}}$				$\#_{\text{best}}$			
		#	L_x^*	L_y^*	CL_x^*	CL_y^*	L_x^*	L_y^*	CL_x^*	CL_y^*
Instance	kroa100	20	**20**	11	**20**	**20**	5	0	**15**	0
	krob100	20	**20**	13	**20**	17	**10**	0	**10**	0
	kroc100	20	**20**	9	**20**	18	6	0	**14**	0
	krod100	20	**20**	7	**20**	18	6	0	**14**	0
	kroe100	20	**20**	10	**20**	19	9	0	**11**	0
	rd100	20	**20**	11	**20**	**20**	**10**	0	**10**	0
	eil101	20	**20**	6	19	10	6	0	**14**	0
	pr107	20	**20**	15	**20**	**20**	0	4	**11**	5
	pr124	20	**20**	10	**20**	16	1	0	**19**	0
	bier127	20	**19**	13	**19**	16	2	0	**16**	2
	gr137	20	**5**	0	4	0	6	0	**14**	0
	pr144	20	16	7	**20**	12	0	0	**19**	1
	kroa150	20	**18**	1	12	4	**11**	0	9	0
	krob150	20	**18**	0	12	1	9	0	**11**	0
	pr152	20	15	2	**19**	6	3	0	**16**	1
	u159	20	**12**	0	10	0	7	0	**13**	0
Clustering Method	geographical	64	52	14	**53**	37	14	0	**50**	0
	grid, $\mu = 3$	64	**50**	2	46	30	27	0	**33**	4
	grid, $\mu = 5$	64	**56**	17	52	37	24	0	**37**	3
	grid, $\mu = 7$	64	**61**	38	60	44	15	1	**47**	1
	grid, $\mu = 10$	64	**64**	44	**64**	49	11	3	**49**	1
Hop Limit (H)	3	80	**76**	45	65	60	**53**	1	20	6
	4	80	**71**	26	67	49	27	1	**50**	2
	5	80	69	23	**70**	45	8	1	**70**	1
	6	80	67	21	**73**	43	3	1	**76**	0

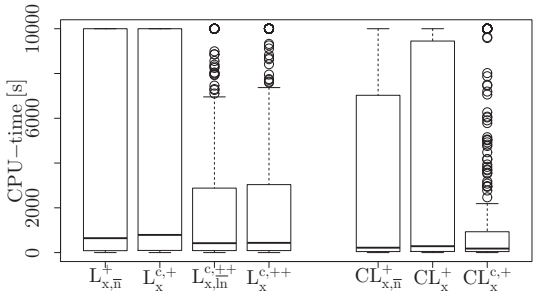


Figure 11: Distributions of CPU-times [s] for variants of L_x and CL_x.

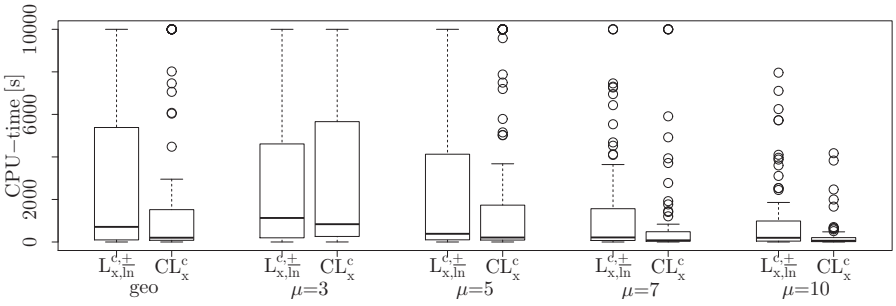


Figure 12: Distributions of CPU-times [s] of $L^{c,+}_{x,\bar{ln}}$ and $CL^{c,+}_{x}$ by clustering method.

On the contrary, we observe that variants making use of the layered cluster graph clearly outperform the others when considering the numbers of cases in which a best performance among all variants is achieved. This trend is particularly clear when the hop limit is not too small, for larger instances and / or when considering instances with not too few nodes per cluster. Overall, though these variants do not include the theoretically strongest model (i.e., $L^{c,++}_{x}$) using the layered cluster graph and therefore reducing the number of layered connectivity cuts seems to pays off in particular for the larger and harder ones among the considered instances. We also note that from Tables 2 and 3 we cannot observe a clear trend whether the variants based on L_x or those CL_x perform better with respect to the total number of instances solved in the given maximum runtime.

To analyze in more detail the performance of the two best performing classes, i.e., the various variants of L_x and CL_x, corresponding runtime distributions over all test instances and parameter are given in Figure 11. We conclude that dynamically separating cutset constraints on the (clustered) layered graph yields a significantly improved performance, i.e., variants $L^{c,++}_{x,\bar{ln}}$, $L^{c,++}_{x}$, and $CL^{c,+}_{x}$ perform better than the simpler variants of the respective class. We also observe that the overall performance of $L^{c,++}_{x,\bar{ln}}$ is slightly better than the one of $L^{c,++}_{x}$ and therefore conclude that considering layered node cuts (9) does not pay off on the considered instances. Overall, $L^{c,++}_{x,\bar{ln}}$ and $CL^{c,+}_{x}$ yield the best performance in their respective class and we will therefore analyze these two in more detail in the following.

Figures 12 and 13 show runtime distributions of $L^{c,++}_{x,\bar{ln}}$ and $CL^{c,+}_{x}$ for different clustering methods and all considered values of H, respectively. We conclude that $CL^{c,+}_{x}$ consistently and significantly outperforms $L^{c,++}_{x,\bar{ln}}$ except for those instances with a very small number of nodes per cluster (i.e., grid clustering with $\mu = 3$) or when $H = 3$. Thus, the results of our computational study clearly show that the concept of a layered cluster graph leads to branch-and-cut algorithms with practically good performance. These findings are also confirmed by Tables 4–8 as well as Tables 9–13 (see Appendix) that detail CPU-times and potentially

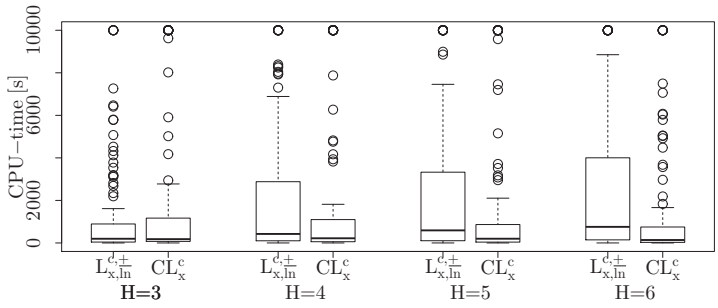


Figure 13: Distributions of CPU-times [s] of $L^{c,+}_{x,ln}$ and $CL^{c,+}_{x}$ by hop limit.

existing optimality gaps after termination for all tested variants and instances.

7 Conclusions

In this article, we studied the generalized hop-constrained minimum spanning tree problem (GHMSTP) which arises in the design of backbone telecommunication networks when the maximum path length may not exceed a predefined upper bound. Different integer programming models based on layered graph reformulations have been discussed including variants that generalize the concept of a cluster graph [22]. Several strengthening valid inequalities have been proposed and a hierarchy with respect to theoretical strength of the proposed formulations (i.e., their LP relaxation values) has been established. The results of our computational study performed on benchmark instance known from other generalized network design problems showed that the branch-and-cut approach utilizing the layered version of the cluster graph outperforms the other variants.

References

[1] Boris V. Cherkassky and Andrew V. Goldberg. On implementing push-relabel method for the maximum flow problem. *Algorithmica*, 19:390–410, 1994.

[2] Geir Dahl. The 2-hop spanning tree problem. *Operations Research Letters*, 23(1-2):21–26, 1998.

[3] Geir Dahl, Luis Gouveia, and Cristina Requejo. On formulations and methods for the hop-constrained minimum spanning tree problem. In Mauricio G. C. Resende and Panos M. Pardalos, editors, *Handbook of Optimization in Telecommunications*, pages 493–515. Springer, 2006.

[4] Corinne Feremans. *Generalized Spanning Trees and Extensions*. PhD thesis, Universite Libre de Bruxelles, 2001.

[5] Corinne Feremans, Martine Labbé, and Gilbert Laporte. The generalized minimum spanning tree problem: Polyhedral analysis and branch-and-cut algorithm. *Networks*, 43(2):71–86, 2004.

[6] Matteo Fischetti, Juan José Salazar González, and Paolo Toth. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research*, 45(3):378–394, 1997.

[7] Bruce Golden, Subramanian Raghavan, and Daliborka Stanojević. Heuristic search for the generalized minimum spanning tree problem. *INFORMS Journal on Computing*, 17(3):290–304, 2005.

[8] Luis Gouveia. Using the Miller-Tucker-Zemlin constraints to formulate a minimal spanning tree problem with hop constraints. *Computers & Operations Research*, 22:959–970, 1995.

[9] Luis Gouveia and Cristina Requejo. A new Lagrangian relaxation approach for the hop-constrained minimum spanning tree problem. *European Journal of Operational Research*, 132(3):539–552, 2001.

[10] Luis Gouveia, Ana Paias, and Dushyant Sharma. Restricted dynamic programming based neighborhoods for the hop-constrained minimum spanning tree problem. *Journal of Heuristics*, 17(1):23–37, 2011.

[11] Luis Gouveia, Luidi Simonetti, and Eduardo Uchoa. Modeling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs. *Mathematical Programming*, 128:123–148, 2011.

[12] Luis Gouveia, Markus Leitner, and Ivana Ljubić. Hop constrained Steiner trees with multiple root nodes. *European Journal of Operational Research*, 236:100–112, 2014.

[13] Luis Gouveia, Markus Leitner, and Ivana Ljubić. The two-level diameter constrained spanning tree problem. *Mathematical Programming*, 2014. to appear.

[14] Bin Hu. *Hybrid Metaheuristics for Generalized Network Design Problems*. PhD thesis, Vienna University of Technology, 2008.

[15] Bin Hu, Markus Leitner, and Günther R. Raidl. Combining variable neighborhood search with integer linear programming for the generalized minimum spanning tree problem. *Journal of Heuristics*, 14(5): 473–499, 2008.

[16] Bin Hu, Markus Leitner, and Günther R. Raidl. The generalized minimum edge biconnected network problem: Efficient neighborhood structures for variable neighborhood search. *Networks*, 55(3):257–275, 2010.

[17] Daniel Karapetyan and Gregory Z. Gutin. Lin–kernighan heuristic adaptations for the generalized traveling salesman problem. *European Journal of Operational Research*, 208(3):221–232, 2011.

[18] Thorsten Koch and Alexander Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32(3):207–232, 1998.

[19] Ivana Ljubić and Stefan Gollowitzer. Layered graph approaches to the hop constrained connected facility location problem. *INFORMS Journal on Computing*, 25:256–270, 2013.

[20] Yound-Soo Myung, Chang-Ho Lee, and Dong-Wan Tcha. On the generalized minimum spanning tree problem. *Networks*, 26(4):231–241, 1995.

[21] Temel Öncan, Jean-Francois Cordeau, and Gilbert Laporte. A tabu search heuristic for the generalized minimum spanning tree problem. *European Journal of Operational Research*, 191(2):306–319, 2008.

[22] Petrica C. Pop. *The Generalized Minimum Spanning Tree Problem*. PhD thesis, University of Twente, 2002.

[23] Petrica C. Pop, Walter Kern, and Georg Still. A new relaxation method for the generalized minimum spanning tree problem. *European Journal of Operational Research*, 170(3):900–908, 2006.

[24] Gerhard Reinelt. TSPLIB–a traveling salesman problem library. *INFORMS Journal on Computing*, 3 (4):376–384, 1991.

[25] Mario Ruthmair. *On Solving Constrained Tree Problems and an Adaptive Layers Framework*. PhD thesis, Vienna University of Technology, 2008.

[26] Mario Ruthmair and Günther R. Raidl. A layered graph model and an adaptive layers framework to solve delay-constrained minimum tree problems. In *Proceedings of the 15th Conference on Integer Programming and Combinatorial Optimization (IPCO XV)*, volume 6655 of *LNCS*, pages 376–388. 2011.

Table 4: Detailed results for smaller instances and geographical clustering. Besides best solution values (Best), CPU-times in seconds are reported whenever an instance is solved to proven optimality by the corresponding method while values in parenthesis are optimality gaps in case of termination due to time or memory (indicated by *) limit.

