Department of Mathematics

In search of good relaxations for the urban snow removal problem

Roghayeh Hajizadeh, Kaj Holmberg

LiTH-MAT-R-2023/03

LINKÖPING UNIVERSITY

https://doi.org/10.3384/LiTH-MAT-R-2023-03
LiTH-MAT-R-2023/03
ISSN 0348-2960

This is a technical report and has not been externally reviewed.

Linköping University Electronic Press

Linköping University
Department of Mathematics
SE-581 83 Linköping

In search of good relaxations for the urban snow removal problem

Roghayeh Hajizadeh
Kaj Holmberg

Department of Mathematics
Linköping Institute of Technology
SE-581 83 Linköping, Sweden

May 29, 2023

Abstract

Snow removal is, in Sweden, an infrequently occurring challenge. Doing the snow removal more efficiently could give much benefits for society. Since the amounts of snow vary a lot from day to day, and from year to year, fixed plans are not the best. Optimization of the snow removal tours could save much money. In this paper, we study the multi-vehicle urban snow removal problem from a mixed integer programming perspective. It is a very hard problem, and obtaining the exact optimum seems to be out of reach. Therefore, we study relaxations of the problem. Our goal is simply to find the best bounds for the optimal objective function value that is possible in limited time. We present some promising possibilities, verified by extensive computational tests.

1 Introduction

Removing snow from city streets is a challenging task in Nordic countries like Sweden. It is rather infrequent, for two reasons. It is only needed in winter, and the amounts of snow vary much from year to year. Therefore, one cannot follow a daily routine and premade tours when doing it. Much can be gained by optimizing the tours for snow removal vehicles.

We have in several papers addressed this problem (see references below). More specifically it concerns the optimization of vehicle tours in a contractor area, which usually is not huge. Nevertheless, a general conclusion is that it is a very hard optimization problem if all relevant details are taken into account. We have so far in general failed in finding an optimal solution for this problem. This is true even if we make some restrictions, such as only considering one type of vehicle, and only considering streets where such vehicles can be used. Usually this removes walking and bicycle paths from consideration. This leaves us with the core of the problem. A number of identical vehicles shall clear an urban area of snow. This requires a partitioning of the area between vehicles, and tours for each vehicle. Considering the nature of city street networks, the partitioning concerns which streets each vehicle shall treat, i.e. remove snow from, but it does not mean that different vehicles never use the same streets for transportation.

In order to find realizable plans, we must include time in the plans. It is important to know when a certain task is performed, since some other tasks cannot be made before that. Our optimization problem is based on unsplittable tasks, such as clearing the right side of a street, or clearing a crossing, and the main decision is when such a task is started and by which vehicle.

One should note that a street is not cleared by only one task. Several tasks are needed, and there are precedence restrictions between some of them. Some tasks are undirected, while others are directed. A specific detail used by local contractors is to do an undirected middle sweep first on ordinary streets, in order to quickly make the streets usable. Another detail is the time that is needed between two tasks in direct sequence, which obviously depends on how the tasks are located in relation to each other. More details are given below, and in [4] where mixed integer programming formulations are considered. How to obtain indata (from OpenStreetMap) is described in [6] and [5].

We have in previous work constructed heuristics for finding feasible solutions for the problem. In [8] and [1], the problem facing one vehicle has been studied. Combining and coordinating the single vehicle tours into complete feasible solutions is discussed in [2]. Elimination of tree parts in the city networks is described in [3].

In our earlier procedures for finding feasible solutions for the detailed snow removal problem, we have no guarantees about the quality of the solutions. For this problem, even "feasible" is not something easily obtained, if all the detailed constraints are taken into account. (This will be evident later in this paper when we describe the model.)

The procedures for finding feasible solutions are rather practical, and often do what one would like to do if approaching this problem "by hand". One main difference is that a computer can check and compare large number of alternatives quickly.

In this paper, our goal is different. As will be shown later in this paper, exact solution of the problem is not possible in reasonable time. So instead we look at different relaxations and approximations of the problem. The goal is obviously to get as close as possible to the exact optimum, but in reasonable time. One thing that must be sacrificed to obtain this is the exact feasibility. Therefore, we look at which constraints that can be ignored in order to obtain as small changes in the solution as possible. We mainly use the objective function value as measure of how close we get to the optimum.

In many cases we don't know the optimal objective function value. Then, we need bounds on it. Upper and lower bounds can possibly give us some information of how close we are. If we have an upper bound on the optimal objective function value, we want it to be as small as possible. If we have a lower bound on the optimal objective function value, we want it to be as large as possible.

This paper uses the mixed integer programming (MIP) formulation as a base for the methods. In previous papers, we didn't even formulate the exact MIP model, but instead worked with the detailed description of the problem. The difference is for example that one might require that a vehicle does a tour, passing a number of streets, before returning to the starting node. In a MIP model, we have to ensure this with the help of integer variables. The reader is probably familiar with all the cycle-breaking constraints one might need in a MIP formulation of a traveling salesman problem. Sometimes the MIP formulations of a certain aspect is not helpful for the understanding of the problem, but rather a technical construction necessary for MIP solution methods.

In this paper, we may look for partial solutions, which might be helpful in practice. One possible scenario is that a certain part of a city network is allocated to a certain vehicle, but it is not prescribed exactly how the vehicle should drive. (This is similar to how it is done today in practice.) We might not call this proper solution, since it might not even be possible for the vehicle to clean that area within the given time. However, given the difficult nature of the problem, such information might still be useful. Even in such a case, it is of interest how long the vehicle may need to do the tasks. Therefore we need good estimations of the objective function value, especially in order to compare solutions, to be able to choose the most promising.

Another aspect, not really present in our previous work, is to have good knowledge of which parts of the model are relaxed. Exactly what parts of the model do we ignore?

Summarizing the goal of this paper, we wish to find good bounds on the optimal objective function value. Often we have a reasonably good solution, obtained by our previous methods, to start from. It is of course of interest to improve that solution, but all in all, the goal of this paper is not as straightforward as it often is. Here we wish to find ways of obtaining good information about the best solutions, not only find *the* best one.

Much of the results in this paper will be used in forthcoming work, in different ways. Either as estimations and comparisons of solutions, or as a base to start finding the best detailed solution from.

Our work in this paper is based on a MIP model, which we believe is exact, i.e. does not disregard anything that we believe is important. The model is time-indexed, and, as many such models, very hard to solve. Previously we have deemed it unsolvable in practice, and therefore not very useful. What has changed since then is that we have developed heuristics that find feasible solutions to the problem, i.e. yield upper bounds on the optimal objective function value. Unfortunately, these heuristics do not give lower bounds on the objective function value, so we have not been able to estimate how good these solutions, and upper bounds are.

However, what these feasible solutions give are upper bounds on the time needed to clear an area. That is very important for a time-indexed model. It enables us to use realistic bounds on the time when dimensioning the time-indexed arrays. It also gives new hope for the usefulness of the MIP model.

The goal is not mainly to find the exact optimal solution, or even the objective function value of it, because this is still out of reach in practice. The main goal is instead to find good lower bounds on the optimal objective function values, thereby enabling us to judge the quality of the previously found solutions. A secondary goal is, if possible, to improve the feasible solutions.

The primal heuristics we previously used are based on the same problem, but do not use the exact MIP model. A feasible solution from the heuristics is of course feasible in the MIP model, but the objective function is not used exactly. We have three parts in the objective function, and in the MIP model they are summed up with suitable weights. In the heuristics we use a more multiobjective inspired approach, where the first objective, minimizing the total time, is the one basically used, while the other two, minimizing the finishing time of all tasks (not only the last), and minimizing the distance traveled by the vehicles, are used in later improvement procedures. This means that the weights in the objective function of MIP model are not clearly defined in the heuristics. However, when a solution is at hand, we can of course calculate the exact objective function value, using the given weights.

In this paper, we first investigate how far we get with the exact model (which is not far) and then investigate many different relaxations. The goal is simply to get as good as possible lower bounds on the optimal objective function values in a reasonable time.

The instances are based on real life networks, but some details, such as exactly how long time that is available for the runs, are not known. Therefore we can not get definite answers to what is best. Instead we describe some possibilities that are promising (and some that are not).

The paper is organized as follows. In section 2, we present the mathematical model of the problem. In section 3 we describe the test instances and give information about their sizes. In the sections after this, computational results are given, and in section 5, some conclusions are presented.

2 Mathematical Model

In this section, the complete mathematical model for the snow removal problem will be presented as a MIP model. We will also give an aggregated model, which can be used when looking for good relaxations of the complete model. Finally, we will define a useful notation for relaxations.

2.1 Exact model

We here present a time-indexed mathematical model for urban snow removal problem facing one contractor in a city network where a number of identical vehicles must remove the snow of an area in the city. In order to formulate, we first discretize the time and number the time periods from 1 to T^{MAX}, where T^{MAX} is an upper bound of the time needed.

Consider a city network with nodes (points, crossing) N, links (streets) L, and undirected links $L^U \subseteq L$. We denote the set of tasks by T, the set of groups by G, and the set of vehicles by Q. In addition, g_j is the group that task j belongs to and G_g is the set of tasks that belong to group g. Furthermore, the depot for vehicle k is D_k.

Each task and link has a starting node and an ending node. In addition, it takes a certain time for a vehicle to do a task or transport itself along a link without removing snow. The time needed for a vehicle to do task j is denoted by d_j^T and to transport itself along link l is denoted by d_l^L. We show the starting node and the ending node of task j by s_j^T and e_j^T respectively. In addition, we denote the starting node and the ending node of link l by s_l^L and e_l^L respectively. Furthermore, we show the set of tasks starting at node i by S_i^T, the set of tasks ending at node i by E_i^T, the set of links starting at node i by S_i^L, and the set of links ending at node i by E_i^L.

Sometimes, there are different link tasks associated to one specific link, i.e. each street needs a number of different sweeps. We introduce T_l^{L2} as the set of link tasks that have the same starting and ending node as link l, and T_l^{L2r} as the set of link tasks that have the same starting and ending nodes as link l, but in reversed order. Usually T_l^{L2} and T_l^{L2r} are nonempty.

