Mixed-Integer Programming Approaches for the Time-Constrained Maximal Covering Routing Problem

Markus Sinnl*[1]

[1] *Institute of Production and Logistics Management, Johannes Kepler University Linz, Linz, Austria*

Abstract

In this paper, we study the recently introduced time-constrained maximal covering routing problem. In this problem, we are given a central depot, a set of facilities, and a set of customers. Each customer is associated with a subset of the facilities which can cover it. A feasible solution consists of p Hamiltonian cycles on subsets of the facilities and the central depot. Each cycle must contain the depot and must respect a given distance limit. The goal is to maximize the number of customers covered by facilities contained in the cycles.

We develop two exact solution algorithms for the problem based on new mixed-integer programming models. One algorithm is based on a compact model, while the other model contains an exponential number of constraints, which are separated on-the-fly, i.e., we use branch-and-cut. We also describe preprocessing techniques and valid inequalities for both models.

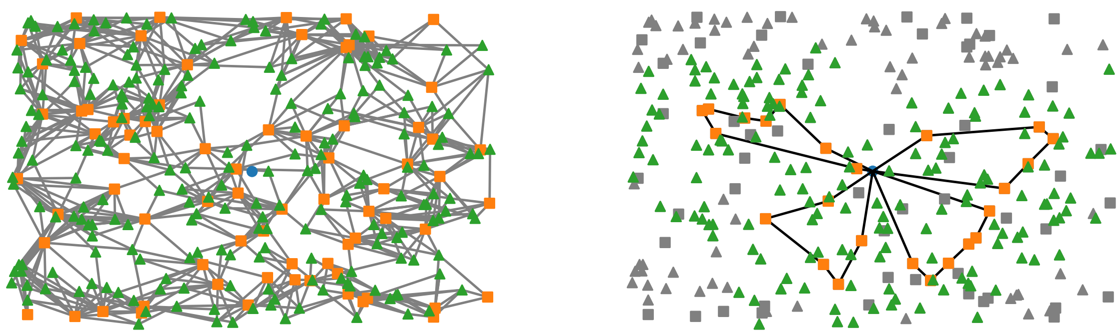
We evaluate our solution approaches on the instances from literature and our algorithms are able to find the provably optimal solution for 264 out of 270 instances, including 120 instances, for which the optimal solution was not known before. Moreover, for most of the instances, our algorithms only take a few seconds, and thus are up to five magnitudes faster than previous approaches. Finally, we also discuss some issues with the instances from literature (e.g., in some instances, up to 90% of customers have no facility associated with them and are thus useless; the optimal solution for many instances contains just all remaining customers) and present some new instances.

1 Introduction

Vehicle routing problems and covering problems are important and fundamental problems in Operations Research and Logistics. In this paper, we study the recently introduced *time-constrained maximal covering routing problem (TCMCRP)*, which is a generalization of well-known routing problems such as the *orienteering problem* (see e.g., [12, 13]), and in particular the *team orienteering problem* (see e.g., [5]), and the *maximal covering location problem* (see e.g., [7]). The problem was introduced in [1] and applications in health care were discussed. In the TCMCRP, we are given a directed graph $G = (V, A)$, where $V = 0 \cup F \cup C$ is the set of vertices. The vertex 0 represents the *central depot*, F the set of *facilities* and C the set of *customers*. The arc set $A = A_{0F} \cup A_{FC}$ is defined as set of *routing arcs* $A_{0F} = \{(i, i') : i, i', \in 0 \cup F\}$ (i.e., the complete directed graph on $0 \cup F$) and *assignment arcs* $A_{FC} \subseteq \{(i, j) : i \in F, j \in C\}$ (i.e., the assignment arcs are a subset of all possible facility/customer connections). Each arc $(i, i') \in A_{0F}$ has a travel distance $d_{ii'} > 0$ associated with it. Moreover, let P represent the set of $p = |P|$ available vehicles, and let L_v be a *distance limit* for each $v \in P$. A feasible solution consists of one Hamiltonian cycle on a subset of $0 \cup F$ for each $v \in P$. Each of these cycles must contain 0 and respect the distance limit L_v, and a facility F can only appear in at most one cycle. A customer is *covered* by a solution, if there exists an assignment arc in A_{FC}

*markus.sinnl@jku.at

between the customer and a facility visited in one of the Hamiltonian cycles. The goal is to find a feasible solution, which maximizes the number of covered customers. Note that in the instances from literature, L_v is the same for all vehicles (and despite allowing for different L_v in the definition in [1], the vehicles are also called *homogeneous* in the definition), and the distance function is Euclidean (both are common assumptions in vehicle routing problems). In this work, we use the first assumption in one of the two presented mixed-integer programming (MIP) models, and the second assumption when deriving preprocessing procedures and valid inequalities for both models. Figure 1 shows an exemplary instance-graph of the TCMCRP and its optimal solution for four vehicles and a given distance limit.



(a) Instance-graph.

(b) Optimal solution.

Figure 1: Exemplary instance-graph of the TCMCRP and its optimal solution for four vehicles and a given distance limit. The blue circle is the central depot, orange boxes are facilities and green triangles are customers. The gray edges in 1a between facilities and customers denote which customers are covered by each facility. For better readability, the arcs between facilities, and facilities and the central depot are not displayed. In the solution 1b, the arcs of the optimal solution are indicated in black, and all facilities and customers not in the solution are grayed out.

Contribution and Paper Outline The TCMCRP was recently introduced in [1], where the authors presented a flow-based MIP model, an iterated local search, a tabu search and a variable neighborhood search for it. They evaluated their algorithms on instances derived from the well-known TSPLIB [23]. In this paper, we develop two exact solution algorithms for the problem based on new MIP-models. One algorithm is based on a compact model, while the other model contains an exponential number of constraints. We also describe preprocessing techniques and valid inequalities for both models. Since for the compact model, the valid inequalities are of exponential size, we use branch-and-cut in both solution algorithms. In a computational study, we evaluate our solution approaches on the instances from [1]. The study reveals that our algorithms are able to find the provably optimal solution for 120 instances, where the optimal solution was not known before, and only 6 instances remain unsolved. Moreover, for most of the instances, our algorithms only take a few seconds, and thus are up to five magnitudes faster than the algorithms presented in [1]. Finally, we also discuss some issues with the instances used in [1] (e.g., not all customers have a facility associated with it, and the optimal solution often contains just all reachable facilities; the calculation of L_v is not the same as described in the paper), and introduce a set of new and more difficult instances.

The paper is organized as follows: In the remainder of this section, we give an overview of related work. In Section 2, we present our two new MIP-models, together with preprocessing/variable-fixing procedures and valid inequalities. In Section 3, we discuss additional details about the developed branch-and-cut framework, such as separation procedures for the valid inequalities. Section 4 contains the computational study, and Section 5 concludes the paper.

Related Work As the studied problem is a quite general routing/covering problem, there is naturally a vast number of related work; the paper introducing the TCMCRP ([1]) contains a quite exhaustive and up-to-date discussion of related problems. We thus focus the discussion about related work on the *team orienteering problem (TOP)*, since both of our models are extensions of models for the TOP. For a general overview on routing problems, we refer to e.g., [24] and for a general overview on facility location/covering problems, we refer to e.g., [17] (Chapter 5).

The TOP is an extension of the *orienteering problem (OP)* to multiple vehicles. The OP was first introduced in [25]. In the OP, we are given a central depot, a set of customers which can be visited (and provide profit if visited), and a distance limit. The goal is to find the most profitable Hamiltonian cycle on a subset of the customers, the cycle must also contain the depot and respect the distance limit. Sometimes the OP is defined with a start depot and an end depot, and a Hamiltonian path on a subset of the customers from start to end is searched. Moreover, there is a variant of the OP called *selective traveling salesman problem* (see [11, 16]), with additional compulsory vertices, which must be in any feasible cycle/path. There exist also many other variants with additional side-constraints (such as capacities or time-windows), for more details, see, e.g., the survey [13]). Regarding successful exact approaches for the OP, there are several papers ([10, 11, 18]) using branch-and-cut approaches based on models with *generalized subtour elimination constraints (GSECs)/connectivity cuts(CCs)*. Similar GSECs/CCs will also be used in our approaches.

The TOP was introduced in [5], where a heuristic was proposed. The TOP extends the OP by introducing p (homogeneous) vehicles, i.e., the goal is now to find p Hamiltonian cycles/paths containing the depot and respecting the distance limit, instead of a single one. Note that the TOP can be seen as a special case of the TCMCRP, with a one-to-one-correspondence between facilities and customers. A variant of the TOP with heterogeneous vehicles (denoted as *multiple tour maximum collection problem*) was considered in [4]. Several column generation, and branch-and-price(-and-cut) approaches were proposed for the TOP, see [3, 4, 14, 22]. An exponential size formulation using GSECs and solved by branch-and-cut was developed in [8, 9]. In [2], the authors presented a compact model for the TOP based on a formulation of [19] for the *sequential ordering problem*. They strengthen the model by separating CCs, and were able to solve additional instances to optimality.

Aside from the TOP, another strongly related problem to the TCMCRP is the *time-constrained maximal covering salesman problem (TCMCSP)*, which was introduced in [20]. The TCMCSP is the single-vehicle variant of the TCMCRP. In [20], the authors presented a flow-based MIP model and some heuristics for the TCMCSP ([1] is basically the extension of the approaches in [20] to the TCMCRP). In [21], an exact solution algorithm based on GSECs for a variant of the TCMCSP was proposed.

2 Mixed Integer Programming Models and Valid Inequalities

We first present the compact model, together with its associated preprocessing and valid inequalities, and then the exponential-sized model, together with its associated preprocessing and valid inequalities. Note that for the compact model, the presented valid inequalities are of exponential size, so we also use branch-and-cut in the algorithm based on the compact model. For later use, for a subset $S \subseteq 0 \cup F$, let $\delta^+(S) = \{(i,i') \in A_{0F} : i \in S, i' \notin S\}$ and $\delta^-(S) = \{(i,i') \in A_{0F} : i \notin S, i' \in S\}$ be the set of outgoing, resp., incoming arcs of the cut induced by S.

2.1 Compact Model

This formulation follows the approach proposed for the TOP in [2]. As for this formulation, we assume that the vehicles P are homogeneous, let $L = L_v, v \in P$. The presented formulation allows for solutions with less than p cycles, as initial tests showed that some instances from literature were infeasible, when a solution with exactly p cycles was sought (i.e., there were less than p facilities within the distance limit L from the central depot). Let binary variables $x_{ii'} = 1$, for $(i,i') \in A_{0F}$, iff arc (i,i') is traveled by a vehicle in the solution. Let binary variables $y_i = 1$, $i \in F$, iff facility i is visited in the solution, and binary variables $z_j = 1$, $j \in C$, iff customer j is covered by the solution. Moreover, let continuous variables $f_{ii'}$ for $(i,i') \in A_{0F}$ indicate the traveled distance from the central depot at facility i' for a vehicle arriving from i. Let integer variable $w \in \{1,\dots,p\}$ indicate the number of vehicles used in the solution. Using these variables, the TCMCRP can be modeled as follows, we denote the model by (C).

$$\max \sum_{j\in C} z_j \tag{C-OBJ}$$

$$s.t. \sum_{(i,j)\in A_{FC}} y_i \geq z_j \qquad \forall j \in C \tag{C-LINK}$$

$$\sum_{(i,i')\in A_{FC}} x_{ii'} = y_i \qquad \forall i \in F \tag{C-OUT}$$

$$\sum_{(i',i)\in A_{FC}} x_{i'i} = y_i \qquad \forall i \in F \tag{C-IN}$$

$$\sum_{(0,i)\in A_{FC}} x_{0i} = w \tag{C-OUT0}$$

$$\sum_{(i,0)\in A_{FC}} x_{i0} = w \tag{C-IN0}$$

$$f_{0i} = d_{0i}x_{0i} \qquad \forall i \in F \tag{C-FLOW0}$$

$$\sum_{i'\in\delta^+(i)} f_{ii'} - \sum_{i'\in\delta^-(i)} f_{i'i} = d_{ii'}x_{ii'} \qquad \forall i \in F \tag{C-FLOW}$$

$$f_{ii'} \leq (L - d_{i'0})x_{ii'} \qquad \forall (i,i') \in A_{0F} \tag{C-DIST}$$

$$y_i \in \{0,1\} \qquad \forall i \in F \tag{C-Y}$$

$$z_j \in \{0,1\} \qquad \forall j \in C \tag{C-Z}$$

$$x_{ii'} \in \{0,1\} \qquad \forall (i,i') \in A_{0F} \tag{C-X}$$

$$f_{ii'} \geq 0 \qquad \forall (i,i') \in A_{0F} \tag{C-F}$$

$$w \in \{1,\dots,p\} \tag{C-W}$$

The objective function (C-OBJ) and constraints (C-LINK) ensure that a customer is only counted in the objective function, if a facility covering it is visited in the solution. Constraints (C-IN) and (C-OUT) ensure that there is one incoming, resp., outgoing arc in the tour, if facility i gets visited. Constraints (C-OUT0) and (C-IN0) (together with (C-W))make sure that there are at most p vehicles leaving and entering the depot. The solution defined by the previous for set of constraints (plus integrality of the variables) will consist of p or more cycles, at most p of them containing the depot, i.e., subtours are possible. Such subtours are prohibited using flow-conservation constraints (C-FLOW0) and (C-FLOW), which ensure that the flow-variables $f_{i'i}$ encode the distance traveled so far the tour arrives at facility $i' \in F$. The distance limit on the tour is modeled by constraints (C-DIST): For any facility in the solution, the traveled distance must allow to go back to the central depot within the distance limit. Finally, constraints (C-Y) to (C-W) define the variables.