Inst.	H	Opt	$L^{+}_{x,\bar{n}}$	$L^{c,+}_{x}$	$L^{c,++}_{x,ln}$	$L^{c,++}_{x}$	L_{y}	$L^{c,+}_{y,ln}$	$L^{c,+}_{y,}$	$CL^{+}_{x,\bar{n}}$	CL^{+}_{x}	$CL^{c,+}_{x}$	CL_{y}	CL^{c}_{y}
att48	3	12139	**0.3**	**0.3**	**0.3**	**0.3**	1.9	14.2	13.1	0.5	0.5	0.5	1.2	1.6
	4	11271	**0.5**	0.6	0.6	0.6	3.0	1121.6	302.1	0.6	0.6	0.6	2.0	1.9
	5	10933	1.1	1.1	1.1	1.1	2.6	7292.6	15.3	**0.5**	**0.5**	**0.5**	1.2	1.3
	6	10923	2.1	2.1	2.1	2.2	2.7	(10.17)	60.3	**0.6**	**0.6**	**0.6**	1.0	2.0
eil51	3	142	5.3	6.0	7.4	7.9	17.3	6821.8	1136.8	**3.4**	3.8	3.7	6.4	6.8
	4	141	11.8	15.0	14.0	15.9	55.6	(20.41)	(10.16)	**4.0**	5.0	5.3	14.1	14.6
	5	139	14.2	15.8	25.4	31.4	48.0	(23.08)	(23.08)	**4.1**	4.5	5.6	18.7	33.4
	6	134	11.8	12.4	12.9	14.5	47.0	(19.85)	(19.86)	**2.7**	**2.7**	3.5	7.5	23.3
eil76	3	213	146.5	173.3	**94.9**	100.2	2262.5	(28.43)	(27.59)	143.2	176.0	166.4	1159.8	848.3
	4	202	**120.3**	125.5	325.3	364.6	5130.0	(32.46)	(32.46)	132.4	161.7	163.3	2549.3	1743.3
	5	198	500.8	664.9	340.6	354.6	(3.18)	(26.06)	(26.06)	**82.7**	107.7	106.8	3360.0	(5.61)*
	6	197	480.1	506.0	577.4	597.9	(7.39)	(29.34)	(29.34)	**93.5**	117.0	150.0	2489.5	(9.99)
st70	3	272	24.9	29.2	**20.9**	21.1	75.4	(13.59)	9995.7	27.4	33.0	53.0	23.8	27.1
	4	248	26.1	26.6	33.2	32.5	121.0	(25.41)	(25.27)	28.2	32.1	**19.8**	35.1	27.9
	5	244	50.1	51.6	66.6	67.1	335.6	(24.46)	(24.45)	**22.4**	27.9	30.5	60.1	54.3
	6	241	73.7	76.5	95.7	92.9	1913.7	(16.44)	(16.44)	**19.0**	22.8	23.5	37.6	77.0
gr96	3	265	**61.9**	81.1	82.0	80.5	767.4	(15.88)	(16.72)	788.5	1046.5	146.1	122.4	183.1
	4	241	142.4	176.7	95.6	**94.3**	2464.2	(25.59)	(24.85)	305.8	479.4	146.9	1878.5	687.4
	5	231	169.9	192.9	189.0	205.8	9489.9	(23.63)	(23.62)	**77.9**	87.5	79.4	1225.3	1592.5
	6	226	213.0	216.3	240.8	220.8	(4.83)	(19.34)	(19.34)	**31.4**	33.2	54.2	1087.7	2696.8
pr76	3	53186	12.9	13.6	**12.3**	12.6	230.1	(18.29)	(19.75)	123.2	135.5	69.2	112.0	90.9
	4	49859	121.0	137.7	112.7	113.0	2776.3	(22.50)	(22.52)	**56.6**	61.0	80.5	703.4	453.4
	5	47621	48.4	47.5	52.0	51.9	3087.3	(20.81)	(20.81)	34.5	37.0	**27.2**	1223.0	999.1
	6	47106	87.5	78.7	95.0	96.3	1886.5	(19.64)	(19.64)	20.8	21.5	**16.9**	428.0	1152.4
rat99	3	533	718.2	937.8	**552.1**	556.1	(4.97)	(29.80)	(29.59)	1174.9	1304.8	762.2	3430.2	4576.9
	4	492	(5.13)	(5.37)	1323.5	**1287.9**	(19.52)	(40.58)	(40.07)	3202.7	3251.9	1817.1	(22.08)*	(7.10)
	5	457	(4.51)	(4.52)	1320.3	1209.9	(19.00)	(43.81)	(43.86)	3587.1	4026.7	**379.3**	(18.85)*	(7.66)
	6	442	(4.73)	(4.39)	1332.6	1286.2	(22.65)	(42.19)	(42.19)	2224.7	3901.6	**374.3**	(18.87)*	(13.32)

[27] John Silberholz and Bruce Golden. The generalized traveling salesman problem: A new genetic algorithm approach. In Edward K. Baker, Anito Joseph, Anuj Mehrotra, and Michael A. Trick, editors, *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*, volume 37 of *Operations Research/Computer Science Interfaces Series*, pages 165–181. 2007.

[28] Stefan Voß. The Steiner tree problem with hop constraints. *Annals of Operations Research*, 86:271–294, 1999.

Appendix

See Tables 4, 5, 6, 7, 8, 9, 10, 11, 12, and 13.

Table 5: Detailed results for smaller instances and grid clustering with $\mu = 3$. Besides solution values (Best), CPU-times in seconds are reported whenever an instance is solved to proven optimality by the corresponding method while values in parenthesis are optimality gaps in case of termination due to time or memory (indicated by *) limit.

Inst.	H	Best	$L^{+}_{x,\bar{n}}$	$L^{c,+}_{x}$	$L^{c,++}_{x,\bar{ln}}$	$L^{c,++}_{x}$	L_{y}	$L^{c,+}_{y,\bar{ln}}$	$L^{c,+}_{y,}$	$CL^{+}_{x,\bar{n}}$	CL^{+}_{x}	$CL^{c,+}_{x}$	CL_{y}	CL^{c}_{y}
att48	3	19437	**1.0**	**1.0**	**1.0**	**1.0**	14.1	197.3	100.4	2.8	3.0	3.0	6.9	4.8
	4	17825	2.3	2.3	2.2	**2.1**	15.9	1693.9	952.2	2.2	2.2	2.2	6.9	7.3
	5	17339	7.2	7.1	5.8	5.8	38.1	(6.32)	(7.09)	**4.3**	4.5	5.0	17.6	16.5
	6	17115	10.9	11.1	11.4	11.4	72.0	(9.86)	(9.86)	**3.8**	4.5	5.0	22.4	29.3
eil51	3	278	5.3	7.5	**5.1**	6.2	30.4	359.3	174.2	12.0	17.8	20.7	30.4	14.2
	4	258	**4.5**	4.9	5.5	5.3	33.8	1160.0	913.7	18.2	25.4	12.3	13.9	14.4
	5	251	8.1	11.2	8.4	9.1	29.2	(3.59)	4093.7	**7.2**	10.3	30.3	9.0	28.4
	6	248	**5.0**	6.2	8.0	8.3	19.3	(2.89)	5703.9	**5.0**	5.9	8.6	9.6	23.1
eil76	3	379	42.1	57.5	**31.2**	31.4	3283.9	(5.29)	(6.08)	587.2	917.7	240.4	1176.0	493.1
	4	350	1159.1	1691.7	131.3	**127.9**	(5.04)	(12.82)	(9.88)	4274.2	6793.8	1148.8	7200.3	(0.91)
	5	329	274.7	375.1	102.8	**97.4**	(0.95)	(18.38)	(18.38)	901.1	1291.5	271.1	7523.8	3451.8
	6	317	52.8	58.7	**50.5**	**50.5**	8013.6	(12.44)	(12.41)	99.7	148.9	138.5	2348.9	2276.1
st70	3	369	7.7	8.3	6.4	**6.3**	212.1	2414.1	1543.0	32.3	46.0	19.8	44.9	21.1
	4	338	74.5	96.2	**33.7**	33.8	383.4	(13.61)	8814.0	59.4	86.4	46.5	238.7	70.7
	5	319	62.5	66.3	42.7	45.5	440.4	(10.64)	(13.51)	32.9	47.2	**27.3**	232.8	80.6
	6	312	76.4	87.2	54.5	53.7	716.9	(10.77)	(10.79)	35.4	48.2	**29.4**	144.2	209.2
gr96	3	391	1589.7	1700.7	440.1	**437.5**	(4.89)*	(13.64)	(13.68)	4569.6	5994.6	1504.4	8939.3	(2.45)*
	4	353	(3.82)	(3.83)	1186.3	1199.2	(10.27)*	(15.81)	(15.75)	(1.79)	(2.68)	**652.6**	(7.48)*	8263.0
	5	331	(3.39)	(3.48)	1321.9	1416.0	(7.84)*	(20.57)	(20.69)	(1.60)	(2.26)	**970.7**	(8.34)*	(4.66)
	6	324	(4.26)*	(4.26)	3815.5	3635.9	(11.32)	(23.92)	(23.97)	(3.23)*	(3.99)	**2968.7**	(9.12)*	(7.26)
pr76	3	72858	**44.0**	54.5	61.4	60.5	2643.7	(12.81)	(4.34)	411.7	585.2	183.7	743.7	403.8
	4	66449	1744.3	2212.1	**196.4**	203.4	(7.69)*	(17.64)	(14.97)	1907.2	2486.6	231.3	(2.89)	1495.8
	5	62936	551.6	642.0	266.3	252.0	(6.40)*	(15.88)	(15.88)	937.7	1336.3	**180.3**	(5.97)*	4568.5
	6	61184	563.9	671.7	360.6	376.7	(5.27)	(19.97)	(19.88)	1210.9	1661.7	**343.3**	(1.06)	(2.46)
rat99	3	791	(3.60)	(4.23)	660.2	**637.8**	(13.37)*	(24.22)	(17.28)	(7.11)	(8.44)	9624.6	(8.83)*	(3.57)
	4	684	(8.97)	(8.93)	1030.4	**1008.7**	(16.19)	(26.32)	(26.36)	(11.26)	(11.48)	(2.86)	(13.13)*	(4.52)
	5	639	(8.59)	(8.85)	2115.3	**1961.2**	(18.25)*	(21.67)	(21.96)	(9.94)	(10.96)	(2.61)	(14.92)*	(9.05)
	6	610	(7.39)	(8.67)	**1412.0**	1611.6	(15.01)	(19.17)	(19.31)	(8.28)*	(8.86)	(2.51)	(13.36)	(8.30)

Table 6: Detailed results for smaller instances and grid clustering with $\mu = 5$. Besides solution values (Best), CPU-times in seconds are reported whenever an instance is solved to proven optimality by the corresponding method while values in parenthesis are optimality gaps in case of termination due to time or memory (indicated by *) limit.