We define ρ_{gj}^{GT}, $\rho_{j'j}^{TT}$, and $\rho_{j'l}^{TL}$ for precedences which can be obtained from task types. $\rho_{gj}^{GT} = 1$ if group g must precede task j and $\rho_{j'j}^{TT} = 1$ if task j' must precede task j. Otherwise, they are zero. Similarly, $\rho_{j'l}^{TL} = 1$ if task j' must precede the usage of link l and zero otherwise.

Some parameters are required for switching times. The switching time between two tasks depends on their corresponding links and which directions are used. Note that switching times can be nonzero only if the corresponding links are adjacent or are the same link. The time needed for a vehicle to switch directly from link l to link l' is denoted by $d_{ll'}^{SLL}$. If a positive time needed for a vehicle to switch from forward use of link l to forward use of link l', we define $\delta_{ll'}^{SLFF} = 1$ and zero otherwise. Similarly, $\delta_{ll'}^{SLFB} = 1$, $\delta_{ll'}^{SLBF} = 1$, and $\delta_{ll'}^{SLBB} = 1$ if a positive time needed for a vehicle to switch respectively from forward use of link l to backward use of link l', from backward use of link l to forward use of link l', and from backward use of link l to backward use of link l'. Otherwise, they have zero values.

We define the main variables as the binary variables x^T, x^{FL}, x^{BL}, and I. $x_{jkt}^T = 1$ if vehicle k starts task j at time t, for any j, t and k, and 0 otherwise. In addition, $x_{lkt}^{FL} = 1$ if transportation of vehicle k starts forwards on link l at time t, for any $l \in L$, k and t, and 0 otherwise. Similarly, $x_{lkt}^{BL} = 1$ if transportation of vehicle k starts backwards on link l at time t, for any $l \in L^U$, k and t, and 0 otherwise. Finally, $I_{ikt} = 1$ if vehicle k stays at node i from time t to the beginning of time $t + 1$ for any i, k, $t = 0, \ldots T^{MAX}$, and 0 otherwise.

In addition, we consider the following continuous variables for finishing times. The time when task j is finished is given by v_j^T and the time when transportation on link l is ready by v_l^L. Similarly, v_k^R gives the time when vehicle k is ready, and v^{TOT} is the time when everything is finished. The detailed time-indexed mixed MIP model is given as below.

$$\begin{aligned}\min \quad Obj \quad &= \alpha_1 v^{TOT} + \alpha_2 (\sum_k \sum_t \sum_j (t + d_j^T) x_{jkt}^T) \\ &+ \alpha_3 (\sum_k \sum_t \sum_l d_l^L x_{lkt}^{FL} + \sum_k \sum_t \sum_{l \in L^U} d_l^L x_{lkt}^{BL})\end{aligned}$$

s.t.

$$\sum_{j \in G_g} \sum_k \sum_t x_{jkt}^T = 1 \qquad \forall g \qquad (1)$$

$$I_{ik,t-1} + \sum_{j \in E_i^T} x_{jk,t-d_j^T}^T + \sum_{l \in E_i^L} x_{lk,t-d_l^L}^{FL} + \sum_{l \in S_i^L \cap L^U} x_{lk,t-d_l^L}^{BL} =$$
$$I_{ikt} + \sum_{j \in S_i^T} x_{jkt}^T + \sum_{l \in S_i^L} x_{lkt}^{FL} + \sum_{l \in E_i^L \cap L^U} x_{lkt}^{BL} \qquad \forall i, k, t \qquad (2)$$

$$I_{D_k,k,0} = 1 \qquad \forall k \qquad (3)$$

$$I_{D_k,k,T^{MAX}} = 1 \qquad \forall k \qquad (4)$$

$$I_{i,k,0} = 0 \qquad \forall k, i \neq D_k \qquad (5)$$

$$I_{i,k,T^{MAX}} = 0 \qquad \forall k, i \neq D_k \qquad (6)$$

$$\sum_j x_{jkt}^T + \sum_l x_{lkt}^{FL} + \sum_{l \in L^U} x_{lkt}^{BL} \leq 1 \qquad \forall k, t \qquad (7)$$

$$\sum_k \sum_{t'=1}^{t + \min_{j' \in G_g} d_{j'}^T} \rho_{gj}^{GT} x_{jkt'}^T \leq \sum_k \sum_{j' \in G_g} \sum_{t'=1}^{t} x_{j'kt'}^T \qquad \forall t, g, j \qquad (8)$$

$$\sum_k \sum_{t'=1}^{t+d_{j'}^T} \rho_{j'j}^{TT} x_{jkt'}^T + \sum_k \sum_{t'=t+1}^{T^{MAX}} x_{j'kt'}^T \leq 1 \qquad \forall t, j, j' \qquad (9)$$

$$\sum_k \sum_{t'=1}^{t+d_{j'}^T} \rho_{j'l}^{TL} (x_{lkt'}^{FL} + x_{lkt'}^{BL}) + \sum_k \sum_{t'=t+1}^{T^{MAX}} x_{j'kt'}^T \leq 1 \qquad \forall t, j', l \in L^U \qquad (10)$$

$$\sum_k \sum_{t'=1}^{t+d_{j'}^T} \rho_{j'l}^{TL} x_{lkt'}^{FL} + \sum_k \sum_{t'=t+1}^{T^{MAX}} x_{j'kt'}^T \leq 1 \qquad \forall t, j', l \in L \setminus L^U \qquad (11)$$

$$\sum_{t'=t}^{t+d_j^T+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLFF} x_{j'kt'}^T + x_{jkt}^T \leq 1 \qquad \forall j \in T_l^{L2}, l, k, t, j' \in T_{l'}^{L2}, l' \qquad (12)$$

$$\sum_{t'=t}^{t+d_j^T+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLBF} x_{j'kt'}^T + x_{jkt}^T \leq 1 \qquad \forall j \in T_l^{L2r}, l \in L^U, k, t, j' \in T_{l'}^{L2}, l' \qquad (13)$$

$$\sum_{t'=t}^{t+d_j^T+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLFB} x_{j'kt'}^T + x_{jkt}^T \leq 1 \qquad \forall j \in T_l^{L2}, l, k, t, j' \in T_{l'}^{L2r}, l' \in L^U \qquad (14)$$

$$\sum_{t'=t}^{t+d_j^T+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLBB} x_{j'kt'}^T + x_{jkt}^T \le 1 \qquad \forall j \in T_l^{L2r}, l \in L^U, k, t, j' \in T_{l'}^{L2r}, l' \in L^U \tag{15}$$

$$\sum_{t'=t}^{t+d_j^T+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLFF} x_{l'kt'}^{FL} + x_{jkt}^T \le 1 \qquad \forall j \in T_l^{L2}, l, k, t, l' \tag{16}$$

$$\sum_{t'=t}^{t+d_j^T+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLBF} x_{l'kt'}^{FL} + x_{jkt}^T \le 1 \qquad \forall j \in T_l^{L2r}, l \in L^U, k, t, l' \tag{17}$$

$$\sum_{t'=t}^{t+d_j^T+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLFB} x_{l'kt'}^{BL} + x_{jkt}^T \le 1 \qquad \forall j \in T_l^{L2}, l, k, t, l' \in L^U \tag{18}$$

$$\sum_{t'=t}^{t+d_j^T+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLBB} x_{l'kt'}^{BL} + x_{jkt}^T \le 1 \qquad \forall j \in T_l^{L2r}, l \in L^U, k, t, l' \in L^U \tag{19}$$

$$\sum_{t'=t}^{t+d_l^L+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLFF} x_{j'kt'}^T + x_{lkt}^{FL} \le 1 \qquad \forall l, k, t, j' \in T_{l'}^{L2}, l' \tag{20}$$

$$\sum_{t'=t}^{t+d_l^L+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLBF} x_{j'kt'}^T + x_{lkt}^{BL} \le 1 \qquad \forall l \in L^U, k, t, j' \in T_{l'}^{L2}, l' \tag{21}$$

$$\sum_{t'=t}^{t+d_l^L+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLFB} x_{j'kt'}^T + x_{lkt}^{FL} \le 1 \qquad \forall l, k, t, j' \in T_{l'}^{L2r}, l' \in L^U \tag{22}$$

$$\sum_{t'=t}^{t+d_l^L+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLBB} x_{j'kt'}^T + x_{lkt}^{BL} \le 1 \qquad \forall l \in L^U, k, t, j' \in T_{l'}^{L2r}, l' \in L^U \tag{23}$$

$$\sum_{t'=t}^{t+d_l^L+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLFF} x_{l'kt'}^{FL} + x_{lkt}^{FL} \le 1 \qquad \forall l, k, t, l' \tag{24}$$

$$\sum_{t'=t}^{t+d_l^L+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLBF} x_{l'kt'}^{FL} + x_{lkt}^{BL} \le 1 \qquad \forall l \in L^U, k, t, l' \tag{25}$$

$$\sum_{t'=t}^{t+d_l^L+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLFB} x_{l'kt'}^{BL} + x_{lkt}^{FL} \le 1 \qquad \forall l, k, t, l' \in L^U \tag{26}$$

$$\sum_{t'=t}^{t+d_l^L+d_{ll'}^{SLL}-1} \delta_{ll'}^{SLBB} x_{l'kt'}^{BL} + x_{lkt}^{BL} \le 1 \qquad \forall l \in L^U, k, t, l' \in L^U \tag{27}$$

$$v_j^T \ge \sum_k \sum_t (t + d_j^T) x_{jkt}^T \qquad \forall j \tag{28}$$

$$v_k^R \geq t(\sum_{j\in E^T_{D_k}} x^T_{jk,t-d^T_j} + \sum_{l\in E^L_{D_k}} x^{FL}_{lk,t-d^L_l} + \sum_{l\in S^L_{D_k}\cap L^U} x^{BL}_{lk,t-d^L_l}) \quad \forall k,t \tag{29}$$

$$v_l^L \geq (t+d_l^L)(x^{FL}_{lkt} + x^{BL}_{lkt}) \quad \forall l \in L^U, k, t \tag{30}$$

$$v_l^L \geq (t+d_l^L)x^{FL}_{lkt} \quad \forall l \in L \setminus L^U, k, t \tag{31}$$