Valid Inequalities Next, we present some valid inequalities for (C), including variable-fixing procedures, which can be applied in a preprocessing step. As mentioned in the introduction, we assume that the distance function is Euclidean, resp., is symmetric and satisfies the triangle inequality (thus the direct connection between two vertices is always the shortest). Some of the valid inequalities will be based on optimality-arguments, i.e., they disallow solutions, which would be valid for the problems, but are provably non-optimal.

The first set of inequalities is concerned with dominance between facilities.

Theorem 1. *Let $i, i' \in F$, and $C(i'') = \{j \in C : (i'', j) \in A_{FC}\}$ for $i'' = i, i'$. Suppose $C(i') \subseteq C(i)$. Then the following* facility dominance inequalities

$$y_i + y_{i'} \leq 1 \tag{C-FD}$$

are valid for (C).

Proof. Due to the triangle inequality, a facility will only be included in an optimal solution, if it allows the coverage of additional customers. As i covers at least the same customers also covered by i', there is no benefit in including i' in any solution containing i. □

The following variable-fixing exploit the distance limit L, similar ideas have been used in [2, 8, 9] for the TOP and in [21] for the TCMCSP, they can be seen as special-case of the *path inequalities* for the OP proposed in [10].

Theorem 2. *Let $i \in F$ with $2d_{0i} > L$. Then*

$$y_i = 0 \tag{C-FIXF}$$

is valid for (C).

Let $i, i' \in F$ with $d_{0i} + d_{ii'} + d_{i'0} > L$. Then

$$x_{ii'} = 0 \tag{C-FIXA}$$

is valid for (C).

Let $F(j) = \{i \in F : (i, j) \in A_{FC}\}$ for $j \in C$. If for all $i \in F(j)$, we have $2d_{0i} > L$, then

$$z_j = 0 \tag{C-FIXC}$$

is valid for (C).

Proof. Obvious, as the distance limit does not allow the shortest cycle containing (i, i'), resp., i. Moreover, if no facility covering j can be reached given the distance limit, j cannot be in any solution. □

The following global constraint (C-DISTG) on the length of all p tours can also be added, as well as constraints (C-FLOWER) which impose lower bounds on the flow-variables $f_{ii'}$ (see [2])

$$\sum_{(i,i') \in A_{0F}} d_{ii'} x_{ii'} \leq wL, \tag{C-DISTG}$$

$$f_{ii'} \geq (d_{0i} + d_{ii'}) x_{ii'}. \tag{C-FLOWER}$$

While the inequalities in the model already ensure that the solution is connected (and hence, consists of p cycles containing the central 0), the model can be strengthened by adding connectivity cuts (C-CC) (see, e.g., [2]). As there exponential many of them, we separate them on-the-fly in a branch-and-cut, see Section 3.2 for the separation.

$$\sum_{(i'',i') \in \delta^-(S)} x_{i''i'} \geq y_i \quad \forall S \subset F, i \in S : |S| \geq 2 \tag{C-CC}$$

Let LB a given lower bound for the objective value, e.g., the value of the current incumbent solution during branch-and-cut. Using LB, an optimality-based lifting of (C-CC) may be possible

Theorem 3. *Let $S \subset F$ with $|S| \geq 2$, and let $i \in S$. Let $Z(S) = |\{j : (i, j) \in A_{FC}, i \notin S\}|$, i.e., the number of customers, which can be served by facilities not in S. Suppose $Z(S) \leq LB$, then the following inequality is valid*

$$\sum_{(i'',i') \in \delta^-(S)} x_{i''i'} \geq 1 \tag{C-CCL}$$

Proof. As $Z(S) \leq LB$, facilities outside of S cannot serve enough customers to provide an improved solution, thus, at least one tour must visit facilities in S to provide a solution with value better than LB. □

Another lifted version of (C-CC) can be obtained using the facility dominance inequalities (C-FD).

Theorem 4. *Let $i, i' \in S \subseteq F$, and $C(i'') = \{j \in C : (i'', j) \in A_{FC}\}$ for $i'' = i, i'$. Suppose $C(i') \subseteq C(i)$. Then the following inequality is valid*

$$\sum_{(i'',i') \in \delta^-(S)} x_{i''i'} \geq y_i + y_{i'} \tag{C-CCL2}$$

Proof. Both i, i' are in S and in any optimal solution, at most one of them will be taken. □

2.2 Exponential-Sized Model

This model follows the formulation of [8, 9] for the TOP and allows for heterogeneous vehicles. Binary variables z_j, $j \in C$ have the same meaning as in model (C). Let binary variables $y_i^v = 1$, $i \in F$, $v \in P$, if facility i gets visited by the tour of vehicle $v \in P$. Moreover, let binary variables $x_{ii'}^v = 1$, for $(i, i') \in A_{0F}$, iff arc (i, i') is traveled by vehicle $v \in P$ in the solution. Let binary variable $w_v = 1, v \in P$ iff vehicle v is used in the solution. Using these variables, the TCMCRP can be modeled as follows, we denote the model by (E).

$$\max \quad \sum_{j \in C} z_j \tag{E-OBJ}$$

$$s.t. \quad \sum_{v \in P} \sum_{(i,j) \in A_{FC}} y_i^v \geq z_j \qquad \forall j \in C \tag{E-LINK}$$

$$\sum_{v \in P} y_i^v \leq 1 \qquad \forall i \in F \tag{E-ONEF}$$

$$\sum_{(i,i') \in A_{0F}} x_{ii'}^v = y_i^v \qquad \forall i \in F, v \in P \tag{E-OUT}$$

$$\sum_{(i',i) \in A_{0F}} x_{i'i}^v = y_i^v \qquad \forall i \in F, v \in P \tag{E-IN}$$

$$\sum_{(0,i') \in A_{0F}} x_{0i'}^v = w_v \qquad \forall v \in P \tag{E-OUT0}$$

$$\sum_{(i',0) \in A_{0F}} x_{i'0}^v = w_v \qquad \forall v \in P \tag{E-IN0}$$

$$\sum_{(i'',i') \in \delta^-(S)} x_{i''i'}^v \geq y_i^v \qquad \forall S \subset F, i \in S : |S| \geq 2, v \in P \tag{E-CC}$$

$$\sum_{(i,i') \in A_{0F}} d_{ii'} x_{ii'}^v \leq L_v w_v \qquad \forall v \in P \tag{E-DIST}$$

$$y_i^v \in \{0, 1\} \qquad \forall i \in F, v \in P \tag{E-Y}$$

$$z_j \in \{0, 1\} \qquad \forall j \in C \tag{E-Z}$$

$$x_{ii'}^v \in \{0, 1\} \qquad \forall (i, i') \in A_{0F}, v \in P \tag{E-X}$$

$$w_v \in \{0, 1\} \qquad v \in P \tag{E-W}$$

The objective function (E-OBJ) and constraints (E-LINK) are the same as in model (C). Constraints (E-ONEF) make sure, that each facility is only visited by one vehicle (in case of distances satisfying the triangle inequality, these constraints are redundant, as using a facility in more than one tour will only result in larger distances). For each vehicle $v \in P$, constraints (E-OUT) and (E-IN) ensure that if a facility is visited by vehicle v, the vehicle enters and leaves the facility. Moreover, constraints (E-OUT0) and (E-IN0) ensure that each vehicle enters and leaves the depot if the vehicle is used. Thus, the previous for set of constraints make sure that the solution for each used vehicle consists of one or more cycles, and one of these cycles starts and ends at the depot. Connectivity cuts (E-CC) ensure, that there is only one cycle for each vehicle. Note that they are of exponential size, separation of them is discussed in Section 3.1. The distance limit is enforced by constraints (E-DIST). The variables are defined by constraints (E-Y) to (E-W).

Valid Inequalities Similar to model (C), several valid inequalities and variable-fixing procedures can be defined. Some of these inequalities are adaptions of the inequalities for (C), however, there are also additional inequalities. We first mention the adapted inequalities, before we give the additional ones.

Facility domination inequalities (C-FD) can be adapted as follows for $i, i' \in F$ fulfilling the conditions to obtain inequalities (E-FD)

$$\sum_{v \in P} (y_i^v + y_{i'}^v) \leq 1. \tag{E-FD}$$

Adaption of the variable-fixings (C-FIXF), (C-FIXA) and (C-FIXC) are straightforward (we denote them as (E-FIXF), (E-FIXA) and (E-FIXF)), and the following conflict-constraints (E-FC) for $i, i' \in F$ with $d_{0i} + d_{ii'} + d_{i'0} > L_v$ can additionally be derived (see also the *incompatibility clique cuts* for the TOP in [8, 9])

$$y_i^v + y_{i'}^v \leq w_v. \tag{E-FC}$$

Inequalities (E-FC) can be used for lifting (E-CC) for $i, i' \in F$ with $d_{0i} + d_{ii'} + d_{i'0} > L_v$ to (E-CCL),

$$\sum_{(i''', i'') \in \delta^-(S)} x_{i'''i''}^v \geq y_i^v + y_{i'}^v \quad \forall S \subset F, i, i' \in S : |S| \geq 2, \tag{E-CCL}$$

which are valid since at most one of y_i^v and $y_{i'}^v$ can be included in a solution. Moreover, a lifted version of (E-CC) using facility dominance, similar to (C-CCL2) can also be defined, we denote it as (E-CCL2).

In case $L = L_v, v \in P$, there can be symmetric solutions. In order to break these symmetries, the following set of inequalities (E-SYM) can be used. Let $v_1, v_2, \ldots, v_{|P|}$ denote an arbitrary ordering of the vehicles. Inequalities (E-SYM) impose that a vehicle with lower index needs to visit at least as many facilities in its tour than a vehicle with a higher index (see also [8, 9] for similar inequalities for the TOP)

$$\sum_{i \in F} y_i^{v_k} \geq \sum_{i \in F} y_i^{v_{k+1}}, \quad 1 \leq k \leq |P| - 1. \tag{E-SYM}$$

.

3 Algorithmic Frameworks

In this section we discuss further details of our solution frameworks based on models (C), resp., (E).

3.1 Separation Algorithms

Model (C) has polynomial size, however, the family of valid inequalities (C-CC) is of exponential size. Thus, we do not add all of them in the beginning, but separate them on-the-fly, when they are violated, i.e., we use branch-and-cut. The same holds for inequalities (E-CC) of model (E). Note that aside from the fact, that inequalities (C-CC) are defined on variables x, y, and (E-CC) on x^v, y^v, $v \in P$, the inequalities, and thus their separation, is the same. We thus only describe separation of inequalities (C-CC). Depending on whether the current (partial) solution (x^*, y^*) is integer or fractional, we use different separation routines

Separation of (C-CC) **for Integer Solutions** In case (x^*, y^*) is integer, the induced solution will consist of p or more cycles (which can be simply detected using e.g., breadth-first-search). For each cycle S not containing the central depot 0, a constraint (C-CC) is added. For i on the right-hand-side, we take a facility with the largest number of associated customers, ties are broken by taking the one with smallest index.