Inst.	H	Best	$L^{+}_{x,\bar{n}}$	$L^{c,+}_{x}$	$L^{c,++}_{x,\bar{ln}}$	$L^{c,++}_{x}$	L_{y}	$L^{c,+}_{y,\bar{ln}}$	$L^{c,+}_{y,}$	$CL^{+}_{x,\bar{n}}$	CL^{+}_{x}	$CL^{c,+}_{x}$	CL_{y}	CL^{c}_{y}
att48	3	14980	**0.6**	**0.6**	**0.6**	**0.6**	3.5	157.9	33.4	0.8	0.9	1.0	1.3	1.0
	4	13940	**1.1**	**1.1**	1.2	1.2	6.5	124.6	120.6	1.4	1.4	1.8	2.6	2.1
	5	13343	2.5	2.5	2.5	2.5	6.4	1586.2	1626.3	1.7	1.9	**1.3**	2.0	2.0
	6	13251	3.6	3.6	3.7	3.6	5.7	(9.09)	(9.09)	**1.1**	**1.1**	1.5	2.8	4.2
eil51	3	178	**3.2**	4.0	4.3	4.4	12.6	1647.7	578.2	3.9	4.7	3.9	8.7	5.3
	4	170	3.4	3.6	5.1	5.1	24.3	(10.64)	2022.6	**3.3**	4.0	4.3	9.4	11.6
	5	165	3.8	3.8	3.8	3.7	27.3	(8.21)	(8.21)	**2.5**	2.6	3.2	5.0	6.3
	6	162	4.0	4.3	4.2	4.2	37.1	(9.74)	(9.74)	**1.6**	1.7	2.0	6.2	10.3
eil76	3	171	**32.0**	37.4	46.3	49.0	302.1	(26.03)	(22.86)	59.1	86.5	37.1	177.9	199.4
	4	162	68.0	80.5	105.9	117.4	1230.0	(30.47)	(30.47)	36.9	50.1	**28.3**	299.5	400.3
	5	157	83.1	100.0	91.3	101.1	1487.4	(30.08)	(30.08)	28.2	37.9	**26.3**	296.8	949.7
	6	152	58.8	53.8	71.9	65.1	1794.6	(23.07)	(23.07)	8.6	**8.5**	14.2	144.8	314.6
st70	3	249	**2.8**	2.9	3.0	3.0	28.2	3579.2	2707.4	4.9	5.2	5.0	12.6	11.3
	4	233	6.8	6.7	**6.3**	6.5	85.1	(12.01)	(8.30)	13.3	17.3	10.3	13.3	48.4
	5	228	25.1	27.0	26.3	25.6	292.4	(19.13)	(19.12)	**10.8**	12.4	13.6	29.6	31.7
	6	222	26.1	26.4	26.1	26.6	663.1	(20.52)	(20.52)	**7.2**	7.5	9.3	19.6	62.0
gr96	3	283	189.7	226.3	99.8	**87.6**	2922.6	(16.04)	(15.87)	463.1	593.5	252.4	2315.1	827.5
	4	264	2091.8	2396.7	228.1	**218.8**	(8.37)*	(20.15)	(20.08)	983.6	1306.1	817.6	(7.78)*	4691.2
	5	253	1194.4	1488.3	588.5	595.8	(10.43)	(35.98)	(35.92)	**492.0**	615.3	867.4	(6.28)*	(3.58)
	6	249	2381.4	2818.1	1599.4	1485.0	(9.73)	(29.83)	(29.83)	1359.0	1791.4	**523.1**	(9.17)*	(7.30)
pr76	3	37376	23.9	27.0	**16.3**	**16.3**	200.8	(21.96)	(20.16)	59.9	81.2	78.0	86.3	107.0
	4	34399	128.3	136.7	**88.0**	88.4	6282.6	(30.50)	(26.34)	128.7	174.0	95.8	678.3	897.1
	5	32204	108.3	114.9	66.9	69.0	1777.4	(29.16)	(29.06)	**28.8**	35.6	35.2	344.2	964.3
	6	31286	109.9	104.0	94.4	92.5	4409.4	(27.11)	(27.11)	25.3	29.4	**19.8**	133.2	2321.2
rat99	3	564	6309.9	6577.2	**474.8**	494.0	5538.6	(24.30)	(25.36)	(5.15)	(4.25)	5019.1	(9.89)*	4448.0
	4	489	3481.0	6071.4	464.9	**446.7**	(13.31)	(25.02)	(25.25)	3759.5	4864.1	1360.6	(9.13)*	4626.6
	5	466	6767.3	9325.9	**653.9**	660.2	(14.31)	(33.99)	(34.11)	7729.4	9822.5	2105.6	(9.60)*	(6.30)
	6	454	(1.56)	(1.26)	**756.5**	834.7	(15.56)	(29.84)	(29.84)	8026.8	(2.31)	5032.0	(12.00)*	(11.76)

Table 7: Detailed results for smaller instances and grid clustering with $\mu = 7$. Besides solution values (Best), CPU-times in seconds are reported whenever an instance is solved to proven optimality by the corresponding method while values in parenthesis are optimality gaps in case of termination due to time or memory (indicated by *) limit.

Inst.	H	Best	$L^{+}_{x,\bar{n}}$	$L^{c,+}_{x}$	$L^{c,++}_{x,\bar{ln}}$	$L^{c,++}_{x}$	L_{y}	$L^{c,+}_{y,\bar{ln}}$	$L^{c,+}_{y,}$	$CL^{+}_{x,\bar{n}}$	CL^{+}_{x}	$CL^{c,+}_{x}$	CL_{y}	CL^{c}_{y}
att48	3	7082	**0.2**	**0.2**	**0.2**	**0.2**	1.5	311.4	3.6	0.3	0.3	0.3	0.6	0.7
	4	6867	0.5	0.4	0.4	0.4	2.2	41.7	45.9	**0.3**	**0.3**	**0.3**	0.7	1.0
	5	6667	0.5	0.5	0.5	0.5	2.4	184.5	198.3	**0.3**	**0.3**	**0.3**	0.7	0.8
	6	6667	0.8	0.9	0.8	0.8	2.0	9.7	21.3	**0.3**	**0.3**	**0.3**	0.9	1.0
eil51	3	100	**0.6**	0.7	0.7	**0.6**	4.0	2287.1	212.2	0.7	0.7	0.7	1.6	1.6
	4	100	1.6	1.5	1.5	1.6	10.2	(13.32)	8632.6	0.9	**0.8**	1.2	2.3	3.8
	5	100	1.9	2.0	1.9	2.0	13.6	(14.33)	(7.92)	**0.8**	**0.8**	**0.8**	2.4	3.6
	6	100	4.3	4.5	4.4	4.5	11.9	(9.89)	(12.32)	**1.0**	1.1	**1.0**	2.4	5.2
eil76	3	171	**33.0**	37.9	46.8	48.4	302.0	(26.03)	(20.82)	59.3	87.3	37.2	185.4	207.5
	4	162	69.0	83.9	107.0	116.6	1240.4	(30.47)	(30.47)	37.9	50.4	**28.3**	293.6	400.1
	5	157	84.8	100.8	90.4	101.5	1468.1	(30.08)	(30.08)	28.1	37.6	**26.2**	291.4	955.8
	6	152	61.4	61.7	74.3	66.9	1782.1	(23.07)	(23.07)	**8.5**	**8.5**	14.4	143.0	311.1
st70	3	249	**2.8**	2.9	3.0	3.0	28.9	3529.1	2819.6	5.0	5.2	5.1	12.5	11.4
	4	233	6.8	7.0	**6.3**	6.5	87.0	(12.01)	(8.15)	13.2	17.1	10.2	13.1	48.1
	5	228	25.1	26.0	25.7	26.0	277.0	(19.22)	(19.01)	**10.6**	12.4	13.2	29.9	31.8
	6	222	27.3	26.4	27.5	27.1	652.5	(20.52)	(20.52)	**7.2**	7.4	9.3	19.6	61.9
gr96	3	204	**34.3**	35.8	42.0	42.3	938.6	(24.95)	(21.23)	76.7	89.0	83.9	277.9	211.5
	4	192	98.8	99.4	121.4	127.7	1420.8	(31.33)	(31.33)	**46.1**	52.4	54.9	476.0	575.5
	5	190	111.6	118.8	330.2	327.5	6090.5	(30.32)	(30.32)	**28.2**	29.9	29.2	725.9	2472.0
	6	187	147.0	144.1	148.4	140.4	3740.9	(25.26)	(25.26)	**23.5**	23.8	26.8	776.3	(1.87)
pr76	3	37376	23.9	27.2	16.3	**16.2**	200.2	(21.96)	(20.11)	59.9	80.9	77.7	84.2	107.7
	4	34399	128.9	136.0	**87.4**	**87.4**	6463.1	(30.50)	(26.34)	129.2	175.0	96.0	736.3	902.6
	5	32204	107.9	114.7	67.7	63.5	2023.7	(29.05)	(29.16)	**28.6**	35.6	35.3	347.4	967.5
	6	31286	107.4	104.2	100.0	92.3	4882.1	(27.11)	(27.11)	25.4	29.4	**19.8**	137.1	2260.4
rat99	3	395	572.8	711.5	**203.6**	206.6	1251.1	(31.56)	(36.25)	593.6	808.4	471.0	1855.1	1041.1
	4	361	2148.7	2190.3	947.5	1028.0	(14.45)*	(41.38)	(41.23)	2115.3	2763.1	**706.0**	(15.23)*	(5.99)*
	5	340	1373.8	1526.3	544.6	614.2	(18.21)	(38.29)	(38.41)	671.2	1098.8	**495.1**	(11.52)*	(15.11)*
	6	328	726.3	876.8	526.3	519.9	(14.00)	(37.00)	(37.00)	224.5	296.2	**204.4**	(16.45)*	(11.31)*

Table 8: Detailed results for smaller instances and grid clustering with $\mu = 10$. Besides solution values (Best), CPU-times in seconds are reported whenever an instance is solved to proven optimality by the corresponding method while values in parenthesis are optimality gaps in case of termination due to time or memory (indicated by *) limit.

Inst.	H	Best	$L^{+}_{x,\bar{n}}$	$L^{c,+}_{x}$	$L^{c,++}_{x,\bar{ln}}$	$L^{c,++}_{x}$	L_{y}	$L^{c,+}_{y,\bar{ln}}$	$L^{c,+}_{y,}$	$CL^{+}_{x,\bar{n}}$	CL^{+}_{x}	$CL^{c,+}_{x}$	CL_{y}	CL^{c}_{y}
att48	3	7082	**0.2**	**0.2**	**0.2**	**0.2**	1.5	274.6	3.8	0.3	0.3	0.3	0.7	0.7
	4	6867	0.4	0.4	0.4	0.4	2.1	41.3	47.7	**0.3**	**0.3**	**0.3**	0.7	0.9
	5	6667	0.5	0.5	0.5	0.5	2.3	188.9	209.0	**0.3**	**0.3**	**0.3**	0.7	0.8
	6	6667	0.8	0.8	0.8	0.8	2.1	9.9	21.6	**0.3**	**0.3**	**0.3**	0.8	1.0
eil51	3	100	**0.7**	**0.7**	**0.7**	**0.7**	4.0	2030.8	212.3	**0.7**	**0.7**	**0.7**	1.6	1.5
	4	100	1.5	1.5	1.6	1.6	10.6	(13.32)	7061.3	**0.8**	0.9	1.3	2.3	3.8
	5	100	2.0	1.8	1.9	2.0	14.9	(14.33)	(7.92)	**0.8**	**0.8**	**0.8**	2.4	3.7
	6	100	4.5	4.4	4.3	4.5	11.8	(9.89)	(11.29)	**1.0**	1.1	1.1	2.4	5.1
eil76	3	101	3.5	3.6	**3.2**	**3.2**	49.4	(32.15)	(30.36)	4.8	5.1	5.6	17.4	16.1
	4	95	4.2	4.0	4.0	4.0	176.6	(30.12)	(30.12)	**2.8**	**2.8**	**2.8**	21.5	39.0
	5	94	14.3	14.2	14.1	14.2	305.7	(31.73)	(31.89)	**3.2**	**3.2**	3.3	30.1	64.2
	6	94	22.8	22.6	23.0	22.9	319.5	(33.43)	(35.90)	**2.7**	**2.7**	**2.7**	36.6	95.2
st70	3	149	2.3	2.3	3.0	3.0	19.8	(34.34)	2999.0	**1.9**	2.0	**1.9**	4.7	3.7
	4	148	3.7	3.8	3.8	3.8	31.5	(23.70)	(23.70)	**3.0**	3.1	3.3	5.5	7.1
	5	147	8.3	8.4	8.9	9.0	66.0	(28.33)	(28.33)	1.6	**1.5**	1.8	11.2	10.6
	6	147	16.3	16.4	17.1	17.1	148.6	(26.30)	(21.96)	**2.6**	**2.6**	**2.6**	6.3	12.9
gr96	3	204	**33.6**	36.5	41.3	41.8	942.2	(24.95)	(20.85)	75.2	89.3	83.4	269.9	221.3
	4	192	92.4	97.9	118.5	120.6	1452.2	(31.33)	(31.33)	**46.8**	54.5	53.7	462.5	570.4
	5	190	123.2	120.7	351.2	345.8	6028.2	(30.32)	(30.32)	**27.3**	29.7	29.2	738.2	2333.6
	6	187	146.1	138.4	147.0	154.1	4049.5	(25.26)	(25.26)	**23.7**	24.2	26.7	759.8	(2.72)
pr76	3	23395	4.6	4.8	**4.0**	**4.0**	45.1	(22.90)	(15.06)	8.8	9.3	10.2	16.5	11.9
	4	22111	10.1	10.6	**8.3**	8.4	161.0	(29.05)	(27.70)	8.6	8.8	9.4	31.2	24.7
	5	21068	16.1	15.7	16.1	15.7	449.5	(30.36)	(30.36)	**4.2**	4.3	**4.2**	20.0	21.4
	6	20743	26.8	26.6	26.2	25.4	371.4	(26.65)	(26.66)	**4.6**	**4.6**	**4.6**	17.2	34.6
rat99	3	395	588.3	753.2	**204.4**	207.4	1343.3	(32.99)	(30.36)	582.5	837.7	476.5	1832.7	1080.8
	4	361	2043.6	2303.8	992.9	931.4	(14.45)*	(41.41)	(41.15)	1903.0	2901.2	**701.5**	(15.23)*	(5.99)*
	5	340	1584.8	1296.3	608.4	583.2	(18.52)	(38.42)	(38.43)	677.6	1094.7	**510.6**	(11.52)*	(15.11)*
	6	328	737.9	887.9	550.6	570.3	(14.71)	(37.00)	(37.00)	224.1	301.5	**205.9**	(16.45)*	(11.78)

Table 9: Detailed results for larger instances and geographical clustering. Besides best solution values (Best) CPU-times in seconds are reported whenever an instance is solved to proven optimality by the corresponding method while values in parenthesis are optimality gaps in case of termination due to time or memory (indicated by *) limit.