$$v^{TOT} \geq v_j^T \quad \forall j \tag{32}$$

$$v^{TOT} \geq v_k^R \quad \forall k \tag{33}$$

$$v^{TOT} \geq v_l^L \quad \forall l \tag{34}$$

$$v_j^T \leq T^{MAX} \quad \forall j \tag{35}$$

$$v_k^R \leq T^{MAX} \quad \forall k \tag{36}$$

$$v^{TOT} \leq T^{MAX} \tag{37}$$

$$x^T_{jkt} \in \{0,1\} \quad \forall j,k,t \tag{38}$$

$$x^{FL}_{lkt} \in \{0,1\} \quad \forall l,k,t \tag{39}$$

$$x^{BL}_{lkt} \in \{0,1\} \quad \forall l \in L^U, k, t \tag{40}$$

$$I_{ikt} \in \{0,1\} \quad \forall i,k,t \tag{41}$$

$$v_j^T \geq 0 \quad \forall j \tag{42}$$

$$v_k^R \geq 0 \quad \forall k \tag{43}$$

$$v_l^L \geq 0 \quad \forall l \tag{44}$$

$$v^{TOT} \geq 0 \tag{45}$$

The objective function is a weighted combination of final end time, the sum of end times for all tasks and the time vehicles are moving without clearing snow, "deadheading". Constraint (1) is the demand constraint which means that one task in each group must be done at one time by one vehicle. Constraint (2) states the node equilibrium for each vehicle at each node and each time. The arrivals at node i of vehicle k at time t plus those waiting from the previous time step should be equal to the departures from node i of vehicle k at time t plus those waiting until the next time step. Constraints (3)-(6) are depot constraints which ensure that the vehicles start and end at their depots. Constraint (7) states that each vehicle may start at most one thing at any time. Precedence constraints are (8)-(11). Constraint (8) states the precedence of group g to task j, constraint (9) gives the precedence of task j' to task j and constraints (10)-(11) indicate the precedence of task j' to transporting on link l. Note that if $\rho^{GT}_{gj} = 0$, $\rho^{TT}_{j'j} = 0$ or $\rho^{TL}_{j'l} = 0$, the corresponding constraint will be redundant. Constraints (12)-(27) are switching constraints. Constraints (12)-(15) state switching from tasks to tasks, (16)-(19) show switching from tasks to links, (20)-(23) indicate switching from links to tasks, and (24)-(27) are for switching from links to links. Each group of switching constraints contains four different sets of constraints, i.e., switching from forward use of link l to forward use of link l', switching from backward to forward, forward to backward, and backward to backward. Finishing times are shown by constraints (28)-(37). Finally, the domain of variables are given in (38)-(45). We denote this model by M10.

Model M10 is huge. The computational results show that in practice the problem is not solvable. The main goal is to find good bounds on the optimal objective function value. In order to find lower bounds, we must consider simplifications of the model. One can ignore some of the constraints.

The problem has also been formulated in aggregated format in [4]. Similarly to the original model, different relaxation of the aggregated model can be considered. For $\alpha_1 = 1$ and $\alpha_2 = \alpha_3 = 0$, one of the relaxed aggregated models, where all precedences and switching constraints are ignored, gives good lower bounds. The model is called M14 and given in section 2.2.

2.2 Aggregated model

In order to formulate the problem in aggregated form, different kind of variables are needed. Here, we only give the variables needed for the relaxed model M14, see [4] for the whole model. Binary variables z^T, z^{FL}, and z^{BL} are introduced. For any j and k, $z^T_{jk} = 1$ if vehicle k does task j. In addition, $z^{FL}_{lk} = 1$ if vehicle k is transported forwards on link l, for any l and k. Similarly, $z^{BL}_{lk} = 1$ if vehicle k is transported backwards on link l, for any $l \in L^U$ and k. Furthermore, continuous time variables v^{VT} and v^{VL} are considered where v^{VT}_k is the total time vehicle k is used for doing tasks and v^{VL}_k is the total time vehicle k is used for transportation. Variables v^{TOT} and v^R_k are defined similar to the time-indexed original model. Then, the aggregated relaxed model, M14 is as blow.

$$\min \quad Obj = v^{TOT}$$
s.t.

$$\sum_{j \in G_g} \sum_{k} z^T_{jk} = 1 \qquad \forall g \qquad (46)$$

$$\sum_{j \in E^T_i} z^T_{jk} + \sum_{l \in E^L_i} z^{FL}_{lk} + \sum_{l \in S^L_i \cap L^U} z^{BL}_{lk} = \sum_{j \in S^T_i} z^T_{jk} + \sum_{l \in S^L_i} z^{FL}_{lk} + \sum_{l \in E^L_i \cap L^U} z^{BL}_{lk} \qquad \forall i, k \qquad (47)$$

$$\sum_{j \in S^T_{D_k}} z^T_{jk} + \sum_{l \in S^L_{D_k}} z^{FL}_{lk} + \sum_{l \in E^L_{D_k} \cap L^U} z^{BL}_{lk} \geq 1 \qquad \forall k \qquad (48)$$

$$\sum_{j \in E^T_{D_k}} z^T_{jk} + \sum_{l \in E^L_{D_k}} z^{FL}_{lk} + \sum_{l \in S^L_{D_k} \cap L^U} z^{BL}_{lk} \geq 1 \qquad \forall k \qquad (49)$$

$$v^{VT}_k = \sum_{j} d^T_j z^T_{jk} \qquad \forall k \qquad (50)$$

$$v^{VL}_k = \sum_{l} d^L_l z^{FL}_{lk} + \sum_{l \in L^U} d^L_l z^{BL}_{lk} \qquad \forall k \qquad (51)$$

$$v^R_k \geq v^{VT}_k + v^{VL}_k \qquad \forall k \qquad (52)$$

$$v^{TOT} \geq v^R_k \qquad \forall k \qquad (53)$$

$$v^R_k \leq T^{MAX} \qquad \forall k \qquad (54)$$

$$v^R_k \geq 0 \qquad \forall k \qquad (55)$$

$$v_k^{VT} \geq 0 \qquad \forall k \qquad (56)$$
$$v_k^{VL} \geq 0 \qquad \forall k \qquad (57)$$
$$v^{TOT} \geq 0 \qquad (58)$$
$$z_{jk}^{T} \in \{0,1\} \qquad \forall j,k \qquad (59)$$
$$z_{lk}^{FL} \in \{0,1\} \qquad \forall l,k \qquad (60)$$
$$z_{lk}^{BL} \in \{0,1\} \qquad \forall l \in L^U, k \qquad (61)$$

Constraint (46) is the demand constraint and constraint (47) is the node equilibrium constraint. In addition, depot constraints are given in (48) and (49). Furthermore, finishing times are given in constraint (50)-(54). Finally, constraints (55)-(61) show the domain of the variables.

Note that the original variables and aggregated variables are closely related. Their relations are as below.

$$z_{jk}^{T} = \sum_{t} x_{jkt}^{T} \qquad \forall j,k \qquad (62)$$
$$z_{lk}^{FL} = \sum_{t} x_{lkt}^{FL} \qquad \forall l,k \qquad (63)$$
$$z_{lk}^{BL} = \sum_{t} x_{lkt}^{BL} \qquad \forall l \in L^U, k \qquad (64)$$

Numerical tests will show that the combination of model M10 and M14 is faster to solve than model M10. We denote the combined model with M1014 which is the model with all x, I, z, and v variables and constraints (1)-(64).

2.3 Notation

We have, as mentioned, defined the basic models M10 and M14, plus M1014, which simply is the two together with the constraint sets (62), (63) and (64) for connecting the variables. Relaxations of the models can be made by simply ignoring certain sets of constraints. We will in principle work with M1014 as basic subproblem, and discuss variants where certain constraint sets in x are present or not, i.e. remove certain parts of M10 or/and coupling constraints.

We will use a notation of a string of zeros and ones, where a zero indicates that a constraint set is not included in the model, and a one indicates that the constraint set is included. We use the following grouping and order:
1. Demand constraints, (1).
2. Node equilibrium constraints, (2) and (3) - (6).
3. "Start one thing at a time" constraints, (7).
4. Coupling constraints between x and z, (62) - (64).
5. Precedence constraints, (8) - (11).
6. Switching constraints, (12) - (27).

Name	m	n	p	g	T^{MAX}
mini-05a	5	5	15	15	40
mini-05b	5	5	19	17	40
mini-05c	5	5	25	20	60
mini-05d	5	5	25	20	1440
mini-06a	9	11	43	37	200
mini-06b	9	11	39	35	70
mini-06c	9	11	49	40	300
mini-08a	9	15	59	49	300
mini-08b	9	15	45	42	300
mini-08c	9	15	47	43	300
mini-09a	18	26	104	87	400
mini-09b	18	26	70	70	400
mini-09c	18	26	122	96	400

Table 1: Network sizes of the mini instances.

The basic model 000000 is M14 plus the x and I variables and with the objective function of M10. The model 111111 is the complete M1014. Both M10 and M1014 yield the correct optimal solution of the whole problem. Removing constraint sets from model 111111 yields various relaxations, and thus various lower bounds.

3 Test instances

We have made tests on two sets of instances, first on a set of small artificial instances, called mini, then on a set of real life small cities in Sweden obtained from OpenStreetMap. The details about the instances and the extraction procedure can be found in [6] and [7]. The instances are available at [9].

3.1 Test instance sizes

The sizes of the test instances can be described in several ways. First we have the sizes of the networks. In table 1, we give the sizes of the mini-networks, as number of nodes m, number of links n, number of tasks p, number of groups g and the dimension of time T^{MAX}.

We note that the a priori given value of T^{MAX} is a dimension of the time-indexed model. Instead of using that crude upper bound of T^{MAX}, one might use the value v^{TOT} of a known feasible solution as an upper bound for the number of time periods needed. From previous work, [2], we have feasible solutions, so in table 2, we give model sizes for mini-instances based on those smaller time dimensions.

We give the MIP model sizes, for different number of vehicles q, namely the number of x-variables (including x^T, x^{FL}, x^{BL} and I) denoted by $|x|$, the number of z-variables (including z^T, z^{FL} and z^{BL}) denoted by $|z|$, the number of v-variables denoted by $|v|$. Furthermore, we give the number of constraints, where c1x denotes the number of constraints for demand, node equilibrium etc., in the x-variables, c1z denotes the number of corresponding constraints in the z-variables, c2 denotes the number of constraints in the

v-variables, and c3 the number of constraints dealing with precedences and switching. These numbers have been calculated by approximate formulas, and might not be exact. However, it is the general sizes of the numbers that are interesting, and their relations.