Separation of (C-CC) **for Fractional Solutions** In case (x^*, y^*) is fractional, it is well-known that connectivity cuts like (C-CC) can be separated using maximum flow computations on a graph, where the arc capacities are set to x^* (see, e.g., [2, 10]). A connectivity cut with facility i on the right-hand-side is violated if the maximum flow from the central depot 0 to i is less than the value of y_i^*. Let S be the set containing i in such a minimum cut. The set S is giving a violated constraint (C-CC). In order to speed up the separation, we sort the facilities in descending order by the values of y_i^* and separate in this order. Whenever we find a violated inequality, we remove all facilities in S for consideration of separation. The maximum flow/minimum cut computation is done using the algorithm of [6]. This algorithm may give back a second minimum cut S', if this happens, we also add the violated constraint induced by S'. Moreover, we add a small $\epsilon = 10^{-5}$ to all capacities before separation to get *minimal cardinality cuts*, see e.g., [15] for more details on the last two techniques.

Separation of Lifted Inequalities (C-CCL)**,** (C-CCL2) (E-CCL)**,** (E-CCL2) We do not explicitly separate the lifted inequalities, but instead try to lift inequalities (C-CC) (resp., (E-CC)) when they are found by the separation routine. This is simply done by checking, if $Z(S) \leq LB$ for the detected set S and the current incumbent objective value LB in case of (C-CC) for lifting to (C-CCL). If lifting to (C-CCL) is not successful, we try lifting to (C-CCL2) by checking all candidates fulfilling the facility dominance condition. If there is more than one candidate, we take the one with largest LP-value, ties are broken by taking the one with smallest index. In case of a violated inequality (E-CC) for $i \in S$, we check all $i' \neq i \in S$, if any of them fulfills the condition $d_{0i} + d_{ii'} + d_{i'0} > L_v$ for lifting to (E-CCL) or the facility dominance condition for lifting to (E-CCL2). Again, if there is more than one candidate for lifting, we take the one with largest LP-value, ties are broken by taking the one with smallest index.

Additional Details of the Separation Routine In order to avoid spending too much time in the separation routines, we limit the number of rounds of the separation-loop to 50 in the root node of the branch-and-cut, and to ten in all other nodes. Moreover, when using formulation (C) we initialize our model with $x_{ii'} + x_{i'i} \leq y_i$ for each $i \in F$ and the five nearest $i' \in F$ to i, these constraints are a special case of (C-CC) for $|S| = 2$. The corresponding inequalities in case of formulation (E) are $x^v_{ii'} + x^v_{i'i} \leq y^v_i$. These inequalities forbid subtours of size two. Note that they do not involve the central depot 0 as one of the vertices, as a tour going to a single facility is feasible, and would be forbidden by a constraint $x_{0i} + x_{i0} \leq y_i$.

3.2 Branching Priorities

Due to the structure of our models, branching on different set of variables will have different impacts on the structure of the solutions obtainable in the nodes of the branch-and-cut tree. CPLEX, which is the branch-and-cut solver we use, allows to give branching priorities to variables. In our implementation, we give the highest branching priorities to the facility variables y (resp., y^v), as for fixed facility variables, the customers covered in the solution can be found by inspection, and fixed facility variables have also implications on the arcs in the solution.

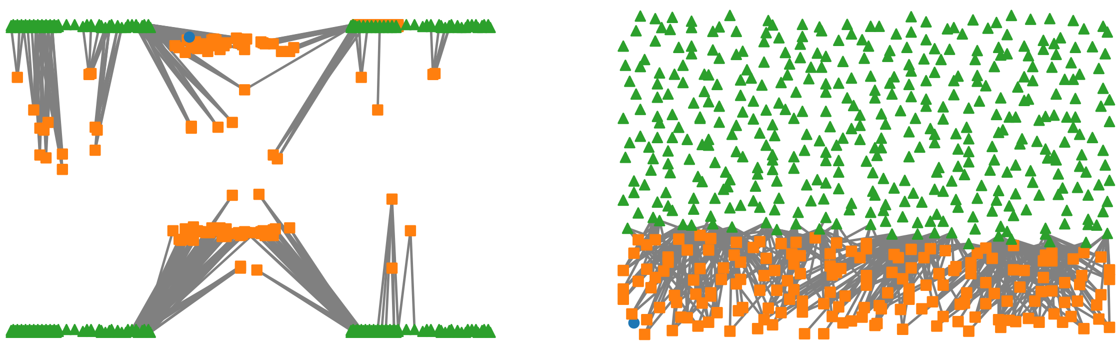
4 Computational Results

The branch-and-cut frameworks were implemented in C++ using CPLEX 12.8 as MIP solver. The runs were carried out on an Intel Xeon X5570 CPU with 2.93GHz and 48GB memory using a single thread, with timelimit for a run set to 600 seconds, and all CPLEX parameters (except branching priorities) are left at their default values.

4.1 Comparison with the MIP Approach of [1]

In this section, we compare our approaches with the MIP approach presented in [1] using the instances introduced in the same paper. We obtained these instances on request from the authors, and made them available at `https://msinnl.github.io/pages/instancescodes.html`. These instances are denoted as `AMSAL` in the following.

They are based on TSPLIB-instances with 52, 76, 100, 150, 200, 318, 417, 575, 657 and 724 vertices For each underlying TSPLIB-instance, three different instances were created by taking 50%, 60% and 70% of the vertices as customers, and the remaining vertices except one as facilities and one vertex as central depot. For each facility, the customers it can cover are randomly chosen from the five nearest ones (the TSPLIB-instances contain coordinates). Here we discovered an issue with the instances, namely, there are often some customers, which cannot be covered by any facility, and thus are useless (this can affect up to 90% of the customers of an instance in some cases). For most of the instances, this even led to the situation, that all customers, which could be covered by some facility in the instance were in the optimal solution (especially after taking account also customers unreachable due to the distance limit as described in Theorem 2). Figure 2 depicts two instances to illustrate this issue (we refer to instances by `|F|-|C|-p-L`). It is extremely pronounced, as the facility/customer-split is not chosen randomly among the vertices of the underlying TSPLIB-instance, but the the first 50% (resp., 40%, 30%) of vertices in the TSPLIB-instance are taken as facilities. From the figures, it can be seen that these vertices are clustered by location.



(a) Instance `125-291-2-9372.69`. (b) Instance `172-402-2-1732.68`.

Figure 2: Two instance-graphs of the TCMCRP from the set `AMSAL` from [1]. The blue circle is the central depot, orange boxes are facilities and green triangles are customers. The gray edges between facilities and customers denote which customers are covered by each facility.

For number of vehicles, $p = 2, 3, 4$ is used, and also three different values for the distance limit L are tested. In [1], the following formula, for $\alpha \in \{1, 0.9, 0.8\}$ is given to define the value for L:

$$L = \frac{\alpha}{p} \cdot |F| \cdot \frac{\sum_{i \in F \cup 0} \sum_{i' \in F \cup 0} d_{ii'}}{(|F| \cdot (|F| + 1))/2}$$

However, when verifying this formula, we noticed that it does not give the correct value, which could be obtained with the following formula instead:

$$L = \cdot \frac{\alpha}{p} \cdot |F| \cdot \frac{\sum_{i \in F \cup 0} \sum_{i' \in F \cup 0} d_{ii'}}{(|F| \cdot (|F| + 1))/4}.$$

The formula could be verified, as the respective values of L are also written explicitly in the obtained instance files and in the result-tables in [1]. However, in some of the instance-files, there were wrong values, and also some entries in the tables in [1] have wrong entries, in particular in Table 3 containing the so-called *large* instances containing 318 nodes and more. We discuss this in more detail later. We corrected these errors in the instance files for our computational study and our uploaded instances also consist of the corrected instances. In total, this set contains $10 \cdot 3 \cdot 3 \cdot 3 = 270$ instances (ten underlying graphs, and the different parameters for $|C|$, p and L). From the discussion above, one can already see, that these instances are maybe not too meaningful for benchmarking. However, as they are the only instances from literature for the problem, we still consider them, but also introduce new instances to evaluate our approaches in the Section 4.2.

First, we are interested in the effect of the different models and of the valid inequalities and branching priorities. We thus compare the following six settings:

- `C`: Model (C) without any variable-fixing and valid inequalities.
- `C+`: Model (C) with variable-fixing (i.e., (C-FIXA),(C-FIXF)) and valid inequalities (C-DISTG). Note that the resulting model is still compact.
- `C++`: `C+` with the separation of connectivity cuts (C-CC) (and the liftings (C-CCL) and (C-CCL2)) and also branching priorities as described in Section 3.2.
- `E`: Model (E) without any variable-fixing and valid inequalities and lifting.
- `E+`: Model (E) with all variable-fixing and valid inequalities (except lifting (C-CC) to (C-CCL) and (E-CCL2) as described in Section 3.1)
- `E++`: `E+` including lifting (C-CC) to (C-CCL) and (E-CCL2) as described in Section 3.1, and branching priorities as described in Section 3.2.

Figure 3 gives a plot of the runtime to optimality for the instances from set `AMSAL` for these settings.

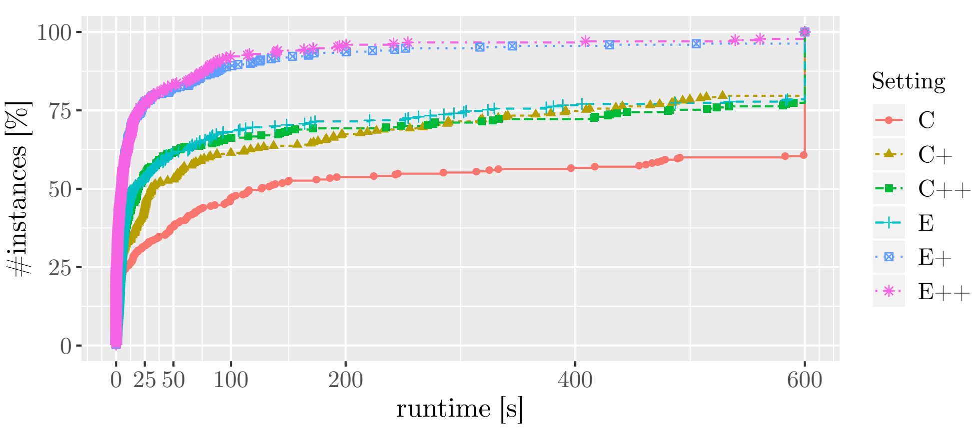


Figure 3: Runtime to optimality for instance set `AMSAL` and different settings.

Looking at Figure 3, the approaches based on model (E) seem more promising, as both `E+` and `E++` manage to solve nearly all of the instance to optimality within the given timelimit (and more than 75% of the instances within 25 seconds). `C+` and `C++` (and `E`) have nearly identical performance (solving about 75% of the instances), and both are considerably better than `C` (solving about 62.5% of the instances). Thus, it seems the variable fixing and the valid inequalities are quite helpful regardless of the model, while connectivity cuts (C-CC) are not really improving the performance for the approach based on (C), and lifting of connectivity cuts (E-CC) does not seem to helpful to improve the approach based on (E). As already mentioned above (and discussed in more detail below), for the instances from this set, the upper bound is often trivial, as the optimal solution will consist of any reachable facility. Thus, an explanation for the observed performance of the different settings could be that once the variable-fixings/valid inequalities are added, both MIP models already achieve (or are very near) the upper bound, and the runtime is mostly dominated by the time to solve the LP-relaxation and for the internal MIP-heuristics of CPLEX to find the optimal solution.

Next, we give a detailed overview of the results for the instance `AMSAL`, and settings `C+` and `E++`, and also compare our results with the ones reported by the MIP approach of [1]. The results in [1] were obtained using CPLEX 12.3 on an Intel Core i7 2.93 GHz processor with 3.49 GB of RAM. Tables 1-10 gives the results for these instances. In the tables, we report for each instance the number of nodes ($|V|$), the number of facilities ($|F|$), the number of customers ($|C|$), the number of vehicles (p), the distance limit (L), the number of customers, for which at least one facility covering it exists ($|C_T|$ "true customers"), the number of customers, for which at least one facility covering it is reachable from the central depot within the given distance limit ($|C_R|$ "reachable customers", i.e., following Theorem 2). For each setting, we report the runtime ($t[s]$), value of the best solution found (z^*), optimality gap ($g[\%]$, calculated as $100 \cdot ((UB - z^*)/z^*)$, where UB is the value of the upper bound found by the setting), and number of branch-and-bound nodes ($\#nBBn$; only for our settings, as these are not reported in [1]). A **bold** entry in z^* indicates that optimality has be proven (as can also be seen by the optimality gap $g[\%]$ being zero). An entry of "-" in z^* and $g[\%]$ indicates that no feasible solution could be found within the timelimit. A **bold** entry in $|C_T|$, resp., $|C_R|$ indicates, that the optimal solution consists of all true, resp., reachable customers. An *italic* entry in the column z^* for the results of [1] indicates that the reported solution value has some issues, which are discussed in detail in the following paragraph.