Inst.	H	Best	$L^{+}_{x,\bar{n}}$	$L^{c,+}_{x}$	$L^{c,++}_{x,\bar{ln}}$	$L^{c,++}_{x}$	L_{y}	$L^{c,+}_{y,\bar{ln}}$	$L^{c,+}_{y}$	$CL^{+}_{x,\bar{n}}$	CL^{+}_{x}	$CL^{c,+}_{x}$	CL_{y}	CL^{c}_{y}
kroa100	3	9016	167.0	190.3	99.2	109.7	1036.7	(24.52)	(24.45)	**87.0**	108.6	133.1	350.7	882.5
	4	8525.0	2014.9	2126.0	411.5	438.4	(11.83)*	(28.03)	(28.03)	196.0	279.8	**163.5**	3711.1	3716.6
	5	8229.0	392.1	396.0	588.5	648.6	(12.97)*	(29.05)	(29.05)	**88.3**	110.2	139.4	5647.5	(4.95)*
	6	8047.0	319.3	309.9	490.5	758.3	(5.96)	(27.80)	(27.80)	**69.4**	79.2	106.3	2598.0	(2.79)
krob100	3	9937	569.5	771.5	**234.9**	245.6	1941.1	(30.11)	(24.17)	281.4	379.2	580.9	1404.7	1407.2
	4	9234.0	(1.02)	(1.26)	1108.3	1190.1	(13.70)*	(35.77)	(35.79)	955.0	1249.8	**602.7**	(7.76)*	6380.2
	5	8548.0	1044.5	1066.3	809.5	767.1	(8.74)	(32.11)	(32.11)	182.1	253.0	**106.3**	2524.5	3047.1
	6	8339.0	496.3	699.8	703.1	766.9	(6.09)	(27.09)	(27.09)	**73.4**	87.1	76.8	2338.3	(1.10)
kroc100	3	9389	48.9	60.6	42.8	**42.1**	3357.2	(20.33)	(24.02)	106.6	136.5	106.5	(0.88)	2331.7
	4	8417.0	55.3	50.1	42.3	53.5	7425.8	(30.50)	(30.51)	23.7	25.9	**21.4**	4004.8	2330.7
	5	8126.0	61.5	61.9	63.0	63.1	(3.01)	(31.29)	(31.29)	15.5	**15.4**	**15.4**	1772.8	5405.5
	6	8041.0	113.7	110.2	109.0	112.1	(7.70)	(31.94)	(31.94)	15.1	**14.7**	**14.7**	3271.3	6845.5
krod100	3	9339	355.7	405.9	117.0	**115.1**	(8.83)*	(18.62)	(20.41)	341.7	404.8	172.2	826.5	1045.9
	4	8393.0	444.4	488.3	253.6	249.1	(8.56)	(29.96)	(29.89)	168.8	213.6	**131.4**	(4.49)*	1968.9
	5	8075.0	424.9	488.1	461.1	480.7	(9.61)*	(27.83)	(27.83)	**178.8**	215.7	226.4	(7.26)*	3980.8
	6	7868.0	241.7	242.3	344.9	367.4	(7.61)	(22.19)	(22.19)	**81.3**	97.2	101.9	3941.6	4512.2
kroe100	3	10194	376.1	486.4	100.7	**96.7**	1165.9	(18.24)	(21.00)	349.7	453.1	204.1	825.0	370.5
	4	9457.0	8859.0	(0.87)	619.8	652.8	(8.09)*	(23.14)	(23.91)	1161.0	1453.4	**302.2**	(6.95)*	4515.9
	5	9008.0	5087.1	7401.8	719.0	761.1	(13.11)	(30.31)	(30.31)	834.0	1147.7	**396.0**	(13.14)*	8675.5
	6	8814.0	6661.4	7264.6	2382.8	2043.2	(11.09)*	(31.76)	(31.76)	613.7	889.1	**443.4**	(10.38)*	(7.97)
rd100	3	3321	**39.7**	43.6	41.4	41.2	1199.7	(19.00)	(17.90)	92.6	123.3	119.4	552.6	367.6
	4	3033.0	96.3	103.2	97.2	95.2	(8.13)*	(31.17)	(31.17)	178.0	241.9	**76.0**	(3.98)*	717.1
	5	2952.0	202.3	206.4	172.5	178.7	(8.18)	(25.54)	(25.53)	160.9	204.1	**100.1**	(8.44)*	7367.0
	6	2857.0	157.2	177.8	185.1	175.1	(12.64)	(26.91)	(26.91)	**46.8**	49.4	56.1	1171.0	5341.0
eil101	3	234	**195.1**	229.0	270.3	279.4	(8.69)*	(21.49)	(34.67)	407.0	485.3	504.3	2844.4	3594.9
	4	221.0	545.4	656.5	838.9	950.1	(16.30)*	(27.48)	(27.48)	515.3	606.2	**384.1**	(15.97)*	(10.10)*
	5	214.0	660.5	794.8	789.4	845.1	(10.55)	(30.19)	(30.19)	**178.8**	215.2	199.1	(8.93)*	(9.77)
	6	212.0	1085.7	1407.7	1866.6	2284.4	(10.06)	(30.28)	(30.28)	111.1	138.7	**101.2**	(7.12)*	(13.23)
pr107	3	27574	(6.83)	(7.53)	980.9	921.8	496.9	(5.55)	(4.18)	613.1	736.6	**203.4**	490.7	263.1
	4	24818.0	(8.15)	(8.35)	2837.8	2846.8	5048.7	(13.96)	(15.93)	5157.3	6719.8	**313.1**	(4.85)*	1002.1
	5	22970.0	(4.44)	(4.14)	3716.8	3476.6	(5.56)	(23.95)	(23.95)	788.0	969.7	**257.0**	(4.79)*	5304.3
	6	21723.0	7670.2	8719.0	1881.6	1642.4	2846.9	(11.47)	(11.47)	382.2	472.4	**237.2**	(3.17)*	3173.1
pr124	3	40583	(3.10)	(3.60)	2345.2	2192.3	8714.8	(12.59)	(16.91)	4154.1	5770.2	**539.3**	(6.68)*	3503.5
	4	36115.0	(5.96)	(6.47)	3399.3	3349.8	(13.44)*	(27.80)	(27.99)	9878.3	(2.40)	**1255.4**	(13.68)*	(4.25)*
	5	34057.0	(6.94)	(7.21)	4822.5	5186.9	(14.84)	(28.54)	(28.54)	(2.88)	(3.37)	**1869.3**	(13.95)*	(7.95)*
	6	32692.0	(4.68)	(4.81)	3600.7	3715.6	(12.86)	(22.02)	(22.02)	(2.55)	(3.10)	**1600.7**	(14.06)*	(10.05)*
bier127	3	65133	5726.4	9213.2	(2.46)	(2.89)	604.7	(4.73)	8314.0	**233.9**	323.1	239.4	480.4	457.2
	4	60228.0	1832.8	3143.2	2200.5	2727.4	961.3	(6.01)	(5.97)	94.7	124.7	**79.3**	292.4	269.2
	5	58905.0	4601.8	6133.1	5998.8	7749.5	2942.4	(7.06)	(6.78)	**79.5**	100.2	107.9	452.1	732.6
	6	58343.0	9360.0	(1.43)	(1.54)	(2.03)	(1.98)*	(8.62)	(8.62)	**63.3**	74.1	79.2	1064.5	1022.3
gr137	3	504	(15.05)	(14.89)	(3.51)	**(3.05)**	(18.97)	(37.45)	(37.49)	(12.47)	(12.36)	(9.21)	(21.59)*	(11.07)*
	4	451.0	(21.53)	(21.53)	(15.43)	(15.86)	(33.50)	(39.24)	(39.79)	(17.29)	(18.44)	**(9.56)**	(29.80)*	(16.62)*
	5	416.0	(21.08)	(21.08)	(17.15)	(16.93)	(35.04)	(38.11)	(38.13)	(20.40)	(20.40)	**(6.16)**	(31.26)*	(14.97)
	6	395.0	(17.89)	(17.89)	(12.21)	(11.59)	(36.81)	(39.60)	(39.61)	(15.39)	(16.01)	**(4.67)**	(32.41)*	(18.20)
pr144	3	51644	(4.56)	(4.58)	3162.6	3095.0	3296.3	(7.75)	(8.38)	464.1	623.5	**350.4**	6182.4	1518.5
	4	46594.0	(7.31)	(6.91)	(2.09)	(2.03)	(6.82)	(21.50)	(21.50)	7162.0	8715.5	**1554.5**	(5.35)*	7594.3
	5	43697.0	(4.30)	(4.30)	6510.2	7109.6	(9.80)	(18.61)	(18.61)	(1.09)	(0.99)	**798.7**	(6.13)*	(1.66)*
	6	42634.0	(3.96)	(3.71)	8559.2	8818.2	(6.43)	(16.08)	(16.08)	(1.22)	(1.69)	**1029.4**	(8.30)*	(2.29)*
kroa150	3	12150	(4.53)	(4.76)	3192.0	**2730.2**	(17.17)*	(41.46)	(41.46)	(2.66)	(4.51)	8019.0	(15.54)*	(13.56)*
	4	11170.0	(4.46)	(4.72)	8311.0	**6850.9**	(24.75)	(42.63)	(42.63)	(5.51)	(5.96)	(3.54)	(20.14)*	(15.67)*
	5	10699.0	(3.55)	(3.55)	**(1.40)**	**(1.40)**	(24.93)	(37.39)	(37.39)	(3.05)	(3.54)	(2.51)	(21.50)*	(17.14)*
	6	10290.0	(1.95)	(3.53)	(2.43)	(2.43)	(33.86)	(37.18)	(37.18)	**5756.6**	9075.9	6036.5	(22.43)*	(18.40)
krob150	3	13022	(2.59)	(3.73)	**2193.9**	2373.3	(18.57)*	(42.05)	(42.05)	(6.77)	(8.41)	(4.41)	(21.80)*	(15.29)*
	4	11760.0	(6.59)	(6.59)	5943.9	**5534.4**	(28.88)	(44.95)	(44.95)	(5.67)	(6.43)	(3.83)	(24.08)*	(17.87)*
	5	11160.0	(6.63)	(6.07)	8989.9	8695.9	(28.07)	(41.96)	(41.96)	(6.79)	(6.15)	**7459.7**	(19.94)*	(20.15)*
	6	10765.0	(8.39)	(7.48)	(1.60)	(1.58)	(28.09)	(33.77)	(33.77)	(4.16)	(3.65)	**4480.3**	(24.77)*	(20.97)
pr152	3	53825	(7.63)	(8.30)	4124.4	4855.7	(7.58)*	(11.17)	(10.80)	(7.04)	(7.54)	**1613.5**	(5.37)*	1973.0
	4	46053.0	(5.81)	(6.01)	8043.4	7599.8	(9.50)	(30.17)	(30.17)	(3.34)	(3.84)	**1488.2**	(7.92)*	2608.4
	5	44033.0	(10.79)	(10.79)	(2.27)	(0.23)	(14.29)	(40.97)	(40.97)	(6.88)	(6.80)	**2953.6**	(13.49)*	(4.54)*
	6	42375.0	(11.41)	(11.41)	(4.84)	(4.02)	(13.32)	(25.56)	(25.56)	(3.61)	(4.25)	**6072.1**	(8.85)	(3.66)
u159	3	26545	(8.10)	(8.38)	**6468.1**	7370.6	(21.20)*	(36.81)	(36.81)	(13.22)	(13.78)	(9.28)	(17.42)*	(12.38)*
	4	23144.0	(11.34)	(11.34)	**8391.1**	9253.5	(27.01)	(42.56)	(42.56)	(10.49)	(9.15)	(6.77)	(22.89)*	(11.63)*
	5	21829.0	(11.53)	(10.36)	(5.28)	(5.24)	(28.19)	(50.32)	(50.32)	(10.38)	(10.07)	**(4.45)**	(25.59)*	(16.65)
	6	20447.0	(5.84)	(5.27)	(11.91)	(12.88)	(37.28)	(48.63)	(48.63)	(5.27)	(6.03)	**7060.2**	(22.94)*	(18.22)

Table 10: Detailed results for larger instances and grid clustering with $\mu = 3$. Besides best solution values (Best) CPU-times in seconds are reported whenever an instance is solved to proven optimality by the corresponding method while values in parenthesis are optimality gaps in case of termination due to time or memory (indicated by *) limit.