We observe that $|x|$ is the bulk of variables in model M10 (and M1014), while M14 contains (only) $|z|$ and $|v|$ variables.

In the left parts of table 3, we give the sizes of the first city instances. In [3], we describe how to eliminate tree parts of the networks, and how to easily find solutions for the eliminated parts. We recall that these networks come from real life small cities, and the tree structures are actually there. It is not a question of making assumptions, but just observing how the networks actually are. Sizes of the remaining networks are given in right parts of the table 3. We add "-te" to the names, signifying that tree elimination has been used. Using v^{TOT} from our best feasible solutions as T^{MAX} yields the model sizes of city instances reported in tables 4 - 7.

In table 5, we find one instance, mantorp-te for 3 vehicles, where no feasible solution was present, which failed to improve T^{MAX}. The difference in size of the x-parts is evident. (The method we used to find feasible solution contains randomness, and yields different solutions every time.)

Picking one example at random, we study atvid (Åtvidaberg), which has 216 nodes and 280 links, 1396 tasks in 1086 groups. Using 5 vehicles and $T^{MAX} = 99999$ yields $|x|$=1 085 989 140 and c1x=108 001 086. We have a feasible solution with $v^{TOT} = 1120$, and using that as time dimension yields $|x|$=12 163 200 and c1x=1 211 766, i.e. a smaller problem.

Using tree elimination reduces the number of nodes to 159 and the number of links to 223, and yields 1103 tasks in 854 groups. For 5 vehicles, we get $|x|$=853 991 460 and c1x=79 500 854. Now we have solution with $v^{TOT} = 909$, and using that yields $|x|$=4 210 220 and c1x=396 049, i.e. an even smaller problem.

One might point out that we do expect to be able to solve this problem exactly with a MIP-code. However, if we plan to use a formulation with the x-variables, we should definitely use these size reducing steps.

4 Computational tests

In this section, we will present various tests that aim to illustrate how various relaxations behave, and may help us finding the best relaxations. There are instances of different size, and different difficulty, so we will not arrive at a single strategy that is always the best. It is better to have a choice of different strategies available for different instances. On the other hand, there are many possible strategies, and some of them will be inferior. Also this is good to know. We will investigate what constraints to keep, and also investigate the use of LP-relaxation.

Name	m	n	q	T^{MAX}	$\|x\|$	$\|z\|$	$\|v\|$	c1x	c1z	c2	c3
mini-05a	5	5	2	17	1020	50	27	229	27	129	15
mini-05a	5	5	3	16	1440	75	30	318	33	173	15
mini-05a	5	5	4	12	1440	100	33	323	39	203	15
mini-05b	5	5	2	24	1632	58	31	315	29	159	23
mini-05b	5	5	3	22	2244	87	34	428	35	211	23
mini-05b	5	5	4	17	2312	116	37	445	41	247	23
mini-05c	5	5	2	25	2000	70	37	330	32	185	35
mini-05c	5	5	3	22	2640	105	40	431	38	241	35
mini-05c	5	5	4	27	4320	140	43	688	44	323	35
mini-05d	5	5	2	29	2320	70	37	378	32	193	35
mini-05d	5	5	3	25	3000	105	40	485	38	250	35
mini-05d	5	5	4	20	3200	140	43	520	44	295	35
mini-06a	9	11	2	45	6660	130	61	955	57	327	57
mini-06a	9	11	3	37	8214	195	64	1174	67	418	57
mini-06a	9	11	4	28	8288	260	67	1193	77	489	57
mini-06b	9	11	2	45	6300	122	57	953	55	311	49
mini-06b	9	11	3	41	8610	183	60	1292	65	410	49
mini-06b	9	11	4	29	8120	244	63	1231	75	469	49
mini-06c	9	11	2	52	8320	142	67	1098	60	365	69
mini-06c	9	11	3	52	12480	213	70	1627	70	493	69
mini-06c	9	11	4	29	9280	284	73	1236	80	529	69
mini-08a	9	15	2	58	11368	178	81	1227	69	437	85
mini-08a	9	15	3	39	11466	267	84	1246	79	532	85
mini-08a	9	15	4	38	14896	356	87	1605	89	661	85
mini-08b	9	15	2	41	6888	150	67	880	62	347	57
mini-08b	9	15	3	35	8820	225	70	1119	72	450	57
mini-08b	9	15	4	30	10080	300	73	1278	82	545	57
mini-08c	9	15	2	43	7396	154	69	921	63	359	61
mini-08c	9	15	3	61	15738	231	72	1900	73	538	61
mini-08c	9	15	4	32	11008	308	75	1359	83	565	61
mini-09a	18	26	2	79	27492	312	137	3125	125	714	146
mini-09a	18	26	3	63	32886	468	140	3732	144	906	146
mini-09a	18	26	4	66	45936	624	143	5175	163	1142	146
mini-09b	18	26	2	54	15120	244	103	2158	108	528	78
mini-09b	18	26	3	45	18900	366	106	2689	127	682	78
mini-09b	18	26	4	39	21840	488	109	3106	146	830	78
mini-09c	18	26	2	160	61440	348	155	6212	134	948	182
mini-09c	18	26	3	119	68544	522	158	6933	153	1164	182
mini-09c	18	26	4	99	76032	696	161	7692	172	1382	182

Table 2: Model sizes of the mini instances with optimized T^{MAX}.

	Before tree elim				After tree elim			
Name	m	n	p	g	m	n	p	g
askeby	53	56	275	220	15	18	87	69
atvid-a-1	91	113	593	455	65	87	465	352
atvid-a-2	71	96	455	359	61	86	405	319
atvid-a-3	34	49	230	181	33	48	225	177
atvid-a	195	259	1281	997	159	223	1103	854
atvid	216	280	1396	1086	159	223	1103	854
bankekind	30	35	168	134	16	21	100	79
borensberg-a	137	171	803	641	94	128	608	479
borensberg-b	103	131	605	485	72	100	470	371
borensberg	172	207	950	768	104	139	654	518
boxholm-1	90	119	570	449	67	96	455	357
boxholm	189	228	1109	877	112	151	732	573
brokind-1	89	101	479	385	43	55	263	208
brokind	163	179	851	686	62	78	380	299
ekangen-1	74	82	380	309	33	41	179	147
ekangen	129	145	683	551	60	76	346	279
grebo	56	57	284	227	5	6	29	23
liu	127	153	559	496	76	102	370	325
mantorp-a	127	152	733	582	81	106	505	399
mantorp-b	68	83	400	317	43	58	275	217
mantorp	165	194	926	738	91	120	567	449
norsholm	64	77	368	293	51	64	305	242
rimforsa-a-1	64	74	354	283	31	41	195	154
rimforsa-a-2	30	34	158	128	15	19	91	72
rimforsa-a	93	108	511	410	48	63	300	237
rimforsa	195	219	1045	839	94	118	588	459
ryde	73	73	279	249	6	6	22	20
skanninge-a-1	68	86	410	325	42	60	282	222
skanninge-a-2	45	55	263	209	39	49	235	186
skanninge-a	111	140	667	529	79	108	511	403
skanninge	123	153	729	579	81	111	525	414
sturefors	138	146	674	552	29	37	169	136
ull1	41	43	199	163	10	12	58	46
ull2	44	47	226	182	23	26	123	99
valla-a	40	55	250	200	33	48	219	174
valla	69	85	393	316	41	57	263	209
vikingstad-a-1	32	35	168	135	20	23	112	89
vikingstad-a-2	63	86	407	321	52	75	352	277
vikingstad-a-3b	10	12	58	46	10	12	58	46
vikingstad-a-3	15	17	83	66	10	12	58	46
vikingstad-a	110	140	666	528	83	113	535	422
vikingstad	158	188	900	717	83	113	535	422

Table 3: Network sizes of the city instances without and with tree elimination.

Name	q	$\|x\|$	$\|z\|$	$\|v\|$	c1x	c1z	c2	c3
askeby-te	2	37536	246	112	4451	101	720	126
askeby-te	3	49266	369	115	5826	117	933	126
askeby-te	4	57960	492	118	6849	133	1124	126
askeby-te	5	107640	615	121	12624	149	1612	126
atvid-a-1-te	2	568832	1278	559	53810	484	3113	713
atvid-a-1-te	3	720192	1917	562	68065	550	3972	713
atvid-a-1-te	4	850432	2556	565	80340	616	4801	713
atvid-a-1-te	5	1115840	3195	568	105287	682	5822	713
atvid-a-2-te	2	427460	1154	498	41981	443	2730	602
atvid-a-2-te	3	478500	1731	501	47002	505	3392	602
atvid-a-2-te	4	495088	2308	504	48675	567	4000	602
atvid-a-2-te	5	484880	2885	507	47744	629	4566	602
atvid-a-3-te	2	99828	642	280	9831	245	1432	336
atvid-a-3-te	3	117882	963	283	11598	279	1809	336
atvid-a-3-te	4	157176	1284	286	15405	313	2246	336
atvid-a-3-te	5	157530	1605	289	15472	347	2573	336
atvid-a-te	2	3200792	3098	1333	301012	1174	7411	1665
atvid-a-te	3	3151260	4647	1336	296531	1334	8936	1665
atvid-a-te	4	3306688	6196	1339	311250	1494	10581	1665
atvid-a-te	5	3544100	7745	1342	333649	1654	12274	1665
atvid-te	2	3105144	3098	1333	292052	1174	7355	1665
atvid-te	3	3535560	4647	1336	332531	1334	9161	1665
atvid-te	4	3518480	6196	1339	331090	1494	10705	1665
atvid-te	5	4210220	7745	1342	396049	1654	12664	1665
bankekind-te	2	52772	284	128	5789	113	849	147
bankekind-te	3	72522	426	131	7930	130	1121	147
bankekind-te	4	59408	568	134	6535	147	1185	147
bankekind-te	5	112970	710	137	12314	164	1671	147
borensberg-a-te	2	1126608	1728	743	112387	669	4258	900
borensberg-a-te	3	1471488	2592	746	146681	764	5487	900
borensberg-a-te	4	1279888	3456	749	127775	859	6156	900
borensberg-a-te	5	1427420	4320	752	142499	954	7179	900
borensberg-b-te	2	611408	1340	577	60667	517	3214	696
borensberg-b-te	3	810264	2010	580	80303	590	4157	696
borensberg-b-te	4	834008	2680	583	82711	663	4864	696
borensberg-b-te	5	920080	3350	586	91251	736	5655	696
borensberg-te	2	1616160	1864	800	164526	728	4881	961
borensberg-te	3	1855476	2796	803	188885	833	6049	961
borensberg-te	4	1984976	3728	806	202114	938	7111	961
borensberg-te	5	2139340	4660	809	217863	1043	8197	961
boxholm-1-te	2	522648	1294	558	50267	493	3042	680
boxholm-1-te	3	668304	1941	561	64206	561	3898	680
boxholm-1-te	4	634032	2588	564	61009	629	4502	680
boxholm-1-te	5	788970	3235	567	75832	697	5371	680
boxholm-te	2	2007792	2068	890	198773	799	5445	1089
boxholm-te	3	1622736	3102	893	160917	912	6148	1089
boxholm-te	4	1916112	4136	896	189957	1025	7443	1089
boxholm-te	5	2727480	5170	899	270073	1138	9190	1089

Table 4: Model sizes of the city instances after tree elimination, with optimized T^{MAX}, part 1.