Note that for instances with up to 200 nodes, the facility-customer-coverage-allocation is the same for each underlying graph and facility/customer split. This means that for the same `|F|-|C|-p`, the optimal value of instances with larger L always gives an upper bound to the optimal value of instances with smaller L. The following instances files contained wrong values for L, and accordingly, there were also wrong results reported for these instances in [1]: instance `15-36-2-1090.05` was the same as instance `15-36-2-1368.07`, and instance `59-140-4-5005.31` was the same as instance `59-140-4-5630.98`. With regard to instance `15-36-2-1368.07`, the value in the instance-file and in the result-table in [1] is also not the same, it is 1386.07 versus 1386.03, with the former being the correct value when verified using the formula mentioned in the beginning of the section. For `15-36-2-1090.05`, the value of L in the table is also not the same as the one obtained by calculation with the formula, the correct value is 1094.46. Moreover, for instances `29-70-2-5574.4`, `29-70-2-4955.06`, `29-70-3-4129.21`, `29-70-3-3716.29` the table in [1] reports an optimal value of 36, while we found different (lower) optimal values. The (optimal) results for instances `25-26-4-894.71` and `25-26-4-795.30` also seem wrong compared to our results, and also conflict with the construction of the instances, as the optimal solution value reported for the instance with lower distance limit is larger than the one for the instance with larger distance limit Finally, in Table 3 in [1] (containing the instances with 318 nodes and more), there are many wrong combinations of `|F|-|C|-p-L` (i.e., not as occurring in the instance files). Moreover, $L = 1061.21$ in the table should read 51061.21, and $L = 7243.53$ is occurring twice in the table, while 6699.96 is occurring in an instance file, but missing in the table. As the values of L are unique in the instances, the result of Table 3 could still be useful for comparison by assuming that the best objective value reported for a given L gives a correct value and only $|F|$, $|C|$ and p was mixed up in the table. However, when we compared the reported solution values for a given L with our obtained solution values for the instance with this L, it was often larger than the optimal value we found and even larger than the number of true customers of the instance. Thus, it is not clear to us how to directly compare the results, resp., which of the solution values in Table 3 of [1] correspond to which instance. Hence we do not report the detailed results of [1] for instances of `AMSAL` with 318 nodes or more, but just note, that for only 38 of these 135 instances, the MIP approach of [1] managed to find a feasible solution within the given timelimit of 18000 seconds, and for 32 of them, optimality is reported (and for 112 of the other instances, leading to a total number of 144 out of 270, for which optimality is reported in [1]).

The results show that `E++` manages to solve 264 out of 270 instances to optimality and `C+` 215 instances. We note that for the larger instances, `C+` often does not manage to leave the root node (see, e.g., Table 10) within the given timelimit, as the size of the model likely becomes prohibitive. For smaller instances, the performance is quite similar to `E++`. Moreover, for 183 of the 270 instances, the value of the optimal solution is identical to the number of true terminals $|C_T|$ and for 236 instances, the value of the optimal solution is identical to the number of reachable terminals $|C_R|$.

Table 1: Detailed results for instance set AMSAL with 52 vertices

						C+				E++				[1]		
$\|F\|$	$\|C\|$	p	L	$\|C_T\|$	$\|C_R\|$	$t[s]$	z^*	$g[\%]$	$\#nBBn$	$t[s]$	z^*	$g[\%]$	$\#nBBn$	$t[s]$	z^*	$g[\%]$
15	36	2	1368.07	23	19	0	**15**	0.00	0	0	**15**	0.00	0	0	**15**	0.00
15	36	2	1231.26	23	**12**	0	**12**	0.00	0	0	**12**	0.00	0	0	**12**	0.00
15	36	2	1094.46	23	**12**	0	**12**	0.00	0	0	**12**	0.00	0	0	***15***	0.00
15	36	3	912.05	23	12	0	**11**	0.00	0	0	**11**	0.00	0	0	**11**	0.00
15	36	3	820.84	23	**9**	0	**9**	0.00	0	0	**9**	0.00	0	0	**9**	0.00
15	36	3	729.64	23	**8**	0	**8**	0.00	0	0	**8**	0.00	0	0	**8**	0.00
15	36	4	684.04	23	**8**	0	**8**	0.00	0	0	**8**	0.00	0	0	**8**	0.00
15	36	4	615.63	23	**8**	0	**8**	0.00	0	0	**8**	0.00	0	0	**8**	0.00
15	36	4	547.23	23	**1**	0	**1**	0.00	0	0	**1**	0.00	0	0	**1**	0.00
20	31	2	1707.37	21	17	0	**15**	0.00	0	0	**15**	0.00	0	4	**15**	0.00
20	31	2	1536.63	21	17	0	**15**	0.00	0	0	**15**	0.00	0	1	**15**	0.00
20	31	2	1365.89	21	17	0	**14**	0.00	0	0	**14**	0.00	8	1	**14**	0.00
20	31	3	1138.24	21	**13**	0	**13**	0.00	0	0	**13**	0.00	0	0	**13**	0.00
20	31	3	1024.42	21	**13**	0	**13**	0.00	0	0	**13**	0.00	0	0	**13**	0.00
20	31	3	910.59	21	13	0	**12**	0.00	0	0	**12**	0.00	0	1	**12**	0.00
20	31	4	853.68	21	**10**	0	**10**	0.00	0	0	**10**	0.00	0	0	**10**	0.00
20	31	4	768.31	21	**9**	0	**9**	0.00	0	0	**9**	0.00	0	0	**9**	0.00
20	31	4	682.95	21	**9**	0	**9**	0.00	0	0	**9**	0.00	0	0	**9**	0.00
25	26	2	1988.24	21	21	0	**20**	0.00	0	0	**20**	0.00	0	30	**20**	0.00
25	26	2	1789.42	21	20	0	**19**	0.00	19	0	**19**	0.00	0	20	**19**	0.00
25	26	2	1590.59	21	20	0	**18**	0.00	50	0	**18**	0.00	0	3	**18**	0.00
25	26	3	1325.49	21	18	0	**17**	0.00	3	0	**17**	0.00	0	57	**17**	0.00
25	26	3	1192.94	21	**15**	0	**15**	0.00	0	0	**15**	0.00	0	2	**15**	0.00
25	26	3	1060.39	21	**15**	0	**15**	0.00	0	0	**15**	0.00	0	0	**15**	0.00
25	26	4	994.12	21	**15**	0	**15**	0.00	0	0	**15**	0.00	0	1	**14**	0.00
25	26	4	894.71	21	15	0	**14**	0.00	0	0	**14**	0.00	0	22	***11***	0.00
25	26	4	795.30	21	**11**	0	**11**	0.00	0	0	**11**	0.00	0	0	***12***	0.00

Table 2: Detailed results for instance set `AMSAL` with 76 vertices

$\|F\|$	$\|C\|$	p	L	$\|C_T\|$	$\|C_R\|$	`C+` $t[s]$	`C+` z^*	`C+` $g[\%]$	`C+` $\#nBBn$	`E++` $t[s]$	`E++` z^*	`E++` $g[\%]$	`E++` $\#nBBn$	[1] $t[s]$	[1] z^*	[1] $g[\%]$
22	53	2	11960.48	16	**12**	0	**12**	0.00	0	0	**12**	0.00	0	0	**12**	0.00
22	53	2	10764.43	16	**12**	0	**12**	0.00	0	0	**12**	0.00	0	0	**12**	0.00
22	53	2	9568.38	16	**10**	0	**10**	0.00	0	0	**10**	0.00	0	0	**10**	0.00
22	53	3	7973.65	16	**10**	0	**10**	0.00	0	0	**10**	0.00	0	0	**10**	0.00
22	53	3	7176.29	16	**7**	0	**7**	0.00	0	0	**7**	0.00	0	0	**7**	0.00
22	53	3	6378.92	16	**7**	0	**7**	0.00	0	0	**7**	0.00	0	0	**7**	0.00
22	53	4	5980.24	16	**5**	0	**5**	0.00	0	0	**5**	0.00	0	0	**5**	0.00
22	53	4	5382.22	16	**5**	0	**5**	0.00	0	0	**5**	0.00	0	0	**5**	0.00
22	53	4	4784.19	16	**5**	0	**5**	0.00	0	0	**5**	0.00	0	0	**5**	0.00
30	45	2	17564.36	17	**16**	0	**16**	0.00	0	0	**16**	0.00	0	3	**16**	0.00
30	45	2	15807.92	17	**16**	0	**16**	0.00	0	0	**16**	0.00	0	1	**16**	0.00
30	45	2	14051.49	17	14	0	**13**	0.00	11	0	**13**	0.00	0	14	**13**	0.00
30	45	3	11709.57	17	**13**	0	**13**	0.00	0	0	**13**	0.00	0	0	**13**	0.00
30	45	3	10538.61	17	**13**	0	**13**	0.00	0	0	**13**	0.00	0	0	**13**	0.00
30	45	3	9367.66	17	**12**	0	**12**	0.00	0	0	**12**	0.00	0	0	**12**	0.00
30	45	4	8782.18	17	**12**	0	**12**	0.00	0	0	**12**	0.00	0	0	**12**	0.00
30	45	4	7903.96	17	**12**	0	**12**	0.00	0	0	**12**	0.00	0	0	**12**	0.00
30	45	4	7025.74	17	**7**	0	**7**	0.00	0	0	**7**	0.00	0	0	**7**	0.00
37	38	2	22400.71	**17**	**17**	0	**17**	0.00	0	0	**17**	0.00	0	5	**17**	0.00
37	38	2	20160.64	**17**	**17**	0	**17**	0.00	0	0	**17**	0.00	0	2	**17**	0.00
37	38	2	17920.57	17	**15**	0	**15**	0.00	0	0	**15**	0.00	0	3	**15**	0.00
37	38	3	14933.80	17	**14**	0	**14**	0.00	0	0	**14**	0.00	0	1	**14**	0.00
37	38	3	13440.42	17	**14**	0	**14**	0.00	0	0	**14**	0.00	0	0	**14**	0.00
37	38	3	11947.04	17	**12**	0	**12**	0.00	0	0	**12**	0.00	0	0	**12**	0.00
37	38	4	11200.35	17	**12**	0	**12**	0.00	0	0	**12**	0.00	0	0	**12**	0.00
37	38	4	10080.32	17	**11**	0	**11**	0.00	0	0	**11**	0.00	0	0	**11**	0.00
37	38	4	8960.28	17	**9**	0	**9**	0.00	0	0	**9**	0.00	0	0	**9**	0.00