Inst.	H	Best	$L^{+}_{x,\bar{n}}$	$L^{c,+}_{x}$	$L^{c,++}_{x,\bar{ln}}$	$L^{c,++}_{x}$	L_{y}	$L^{c,+}_{y,\bar{ln}}$	$L^{c,+}_{y,}$	$CL^{+}_{x,\bar{n}}$	CL^{+}_{x}	$CL^{c,+}_{x}$	CL_{y}	CL^{c}_{y}
kroa100	3	15651	241.9	386.8	72.3	**71.7**	3990.7	(2.44)	6757.3	197.1	309.6	165.2	529.9	173.2
	4	13831.0	419.3	650.0	**142.8**	147.4	(0.66)	(5.30)	(6.17)	1541.3	2475.6	201.1	3089.7	342.5
	5	13133.0	1315.0	1960.5	308.7	295.2	(4.03)*	(17.11)	(17.24)	821.3	1289.3	**257.6**	5693.5	1289.6
	6	12736.0	522.0	562.5	381.0	385.5	(2.01)	(13.67)	(13.57)	823.5	1227.7	**211.3**	4667.6	2511.7
krob100	3	17200	3911.3	6533.9	**305.1**	309.2	(8.35)*	(8.81)	(6.94)	(1.15)	(2.64)	1431.4	(3.80)*	5745.5
	4	15198.0	(2.55)	(3.07)	292.6	**285.1**	(10.05)*	(9.91)	(18.80)	(2.40)	(3.51)	1049.5	(7.63)*	(0.23)
	5	14438.0	(4.08)	(4.42)	1436.7	**1348.0**	(11.84)*	(16.51)	(16.44)	(5.07)	(5.56)	3524.2	(10.55)*	(2.60)
	6	13880.0	(2.31)	(2.81)	1538.4	**1463.0**	(10.16)*	(18.55)	(18.50)	(3.27)	(4.06)	1835.7	(9.01)*	(4.32)
kroc100	3	16775	4792.9	6498.6	**187.2**	205.0	(3.61)*	(6.66)	(5.55)	2348.4	3958.8	374.9	5165.0	282.2
	4	14916.0	(2.81)	(2.99)	557.2	600.2	(6.73)*	(5.86)	(7.12)	(1.26)	(2.71)	**335.2**	(3.66)	1273.9
	5	13911.0	(2.27)	(2.59)	967.8	1013.3	(5.87)	(12.10)	(18.45)	(1.86)	(2.81)	**307.2**	(3.86)*	401.3
	6	13362.0	(1.34)	(1.59)	1015.6	1032.2	(3.64)*	(5.45)	(5.23)	3531.5	5184.1	**414.9**	(2.55)*	1249.6
krod100	3	16809	1564.0	2141.4	**197.2**	201.0	(3.27)	(6.08)	(4.09)	3129.7	4562.5	579.3	4013.2	649.6
	4	14861.0	(3.83)	(4.01)	965.8	1072.6	(10.26)*	(10.20)	(9.10)	(4.21)	(5.51)	**809.6**	(6.77)*	3271.3
	5	13874.0	(4.41)	(4.60)	1206.9	1437.3	(8.65)	(10.42)	(10.39)	(4.28)	(5.54)	**550.7**	(7.70)*	3647.2
	6	13284.0	(3.88)*	(4.20)	1727.3	1639.2	(7.75)*	(9.80)	(9.19)	(2.90)	(3.69)	**609.4**	(6.06)*	(2.92)
kroe100	3	17201	3053.4	4923.2	93.9	**93.7**	(5.08)*	(6.57)	(6.06)	2653.8	4391.1	560.5	4828.8	724.8
	4	15212.0	(4.03)	(4.65)	423.5	**390.7**	(10.09)*	(18.56)	(18.46)	8725.5	(1.75)	792.1	(6.35)*	6176.4
	5	14225.0	(4.22)	(4.52)	957.4	1074.0	(9.89)*	(15.74)	(15.62)	(3.51)	(4.83)	**865.0**	(7.24)*	8388.8
	6	13681.0	(3.98)	(4.19)	1338.1	1392.5	(8.99)*	(14.08)	(14.29)	(2.85)	(3.58)	**938.2**	(7.45)*	7452.0
rd100	3	5564	1073.9	1795.0	158.6	**156.0**	(6.00)*	(10.65)	(9.28)	2229.2	3562.4	264.6	6180.8	2892.7
	4	4917.0	(1.30)	(2.18)	348.2	**347.1**	(9.10)*	(21.44)	(21.44)	6891.3	(1.59)	443.0	(4.80)*	3006.1
	5	4592.0	9026.3	(0.79)	801.9	691.5	(8.87)	(18.42)	(18.26)	(0.61)	(2.21)	**437.3**	(7.70)*	4768.4
	6	4371.0	1408.6	1974.6	**273.9**	278.6	(5.42)	(18.51)	(18.09)	1630.3	2655.1	318.7	(2.40)	3056.5
eil101	3	351	693.5	785.3	**446.7**	479.9	(5.95)	(15.35)	(13.73)	(0.95)	(2.09)	(1.84)	(4.03)	(5.62)*
	4	324.0	1934.8	2728.8	**1073.3**	1089.5	(9.32)	(22.77)	(22.76)	(1.80)	(2.52)	4820.7	(9.95)*	(6.66)
	5	311.0	1851.8	**1189.4**	1426.5	1572.0	(10.07)	(17.47)	(17.55)	2607.9	3640.8	3178.3	(7.74)*	(5.67)
	6	306.0	1617.1	2194.2	2156.7	2737.2	(8.83)	(19.43)	(19.43)	**1226.2**	1927.6	1457.1	(7.68)*	(8.40)
pr107	3	36037	(8.19)	(9.04)	879.3	873.5	(8.82)*	6074.8	5249.3	9333.9	(2.54)	165.5	1715.6	**51.1**
	4	31195.0	(16.10)*	(15.69)	2383.1	2250.9	(14.72)*	(4.06)	(1.67)	(12.05)	(13.13)	367.0	(10.81)*	**143.6**
	5	29121.0	(15.50)	(15.67)	4149.4	3003.2	(17.52)*	(13.36)	(13.37)	(15.79)*	(16.40)	515.3	(16.97)*	**487.8**
	6	27501.0	(13.14)	(12.65)	5758.0	5843.3	(13.93)*	(12.04)	(12.83)	(11.62)*	(11.97)	**663.2**	(12.46)*	747.5
pr124	3	52279	5045.9	7111.2	595.5	610.6	(3.97)*	(0.82)	(0.77)	2953.8	4835.7	**211.2**	6838.5	385.7
	4	46911.0	(7.65)	(7.91)	1500.8	1314.9	(12.16)*	(21.91)	(21.88)	(5.70)	(6.70)	**654.6**	(8.69)*	2495.2
	5	44119.0	(6.99)	(6.92)	2186.7	2266.7	(10.13)	(12.29)	(13.08)	(5.74)	(6.64)	**758.8**	(10.55)*	9294.2
	6	41999.0	(5.25)	(5.41)	3776.0	3883.5	(9.01)	(17.53)	(17.48)	(4.57)	(4.93)	**655.5**	(9.49)*	(1.87)
bier127	3	89013	**1459.6**	2048.4	2818.1	3454.6	(6.17)*	(8.96)	(7.30)	(1.09)	(1.66)	(0.68)	(4.77)*	(3.06)*
	4	81024.0	(1.67)	(2.23)	**3422.3**	4101.0	(7.39)*	(10.51)	(11.29)	(2.00)	(2.49)	6267.0	(4.91)*	(4.30)*
	5	76793.0	(2.11)	(2.24)	2939.9	3684.6	(6.04)*	(13.02)	(13.01)	(0.65)	(1.24)	**1989.3**	(5.12)*	(3.12)*
	6	74595.0	(1.79)	(2.30)	2688.5	3540.2	(5.70)	(14.61)	(14.63)	(0.75)	(1.15)	**2187.0**	(4.57)*	(2.70)
gr137	3	651	(12.66)	(12.73)	(1.39)	**(1.21)**	(25.51)*	(41.06)	(41.03)	(20.24)	(21.22)	(9.23)	(19.38)	(10.53)
	4	572.0	(23.08)	(23.08)	(11.82)	(12.65)	(36.21)	(51.34)	(52.00)	(23.19)	(23.94)	**(10.55)**	(31.21)*	(14.46)
	5	529.0	(26.58)	(26.58)	(20.26)	(19.60)	(32.16)	(44.20)	(43.65)	(20.34)	(20.34)	**(9.89)**	(30.16)*	(17.15)
	6	503.0	(22.64)	(22.64)	(22.42)	(23.19)	(35.33)	(40.53)	(40.60)	(18.46)	(19.91)	**(8.13)**	(32.66)*	(19.24)
pr144	3	61456	(7.03)	(7.29)	3534.4	3621.3	(7.65)*	(4.31)	(3.53)	(1.76)	(2.67)	**1612.4**	(4.63)*	3059.1
	4	53301.0	(9.82)	(9.70)	(1.45)	(2.50)	(13.27)*	(19.92)	(20.19)	(6.49)	(7.20)	**4169.8**	(10.71)*	(0.45)
	5	49232.0	(11.13)	(9.89)	(10.71)	(10.56)	(13.36)*	(20.34)	(20.98)	(6.91)	(7.03)	**1472.1**	(8.78)*	(1.01)
	6	47284.0	(5.84)	(6.11)	(13.50)	(13.78)	(11.95)	(22.07)	(22.07)	(3.20)	(3.88)	**1663.1**	(8.03)*	(2.48)
kroa150	3	19073	(4.92)	(5.37)	**2681.1**	2721.9	(13.35)	(19.39)	(19.47)	(8.87)	(10.27)	(5.15)	(12.21)*	(6.99)
	4	17041.0	(7.77)	(7.84)	**6887.3**	7045.5	(18.09)	(31.35)	(31.44)	(10.23)	(10.36)	(4.65)	(15.72)*	(9.31)
	5	15913.0	(6.04)	(6.17)	6645.5	**6643.8**	(15.39)	(26.54)	(26.44)	(7.19)	(7.60)	(1.25)	(13.39)*	(8.44)
	6	15270.0	(5.16)	(5.88)	**6695.6**	7714.3	(17.49)	(25.23)	(25.23)	(5.45)	(5.69)	(1.13)	(13.78)*	(10.86)
krob150	3	20310	(7.41)	(7.61)	5073.2	**4220.3**	(17.42)*	(25.62)	(25.70)	(10.60)	(12.17)	(6.88)	(12.91)	(7.43)
	4	17605.0	(10.80)	(11.96)	7926.1	**7084.2**	(18.75)	(30.56)	(29.90)	(10.87)	(13.51)	(5.22)	(18.28)*	(7.47)
	5	16326.0	(7.55)	(8.84)	**8857.0**	9116.1	(19.76)	(31.24)	(31.28)	(8.48)	(9.04)	(1.43)	(16.47)*	(9.83)
	6	15728.0	(7.67)	(7.67)	(25.60)	(25.44)	(22.13)	(29.42)	(29.42)	(8.01)	(8.71)	**(4.35)**	(12.31)	(10.49)
pr152	3	73072	(16.05)	(16.05)	5781.3	6741.2	(14.60)*	(14.64)	(14.03)	(10.94)	(11.63)	2943.8	(8.90)	**1020.1**
	4	58836.0	(20.39)	(20.43)	(17.71)	(17.12)	(21.90)*	(26.43)	(26.71)	(12.33)	(12.37)	**4761.2**	(11.84)	7457.9
	5	53112.0	(15.67)	(15.99)	(25.21)	(25.23)	(15.38)*	(29.23)	(29.37)	(11.55)	(11.20)	**3080.7**	(13.33)*	(1.11)
	6	50502.0	(14.18)	(14.18)	(19.18)	(19.89)	(13.72)	(29.85)	(30.23)	(8.14)	(8.47)	**5050.5**	(10.67)*	(0.64)
u159	3	37096	(9.75)	(10.57)	5798.5	**5456.4**	(18.88)*	(29.71)	(29.91)	(12.50)	(12.73)	(4.72)	(16.83)*	(5.45)
	4	31808.0	(16.40)	(16.40)	(8.94)	(10.91)	(26.11)	(31.00)	(30.71)	(15.73)	(17.63)	**(2.40)**	(23.04)*	(10.71)
	5	29863.0	(19.23)	(17.33)	(13.92)	(14.27)	(29.73)	(45.66)	(45.79)	(18.00)	(17.68)	**(3.97)**	(24.81)*	(14.62)
	6	28860.0	(20.84)	(20.84)	(17.68)	(16.90)	(33.22)	(33.78)	(33.72)	(13.74)	(14.44)	**(4.53)**	(23.89)*	(17.65)