Name	q	$\lvert x \rvert$	$\lvert z \rvert$	$\lvert v \rvert$	c1x	c1z	c2	c3
brokind-1-te	2	243776	746	325	26078	296	1923	385
brokind-1-te	3	287040	1119	328	30697	340	2405	385
brokind-1-te	4	307840	1492	331	32940	384	2833	385
brokind-1-te	5	374400	1865	334	40023	428	3371	385
brokind-te	2	767832	1072	465	81315	425	3204	558
brokind-te	3	701454	1608	468	74384	488	3634	558
brokind-te	4	691288	2144	471	73375	551	4158	558
brokind-te	5	920920	2680	474	97629	614	5083	558
ekangen-1-te	2	146412	522	227	17145	215	1429	251
ekangen-1-te	3	284004	783	230	33090	249	2163	251
ekangen-1-te	4	244608	1044	233	28567	283	2295	251
ekangen-1-te	5	248430	1305	236	29042	317	2574	251
ekangen-te	2	547956	996	429	60301	401	2756	496
ekangen-te	3	515592	1494	432	56823	462	3201	496
ekangen-te	4	627192	1992	435	69083	523	3904	496
ekangen-te	5	876060	2490	438	96349	584	4853	496
grebo-te	2	9844	82	42	1317	35	370	42
grebo-te	3	13938	123	45	1856	41	505	42
grebo-te	4	13984	164	48	1867	47	552	42
grebo-te	5	29670	205	51	3918	53	939	42
liu-te	2	564200	1148	479	67313	479	2868	486
liu-te	3	594750	1722	482	71008	556	3494	486
liu-te	4	694200	2296	485	82865	633	4226	486
liu-te	5	945750	2870	488	112740	710	5192	486
mantorp-a-te	2	793212	1434	618	82069	563	3554	742
mantorp-a-te	3	857052	2151	621	88710	645	4356	742
mantorp-a-te	4	880992	2868	624	91251	727	5108	742
mantorp-a-te	5	1045380	3585	627	108224	809	6036	742
mantorp-b-te	2	234360	782	340	24063	305	1940	406
mantorp-b-te	3	261702	1173	343	26878	349	2399	406
mantorp-b-te	4	328104	1564	346	33653	393	2948	406
mantorp-b-te	5	351540	1955	349	36072	437	3398	406
mantorp-te	2	1381124	1614	694	142127	633	4416	832
mantorp-te	3	269397306	2421	697	27600446	725	303687	832
mantorp-te	4	1831920	3228	700	188493	817	6542	832
mantorp-te	5	1728650	4035	703	178004	909	7239	832
norsholm-te	2	248776	866	376	27072	346	2064	444
norsholm-te	3	389136	1299	379	42203	398	2792	444
norsholm-te	4	394944	1732	382	42878	450	3242	444
norsholm-te	5	396880	2165	385	43137	502	3684	444
rimforsa-a-1-te	2	129360	554	243	13656	218	1415	287
rimforsa-a-1-te	3	185724	831	246	19543	250	1880	287
rimforsa-a-1-te	4	179872	1108	249	18966	282	2143	287
rimforsa-a-1-te	5	244860	1385	252	25749	314	2636	287
rimforsa-a-2-te	2	28224	258	117	3238	104	665	133
rimforsa-a-2-te	3	45360	387	120	5157	120	918	133
rimforsa-a-2-te	4	35136	516	123	4036	136	981	133
rimforsa-a-2-te	5	47520	645	126	5427	152	1201	133
rimforsa-a-te	2	284400	852	370	29733	335	2125	441
rimforsa-a-te	3	354078	1278	373	36984	384	2703	441
rimforsa-a-te	4	475896	1704	376	49625	433	3391	441
rimforsa-a-te	5	447930	2130	379	46782	482	3763	441

Table 5: Model sizes of the city instances after tree elimination, with optimized T^{MAX}, part 2.

Name	q	$\lvert x\rvert$	$\lvert z\rvert$	$\lvert v\rvert$	c1x	c1z	c2	c3
rimforsa-te	2	1147500	1648	713	119397	649	4202	870
rimforsa-te	3	1605582	2472	716	166896	744	5530	870
rimforsa-te	4	1960848	3296	719	203755	839	6746	870
rimforsa-te	5	1982880	4120	722	206129	934	7599	870
ryde-te	2	8560	68	35	1530	34	342	26
ryde-te	3	12840	102	38	2285	41	488	26
ryde-te	4	12640	136	41	2256	48	522	26
ryde-te	5	15600	170	44	2780	55	635	26
skanninge-a-1-te	2	270840	804	349	26536	308	2048	420
skanninge-a-1-te	3	355644	1206	352	34791	351	2646	420
skanninge-a-1-te	4	303696	1608	355	29802	394	2936	420
skanninge-a-1-te	5	470640	2010	358	46012	437	3719	420
skanninge-a-2-te	2	168888	666	291	18424	266	1649	343
skanninge-a-2-te	3	172980	999	294	18903	306	1998	343
skanninge-a-2-te	4	193440	1332	297	21142	346	2391	343
skanninge-a-2-te	5	303180	1665	300	32981	386	3024	343
skanninge-a-te	2	751192	1454	626	75121	563	3526	756
skanninge-a-te	3	972036	2181	629	97120	643	4532	756
skanninge-a-te	4	951080	2908	632	95119	723	5238	756
skanninge-a-te	5	1192880	3635	635	119198	803	6270	756
skanninge-te	2	919080	1494	643	91596	578	3775	777
skanninge-te	3	1107864	2241	646	110373	660	4755	777
skanninge-te	4	1079712	2988	649	107666	742	5473	777
skanninge-te	5	1544220	3735	652	153749	824	6786	777
sturefors-te	2	217056	486	213	24134	196	1669	243
sturefors-te	3	270912	729	216	30103	226	2115	243
sturefors-te	4	336192	972	219	37332	256	2603	243
sturefors-te	5	371280	1215	222	41231	286	2980	243
ull1-te	2	26680	164	77	3256	68	592	84
ull1-te	3	31464	246	80	3838	79	731	84
ull1-te	4	49312	328	83	5982	90	1012	84
ull1-te	5	49680	410	86	6036	101	1103	84
ull2-te	2	72468	350	156	8929	147	998	174
ull2-te	3	113454	525	159	13920	171	1385	174
ull2-te	4	78408	700	162	9695	195	1388	174
ull2-te	5	99000	875	165	12214	219	1672	174
valla-a-te	2	130152	630	274	12956	242	1500	324
valla-a-te	3	146160	945	277	14553	276	1866	324
valla-a-te	4	168432	1260	280	16762	310	2250	324
valla-a-te	5	153120	1575	283	15299	344	2526	324
valla-te	2	219032	754	327	22299	293	1871	387
valla-te	3	274626	1131	330	27926	335	2386	387
valla-te	4	297616	1508	333	30277	377	2823	387
valla-te	5	332310	1885	336	33804	419	3288	387

Table 6: Model sizes of the city instances after tree elimination, with optimized T^{MAX}, part 3.

Name	q	$\|x\|$	$\|z\|$	$\|v\|$	c1x	c1z	c2	c3
vikingstad-a-1-te	2	45924	316	142	5547	131	831	161
vikingstad-a-1-te	3	45390	474	145	5504	152	991	161
vikingstad-a-1-te	4	70488	632	148	8485	173	1295	161
vikingstad-a-1-te	5	69420	790	151	8379	194	1452	161
vikingstad-a-2-te	2	312456	1004	434	30273	383	2357	525
vikingstad-a-2-te	3	320766	1506	437	31120	436	2879	525
vikingstad-a-2-te	4	361208	2008	440	35041	489	3459	525
vikingstad-a-2-te	5	459820	2510	443	44527	542	4144	525
vikingstad-a-3b-te	2	8648	164	77	1100	68	396	84
vikingstad-a-3b-te	3	11040	246	80	1396	79	509	84
vikingstad-a-3b-te	4	15456	328	83	1934	90	644	84
vikingstad-a-3b-te	5	11960	410	86	1526	101	693	84
vikingstad-a-3-te	2	11776	164	77	1474	68	430	84
vikingstad-a-3-te	3	16008	246	80	1990	79	563	84
vikingstad-a-3-te	4	25392	328	83	3122	90	752	84
vikingstad-a-3-te	5	19780	410	86	2461	101	778	84
vikingstad-a-te	2	705584	1522	655	70812	590	3551	791
vikingstad-a-te	3	822900	2283	658	82571	674	4456	791
vikingstad-a-te	4	850752	3044	661	85426	758	5255	791
vikingstad-a-te	5	1109860	3805	664	111297	842	6328	791
vikingstad-te	2	855816	1522	655	85764	590	3729	791
vikingstad-te	3	1015332	2283	658	101723	674	4684	791
vikingstad-te	4	1238992	3044	661	124066	758	5715	791
vikingstad-te	5	1266000	3805	664	126837	842	6513	791

Table 7: Model sizes of the city instances after tree elimination, with optimized T^{MAX}, part 4.