Table 3: Detailed results for instance set `AMSAL` with 100 vertices

						C+				E++				[1]		
$\|F\|$	$\|C\|$	p	L	$\|C_T\|$	$\|C_R\|$	$t[s]$	z^*	$g[\%]$	$\#nBBn$	$t[s]$	z^*	$g[\%]$	$\#nBBn$	$t[s]$	z^*	$g[\%]$
29	70	2	6193.82	43	43	58	**36**	0.00	5954	3	**36**	0.00	107	350	**36**	0.00
29	70	2	5574.44	43	40	1	**34**	0.00	227	1	**34**	0.00	0	6	***36***	0.00
29	70	2	4955.06	43	35	0	**30**	0.00	19	1	**30**	0.00	21	3	***36***	0.00
29	70	3	4129.21	43	**27**	0	**27**	0.00	0	0	**27**	0.00	0	0	***36***	0.00
29	70	3	3716.29	43	**24**	0	**24**	0.00	0	0	**24**	0.00	0	0	***36***	0.00
29	70	3	3303.37	43	**21**	0	**21**	0.00	0	0	**21**	0.00	0	0	**21**	0.00
29	70	4	3096.91	43	**21**	0	**21**	0.00	0	0	**21**	0.00	0	0	**21**	0.00
29	70	4	2787.22	43	**16**	0	**16**	0.00	0	0	**16**	0.00	0	0	**16**	0.00
29	70	4	2477.53	43	**14**	0	**14**	0.00	0	0	**14**	0.00	0	0	**14**	0.00
39	60	2	8326.08	**51**	**51**	1	**51**	0.00	0	1	**51**	0.00	0	92	**51**	0.00
39	60	2	7493.47	**51**	**51**	1	**51**	0.00	0	1	**51**	0.00	0	2	**51**	0.00
39	60	2	6660.87	51	51	379	**49**	0.00	13695	1	**49**	0.00	0	10564	**49**	0.00
39	60	3	5550.72	51	49	339	**48**	0.00	30103	1	**48**	0.00	3	18000	48	2.08
39	60	3	4995.65	51	47	435	**40**	0.00	35216	16	**40**	0.00	261	5434	**40**	0.00
39	60	3	4440.58	51	39	20	**38**	0.00	3519	2	**38**	0.00	44	314	**38**	0.00
39	60	4	4163.04	51	**34**	0	**34**	0.00	0	0	**34**	0.00	0	10	**34**	0.00
39	60	4	3746.74	51	**32**	0	**32**	0.00	0	0	**32**	0.00	0	1	**32**	0.00
39	60	4	3330.43	51	**27**	0	**27**	0.00	0	0	**27**	0.00	0	2	**27**	0.00
49	50	2	10458.99	**45**	**45**	5	**45**	0.00	0	0	**45**	0.00	0	8	**45**	0.00
49	50	2	9413.09	**45**	**45**	1	**45**	0.00	0	1	**45**	0.00	0	27	**45**	0.00
49	50	2	8367.19	**45**	**45**	3	**45**	0.00	0	0	**45**	0.00	0	37	**45**	0.00
49	50	3	6972.66	**45**	**45**	18	**45**	0.00	133	1	**45**	0.00	0	1018	**45**	0.00
49	50	3	6275.93	**45**	**45**	1	**45**	0.00	0	1	**45**	0.00	0	1175	**45**	0.00
49	50	3	5578.13	45	43	600	42	2.38	14369	19	**42**	0.00	17	18000	42	2.38
49	50	4	5229.49	45	43	600	42	2.38	30545	3	**42**	0.00	38	18000	42	2.38
49	50	4	4706.55	45	**37**	1	**37**	0.00	0	0	**37**	0.00	0	127	**37**	0.00
49	50	4	4183.60	45	**33**	0	**33**	0.00	0	1	**33**	0.00	0	12	**33**	0.00

Table 4: Detailed results for instance set AMSAL with 150 vertices

$\|F\|$	$\|C\|$	p	L	$\|C_T\|$	$\|C_R\|$	C+ $t[s]$	C+ z^*	C+ $g[\%]$	C+ $\#nBBn$	E++ $t[s]$	E++ z^*	E++ $g[\%]$	E++ $\#nBBn$	[1] $t[s]$	[1] z^*	[1] $g[\%]$
44	105	2	9210.94	**72**	**72**	24	**72**	0.00	465	1	**72**	0.00	0	454	**72**	0.00
44	105	2	8289.84	**72**	**72**	117	**72**	0.00	2265	2	**72**	0.00	0	1729	**72**	0.00
44	105	2	7368.75	72	72	600	70	2.86	11395	28	**70**	0.00	153	18000	70	2.86
44	105	3	6140.62	**72**	**72**	509	**72**	0.00	10824	4	**72**	0.00	4	18000	69	4.35
44	105	3	5526.56	72	72	2	**69**	0.00	0	4	**69**	0.00	0	56	**69**	0.00
44	105	3	4912.50	72	68	600	64	4.51	18950	81	**64**	0.00	523	18000	64	1.56
44	105	4	4605.47	72	67	600	66	1.52	28826	72	**66**	0.00	44	18000	66	1.52
44	105	4	4144.92	72	64	600	58	1.72	37624	409	**58**	0.00	2969	18000	58	2.58
44	105	4	3684.37	72	54	65	**49**	0.00	7497	26	**49**	0.00	211	1555	**49**	0.00
59	90	2	12364.81	**71**	**71**	68	**71**	0.00	942	2	**71**	0.00	9	45	**71**	0.00
59	90	2	11128.36	**71**	**71**	8	**71**	0.00	10	1	**71**	0.00	0	614	**71**	0.00
59	90	2	9891.85	**71**	**71**	227	**71**	0.00	1585	2	**71**	0.00	0	2193	**71**	0.00
59	90	3	8243.20	**71**	**71**	7	**71**	0.00	0	8	**71**	0.00	0	18000	70	1.43
59	90	3	7418.88	**71**	**71**	6	**71**	0.00	0	5	**71**	0.00	4	6268	**71**	0.00
59	90	3	6594.56	**71**	**71**	413	**71**	0.00	3378	4	**71**	0.00	0	13768	69	2.90
59	90	4	6182.40	**71**	**71**	19	**71**	0.00	121	9	**71**	0.00	0	2920	**71**	0.00
59	90	4	5564.16	**71**	**71**	33	**71**	0.00	100	3	**71**	0.00	0	5404	**71**	0.00
59	90	4	4945.92	71	68	600	65	3.08	8771	600	65	3.08	250	18000	65	4.43
74	75	2	15315.62	**64**	**64**	5	**64**	0.00	0	1	**64**	0.00	0	32	**64**	0.00
74	75	2	13784.06	**64**	**64**	53	**64**	0.00	200	1	**64**	0.00	0	24	**64**	0.00
74	75	2	12252.49	**64**	**64**	6	**64**	0.00	0	2	**64**	0.00	2	174	**64**	0.00
74	75	3	10210.41	**64**	**64**	82	**64**	0.00	505	3	**64**	0.00	0	176	**64**	0.00
74	75	3	9189.37	**64**	**64**	289	**64**	0.00	1112	7	**64**	0.00	0	279	**64**	0.00
74	75	3	8168.33	**64**	**64**	52	**64**	0.00	160	12	**64**	0.00	0	18000	63	1.59
74	75	4	7657.81	**64**	**64**	70	**64**	0.00	159	11	**64**	0.00	5	3799	**64**	0.00
74	75	4	6892.03	**64**	**64**	111	**64**	0.00	320	9	**64**	0.00	0	417913	**64**	0.00
74	75	4	6126.25	**64**	**64**	13	**64**	0.00	0	193	**64**	0.00	12	7620	**64**	0.00

Table 5: Detailed results for instance set `AMSAL` with 200 vertices

$\|F\|$	$\|C\|$	p	L	$\|C_T\|$	$\|C_R\|$	`C+` $t[s]$	`C+` z^*	`C+` $g[\%]$	`C+` $\#nBBn$	`E++` $t[s]$	`E++` z^*	`E++` $g[\%]$	`E++` $\#nBBn$	[1] $t[s]$	[1] z^*	[1] $g[\%]$
59	140	2	12513.29	**104**	**104**	474	**104**	0.00	8092	7	**104**	0.00	7	89	**104**	0.00
59	140	2	11261.96	**104**	**104**	528	**104**	0.00	4292	41	**104**	0.00	28	173	**104**	0.00
59	140	2	10010.63	**104**	**104**	600	102	1.96	5104	96	**104**	0.00	60	18000	102	1.96
59	140	3	8342.19	**104**	**104**	600	98	6.12	4872	242	**104**	0.00	145	18000	92	13.04
59	140	3	7507.97	**104**	**104**	600	102	1.96	2342	254	**104**	0.00	90	10308	**104**	0.00
59	140	3	6673.75	104	103	600	99	4.04	3289	600	101	1.98	77	18000	101	1.98
59	140	4	6256.64	104	103	600	95	7.37	3623	600	89	14.61	267	18000	100	2.74
59	140	4	5630.98	104	90	600	89	1.12	7288	600	89	1.12	313	18000	88	2.27
59	140	4	5005.31	104	87	3	**83**	0.00	0	116	**83**	0.00	137	18000	88	2.27
79	120	2	16673.37	**107**	**107**	66	**107**	0.00	200	2	**107**	0.00	0	60	**107**	0.00
79	120	2	15006.03	**107**	**107**	317	**107**	0.00	1000	2	**107**	0.00	0	89	**107**	0.00
79	120	2	13338.69	**107**	**107**	14	**107**	0.00	0	15	**107**	0.00	21	262	**107**	0.00
79	120	3	11115.58	**107**	**107**	112	**107**	0.00	300	53	**107**	0.00	24	9168	**107**	0.00
79	120	3	10004.02	**107**	**107**	219	**107**	0.00	538	45	**107**	0.00	0	18000	103	3.88
79	120	3	8892.46	**107**	**107**	289	**107**	0.00	729	172	**107**	0.00	57	18000	100	7.00
79	120	4	8336.68	**107**	**107**	600	93	15.05	1497	561	**107**	0.00	145	18000	104	2.89
79	120	4	7503.01	107	107	600	103	3.88	972	600	-	-	42	18000	103	3.88
79	120	4	6669.35	107	106	600	105	0.95	1121	600	-	-	4	18000	95	11.58
99	100	2	20883.45	**94**	**94**	500	**94**	0.00	896	3	**94**	0.00	0	101	**94**	0.00
99	100	2	18795.10	**94**	**94**	275	**94**	0.00	486	2	**94**	0.00	0	178	**94**	0.00
99	100	2	16706.76	**94**	**94**	20	**94**	0.00	0	3	**94**	0.00	0	254	**94**	0.00
99	100	3	13922.30	**94**	**94**	465	**94**	0.00	360	5	**94**	0.00	0	96	**94**	0.00
99	100	3	12530.07	**94**	**94**	366	**94**	0.00	485	5	**94**	0.00	0	460	**94**	0.00
99	100	3	11137.84	**94**	**94**	600	82	14.63	1023	30	**94**	0.00	0	646	**94**	0.00
99	100	4	10441.72	**94**	**94**	129	**94**	0.00	200	21	**94**	0.00	0	1058	**94**	0.00
99	100	4	9397.55	**94**	**94**	487	**94**	0.00	972	45	**94**	0.00	10	8875	**94**	0.00
99	100	4	8353.38	**94**	**94**	600	88	6.82	693	140	**94**	0.00	29	3986	**94**	0.00

Table 6: Detailed results for instance set AMSAL for instances with 318 vertices

$\|F\|$	$\|C\|$	p	L	$\|C_T\|$	$\|C_R\|$	C+ $t[s]$	C+ z^*	C+ $g[\%]$	C+ $\#nBBn$	E++ $t[s]$	E++ z^*	E++ $g[\%]$	E++ $\#nBBn$
95	222	2	10719.94	**43**	**43**	100	**43**	0.00	160	1	**43**	0.00	0
95	222	3	7146.63	**43**	**43**	21	**43**	0.00	0	6	**43**	0.00	0
95	222	4	5359.97	42	**36**	18	**36**	0.00	0	3	**36**	0.00	0
95	222	2	12059.93	**40**	**40**	13	**40**	0.00	0	1	**40**	0.00	0
95	222	3	8039.95	**39**	**39**	77	**39**	0.00	110	5	**39**	0.00	0
95	222	4	6029.97	42	**40**	18	**40**	0.00	0	15	**40**	0.00	0
95	222	2	13399.92	**43**	**43**	9	**43**	0.00	0	1	**43**	0.00	0
95	222	3	8933.28	**39**	**39**	5	**39**	0.00	0	2	**39**	0.00	0
95	222	4	6699.96	**41**	**41**	24	**41**	0.00	0	13	**41**	0.00	0
127	190	2	17407.31	**36**	**36**	10	**36**	0.00	0	2	**36**	0.00	0
127	190	3	11604.87	**41**	**41**	26	**41**	0.00	0	8	**41**	0.00	0
127	190	4	8703.65	**37**	**37**	27	**37**	0.00	0	7	**37**	0.00	0
127	190	2	19583.22	**39**	**39**	22	**39**	0.00	10	1	**39**	0.00	0
127	190	3	13055.48	**43**	**43**	22	**43**	0.00	21	3	**43**	0.00	0
127	190	4	9791.61	**44**	**44**	30	**44**	0.00	0	4	**44**	0.00	0
127	190	2	21759.13	**40**	**40**	25	**40**	0.00	12	1	**40**	0.00	0
127	190	3	14506.09	**40**	**40**	8	**40**	0.00	0	2	**40**	0.00	0
127	190	4	10879.57	**38**	**38**	25	**38**	0.00	0	7	**38**	0.00	0
158	159	2	22073.87	**40**	**40**	13	**40**	0.00	0	1	**40**	0.00	0
158	159	3	14715.91	**37**	**37**	22	**37**	0.00	0	6	**37**	0.00	0
158	159	4	11036.93	**37**	**37**	59	**37**	0.00	11	10	**37**	0.00	0
158	159	2	24833.10	**41**	**41**	16	**41**	0.00	0	2	**41**	0.00	0
158	159	3	16555.40	**42**	**42**	74	**42**	0.00	30	4	**42**	0.00	0
158	159	4	12416.55	**39**	**39**	81	**39**	0.00	0	5	**39**	0.00	0
158	159	2	27592.33	**40**	**40**	9	**40**	0.00	0	2	**40**	0.00	0
158	159	3	18394.89	**39**	**39**	45	**39**	0.00	0	4	**39**	0.00	0
158	159	4	13796.17	**39**	**39**	53	**39**	0.00	0	10	**39**	0.00	4