Table 11: Detailed results for larger instances and grid clustering with $\mu = 5$. Besides best solution values (Best) CPU-times in seconds are reported whenever an instance is solved to proven optimality by the corresponding method while values in parenthesis are optimality gaps in case of termination due to time or memory (indicated by *) limit.

Inst.	H	Best	$L^{+}_{x,\bar{n}}$	$L^{c,+}_{x}$	$L^{c,++}_{x,\bar{ln}}$	$L^{c,++}_{x}$	L_{y}	$L^{c,+}_{y,\bar{ln}}$	$L^{c,+}_{y,}$	$CL^{+}_{x,\bar{n}}$	CL^{+}_{x}	$CL^{c,+}_{x}$	CL_{y}	CL^{c}_{y}
kroa100	3	9379	**58.7**	67.1	60.8	63.8	685.6	(13.26)	(13.12)	77.0	84.6	94.0	87.3	67.1
	4	8798.0	404.8	465.8	177.8	179.9	(5.48)*	(19.22)	(27.18)	260.1	347.0	**120.1**	1083.9	567.6
	5	8504.0	422.0	760.3	251.4	258.5	(6.67)	(25.32)	(25.30)	218.8	290.5	**200.3**	3322.0	1844.8
	6	8333.0	498.4	552.6	809.0	641.5	(4.30)	(27.47)	(27.47)	**164.7**	234.2	367.6	2604.6	(2.41)
krob100	3	9837	277.3	447.9	**118.3**	126.3	492.0	(19.15)	(12.58)	188.3	302.5	132.2	129.4	198.2
	4	8943.0	501.8	702.9	345.4	384.0	2961.7	(20.27)	(20.29)	307.9	463.0	**219.7**	423.1	785.7
	5	8603.0	1233.6	1512.9	396.3	334.8	8307.2	(19.49)	(19.49)	628.8	981.4	**211.3**	1214.9	882.1
	6	8265.0	256.0	262.5	333.4	382.6	3671.3	(20.36)	(20.37)	**102.0**	138.3	157.4	219.1	1045.0
kroc100	3	10183	172.6	225.3	**132.0**	135.3	4672.7	(22.65)	(23.57)	367.6	554.8	357.9	4920.1	2771.5
	4	8943.0	255.8	297.4	**122.2**	125.0	(9.05)*	(43.09)	(43.59)	284.3	367.0	178.9	(1.14)	9636.2
	5	8534.0	386.8	459.1	305.1	285.4	(9.28)	(40.67)	(40.67)	**156.0**	211.9	177.4	(6.27)*	(8.51)
	6	8295.0	335.2	328.6	345.5	344.9	(11.84)	(32.91)	(32.91)	**89.3**	109.4	132.3	(6.07)*	(6.17)
krod100	3	10212	**18.1**	19.8	20.0	19.7	625.8	(16.89)	(13.70)	112.4	132.9	139.3	79.2	190.6
	4	9506.0	236.0	297.2	**129.9**	134.8	(3.82)*	(14.44)	(16.32)	331.9	505.1	131.1	3321.2	559.6
	5	9165.0	285.9	341.9	289.5	263.4	(5.27)	(18.41)	(18.40)	**138.4**	172.9	167.5	(4.25)*	2887.9
	6	8998.0	758.5	882.1	588.2	559.2	(4.01)	(21.68)	(21.68)	281.6	419.6	**162.0**	(1.14)	(4.23)
kroe100	3	11323	343.4	476.3	**118.2**	124.0	657.4	(14.77)	(15.36)	228.3	308.8	179.2	308.9	466.0
	4	10019.0	919.5	1103.0	**248.5**	278.3	(4.79)	(21.25)	(21.53)	1264.9	1750.2	274.2	3072.0	278.4
	5	9364.0	1019.5	1379.5	287.4	255.4	(5.38)*	(18.97)	(17.75)	311.4	451.3	**166.8**	(4.28)*	452.6
	6	9035.0	305.2	368.3	213.1	229.2	(1.78)	(21.69)	(21.69)	166.6	238.0	**99.9**	2897.4	1260.5
rd100	3	4047	594.2	771.2	116.4	**116.1**	2702.1	(17.11)	(16.09)	645.0	983.6	277.7	1773.5	528.0
	4	3652.0	3693.7	4432.9	**157.8**	167.6	(4.24)	(22.02)	(21.83)	1561.1	2403.6	208.5	(4.71)*	6306.9
	5	3341.0	76.3	88.4	84.4	76.0	(5.01)	(19.60)	(19.96)	140.3	208.8	**68.0**	2480.5	1849.5
	6	3234.0	145.2	158.7	143.3	140.8	6729.5	(11.51)	(11.51)	**41.0**	48.7	73.1	632.4	2801.4
eil101	3	257	343.2	427.0	300.4	**298.4**	(1.71)	(41.21)	(41.21)	1090.4	1368.5	924.5	1110.5	4034.8
	4	241.0	1098.1	1175.0	**828.7**	899.0	(14.29)*	(35.93)	(35.93)	1273.1	2169.9	1048.1	(8.90)*	(6.94)*
	5	230.0	624.4	665.7	727.2	693.9	(11.62)	(26.93)	(26.93)	595.8	1043.6	**195.8**	(7.00)*	(9.46)
	6	228.0	554.0	583.2	1669.8	1709.1	(11.32)	(29.23)	(29.23)	421.0	634.1	**213.1**	(7.29)*	(13.61)
pr107	3	27086	(4.96)*	(5.20)	374.0	377.6	55.4	5028.2	2802.5	110.5	122.6	121.8	32.0	**10.6**
	4	24782.0	(8.13)	(7.80)	2831.7	3029.7	3203.0	(4.06)	(3.94)	566.4	733.8	252.4	527.1	**156.5**
	5	23044.0	(6.65)	(6.73)	3269.6	3826.7	(1.61)*	(9.29)	(8.49)	379.0	481.0	**130.2**	1511.1	332.3
	6	21838.0	(3.61)	(3.65)	4068.7	4653.6	1716.9	(9.90)	(9.77)	199.3	252.1	**85.9**	1413.2	166.0
pr124	3	36124	1503.6	2200.3	267.1	241.8	2998.8	(13.10)	(14.03)	312.8	411.3	**196.4**	1796.6	458.3
	4	32395.0	6459.1	6518.8	758.6	703.1	(2.53)	(19.46)	(19.73)	1925.8	3004.1	**321.7**	(8.86)*	4464.3
	5	30522.0	5748.9	6965.1	591.5	556.4	(7.16)	(27.47)	(27.46)	2647.2	4029.1	**455.2**	(5.48)*	2095.4
	6	29284.0	1736.6	1832.5	936.6	995.1	(7.07)	(13.48)	(13.48)	303.2	500.5	**88.1**	7683.6	(4.18)*
bier127	3	65227	(1.67)	(2.96)	(2.96)	(2.73)	8430.0	(11.54)	(7.34)	565.6	825.9	**430.8**	1788.9	992.9
	4	60944.0	2796.0	4585.7	5668.7	7636.5	6075.9	(15.28)	(14.81)	**162.2**	215.0	247.2	(2.70)*	2083.0
	5	59934.0	5078.0	5649.8	5038.9	7015.5	(2.45)*	(12.74)	(12.74)	**146.3**	188.5	271.9	(2.14)*	9565.1
	6	59544.0	7188.4	9406.0	4405.9	6119.1	(2.69)	(10.68)	(10.68)	**110.3**	147.1	149.3	4961.1	(1.32)
gr137	3	521	(12.69)	(12.69)	**(1.48)**	(1.65)	(24.58)	(41.70)	(41.71)	(17.17)	(15.27)	(7.77)	(17.72)*	(11.72)
	4	450.0	(21.80)	(21.80)	(16.46)	(16.41)	(36.82)	(40.83)	(39.92)	(19.30)	(21.69)	**(7.84)**	(22.97)	(15.47)
	5	431.0	(17.52)	(17.52)	(12.82)	(12.50)	(35.86)	(59.17)	(59.43)	(19.20)	(19.03)	**(7.64)**	(31.82)*	(19.49)
	6	410.0	(23.83)	(23.83)	(12.29)	(11.81)	(37.25)	(56.58)	(56.63)	(15.63)	(16.24)	**(6.82)**	(30.65)*	(21.30)
pr144	3	44716	(0.47)	(1.27)	760.4	795.1	1249.7	(3.08)	(3.39)	132.9	145.5	167.1	110.5	**50.5**
	4	40665.0	(4.57)	(4.78)	6111.2	6055.9	(7.77)*	(31.51)	(31.52)	2509.5	3690.0	**536.2**	(5.39)*	3710.5
	5	37666.0	(0.49)	(0.48)	4318.1	4572.3	(5.26)	(22.95)	(22.96)	750.7	772.3	**469.4**	(4.63)*	(0.04)
	6	36903.0	2312.6	2100.9	2303.0	2108.3	(5.68)	(17.65)	(17.65)	488.9	582.2	**419.1**	(4.70)*	4980.5
kroa150	3	13064	(8.07)	(7.85)	**6410.2**	7262.8	(17.06)*	(29.58)	(29.37)	(8.55)	(9.22)	(9.19)	(15.89)*	(13.59)*
	4	11588.0	(6.13)	(7.18)	**8248.1**	(0.49)	(20.97)	(35.12)	(34.89)	(6.17)	(6.61)	(2.43)	(15.74)*	(11.20)*
	5	10805.0	(2.30)	(2.52)	4528.1	**3924.1**	(20.15)	(35.52)	(35.51)	9423.6	(1.64)	5153.1	(12.27)*	(11.70)
	6	10565.0	(0.42)	(0.87)	8011.7	(0.94)	(16.33)	(31.67)	(31.67)	**3479.2**	5011.3	5785.8	(14.31)*	(13.65)
krob150	3	13015	(3.42)	(4.30)	**1507.4**	1557.2	(16.35)*	(34.21)	(32.54)	(6.06)	(7.36)	(1.46)	(18.42)*	(13.41)*
	4	11651.0	(7.29)	(7.29)	**4186.6**	4495.1	(24.22)	(36.70)	(36.70)	(7.65)	(10.21)	(5.71)	(18.90)*	(14.18)*
	5	10885.0	(5.97)	(5.59)	6913.5	**6418.3**	(28.24)	(43.67)	(43.67)	(4.90)	(6.10)	7193.9	(17.81)*	(15.76)
	6	10513.0	(7.45)	(4.19)	8851.6	8414.8	(28.65)	(36.03)	(36.03)	(4.30)	(4.62)	**3677.5**	(17.01)*	(17.99)
pr152	3	53602	(10.24)	(11.08)	4322.9	**3541.3**	(11.51)*	(19.80)	(19.79)	(6.68)	(7.59)	(0.58)	(11.28)*	(3.13)*
	4	44870.0	(7.95)	(9.00)	4807.1	**4518.1**	(14.09)*	(41.41)	(40.84)	(5.94)	(6.21)	7876.3	(12.37)*	(2.09)*
	5	42279.0	(5.51)	(5.57)	**6953.8**	7951.3	(13.60)	(31.76)	(31.76)	(4.03)	(4.68)	9584.2	(11.75)*	(6.23)*
	6	40402.0	(2.68)	(2.49)	4046.1	3825.1	(14.85)	(28.14)	(28.14)	(1.62)	(2.08)	**3570.2**	(10.37)*	(5.44)
u159	3	24306	(5.11)	(5.12)	3779.7	**3381.7**	(14.63)	(34.64)	(34.79)	(7.86)	(8.66)	(1.47)	(14.78)*	(13.94)*
	4	22055.0	(12.14)	(12.14)	**(4.89)**	(5.22)	(34.33)	(42.42)	(42.41)	(12.87)	(14.89)	(5.11)	(26.72)*	(19.27)*
	5	20062.0	(13.98)	(13.98)	(9.67)	(9.19)	(30.18)	(39.80)	(39.80)	(9.45)	(9.93)	**(0.75)**	(28.68)*	(19.52)
	6	19157.0	(7.27)	(7.32)	(2.35)	(3.55)	(39.59)	(39.99)	(39.99)	(4.41)	(5.39)	**7494.5**	(22.98)*	(18.71)