4.1 Software and hardware

The implementation was done in Python, and was run on a desktop computer with an Intel Core i7 processor (8-Core, 16-Thread, 2.5/4.9GHz), a 16GB 2666 MHz DDR4, and a 500GB SSD, running Fedora Core 37. As MIP-solver we use Gurobi v10, with gurobipy as interface to Python.

4.2 Tests on the instance askeby-te

Here we report some preliminary tests on the instance askeby-te, i.e. askeby after tree elimination. The size of the instance is 15 nodes, 18 links, 87 tasks in 69 groups, and 4 vehicles. An earlier found feasible solution yields the objective function value 208 and $v^{TOT} = 130$. The latter is used for dimensioning the x and I variables. The number of variables (columns) in all models is 72362, and includes all x, I and z variables.

The basic subproblem 000000 has 10217 rows and 81089 nonzeros. In table 8, we give the results for the various MIP models for instances askeby and askeby-te. The runtime was limited to 1000 seconds.

For askeby-te we note that the eight first models give the optimal solution rather fast, less than 13 seconds. The other two do not produce any feasible solution in 1000 seconds. We did similar tests with the instance askeby, without the tree elimination. The size of

	askeby-te			askeby		
Model	Rows	Obj	Time	Rows	Obj	Time
000000	10217	124.0	8.188	30694	90.0	25.811
100000	10286	340.0	8.203	30914	249.5	27.107
010000	18017	124.0	9.044	58254	90.0	29.922
001000	10737	124.0	8.300	31214	90.0	28.045
000100	10709	339.5	8.262	32242	252.5	27.312
000010	24257	124.0	12.984	73854	90.0	43.199
000001	178697	124.0	10.151	549654	90.0	32.872
011000	18537	124.0	9.287	58774	90.0	31.097
010100	18509	-	>1000	59802	-	>1000
011100	19029	-	>1000	60322	-	>1000

Table 8: Solution of several models for askeby-te and askeby with $q = 4$.

the instance now is 53 nodes, 56 links, 275 tasks in 220 groups, so it is larger. The MIP model has 230684 columns. For askeby, the pattern is similar, but the times are larger. One may note that 000001 has a large number of rows, but is not very much harder (or stronger).

Concerning the objective function values, we find that 100000 and 000100 give much higher values than the others (that were solved), although they don't take significantly longer to solve. These are thus better to use than the others. We conclude that adding the demand constraints or the coupling constraints, we have a rather easily solvable but strong problem.

We might mention that the model 100100 is uninteresting, since the coupling constraints plus the demand constraints in z (which are present in all these models) make the demand constraints in x redundant.

The two sets of constraints controlling the precedences and switching are quite numerous, and they probably add to the difficulty a lot, if combined with the others. It would be more or less out of the question to solve the problem with these constraints (as constraints).

4.3 Tests of the chosen subproblem

In this section, we simply solve the subproblem that we found to be best in the previous tests, model 100000, namely model M14 plus the x and I variables plus the demand constraints in x for all the instances of interest. Then, we solve the same problem, but without the demand constraints in x, to see if it is faster. To our surprise, it is not always faster. Finally, we solve the subproblem without the x-variables. Then we had to change the objective function, since the one we use is based on the x-variables. Instead, we use v^{TOT} as objective function. We mostly do this test to compare solution times. See tables 9 and 10, where obj_1 and time1 refer to the model with x-variables and demand constraints, obj_2 and time2 to the model with x-variables but without demand constraints, and obj_3 and time3 refer to the model without x-variables and demand constraints.

Network	obj_1	time1	obj_2	time2	obj_3	time3
mini-05a	8.280	0.0075	8.000	0.0032	8.000	0.0029
mini-05b	10.310	0.0120	10.000	0.0104	10.000	0.0094
mini-05c	10.360	0.0077	10.000	0.0060	10.000	0.0043
mini-05d	10.360	0.0060	10.000	0.0054	10.000	0.0043
mini-06a	18.660	0.0170	18.000	0.0145	18.000	0.0114
mini-06b	18.620	0.0181	18.000	0.0086	18.000	0.0057
mini-06c	19.700	0.0206	19.000	0.0111	19.000	0.0079
mini-08a	19.710	0.0602	19.000	0.0155	19.000	0.0102
mini-08b	16.610	0.0180	16.000	0.0109	16.000	0.0062
mini-08c	17.630	0.0399	17.000	0.0391	17.000	0.0347
mini-09a	34.240	0.1330	33.000	0.0684	33.000	0.0939
mini-09b	27.020	0.0171	26.000	0.0161	26.000	0.0070
mini-09c	37.340	0.1269	36.000	0.3124	36.000	0.2706

Table 9: Tests with a few versions of the subproblem for the mini instances.

The times differences are in general smaller than what we expected. It seems to be less costly than one may think to keep the numerous x-variables, as long as we don't have too many constraints on them.

4.4 Tests of weights

Next we did some tests to see what effects different values of the weights in the objective function, α_1, α_2 and α_3, would give. We recall that α_1 is the weight of v^{TOT}, α_2 is the weight of the sum of ending times of the tasks, and α_3 is the weight of the costs for transportation. Obviously, different weights give different solutions, but now we are mostly interested in the solution time. We are using the model M14 with all the variables, including x and I, plus the demand constraints. As a test instance we have chosen valla-te, which has 41 nodes, 57 links, 263 tasks in 209 groups. The model has 58725 rows, 416497 columns and 705307 nonzeros.

In table 11, the first row contains our standard setting used in previous work. A large weight, α_1, on v^{TOT}, and smaller weights on the other aspects. The table first gives the three values of α, then the total objective function value (with those weights), then the four parts of the objective function values, Obj1 is v^{TOT}, the time when everything is ready, Obj2, the sum of all ending time for the tasks, and Obj3a and Obj3b, the total transportation times, first forwards, then backwards, in the links. Here we see that there is no transportation, which is good. Obj2 is almost constant, while Obj1 varies between 127 and 248.

The computational times have for some reason a maximum for the first row. Several other settings yield shorter solution times. We especially note that $\alpha = (1, 0, 0)$, i.e. all weight on v^{TOT}, yields the same solution, but faster. The same is true for $\alpha = (1, 0.5, 0.1)$, which we pick as our favorite setting, since it reflects our general priorities.

Network	obj_1	time1	obj_2	time2	obj_3	time3
askeby-te	128.280	0.0556	124.000	0.0543	124.000	0.0328
atvid-a-1-te	174.530	3.4372	168.000	3.5281	168.000	3.4639
atvid-a-2-te	130.800	5.3474	126.000	6.3729	126.000	3.4927
atvid-a-3-te	60.200	0.6334	58.000	1.3713	58.000	2.1234
atvid-a-te	366.510	25.3509	352.000	25.9759	352.000	18.7197
atvid-te	417.490	22.3654	402.000	31.1500	402.000	16.9692
bankekind-te	86.170	0.0614	84.000	0.0526	84.000	0.0260
borensberg-a-te	253.450	11.1446	246.000	19.6797	245.000	11.7287
borensberg-b-te	187.990	8.0513	181.000	11.6933	181.000	6.1314
borensberg-te	314.720	14.4216	303.000	14.2375	303.000	13.1115
boxholm-1-te	170.340	8.0743	164.000	6.4539	164.000	5.8856
boxholm-te	367.790	12.8206	354.000	14.6498	354.000	14.1625
brokind-1-te	174.570	0.6184	168.000	1.3898	168.000	1.3283
brokind-te	321.130	3.1329	309.000	0.7559	309.000	0.4565
ekangen-1-te	146.500	0.4126	141.000	0.2933	141.000	0.2034
ekangen-te	256.700	1.0704	247.000	1.2919	247.000	0.8118
grebo-te	266.940	0.1491	262.000	0.0386	262.000	0.0039
liu-te	198.490	1.1089	191.000	0.5371	191.000	0.2724
mantorp-a-te	246.220	4.5370	237.000	6.1870	237.000	4.5575
mantorp-b-te	130.940	1.0381	126.000	0.6603	126.000	0.5840
mantorp-te	324.160	8.3651	312.000	8.8131	312.000	7.4210
norsholm-te	112.130	0.4808	108.000	1.2813	108.000	0.7583
rimforsa-a-1-te	152.810	0.1502	148.000	0.1599	148.000	0.0598
rimforsa-a-2-te	56.130	0.0379	54.000	0.0395	54.000	0.0211
rimforsa-a-te	180.810	1.5472	174.000	1.3615	174.000	1.4684
rimforsa-te	370.990	2.7931	357.000	4.0515	357.000	3.1439
ryde-te	384.700	0.0836	380.000	0.0305	380.000	0.0043
skanninge-a-1-te	143.960	0.3039	138.000	0.2718	138.000	0.1622
skanninge-a-2-te	75.790	2.3553	73.000	1.3566	73.000	2.1420
skanninge-a-te	206.680	9.9497	199.000	8.8066	199.000	7.5843
skanninge-te	232.690	10.0095	224.000	7.7640	224.000	6.4910
sturefors-te	432.830	0.5278	420.000	0.2902	420.000	0.0462
ull1-te	121.140	0.0337	118.000	0.0326	118.000	0.0129
ull2-te	84.160	0.0979	81.000	0.0946	81.000	0.0459
valla-a-te	72.650	2.0492	70.000	1.1896	70.000	1.4400
valla-te	131.860	2.0369	127.000	1.8214	127.000	2.1961
vikingstad-a-1-te	59.210	0.4495	57.000	0.8010	57.000	0.2198
vikingstad-a-2-te	119.370	2.6874	115.000	2.7964	115.000	3.5407
vikingstad-a-3b-te	15.560	0.1775	15.000	0.2055	15.000	0.2016
vikingstad-a-3-te	29.060	0.1022	28.000	0.0625	28.000	0.1209
vikingstad-a-te	207.650	6.1472	200.000	4.7759	200.000	7.6011
vikingstad-te	401.870	1.1893	390.000	0.9131	390.000	0.3175

Table 10: Tests with a few versions of the subproblem for the city instances.