Table 7: Detailed results for instance set AMSAL for instances with 417 vertices

$\|F\|$	$\|C\|$	p	L	$\|C_T\|$	$\|C_R\|$	C+ $t[s]$	C+ z^*	C+ $g[\%]$	C+ $\#nBBn$	E++ $t[s]$	E++ z^*	E++ $g[\%]$	E++ $\#nBBn$
125	291	2	9372.69	**75**	**75**	30	**75**	0.00	20	1	**75**	0.00	0
125	291	3	6248.46	**72**	**72**	441	**72**	0.00	483	4	**72**	0.00	0
125	291	4	4686.35	**75**	**75**	14	**75**	0.00	0	11	**75**	0.00	0
125	291	2	10544.28	**73**	**73**	514	**73**	0.00	540	1	**73**	0.00	0
125	291	3	7029.52	**73**	**73**	288	**73**	0.00	509	5	**73**	0.00	0
125	291	4	5272.14	**71**	**71**	256	**71**	0.00	330	11	**71**	0.00	0
125	291	2	11715.87	**75**	**75**	30	**75**	0.00	20	4	**75**	0.00	0
125	291	3	7810.58	**72**	**72**	213	**72**	0.00	171	2	**72**	0.00	0
125	291	4	5857.93	**67**	**67**	7	**67**	0.00	0	5	**67**	0.00	0
166	250	2	12877.39	**63**	**63**	176	**63**	0.00	90	1	**63**	0.00	0
166	250	3	8584.93	**64**	**64**	18	**64**	0.00	0	9	**64**	0.00	0
166	250	4	6438.69	**65**	**65**	158	**65**	0.00	40	9	**65**	0.00	0
166	250	2	14487.06	**63**	**63**	28	**63**	0.00	0	3	**63**	0.00	0
166	250	3	9658.04	**61**	**61**	27	**61**	0.00	0	4	**61**	0.00	0
166	250	4	7243.53	**65**	**65**	31	**65**	0.00	0	10	**65**	0.00	0
166	250	2	16096.74	**58**	**58**	56	**58**	0.00	20	2	**58**	0.00	0
166	250	3	10731.16	**62**	**62**	26	**62**	0.00	0	6	**62**	0.00	0
166	250	4	8048.37	**63**	**63**	26	**63**	0.00	0	8	**63**	0.00	0
208	208	2	17827.96	**58**	**58**	39	**58**	0.00	0	3	**58**	0.00	0
208	208	3	11885.31	**54**	**54**	38	**54**	0.00	0	8	**54**	0.00	0
208	208	4	8913.98	**58**	**58**	197	**58**	0.00	40	21	**58**	0.00	0
208	208	2	20056.46	**58**	**58**	25	**58**	0.00	0	2	**58**	0.00	0
208	208	3	13370.97	**58**	**58**	25	**58**	0.00	0	5	**58**	0.00	0
208	208	4	10028.23	**57**	**57**	88	**57**	0.00	19	17	**57**	0.00	2
208	208	2	22284.96	**57**	**57**	74	**57**	0.00	10	3	**57**	0.00	0
208	208	3	14856.64	**59**	**59**	50	**59**	0.00	2	14	**59**	0.00	0
208	208	4	11142.48	**60**	**60**	333	**60**	0.00	120	13	**60**	0.00	0

Table 8: Detailed results for instance set AMSAL for instances with 575 vertices

						C+				E++			
$\|F\|$	$\|C\|$	p	L	$\|C_T\|$	$\|C_R\|$	$t[s]$	z^*	$g[\%]$	$\#nBBn$	$t[s]$	z^*	$g[\%]$	$\#nBBn$
172	402	2	1732.69	**27**	**27**	33	**27**	0.00	4	1	**27**	0.00	0
172	402	3	1155.12	**26**	**26**	11	**26**	0.00	0	6	**26**	0.00	0
172	402	4	866.34	**30**	**30**	16	**30**	0.00	0	6	**30**	0.00	0
172	402	2	1949.27	**30**	**30**	13	**30**	0.00	0	1	**30**	0.00	0
172	402	3	1299.51	**27**	**27**	6	**27**	0.00	0	3	**27**	0.00	0
172	402	4	974.64	**28**	**28**	5	**28**	0.00	0	3	**28**	0.00	0
172	402	2	2165.86	**26**	**26**	16	**26**	0.00	0	1	**26**	0.00	0
172	402	3	1443.91	**29**	**29**	34	**29**	0.00	8	6	**29**	0.00	0
172	402	4	1082.93	**28**	**28**	7	**28**	0.00	0	4	**28**	0.00	0
229	345	2	2589.77	**27**	**27**	24	**27**	0.00	0	5	**27**	0.00	0
229	345	3	1726.51	**27**	**27**	24	**27**	0.00	0	6	**27**	0.00	0
229	345	4	1294.88	**30**	**30**	137	**30**	0.00	0	10	**30**	0.00	0
229	345	2	2913.49	**28**	**28**	53	**28**	0.00	0	2	**28**	0.00	0
229	345	3	1942.32	**29**	**29**	24	**29**	0.00	0	10	**29**	0.00	1
229	345	4	1456.74	**27**	**27**	26	**27**	0.00	0	12	**27**	0.00	0
229	345	2	3237.21	**26**	**26**	29	**26**	0.00	0	6	**26**	0.00	0
229	345	3	2158.14	**30**	**30**	57	**30**	0.00	0	15	**30**	0.00	0
229	345	4	1618.60	**29**	**29**	16	**29**	0.00	0	26	**29**	0.00	0
287	287	2	3624.82	**27**	**27**	50	**27**	0.00	0	6	**27**	0.00	0
287	287	3	2416.55	**29**	**29**	30	**29**	0.00	0	14	**29**	0.00	0
287	287	4	1812.41	**28**	**28**	189	**28**	0.00	0	36	**28**	0.00	0
287	287	2	4077.93	**30**	**30**	233	**30**	0.00	10	14	**30**	0.00	0
287	287	3	2718.62	**31**	**31**	26	**31**	0.00	0	13	**31**	0.00	0
287	287	4	2038.96	**30**	**30**	59	**30**	0.00	2	34	**30**	0.00	0
287	287	2	4531.03	**30**	**30**	26	**30**	0.00	0	5	**30**	0.00	0
287	287	3	3020.69	**30**	**30**	600	24	25.00	0	11	**30**	0.00	0
287	287	4	2265.51	**30**	**30**	29	**30**	0.00	0	16	**30**	0.00	0

Table 9: Detailed results for instance set AMSAL for instances with 657 vertices

						C+				E++			
$\|F\|$	$\|C\|$	p	L	$\|C_T\|$	$\|C_R\|$	$t[s]$	z^*	$g[\%]$	$\#nBBn$	$t[s]$	z^*	$g[\%]$	$\#nBBn$
197	459	2	14375.04	**63**	**63**	124	**63**	0.00	0	9	**63**	0.00	0
197	459	3	9583.36	**67**	**67**	341	**67**	0.00	10	16	**67**	0.00	0
197	459	4	7187.52	**68**	**68**	600	-	-	0	539	**68**	0.00	61
197	459	2	16171.92	**65**	**65**	189	**65**	0.00	0	7	**65**	0.00	0
197	459	3	10781.28	**67**	**67**	271	**67**	0.00	0	19	**67**	0.00	0
197	459	4	8085.96	**68**	**68**	391	**68**	0.00	0	139	**68**	0.00	0
197	459	2	17968.80	**68**	**68**	55	**68**	0.00	0	5	**68**	0.00	0
197	459	3	11979.20	**66**	**66**	392	**66**	0.00	0	14	**66**	0.00	0
197	459	4	8984.40	**67**	**67**	319	**67**	0.00	40	87	**67**	0.00	0
262	394	2	22208.68	**76**	**76**	196	**76**	0.00	0	8	**76**	0.00	0
262	394	3	14805.79	**76**	**76**	600	64	18.75	10	36	**76**	0.00	0
262	394	4	11104.34	**78**	**78**	600	-	-	0	90	**78**	0.00	0
262	394	2	24984.76	**77**	**77**	182	**77**	0.00	0	10	**77**	0.00	0
262	394	3	16656.51	**78**	**78**	170	**78**	0.00	0	48	**78**	0.00	3
262	394	4	12492.38	**77**	**77**	600	63	22.22	21	69	**77**	0.00	0
262	394	2	27760.85	**76**	**76**	267	**76**	0.00	0	13	**76**	0.00	0
262	394	3	18507.23	**73**	**73**	484	**73**	0.00	20	20	**73**	0.00	0
262	394	4	13880.42	**81**	**81**	88	**81**	0.00	0	21	**81**	0.00	0
328	328	2	30838.17	**83**	**83**	494	**83**	0.00	0	15	**83**	0.00	0
328	328	3	20558.78	**84**	**84**	600	-	-	0	27	**84**	0.00	0
328	328	4	15419.08	**83**	**83**	600	-	-	0	73	**83**	0.00	0
328	328	2	34692.94	**82**	**82**	600	-	-	2	23	**82**	0.00	4
328	328	3	23128.62	**83**	**83**	600	-	-	1	63	**83**	0.00	0
328	328	4	17346.47	**80**	**80**	600	-	-	1	83	**80**	0.00	4
328	328	2	38547.71	**78**	**78**	183	**78**	0.00	0	14	**78**	0.00	0
328	328	3	25698.47	**82**	**82**	172	**82**	0.00	0	42	**82**	0.00	3
328	328	4	19273.85	**81**	**81**	409	**81**	0.00	0	31	**81**	0.00	0

Table 10: Detailed results for instance set AMSAL for instances with 724 vertices

						C+				E++			
$\|F\|$	$\|C\|$	p	L	$\|C_T\|$	$\|C_R\|$	$t[s]$	z^*	$g[\%]$	$\#nBBn$	$t[s]$	z^*	$g[\%]$	$\#nBBn$
217	506	2	19912.23	**176**	**176**	600	90	95.56	3	6	**176**	0.00	0
217	506	3	13274.82	**187**	**187**	600	-	-	2	31	**187**	0.00	0
217	506	4	9956.12	**187**	**187**	600	-	-	32	81	**187**	0.00	0
217	506	2	22401.26	**179**	**179**	600	154	16.23	5	14	**179**	0.00	0
217	506	3	14934.17	**179**	**179**	600	133	34.59	3	67	**179**	0.00	10
217	506	4	11200.63	**184**	**184**	600	122	50.82	0	64	**184**	0.00	0
217	506	2	24890.29	**180**	**180**	600	178	1.12	100	17	**180**	0.00	0
217	506	3	16593.53	**183**	**183**	600	165	10.91	1	41	**183**	0.00	5
217	506	4	12445.15	**176**	**176**	600	97	81.44	10	81	**176**	0.00	0
289	434	2	29871.47	**187**	**187**	600	-	-	1	88	**187**	0.00	3
289	434	3	19914.32	**183**	**183**	600	-	-	0	88	**183**	0.00	3
289	434	4	14935.74	**175**	**175**	600	-	-	0	195	**175**	0.00	0
289	434	2	33605.41	**184**	**184**	600	145	26.90	1	26	**184**	0.00	0
289	434	3	22403.60	**176**	**176**	600	-	-	0	52	**176**	0.00	0
289	434	4	16802.70	**183**	**183**	600	-	-	1	200	**183**	0.00	0
289	434	2	37339.34	**177**	**177**	600	-	-	1	23	**177**	0.00	0
289	434	3	24892.89	**182**	**182**	600	-	-	0	80	**182**	0.00	0
289	434	4	18669.67	**174**	**174**	600	-	-	0	114	**174**	0.00	0
361	362	2	40848.98	**205**	**205**	600	-	-	0	76	**205**	0.00	0
361	362	3	27232.65	**200**	**200**	600	-	-	0	77	**200**	0.00	0
361	362	4	20424.49	**214**	**214**	600	-	-	0	164	**214**	0.00	0
361	362	2	45955.09	**195**	**195**	600	-	-	0	96	**195**	0.00	0
361	362	3	30636.73	**207**	**207**	600	-	-	0	141	**207**	0.00	0
361	362	4	22977.55	**204**	**204**	600	-	-	0	68	**204**	0.00	0
361	362	2	51061.22	**208**	**208**	600	-	-	0	61	**208**	0.00	3
361	362	3	34040.81	**206**	**206**	600	-	-	0	73	**206**	0.00	0
361	362	4	25530.61	**216**	**216**	600	-	-	0	100	**216**	0.00	0

4.2 Evaluating our MIP Approaches on New Instances

As the previous section has shown, due to the structure of the instances from AMSAL, it is quite easy to find tight upper bounds. Thus, the main difficulty to solve these instances to optimality is to find the optimal solutions and our approaches seem quite effective also for this purpose. In this section, we introduce a new set of instances, denoted as NEW. The instances are designed in such a way, that for all customers, there are at least two facilities covering it in the underlying graph. Note that due to the nature of the problem, it can still happen, that after applying the distance limit based variable-fixing/preprocessing as described in Theorem 2 that some customers cannot be reached for some values of L.