Table 12: Detailed results for larger instances and grid clustering with $\mu = 7$. Besides best solution values (Best) CPU-times in seconds are reported whenever an instance is solved to proven optimality by the corresponding method while values in parenthesis are optimality gaps in case of termination due to time or memory (indicated by *) limit.

Inst.	H	Best	$L^{+}_{x,\bar{n}}$	$L^{c,+}_{x}$	$L^{c,++}_{x,\bar{ln}}$	$L^{c,++}_{x}$	L_{y}	$L^{c,+}_{y,\bar{ln}}$	$L^{c,+}_{y,}$	$CL^{+}_{x,\bar{n}}$	CL^{+}_{x}	$CL^{c,+}_{x}$	CL_{y}	CL^{c}_{y}
kroa100	3	6716	**57.1**	66.7	225.4	222.9	613.1	(38.97)	(38.97)	68.6	79.0	83.9	170.4	255.1
	4	6276.0	397.1	525.0	149.9	174.5	4445.2	(29.75)	(29.75)	72.4	94.1	**62.7**	596.4	1290.7
	5	6154.0	368.1	390.5	477.3	432.1	3981.5	(31.67)	(31.67)	**64.1**	75.7	67.5	667.9	1415.4
	6	6092.0	267.8	293.6	918.0	1037.7	5200.1	(31.55)	(31.55)	**33.4**	37.1	59.9	566.4	1074.2
krob100	3	6859	16.3	16.5	**15.6**	16.0	111.8	(22.20)	(22.15)	60.0	72.0	65.8	54.5	46.6
	4	6506.0	105.2	117.3	116.2	**104.8**	2381.9	(27.91)	(27.91)	108.7	136.8	137.0	1020.1	224.2
	5	6277.0	95.4	100.2	97.9	104.8	3064.6	(25.93)	(25.97)	**42.0**	46.0	**42.0**	933.9	980.0
	6	6217.0	189.3	183.2	308.1	294.8	8723.4	(25.42)	(25.42)	**47.9**	58.9	58.1	491.1	4405.9
kroc100	3	6620	65.0	86.0	**41.0**	41.5	317.4	(28.90)	(28.90)	95.9	124.1	96.5	96.5	120.6
	4	6056.0	124.3	136.9	86.6	87.7	1986.5	(22.57)	(22.57)	55.0	71.0	**41.8**	212.1	120.5
	5	5819.0	135.2	140.4	151.2	148.4	2453.1	(26.71)	(26.71)	**32.5**	36.5	37.8	221.5	124.5
	6	5722.0	131.0	133.2	128.7	132.3	(0.71)	(23.14)	(23.14)	**29.2**	33.4	30.2	120.6	552.1
krod100	3	6911	44.1	51.3	44.8	**42.2**	332.3	(17.86)	(17.51)	136.8	160.1	87.6	150.6	99.4
	4	6388.0	239.1	247.8	129.7	149.7	(10.75)*	(25.11)	(25.18)	**107.4**	130.4	138.6	737.8	177.6
	5	6035.0	130.8	123.4	139.0	125.9	6909.9	(28.61)	(28.61)	47.4	**47.3**	57.0	885.5	461.2
	6	5934.0	187.4	178.5	165.1	151.2	6575.6	(18.59)	(18.59)	27.2	**26.8**	27.9	691.3	1549.8
kroe100	3	7974	**36.6**	45.8	72.9	71.8	205.2	(36.06)	(36.06)	60.9	73.3	58.4	94.5	60.0
	4	7251.0	**61.8**	70.6	68.7	69.0	2417.5	(33.33)	(33.33)	94.5	114.2	65.7	160.8	498.8
	5	6919.0	163.2	187.0	110.7	107.5	5620.3	(33.32)	(33.32)	**45.2**	58.0	45.6	440.6	525.9
	6	6770.0	184.6	210.5	225.5	240.4	6103.5	(26.80)	(26.80)	**33.4**	40.0	35.8	877.8	2470.6
rd100	3	2805	**32.5**	40.1	49.6	50.8	482.5	(26.11)	(22.51)	101.6	135.3	85.5	269.5	221.5
	4	2520.0	**61.8**	80.4	66.2	64.5	1394.6	(32.03)	(31.55)	70.1	89.8	75.0	429.4	255.7
	5	2411.0	78.8	70.3	72.3	73.6	5193.9	(35.52)	(35.52)	48.6	60.5	**44.2**	967.6	337.4
	6	2338.0	85.1	86.8	87.7	92.4	4379.7	(30.60)	(30.60)	16.3	16.2	**14.6**	192.0	563.9
eil101	3	161	431.4	491.1	345.4	331.7	2938.2	(35.47)	(35.48)	**154.6**	179.2	204.1	871.6	964.8
	4	148.0	139.2	142.7	685.5	790.9	3476.3	(34.57)	(34.57)	56.4	66.6	**52.6**	1609.2	1177.8
	5	145.0	851.1	860.4	930.0	938.2	6960.6	(26.35)	(26.35)	**81.4**	100.1	84.9	1631.4	3018.6
	6	144.0	365.6	459.3	1359.7	1702.3	(8.40)	(23.93)	(23.93)	**45.1**	52.2	62.9	941.6	(7.64)
pr107	3	20809	1801.6	2024.0	346.8	346.7	**44.8**	(6.45)	(3.88)	70.2	71.9	90.7	125.3	51.2
	4	19219.0	1852.5	1859.8	549.3	593.7	901.9	(13.25)	(13.44)	224.2	244.1	**186.1**	648.3	363.8
	5	18196.0	642.1	652.4	564.5	611.4	354.7	(11.67)	(11.67)	195.6	**186.7**	211.1	1601.5	1062.5
	6	17801.0	645.6	638.5	586.3	718.4	349.0	(10.96)	(10.96)	**161.5**	161.7	165.3	649.9	986.9
pr124	3	28429	**30.4**	38.1	35.1	33.7	307.3	(13.55)	(13.36)	49.8	59.6	42.6	311.2	396.8
	4	25839.0	113.5	103.5	112.4	123.7	1131.6	(14.76)	(14.81)	82.8	97.1	**48.8**	609.5	891.6
	5	24577.0	132.2	139.4	137.2	123.8	1901.2	(16.90)	(16.90)	64.0	74.9	**49.8**	2771.9	436.3
	6	23881.0	271.3	227.3	225.9	211.2	1441.5	(9.16)	(9.16)	34.9	39.7	**22.9**	294.4	565.5
bier127	3	54843	661.9	1016.0	765.3	923.7	398.6	(11.34)	(6.72)	154.3	160.3	133.3	68.3	**51.2**
	4	52809.0	2248.6	2482.7	2922.7	3645.5	882.0	(14.89)	(12.42)	**103.0**	108.1	228.9	283.5	367.5
	5	52097.0	2834.0	3093.3	3384.5	3043.3	7987.7	(9.77)	(9.77)	80.3	86.2	**79.5**	(3.01)*	682.1
	6	52097.0	2882.4	3322.7	4683.1	5272.3	5462.4	(8.94)	(8.94)	**64.0**	66.8	86.5	853.7	1907.0
gr137	3	363	(8.41)	(8.33)	7264.7	**6955.1**	(21.89)*	(60.57)	(60.63)	(7.95)	(9.59)	(4.59)	(17.09)*	(17.98)*
	4	323.0	(11.16)	(11.16)	(3.07)	**(0.79)**	(34.63)	(54.83)	(54.84)	(9.94)	(11.88)	(5.88)	(34.69)*	(23.55)*
	5	300.0	(14.86)	(14.86)	(16.97)	(17.46)	(32.73)	(48.49)	(48.49)	(9.69)	(8.32)	**(3.53)**	(33.89)*	(22.00)*
	6	287.0	(10.14)	(7.37)	(3.18)	(3.12)	(34.66)	(38.73)	(38.73)	(5.17)	(5.03)	**(2.95)**	(29.76)*	(27.03)
pr144	3	40511	(1.41)	(1.62)	362.3	353.9	517.6	(9.92)	(12.16)	283.8	340.4	**172.1**	1515.4	574.4
	4	37750.0	(4.44)	(4.66)	2808.1	2245.3	(5.78)*	(23.05)	(22.78)	1425.5	1848.1	**570.3**	(4.89)*	(5.54)*
	5	35212.0	1980.8	2139.7	1736.1	1740.7	(8.55)	(16.91)	(16.91)	296.3	356.4	**215.7**	(3.88)*	(1.64)*
	6	34626.0	2748.5	3149.6	1572.6	1750.3	(5.28)	(16.09)	(16.09)	**216.0**	248.2	238.8	(5.36)*	7655.2
kroa150	3	9426	1481.6	1740.3	**905.6**	941.1	(2.68)	(33.12)	(33.12)	3422.1	3507.6	1410.9	(5.67)*	8794.0
	4	8801.0	7161.7	9186.3	1951.5	2197.4	(17.96)	(23.84)	(23.84)	3114.3	3330.0	**831.7**	(14.57)*	(7.91)*
	5	8493.0	6863.9	7025.9	3641.5	3058.3	(16.92)	(29.02)	(29.02)	4070.9	5125.9	**1207.3**	(15.41)*	(12.87)*
	6	8248.0	3662.6	2830.6	4094.7	3773.9	(19.19)	(20.23)	(20.23)	1765.8	2439.6	**655.6**	(11.79)*	(13.56)
krob150	3	9365	(3.60)	(6.59)	**1558.5**	1601.3	(22.13)*	(37.84)	(37.84)	(4.02)	(5.15)	5909.0	(22.31)*	(6.89)*
	4	8314.0	(3.54)	(5.61)	3043.6	3359.4	(22.56)	(50.11)	(50.11)	(1.49)	(1.92)	**1767.7**	(22.14)*	(18.23)*
	5	7965.0	(3.79)	(4.44)	5536.6	5834.3	(28.35)	(39.23)	(39.23)	(2.14)	(2.77)	**3714.1**	(16.48)*	(17.74)
	6	7754.0	(2.94)	(3.65)	6434.6	4754.6	(33.55)	(40.23)	(40.23)	**2931.0**	4495.4	4922.9	(22.31)*	(24.27)
pr152	3	46009	(8.88)	(8.84)	4526.4	4848.7	(3.94)*	(26.02)	(26.12)	5130.1	6441.1	**1449.3**	2859.6	(4.06)*
	4	38609.0	(1.95)	(1.94)	4123.3	3965.8	(9.34)	(17.75)	(17.75)	1041.0	1177.5	**660.3**	(8.72)*	(7.10)*
	5	36934.0	(0.95)	(1.00)	3216.8	4137.5	(9.28)	(18.86)	(18.86)	867.1	950.8	**637.2**	(8.82)*	(6.27)*
	6	36172.0	3798.3	3409.0	3034.3	3632.2	(12.03)	(16.37)	(16.37)	1113.0	1470.1	**836.8**	(7.41)*	(5.69)*
u159	3	16948	1110.1	1021.2	758.1	**707.2**	(13.50)*	(44.60)	(44.60)	6151.6	6358.6	2774.1	(9.93)*	(6.84)*
	4	15589.0	(8.65)	(7.74)	7303.9	6747.8	(26.34)	(43.22)	(43.22)	(4.45)	(4.97)	**3935.2**	(24.83)*	(17.40)*
	5	14540.0	(2.30)	(2.31)	7458.0	8639.9	(29.49)	(40.18)	(40.18)	(2.99)	(3.90)	**1919.9**	(19.15)*	(19.89)*
	6	13793.0	6783.3	5534.0	6955.0	5948.6	(28.20)	(41.09)	(41.09)	2680.1	3384.8	**838.0**	(24.13)*	(17.30)

Table 13: Detailed results for larger instances and grid clustering with $\mu = 10$. Besides best solution values (Best) CPU-times in seconds are reported whenever an instance is solved to proven optimality by the corresponding method while values in parenthesis are optimality gaps in case of termination due to time or memory (indicated by *) limit.