α_1	α_2	α_3	Obj	Obj1	Obj2	Obj3a	Obj3b	Time
1	0.01	0.001	131.9	127.00	486.00	0.00	0.00	2.2610
1	0	0	127.0	127.00	486.00	0.00	0.00	1.8306
0	1	0	486.0	248.00	486.00	0.00	0.00	0.2685
0	0	1	0.0	248.00	668.00	0.00	0.00	0.2970
1	1	0	615.0	129.00	486.00	0.00	0.00	0.3374
1	1	1	615.0	129.00	486.00	0.00	0.00	0.3720
0	1	1	486.0	248.00	486.00	0.00	0.00	0.2606
1	0	1	127.0	127.00	486.00	0.00	0.00	1.8632
1	0.5	0.1	370.0	127.00	486.00	0.00	0.00	0.4256
1	0.3	0.2	272.8	127.00	486.00	0.00	0.00	1.7724

Table 11: Tests with the subproblem for different weights.

Model	Rows	Obj	Time	Model	Rows	Obj	Time
000000	10940	84.0	9.199	101100	19668	579.4	12.801
100000	11019	192.5	9.245	101010	200540	567.0	38.724
010000	18620	84.0	10.237	101001	34308	563.0	12.314
001000	11420	84.0	9.589	100110	200628	192.9	26.096
000100	11508	192.9	9.913	100101	215180	192.9	17.604
000010	26060	84.0	14.376	100011	27108	192.5	37.812
000001	192380	84.0	11.187	011100	193428	-	>1000
110000	18699	-	>1000	011010	208068	84.0	26.180
101000	11587	563.0	9.590	011001	19747	84.0	15.075
100100	26139	192.9	9.395	010110	34299	-	>1000
100010	192459	192.5	16.880	010101	200619	-	>1000
100001	19100	192.5	12.334	010011	34387	84.0	47.415
011000	19188	84.0	10.141	001110	200707	579.5	100.218
010100	33740	-	>1000	110101	27187	579.4	19.862
010010	200060	84.0	26.229	001101	193507	192.9	53.644
010010	11988	84.0	26.557	000111	34788	-	>1000
010001	26540	84.0	14.931	111100	201108	-	>1000
001100	192860	580.5	15.059	111010	215660	-	>1000
001010	26628	84.0	14.713	111001	215748	-	>1000
001001	192948	84.0	11.910	110110	208548	-	>1000
000110	207500	192.9	28.453	101110	34867	579.5	185.438
000101	19179	192.9	18.992	101101	201187	579.5	20.369
000011	26619	84.0	16.581	011110	215739	-	>1000
111000	192939	-	>1000	011101	215827	-	>1000
110100	26707	-	>1000	011011	208627	84.0	51.881
110010	193027	-	>1000	010111	216228	-	>1000
110001	207579	-	>1000	001111	216307	579.4	261.666

Table 12: Solution of several MIP models for bankekind-te with $q = 4$.

4.5 Tests on bankekind

With the chosen setting of the weights, $\alpha = (1, 0.5, 0.1)$, we did some more tests on a single instance, this time Bankekind. The instance bankekind-te has (after tree elimination) 16 node, 21 links, 100 tasks in 79 groups, and we used 4 vehicles. We solved the models in which different sets of constraints were included. We used a maximum time of 1000 seconds.

In table 12, we give the following. First, we give what constraints were left in the subproblem, then we give the number of rows in the model, and finally the optimal objective function value and the solution time for Gurobi. All models in table 12 have 76534 columns, consisting of 126 continuous and 76408 integer variables. The number of rows vary, and are given in the table.

We find that there are large variations in objective function value. First there are a number of models yielding the rather bad lower bound 84. Including constraint set 1 (demand) increases the objective function value to 192, while the solution time is still quite small. Including constraint 4 (coupling) gives a similar objective function value in slightly longer time. (Including both is, as predicted, the same as including only the coupling constraints.)

Including both constraint sets 1 and 2 (demand and node equilibrium), the model is

		MIP		LP	
Network	q	Obj	Time	Obj	Time
mini-05a	2	74.2	0.606	69.926	0.097
mini-05b	2	92.0	2.506	82.808	0.831
mini-05c	2	116.1	6.664	102.503	0.446
mini-05d	2	116.1	4.448	102.503	0.390
mini-06a	2	337.2	1153.712	309.684	1.409
mini-06b	2	305.1	581.932	279.310	0.902
mini-06c	2	-	>2000	344.322	1.946
mini-08a	2	-	>2000	406.880	3.619
mini-08b	2	-	>2000	315.136	0.952
mini-08c	2	-	>2000	332.555	1.116
mini-09a	2	-	>2000	1210.401	99.609
mini-09b	2	-	>2000	830.846	1.965
mini-09c	2	-	>2000	1408.202	88.306
mini-08a	3	-	>2000	281.531	1.436
mini-08b	3	227.9	833.881	220.758	0.985
mini-08c	3	-	>2000	232.857	1.155
mini-09a	3	-	>2000	838.305	29.942
mini-09b	3	-	>2000	581.638	1.910
mini-09c	3	-	>2000	956.960	80.804

Table 13: Solution of M10 for mini-problems.

not solved in 1000 seconds. Finally, we have some models that give objective function value around 580. All these models include constraint sets 3 and 4 (start one at a time and coupling). Adding constraint sets 1 and 5 (demand and precedences) increases the solution time much.

We thus have found a number of interesting models. For simple problems, as this one, using all constraints except 2, 5 and 6 (node equilibrium, precedences and switching) gives a good lower bound rather quickly. For a harder problem, using constraints 1 and 2 (demand and node equilibrium) gives a rather good lower bound. For even harder problems, including only constraints 1 (demand) or constraints 4 (coupling) is a possibility.

One should note that we are talking about the same instance, so the complete optimal objective function value is above 582. At this stage, we had not found any primal feasible solutions to the complete model.

4.6 Comparison of MIP and LP models

Now we want to see how far we get with the complete model M10, and also with its LP-relaxation. Furthermore, we look at the combination model M1014.

In table 13, we give the results of solution of model M10, the basic complete model, for some smaller instances. We find that only a few could be solved as MIP within 2000 seconds, but the solution of the LP-relaxation was obtained for all of them.

In table 14, we do the same, but with model M1014. Since model M10 is exact, the solution should not be changed by this. The only difference should be that the model

		MIP		LP	
Network	q	Obj	Time	Obj	Time
mini-05a	2	74.2	0.408	71.709	0.211
mini-05b	2	92.0	1.067	84.497	0.290
mini-05c	2	116.1	5.365	105.653	0.583
mini-05d	2	116.1	4.442	105.653	0.452
mini-06a	2	337.2	112.140	314.167	2.184
mini-06b	2	306.6	99.009	282.029	1.222
mini-06c	2	379.2	858.534	347.493	2.436
mini-08a	2	427.0	1246.689	415.466	3.111
mini-08b	2	326.7	31.986	321.496	1.239
mini-08c	2	347.5	327.168	339.269	1.609
mini-09a	2	-	>2000	1222.424	109.293
mini-09b	2	860.7	1132.250	838.867	2.035
mini-09c	2	-	>2000	1419.267	62.695
mini-08a	3	-	>2000	286.575	2.436
mini-08b	3	228.4	125.353	224.724	1.368
mini-08c	3	242.2	588.574	237.114	1.684
mini-09a	3	-	>2000	846.033	58.169
mini-09b	3	599.5	353.849	587.741	2.126
mini-09c	3	-	>2000	964.311	231.409

Table 14: Solution of M1014 for mini-problems.

		M10		M1014	
Network	q	Obj	Time	Obj	Time
grebo-te	2	631.5	119.571	631.5	182.951
grebo-te	3	573.2	138.129	577.5	100.807
ryde-te	2	716.3	102.149	712.0	89.456
ryde-te	3	671.8	86.775	671.8	106.940
ull1-te	4	-	-	459.4	201.491
vikingstad-a-3-te	2	396.9	1680.143	397.1	1446.578
vikingstad-a-3-te	3	281.6	740.784	281.6	1789.342
vikingstad-a-3-te	4	224.3	51.153	225.0	85.608

Table 15: Solution of M10 and M1014 for a few small city-problems.

becomes larger. However the question is if this model has advantages when it comes to MIP methods.

We find that M1014 indeed is better than M10. More of the instances were solved as MIP within 2000 seconds. One example is mini-06a, for which M10 was solved in 1153 seconds, while M1014 only took 112 seconds. On the other hand, we see that solving the LP-relaxation of M1014 usually takes more time than solving the LP-relaxation of M10. We conclude that adding M14 to M10 improves the integer structure.

In table 15, we do the same for the smallest city-instances. Here we find that the results are mixed. M1014 is not always better than M10. We should perhaps recall that a tolerance of 0.01 was used when solving the MIPs with Gurobi, i.e. branches were cut if they did not improve the objective function value with more than 1%. Therefore differences smaller than that between the objective function values may appear.

One thing to point out here is that the difficulty in these instances might lie more in the LP-problem than in the integer structure. For several of these instances, the branch-

		LP of M10		LP of M1014	
Network	q	Obj	Time	Obj	Time
askeby-te	2	1152.854	515.877	1154.354	475.897
askeby-te	3	844.392	186.875	844.392	487.403
askeby-te	4	749.670	23.231	749.670	140.359
atvid-a-2-te	4	6954.053	913.981	-	-
atvid-a-3-te	2	4229.263	800.466	4229.263	1497.363
atvid-a-3-te	3	2842.589	252.408	-	-
atvid-a-3-te	4	2152.800	68.305	2152.800	767.303
bankekind-te	2	1049.269	50.632	1049.269	59.809
bankekind-te	3	752.406	45.148	752.406	545.231
bankekind-te	4	608.725	28.480	608.725	60.813
grebo-te	2	628.300	46.466	628.300	36.981
grebo-te	3	567.100	11.144	567.100	15.181
rimforsa-a-2-te	2	915.550	64.065	922.530	109.662
rimforsa-a-2-te	3	640.432	22.727	641.837	384.997
rimforsa-a-2-te	4	511.100	14.103	511.100	25.073
ryde-te	2	706.512	11.673	706.512	12.491
ryde-te	3	667.667	16.519	667.667	18.523
ull1-te	2	693.050	594.565	693.050	98.576
ull1-te	3	523.600	29.741	523.600	124.449
ull1-te	4	447.100	14.225	447.100	20.445
ull2-te	2	1737.152	460.034	1740.687	935.452
ull2-te	3	1205.435	71.608	1205.435	1174.481
ull2-te	4	941.222	30.720	941.222	230.059
vikingstad-a-1-te	2	1342.286	53.044	1371.733	141.252
vikingstad-a-1-te	3	918.150	21.708	936.343	110.106
vikingstad-a-1-te	4	707.493	22.903	720.179	85.735
vikingstad-a-3b-te	2	315.440	5.540	321.640	5.624
vikingstad-a-3b-te	3	216.842	4.571	220.942	5.373
vikingstad-a-3b-te	4	172.163	5.044	175.650	5.824
vikingstad-a-3-te	2	381.600	7.273	386.855	12.110
vikingstad-a-3-te	3	268.750	5.292	269.479	7.169
vikingstad-a-3-te	4	218.400	7.221	218.400	8.754

Table 16: Solution of LP-relaxation of M10 and M1014 for small city-problems.

and-bound tree in Gurobi was quite small. Most of the time was spent in solving the first LP-problem. Then it matters that the LP-problem in M1014 is larger than that in M10. It is possible that the advantages of the better integer structure of M1014 is more important when the integrality is more challenging.