The instances are made available at https://msinnl.github.io/pages/instancescodes.html and are constructed as follows: $|F|$ facilities and $|C|$ customers are picked by taking random integers within $[0, 1000]$ as the location of them in the Euclidean plane. The central depot is placed at point $(500, 500)$. This is done for the following pairs of $(|F|, |C|)$: $(75, 225), (100, 300), (125, 375)$, for each pair three graphs are constructed. For the facility-customer coverage, we randomly pick between two and five of the nearest facilities for each customer. With this we ensure, that each customer can be covered (and also no trivial one-to-one relationship between some facility and customer can occur). Based on these underlying graphs, instances are created by choosing $p \in \{2, 3, 4\}$, and setting L using the following scheme for $\alpha = \{0.1, 0.15, 0.2, 0.25, 0.3\}$: Let $L(\alpha)$ be the sum of the distances of the $\lceil \alpha|F| \rceil$ nearest facilities to the central depot. The value of L is set as $L(\alpha)/p$. In total, this set contains $3 \cdot 3 \cdot 3 \cdot 5 = 135$ instances (three underlying graphs for each $(|F|, |C|)$ and the different parameters for p and L).

Again, we first analyze the effect of our different settings for the algorithms. Figure 4 gives a plot of the runtime to optimality and Figure 5 gives plot of the optimality gap for the instances from set NEW and our settings.

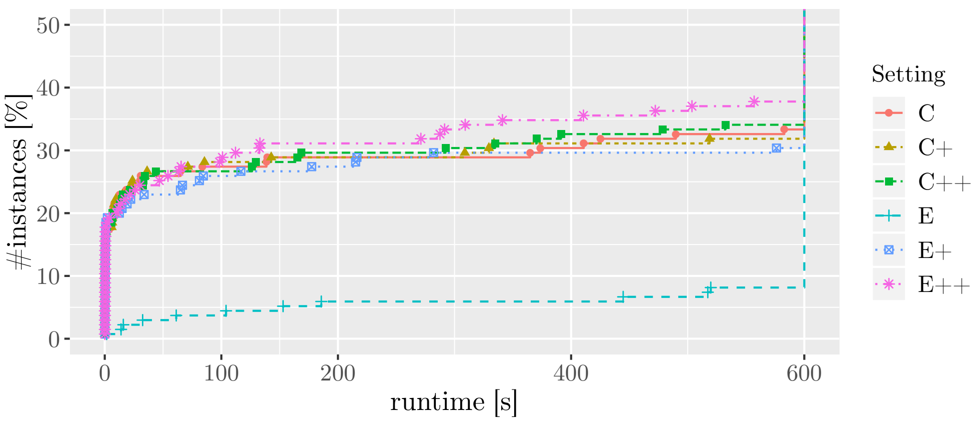


Figure 4: Runtime to optimality for instance set NEW and different settings. For better readability, the y-axis only goes up to 50% of the instances.

From the plots, we can see that these instances seem much more difficult to solve to optimality than the ones from set AMSAL. The best performing setting, E++ only manages to solve about 38% of the instances to optimality within the timelimit. In general, all settings except E, have a quite similar performance, however, when looking also the optimality gap, the three settings based on (C) seem a little better than E+ and E++.

Tables 11 to 13 give detailed results for the instances of set NEWand settings C+ and E++. In the tables, we report for each instance the number of vehicles (p), the distance limit (L) and the number of customers, for which at least one facility covering it is reachable from the central depot within the given distance limit ($|C_R|$ "reachable customers", i.e., following Theorem 2). For each setting, we report the runtime ($t[s]$), value of the best solution found (z^*), optimality gap ($g[\%]$), and number of branch-and-bound nodes ($\#nBBn$). A **bold** entry in z^* indicates that optimality has be proven (as can also be seen by the optimality gap $g[\%]$ being zero). An entry of "-" in z^* and $g[\%]$ indicates that no feasible solution could be found within the

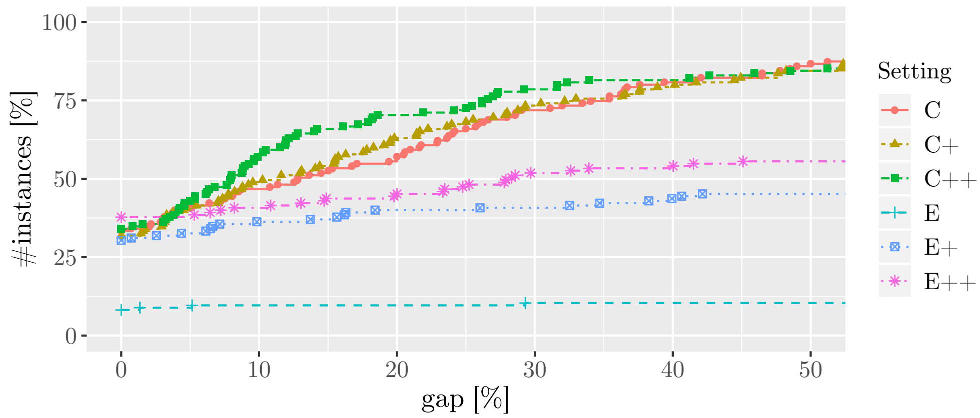


Figure 5: Optimality gap for instance set `NEW` and different settings. For better readability, the x-axis only goes up to 50% gap.

timelimit. A **bold** entry in $|C_R|$ indicates, that the optimal solution consists of all reachable customers.

In the tables, we can see that the instances with smaller distance limit seem to be easier, as more of them can be solved to optimality compared to instances with larger distance limit. This is not too surprising, as with smaller distance limit, the feasible region of the problem becomes smaller, and a smaller number of facilities is reachable within the distance limit. There are ten out of 135 instances, where the optimal solution value is equal to $|C_R|$. In general, the instances seem to become harder when the number of nodes is larger. Setting `E++` often does not manage to leave the root node within the timelimit and also struggles to find feasible solutions. On the other hand, `C+` manages to enumerate a few thousand branch-and-bound nodes quite often. However, the number of branch-and-bound nodes `C+` manages to enumerate seems to depend on the distance limit L, as can, e.g., be seen in Table 13, for smaller limits considerably more nodes can be enumerated, and for intermediate limits the fewest.

Table 11: Detailed results for instance set `NEW` for instances with 301 vertices

			C+				E++			
p	L	$\lvert C_R \rvert$	$t[s]$	z^*	$g[\%]$	$\#nBBn$	$t[s]$	z^*	$g[\%]$	$\#nBBn$
2	422.52	45	0	**36**	0.00	48	0	**36**	0.00	18
2	465.91	58	0	**49**	0.00	3	1	**49**	0.00	28
2	538.19	56	0	**46**	0.00	102	0	**46**	0.00	0
2	824.88	145	71	**78**	0.00	6858	30	**78**	0.00	485
2	900.85	206	600	99	3.72	24228	133	**99**	0.00	1593
2	1048.14	224	600	135	15.52	7370	557	**135**	0.00	2309
2	1188.67	225	600	118	37.58	6497	600	122	40.03	1217
2	1277.91	225	600	120	52.23	4373	600	145	32.60	1307
2	1461.77	225	600	162	29.25	2826	600	165	27.89	970
2	1712.49	225	600	152	41.71	2122	600	154	45.10	860
2	1801.24	225	600	162	36.56	1322	600	187	19.79	302
2	2047.24	225	600	182	23.63	627	600	-	-	11
2	2336.58	225	600	162	38.89	659	600	203	10.84	340
2	2374.58	**225**	600	176	27.84	509	65	**225**	0.00	17
2	2664.08	**225**	600	218	3.21	949	63	**225**	0.00	37
3	258.17	39	0	**34**	0.00	0	0	**34**	0.00	0
3	285.40	**27**	0	**27**	0.00	0	0	**27**	0.00	0
3	349.69	**45**	0	**45**	0.00	0	0	**45**	0.00	0
3	448.37	63	0	**58**	0.00	40	0	**58**	0.00	0
3	552.02	78	1	**61**	0.00	741	0	**61**	0.00	0
3	598.50	121	6	**93**	0.00	3059	17	**93**	0.00	445
3	641.46	136	36	**100**	0.00	5447	14	**100**	0.00	277
3	789.53	179	24	**104**	0.00	1982	133	**104**	0.00	1286
3	816.54	192	329	**138**	0.00	32221	98	**138**	0.00	882
3	959.52	213	600	152	3.23	15092	600	136	28.32	1725
3	1145.94	224	600	175	7.05	4992	600	161	27.78	394
3	1151.62	225	600	189	9.55	8254	600	176	23.46	464
3	1319.82	225	600	198	8.80	7270	600	182	23.33	287
3	1522.41	225	600	189	19.05	937	600	-	-	10
3	1574.57	225	600	206	9.22	608	600	175	28.57	113
4	211.30	**23**	0	**23**	0.00	0	0	**23**	0.00	0
4	219.33	**20**	0	**20**	0.00	0	0	**20**	0.00	0
4	241.75	**17**	0	**17**	0.00	0	0	**17**	0.00	0
4	401.70	54	0	**50**	0.00	147	0	**50**	0.00	0
4	415.60	59	0	**58**	0.00	21	0	**58**	0.00	4
4	449.44	44	0	**43**	0.00	0	0	**43**	0.00	0
4	556.82	90	8	**78**	0.00	2049	21	**78**	0.00	503
4	604.39	121	15	**96**	0.00	5630	4	**96**	0.00	69
4	644.47	123	6	**94**	0.00	1018	0	**94**	0.00	0
4	777.57	163	309	**130**	0.00	30637	600	128	8.20	1099
4	884.61	185	600	154	1.59	40817	600	150	6.45	896
4	920.99	199	519	**157**	0.00	24632	600	137	29.70	703
4	1041.71	222	600	169	8.71	10863	600	141	41.51	79
4	1176.85	225	600	195	7.03	10584	600	-	-	1
4	1219.09	224	600	189	13.08	9961	600	165	33.92	191