Inst.	H	Best	$L^{+}_{x,\bar{n}}$	$L^{c,+}_{x}$	$L^{c,++}_{x,ln}$	$L^{c,++}_{x}$	L_{y}	$L^{c,+}_{y,ln}$	$L^{c,+}_{y,}$	$CL^{+}_{x,\bar{n}}$	CL^{+}_{x}	$CL^{c,+}_{x}$	CL_{y}	CL^{c}_{y}
kroa100	3	6716	**54.9**	63.9	213.6	228.3	618.6	(38.97)	(38.97)	66.9	78.0	83.2	166.5	251.4
	4	6276.0	398.1	535.9	145.5	184.4	4661.6	(29.75)	(29.75)	72.7	94.3	**62.1**	598.1	1282.7
	5	6154.0	361.9	411.4	416.8	455.8	4348.3	(31.67)	(31.67)	**63.4**	75.8	66.0	640.8	1413.7
	6	6092.0	274.8	288.1	1065.7	1004.6	5861.6	(31.55)	(31.55)	**33.0**	36.5	60.6	572.3	1043.0
krob100	3	6859	**15.0**	17.0	**15.0**	15.2	114.5	(22.14)	(22.14)	62.5	71.6	61.9	55.4	46.1
	4	6506.0	101.9	120.5	102.7	**97.1**	2547.7	(27.91)	(27.91)	109.4	136.9	135.6	971.9	230.0
	5	6277.0	93.9	100.0	98.8	96.4	3191.3	(25.93)	(25.93)	41.5	47.3	**41.4**	879.4	967.1
	6	6217.0	195.1	191.5	300.5	282.8	9091.5	(25.42)	(25.42)	**46.1**	59.0	56.5	478.3	4449.2
kroc100	3	6620	65.7	86.5	**40.9**	41.3	312.3	(28.90)	(28.90)	95.5	125.6	92.5	96.2	116.8
	4	6056.0	125.0	134.0	87.6	86.5	1997.2	(22.57)	(22.57)	56.0	70.4	**42.4**	211.7	119.2
	5	5819.0	130.1	133.4	150.2	145.6	2506.8	(26.71)	(26.71)	**32.6**	37.6	37.9	216.3	124.5
	6	5722.0	127.5	135.9	126.8	126.3	(1.44)	(23.14)	(23.14)	29.3	31.9	**28.7**	118.6	559.9
krod100	3	6911	45.0	50.0	42.7	**42.2**	339.7	(17.86)	(17.51)	141.0	160.0	86.7	148.0	99.5
	4	6388.0	238.4	251.6	131.6	131.9	(10.75)*	(25.42)	(25.15)	**107.2**	129.5	133.0	740.3	172.5
	5	6035.0	125.2	132.6	138.0	144.5	7331.4	(28.61)	(28.61)	**45.7**	48.4	57.4	906.0	474.1
	6	5934.0	184.6	195.8	159.9	169.1	7085.8	(18.59)	(18.59)	**26.2**	26.3	26.8	686.3	1540.3
kroe100	3	7974	**36.1**	45.1	72.1	73.6	215.1	(36.06)	(36.06)	61.1	73.3	59.3	90.6	59.2
	4	7251.0	**63.9**	73.5	74.8	70.8	2645.0	(33.33)	(33.33)	94.6	111.7	64.8	162.7	501.1
	5	6919.0	161.6	198.5	107.2	116.5	5841.7	(33.32)	(33.32)	**45.0**	58.5	46.0	462.0	548.3
	6	6770.0	193.1	223.9	240.6	237.9	6044.8	(26.80)	(26.80)	**35.5**	40.1	36.0	844.6	2525.3
rd100	3	2805	**31.2**	39.3	48.9	49.0	477.3	(26.11)	(22.51)	100.6	136.4	84.4	271.1	223.0
	4	2520.0	**66.0**	71.0	71.0	67.7	1359.7	(31.53)	(32.02)	69.8	90.4	74.6	418.3	253.2
	5	2411.0	68.5	73.1	82.8	80.3	5243.3	(35.52)	(35.52)	49.3	61.2	**44.2**	964.2	312.3
	6	2338.0	89.8	72.0	82.5	82.7	3366.0	(30.60)	(30.60)	15.8	15.9	**14.8**	193.1	557.3
eil101	3	161	399.8	474.8	338.6	347.7	2978.8	(35.47)	(35.48)	**155.8**	175.3	207.1	786.0	985.4
	4	148.0	132.6	145.5	653.8	759.6	3680.6	(34.57)	(34.57)	56.7	65.4	**51.6**	1639.0	1175.0
	5	145.0	824.3	880.3	1036.9	771.6	7402.5	(26.35)	(26.35)	82.7	99.8	**79.0**	1685.7	2832.0
	6	144.0	421.2	411.9	1712.8	1750.0	(8.61)	(23.93)	(23.93)	**46.2**	50.4	64.2	955.9	(7.82)
pr107	3	18939	202.9	227.6	191.3	193.9	58.2	(9.02)	(5.55)	19.2	19.4	**18.6**	40.8	71.7
	4	17542.0	171.8	170.4	172.1	170.3	**62.9**	(10.65)	(10.65)	75.5	75.3	74.8	117.8	140.8
	5	16802.0	300.6	314.7	301.4	311.3	**43.2**	(7.48)	(7.48)	223.8	236.6	232.5	136.8	89.0
	6	16756.0	579.9	587.3	541.3	542.4	**127.1**	(7.57)	(7.57)	158.2	151.4	152.2	167.4	140.8
pr124	3	19953	39.5	59.5	59.8	63.1	94.9	(18.04)	(10.31)	**21.9**	27.1	26.4	45.2	48.8
	4	18682.0	43.5	50.5	100.8	115.7	185.6	(22.38)	(22.37)	**11.1**	12.2	12.2	82.5	88.0
	5	18600.0	163.6	208.0	225.1	269.7	287.1	(18.42)	(18.42)	**15.5**	16.5	19.3	200.7	258.5
	6	18554.0	211.1	201.4	302.8	325.9	606.1	(18.85)	(18.85)	**16.0**	18.1	17.7	315.6	953.3
bier127	3	44909	761.8	1573.9	1007.3	1243.3	78.9	(10.46)	3439.9	49.9	60.8	59.6	57.5	**41.5**
	4	43825.0	574.7	934.6	1237.4	1035.8	360.3	(13.30)	(12.61)	**21.0**	23.7	25.6	48.5	79.8
	5	43778.0	833.3	1309.9	1414.1	1864.0	1723.0	(12.63)	(12.80)	**23.0**	29.8	26.3	85.7	94.1
	6	43778.0	582.1	867.0	1859.1	2307.0	543.1	(12.55)	(12.33)	**30.7**	41.0	37.2	186.8	181.8
gr137	3	261	(7.53)	(7.37)	3107.4	**3072.5**	(13.63)*	(46.20)	(46.05)	3422.0	3723.6	4175.1	(19.28)*	(15.85)*
	4	232.0	(6.70)	(7.40)	3876.1	4017.3	(18.24)	(48.42)	(48.42)	3413.3	3293.5	**1662.7**	(22.69)*	(23.88)*
	5	215.0	(3.76)	(3.40)	2532.0	2764.7	(23.09)	(49.41)	(49.41)	1619.6	2153.7	**652.2**	(30.29)*	(25.45)*
	6	208.0	(4.85)	(5.60)	3602.9	3996.6	(26.34)	(48.48)	(48.48)	1675.5	2108.0	**592.2**	(26.58)*	(27.74)*
pr144	3	37199	1622.4	1963.8	204.5	205.1	112.4	(5.44)	(5.40)	57.1	60.1	**44.0**	88.0	77.9
	4	34497.0	548.6	572.1	433.1	397.7	865.3	(9.12)	(9.12)	**122.7**	125.2	134.4	237.8	260.2
	5	33741.0	2279.7	2210.5	845.9	775.8	2538.4	(12.21)	(12.21)	122.0	130.9	**120.1**	2322.6	910.9
	6	33021.0	323.5	280.8	379.9	439.7	809.0	(6.67)	(6.67)	**45.4**	45.5	54.0	697.4	1200.0
kroa150	3	6016	546.9	636.8	1614.2	1344.6	4360.2	(41.78)	(41.78)	284.2	336.3	**237.3**	1349.3	1457.3
	4	5574.0	949.8	980.3	7961.1	9321.4	(12.59)	(38.96)	(38.96)	534.0	719.4	**369.1**	(11.24)*	(15.32)*
	5	5413.0	2374.3	2180.8	2535.7	3288.5	(22.70)	(37.29)	(37.29)	310.6	341.6	**273.8**	(10.22)*	5289.7
	6	5337.0	9787.2	8888.5	3961.3	4016.9	(23.11)	(42.28)	(42.28)	130.3	156.5	**113.0**	2969.2	(12.82)*
krob150	3	6390	705.8	959.2	558.7	457.2	(11.02)*	(37.88)	(37.88)	470.8	517.9	**439.8**	1392.4	2119.8
	4	5889.0	1661.6	2053.2	1072.7	866.0	(16.47)	(45.88)	(45.88)	410.5	450.8	**302.1**	(8.25)	(9.97)*
	5	5743.0	1447.5	1420.9	981.7	1134.3	(17.16)	(37.67)	(37.67)	425.9	470.4	**314.3**	(11.11)*	(17.35)*
	6	5668.0	1184.1	1276.1	4099.6	2956.4	(18.95)	(36.08)	(36.08)	**218.3**	235.4	245.0	(12.48)*	(18.92)*
pr152	3	38206	4115.0	4198.1	833.3	780.2	3042.8	(16.93)	(16.93)	868.2	922.3	**477.2**	(5.16)*	3946.9
	4	34874.0	2028.8	2890.1	930.1	1019.2	4166.4	(13.70)	(13.70)	**170.0**	175.4	184.8	(3.54)*	(4.94)*
	5	33785.0	3249.8	3156.5	2446.0	2547.7	(0.60)	(18.93)	(18.93)	442.5	471.8	**408.8**	(7.77)*	(4.33)*
	6	33479.0	4045.8	4548.2	5727.1	5876.4	(12.52)	(16.31)	(16.31)	**186.8**	193.4	214.7	(6.98)*	(6.95)*
u159	3	16948	994.1	1437.4	**620.3**	715.6	(13.50)*	(44.60)	(44.60)	5117.0	5705.7	2469.5	(9.93)*	(6.84)*
	4	15589.0	(6.72)	(6.86)	6249.8	6498.3	(25.11)	(43.22)	(43.22)	(2.84)	(3.53)	**3828.4**	(24.83)*	(17.40)*
	5	14540.0	(1.58)	(1.86)	7105.5	5996.4	(28.47)	(40.18)	(40.18)	(1.48)	(3.35)	**1998.1**	(19.15)*	(19.89)*
	6	13793.0	5041.7	5425.1	5721.7	5855.9	(27.12)	(41.09)	(41.09)	3431.2	3081.4	**643.9**	(24.13)*	(16.06)