Table 12: Detailed results for instance set NEW for instances with 401 vertices

			C+				E++			
p	L	$\|C_R\|$	$t[s]$	z^*	$g[\%]$	$\#nBBn$	$t[s]$	z^*	$g[\%]$	$\#nBBn$
2	486.16	91	8	**53**	0.00	1446	12	**53**	0.00	159
2	620.83	128	34	**80**	0.00	5085	26	**80**	0.00	251
2	784.34	208	600	88	8.22	16643	271	**89**	0.00	2140
2	927.20	262	600	128	15.48	6879	600	126	25.23	1315
2	1086.09	294	600	154	19.50	6005	600	130	71.18	704
2	1345.72	300	600	143	73.29	764	600	153	71.02	318
2	1435.36	300	600	150	82.91	536	600	53	414.24	139
2	1672.82	300	600	139	115.83	348	600	21	1314.66	195
2	1973.23	300	600	197	52.28	236	600	177	69.49	209
2	2019.76	300	600	153	96.08	114	600	250	20.00	139
2	2309.20	300	600	188	59.57	76	600	-	-	9
2	2638.19	300	600	251	19.52	118	600	-	-	18
2	2690.88	300	600	242	23.97	129	600	-	-	3
2	3004.45	300	600	218	37.61	276	600	-	-	49
2	3331.40	**300**	600	287	4.53	338	504	**300**	0.00	69
3	357.90	44	0	**41**	0.00	0	0	**41**	0.00	0
3	377.34	53	0	**46**	0.00	0	0	**46**	0.00	0
3	400.40	75	0	**65**	0.00	67	0	**65**	0.00	0
3	699.51	186	86	**119**	0.00	9326	291	**119**	0.00	2263
3	711.49	189	600	124	1.73	31478	600	123	7.22	1787
3	724.46	178	143	**117**	0.00	13604	411	**117**	0.00	2243
3	1066.84	300	600	153	44.45	3602	600	89	181.09	50
3	1120.62	293	600	188	21.66	3556	600	81	211.68	69
3	1155.64	297	600	196	17.81	3200	600	36	621.13	55
3	1459.29	300	600	227	32.16	315	600	-	-	1
3	1592.28	300	600	178	68.54	64	600	-	-	1
3	1624.41	300	600	191	57.07	288	600	-	-	1
3	1923.85	300	600	172	74.42	393	600	-	-	1
3	2100.66	300	600	258	16.28	186	600	-	-	1
3	2131.08	300	600	239	25.52	170	600	-	-	1
4	260.83	**44**	0	**44**	0.00	0	0	**44**	0.00	0
4	308.07	**43**	0	**43**	0.00	0	0	**43**	0.00	0
4	331.83	**44**	0	**44**	0.00	0	0	**44**	0.00	0
4	489.73	102	9	**82**	0.00	2431	10	**82**	0.00	181
4	571.59	88	8	**82**	0.00	1031	1	**82**	0.00	0
4	593.91	128	6	**103**	0.00	833	47	**103**	0.00	528
4	772.92	209	600	159	3.14	20309	600	153	14.87	586
4	876.54	258	600	163	14.40	17869	600	50	309.02	176
4	909.80	266	600	199	7.77	25814	600	179	24.77	478
4	1097.81	289	600	226	10.25	4385	600	-	-	1
4	1235.17	300	600	231	21.43	1426	600	-	-	0
4	1265.89	300	600	248	15.41	1524	600	-	-	1
4	1456.87	300	600	230	30.43	331	600	-	-	1
4	1619.40	300	600	207	44.93	178	600	-	-	1
4	1673.74	300	600	263	14.07	252	600	-	-	1

Table 13: Detailed results for instance set NEW for instances with 501 vertices

			C+				E++			
p	L	$\|C_R\|$	$t[s]$	z^*	$g[\%]$	$\#nBBn$	$t[s]$	z^*	$g[\%]$	$\#nBBn$
2	670.15	175	600	91	3.29	31892	112	**91**	0.00	806
2	741.09	247	600	100	7.35	13029	288	**100**	0.00	1670
2	768.77	229	600	106	5.46	10831	600	106	13.03	1735
2	1246.33	375	600	134	119.14	286	600	16	1811.98	45
2	1321.79	375	600	156	89.46	540	600	19	1585.15	46
2	1336.87	375	600	134	131.42	360	600	-	-	33
2	1952.41	375	600	212	76.89	2	600	44	728.34	36
2	1992.25	375	600	-	-	4	600	-	-	48
2	2104.88	375	600	254	47.64	0	600	-	-	4
2	2875.80	375	600	253	48.22	24	600	-	-	1
2	2885.57	375	600	167	124.55	3	600	-	-	1
2	3072.01	375	600	188	99.47	40	600	-	-	1
2	3751.65	**375**	600	348	7.76	64	341	**375**	0.00	7
2	3776.04	375	600	-	-	62	600	-	-	1
2	3966.92	375	600	336	11.61	139	600	-	-	5
3	556.28	127	20	**95**	0.00	2698	54	**95**	0.00	330
3	595.57	133	23	**90**	0.00	2093	101	**90**	0.00	512
3	673.59	171	334	**118**	0.00	54340	309	**118**	0.00	2888
3	940.28	323	600	181	21.41	4727	600	21	1085.98	100
3	956.60	339	600	164	26.99	6228	600	25	858.37	53
3	1175.17	374	600	196	52.36	593	600	28	1094.20	14
3	1355.44	375	600	214	65.99	376	600	21	1625.87	14
3	1409.29	375	600	204	76.34	203	600	-	-	11
3	1726.52	375	600	282	32.98	8	600	-	-	0
3	1959.48	375	600	254	47.64	1	600	-	-	1
3	2020.80	375	600	275	36.36	0	600	-	-	1
3	2423.42	375	600	324	15.74	45	600	-	-	1
3	2532.45	375	600	313	19.81	59	600	-	-	1
3	2598.26	375	600	317	18.30	29	600	-	-	9
3	3085.30	375	600	358	4.75	90	600	-	-	7
4	378.59	76	0	**70**	0.00	0	1	**70**	0.00	0
4	393.43	77	0	**67**	0.00	95	0	**67**	0.00	0
4	399.85	74	0	**69**	0.00	294	2	**69**	0.00	10
4	671.32	192	600	137	1.43	56926	472	**137**	0.00	2150
4	673.95	193	600	129	2.92	38606	600	126	14.71	1030
4	704.28	174	600	132	3.03	40595	600	132	5.30	1661
4	1012.70	355	600	220	22.25	4180	600	-	-	0
4	1042.77	355	600	218	40.15	3734	600	-	-	0
4	1080.47	365	600	224	28.76	2424	600	-	-	0
4	1487.78	375	600	319	17.55	10	600	-	-	0
4	1562.10	375	600	290	29.31	40	600	-	-	0
4	1588.27	375	600	293	27.99	0	600	-	-	0
4	1939.26	375	600	336	11.61	100	600	-	-	1
4	2058.55	375	600	300	25.00	58	600	-	-	0
4	2098.21	375	600	332	12.95	50	600	-	-	0

5 Conclusion

In this paper, we studied the recently introduced *time-constrained maximal covering routing problem*. The problem is a generalization of well-known problems such as the *(team) orienteering problem* and *maximal covering location*. In the problem, we are given a central depot, facilities and customers. Each customer can be served by a subset of the facilities. Moreover, we are given distances between the facilities (and central depot), a distance limit L and a number of vehicles p. A feasible solution consists of p Hamiltonian cycles on subsets of the facilities and the central depot. All cycles must contain the central depot and respect the distance limit L. The goal is to maximize the number of customers covered by the facilities in the solution. The problem was introduced in [1], where an exact mixed-integer programming (MIP) approach and several metaheuristics were proposed.

We introduced two new MIP formulations and presented exact solution frameworks based on these MIP formulations. We evaluated our solution approaches on the instances from literature for the problem (from [1]). The computational study revealed, that our algorithms were able to find the provably optimal solution for 264 out of 270 instances, including 120 instances, for which the optimal solution was not known before. Moreover, for most of the instances, our algorithms only took a few seconds, being up to five magnitude faster than the exact MIP approach presented in [1]. The computational study also showed that the instances from [1] have some issues, which potentially decrease their usefulness as benchmark instances. In particular, there are often many customers, which are not associated with any facility in the instance, and thus can never be in any feasible solution. In many instances, the value of the optimal solution is then simply similar to the number of the remaining customers. Moreover, there are also some wrong entries in the result-tables of [1]. We thus also introduced a new set of more challenging instances.

There are several avenues for further work: Naturally, similar to other routing problems, additional side-constraints based on real-life considerations can be added, such as multiple-depots, time-windows, capacities, or uncertainty. Moreover, weighted customers and assignment-costs for customers could also be interesting extensions. To deal with more difficult instances, such as the ones introduced in this paper, the design and (re-)evaluation of metaheuristics could be an promising topic. We note that [1] already proposed and tested some metaheuristics for the problem, however, as discussed, the instances used by them to evaluate their approaches had some issues. Investigating an exact approach based on a formulation with exponentially many variables (i.e., column generation/branch-and-price) could also be a fruitful topic, as such approaches often work quite well for routing problems of similar type.

References

[1] Amiri, A., Salari, M.: Time-constrained maximal covering routing problem. OR Spectrum **41**(2), 415–468 (2019)

[2] Bianchessi, N., Mansini, R., Speranza, M.G.: A branch-and-cut algorithm for the team orienteering problem. International Transactions in Operational Research **25**(2), 627–635 (2018)

[3] Boussier, S., Feillet, D., Gendreau, M.: An exact algorithm for team orienteering problems. 4OR **5**(3), 211–230 (2007)

[4] Butt, S.E., Ryan, D.M.: An optimal solution procedure for the multiple tour maximum collection problem using column generation. Computers & Operations Research **26**(4), 427–441 (1999)

[5] Chao, I., Golden, B.L., Wasil, E.A.: The team orienteering problem. European Journal of Operational Research **88**(3), 464–474 (1996)

[6] Cherkassky, B.V., Goldberg, A.V.: On implementing push-relabel method for the maximum flow problem. In: International Conference on Integer Programming and Combinatorial Optimization, pp. 157–171. Springer (1995)

[7] Church, R., Velle, C.R.: The maximal covering location problem. Papers in Regional Science **32**(1), 101–118 (1974)

[8] Dang, D.C., El-Hajj, R., Moukrim, A.: A branch-and-cut algorithm for solving the team orienteering problem. In: International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, pp. 332–339. Springer (2013)

[9] El-Hajj, R., Dang, D.C., Moukrim, A.: Solving the team orienteering problem with cutting planes. Computers & Operations Research **74**, 21–30 (2016)

[10] Fischetti, M., Gonzalez, J.J.S., Toth, P.: Solving the orienteering problem through branch-and-cut. INFORMS Journal on Computing **10**(2), 133–148 (1998)

[11] Gendreau, M., Laporte, G., Semet, F.: A branch-and-cut algorithm for the undirected selective traveling salesman problem. Networks: An International Journal **32**(4), 263–273 (1998)

[12] Golden, B.L., Levy, L., Vohra, R.: The orienteering problem. Naval Research Logistics **34**(3), 307–318 (1987)

[13] Gunawan, A., Lau, H.C., Vansteenwegen, P.: Orienteering problem: A survey of recent variants, solution approaches and applications. European Journal of Operational Research **255**(2), 315–332 (2016)

[14] Keshtkaran, M., Ziarati, K., Bettinelli, A., Vigo, D.: Enhanced exact solution methods for the team orienteering problem. International Journal of Production Research **54**(2), 591–601 (2016)

[15] Koch, T., Martin, A.: Solving steiner tree problems in graphs to optimality. Networks: An International Journal **32**(3), 207–232 (1998)

[16] Laporte, G., Martello, S.: The selective travelling salesman problem. Discrete Applied Mathematics **26**(2-3), 193–207 (1990)

[17] Laporte, G., Nickel, S., da Gama, F.S.: Location science, vol. 528. Springer (2015)

[18] Leifer, A.C., Rosenwein, M.B.: Strong linear programming relaxations for the orienteering problem. European Journal of Operational Research **73**(3), 517–523 (1994)

[19] Maffioli, F., Sciomachen, A.: A mixed-integer model for solving ordering problems with side constraints. Annals of Operations Research **69**, 277–297 (1997)

[20] Naji-Azimi, Z., Salari, M.: The time constrained maximal covering salesman problem. Applied Mathematical Modelling **38**(15-16), 3945–3957 (2014)

[21] Ozbaygin, G., Yaman, H., Karasan, O.E.: Time constrained maximal covering salesman problem with weighted demands and partial coverage. Computers & Operations Research **76**, 226–237 (2016)

[22] Poggi, M., Viana, H., Uchoa, E.: The team orienteering problem: Formulations and branch-cut and price. In: 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'10). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2010)

[23] Reinelt, G.: TSPLIB - a traveling salesman problem library. ORSA Journal on Computing **3**(4), 376–384 (1991)

[24] Toth, P., Vigo, D.: Vehicle routing: problems, methods, and applications. SIAM (2014)

[25] Tsiligirides, T.: Heuristic methods applied to orienteering. Journal of the Operational Research Society **35**(9), 797–809 (1984)