For larger instances, we were not able to solve M10. Therefore, we restricted the tests to solving the LP-relaxations, as shown in table 16. One point is that even if two problems have the same integer optimal objective function value, their LP-relaxations might not. However, the results of this test could best be described as inconclusive.

In order to get further, we now study combining the LP-relaxations with selection of constraints to keep. In other words, we do two different relaxations at the same time. The main goal here is to obtain a good lower bound on the optimal objective function value in reasonable time. It also enables us to tackle larger problems.

In tables 17 and 18, we give the results for the mini-problems, and in tables 19 - 21 we give the results for the smaller city-problems. Tables 22 - 24 and 25 give the results for larger city-instances.

Network	q	Model	Obj	Time	Network	q	Model	Obj	Time
mini-05a	2	000000	14.000	0.277	mini-06a	2	000000	33.000	0.424
mini-05a	2	100000	28.000	0.041	mini-06a	2	100000	66.000	0.278
mini-05a	2	110000	69.727	0.053	mini-06a	2	110000	311.000	0.389
mini-05a	2	111000	69.727	0.059	mini-06a	2	111000	311.000	0.355
mini-05a	2	111100	70.233	0.071	mini-06a	2	111100	312.710	0.384
mini-05a	2	111110	71.709	0.093	mini-06a	2	111110	314.167	1.086
mini-05a	2	111111	71.709	0.133	mini-06a	2	111111	314.167	2.173
mini-05b	2	000000	15.500	0.118	mini-06b	2	000000	31.000	0.242
mini-05b	2	100000	31.000	0.059	mini-06b	2	100000	62.000	0.214
mini-05b	2	110000	82.850	0.074	mini-06b	2	110000	278.996	0.303
mini-05b	2	111000	82.850	0.077	mini-06b	2	111000	278.996	0.351
mini-05b	2	111100	83.700	0.098	mini-06b	2	111100	281.404	0.308
mini-05b	2	111110	84.497	0.805	mini-06b	2	111110	282.029	0.665
mini-05b	2	111111	84.497	0.209	mini-06b	2	111111	282.029	1.199
mini-05c	2	000000	18.000	0.170	mini-06c	2	000000	35.000	0.420
mini-05c	2	100000	36.000	0.107	mini-06c	2	100000	70.000	0.388
mini-05c	2	110000	104.362	0.131	mini-06c	2	110000	344.853	0.496
mini-05c	2	111000	104.362	0.140	mini-06c	2	111000	344.853	0.503
mini-05c	2	111100	105.633	0.163	mini-06c	2	111100	347.493	0.505
mini-05c	2	111110	105.653	0.331	mini-06c	2	111110	347.493	1.506
mini-05c	2	111111	105.653	0.520	mini-06c	2	111111	347.493	2.206
mini-05d	2	000000	18.000	0.161					
mini-05d	2	100000	36.000	0.103					
mini-05d	2	110000	104.362	0.122					
mini-05d	2	111000	104.362	0.126					
mini-05d	2	111100	105.633	0.152					
mini-05d	2	111110	105.653	0.260					
mini-05d	2	111111	105.653	0.440					

Table 17: Solution of the LP-relaxation of several models for the mini-instances 1.

For instances missing from the tables, we were unable to solve the LP-problem in the maximal time, 2000 seconds. Often the code crashed because of the size of the problem. For the larger city-instances, we often did not manage to solve models 111111 and 111110, and for the largest we didn't even try.

General conclusions from these tables are that model 000000 has a very low optimal objective function value. Model 100000 also has a significantly lower optimal objective function value than the rest of the models. The models 110000, 111000, 111100, 111110 and 111111 have quite similar optimal objective function values. Since model 111111 gives the true optimal value, this tells us that omitting the precedence constraints, the switching constraints, the coupling constraints and the "start one at a time" constraints does not affect the objective function value too much for the LP-relaxation.

On the other hand, the solution time for the complete model 111111 is very large compared to the others. For many instances, we were unable to solve this model. Also model 111110 requires much time to solve, and could not always be solved.

From a practical point of view, the models 110000, 111000 and 11100 seem to be good choices. They give good objective function values in relatively short times. In words, it seems good to include the demand constraints and the node equilibrium constraints, but maybe not worthwhile to include the switching and precedence constraints. However, one must keep in mind that this is for the LP-relaxation. We expect the increase in

		Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$	
mini-08a	000000	35.500	0.496	23.667	0.464
mini-08a	100000	71.000	0.455	59.167	0.420
mini-08a	110000	414.200	0.522	285.400	0.528
mini-08a	111000	414.200	0.548	285.400	0.559
mini-08a	111100	415.200	0.568	286.178	0.580
mini-08a	111110	415.466	1.534	286.575	1.418
mini-08a	111111	415.466	3.028	286.575	2.387
mini-08b	000000	30.500	0.265	20.333	0.329
mini-08b	100000	61.000	0.248	50.833	0.316
mini-08b	110000	320.640	0.372	222.156	0.394
mini-08b	111000	320.640	0.331	222.156	0.404
mini-08b	111100	321.233	0.351	223.171	0.437
mini-08b	111110	321.496	0.757	224.724	0.780
mini-08b	111111	321.496	1.186	224.724	1.239
mini-08c	000000	31.500	0.278	21.000	0.355
mini-08c	100000	63.000	0.262	52.500	0.332
mini-08c	110000	338.147	0.393	234.820	0.417
mini-08c	111000	338.147	0.361	234.820	0.427
mini-08c	111100	338.770	0.394	235.822	0.483
mini-08c	111110	339.269	0.964	237.114	0.871
mini-08c	111111	339.269	1.523	237.114	1.427
mini-09a	000000	62.000	1.821	41.333	1.870
mini-09a	100000	124.000	1.844	103.333	1.846
mini-09a	110000	1218.479	2.381	842.880	2.230
mini-09a	111000	1218.479	2.435	842.880	2.322
mini-09a	111100	1221.465	2.596	844.878	2.411
mini-09a	111110	1222.424	21.197	846.033	8.869
mini-09a	111111	1222.424	107.626	846.033	51.244
mini-09b	000000	51.000	0.516	34.000	0.654
mini-09b	100000	102.000	0.519	85.000	0.664
mini-09b	110000	837.642	0.665	584.863	0.818
mini-09b	111000	837.642	0.683	584.863	0.848
mini-09b	111100	838.099	0.724	586.508	0.879
mini-09b	111110	838.867	1.280	587.741	1.324
mini-09b	111111	838.867	1.959	587.741	1.931
mini-09c	000000	67.000	2.071	44.667	2.442
mini-09c	100000	134.000	2.134	111.667	2.477
mini-09c	110000	1416.744	2.729	961.882	3.004
mini-09c	111000	1416.744	2.806	961.882	3.103
mini-09c	111100	1419.265	2.911	964.123	3.613
mini-09c	111110	1419.267	27.671	964.311	34.318
mini-09c	111111	1419.267	62.558	964.311	202.227

Table 18: Solution of the LP-relaxation of several models for the mini-instances 2.

solution time to be much worse for the integer problem.

Another interesting observation is that the problem does not always become more difficult when the number of vehicles increases. There are several examples where it is faster to solve the problem with 4 vehicles than 3. This tells us that the difficulty is not only a matter of size.

In tables 26 - 28, we compare different partial LP-relaxations to the integer solutions for three instances. In these test we used a maximal time of 1000 seconds.

The instance bankekind-te has 16 nodes, 21 links and 100 tasks in 79 groups. The model has 60110 variables, most of them binary. In the LP-relaxation they are obviously all continuous. We also study two other relaxations, one where the x-variables and I-variables are continuous, called LPX, and one where the z-variables are continuous, called LPZ.

The instance rimforsa-a-2-te has 15 nodes, 19 links and 91 tasks in 72 groups. The model has 35775 variables, of which 35652 are binary and 123 continuous. In LPX, 35259 variables are continuous, and 516 binary, so the integrality requirements are restricted to the relatively few z variables. In LPZ, 639 variables are continuous, and 35136 binary, so the integrality requirements apply to the many x and I variables.

The instance vikingstad-a-3-te has 10 nodes, 12 links and 58 tasks in 46 groups. The model has 25720 variables, of which 25803 are binary and 83 continuous in the basic MIP model. In LPX, 25475 variables are continuous, and 328 binary, and in LPZ, 411 variables are continuous, and 25392 binary.

For bankekind-te there are quite small differences between the solution times, with the exception that models 110000 and 111000 were solved by LPX and LP in around 10 seconds, while MIP and LPZ failed to solve them in 1000 seconds. MIP and LPZ failed to solve all models from 010100 and upwards (in difficulty). LPX and LP failed to solve five of them.

The objective function values are quite different, with LP and LPZ on one hand, and MIP and LPX on the other. For the easier models LP and LPZ give 54.2, while MIP and LPX give 84.0. For the more difficult models, LP solves all of them, with objective function values around 606. This is the same as LPX gives for 110000 and 111000. The best choice here would be LPX, since it gives the same objective function value as MIP for the easier problems, but better for some more difficult.

For rimforsa-a-2-te, we make the same observation for the easier models, but the differences are smaller. For the more difficult models, the result is mixed. For model 0101000, MIP and LPZ give 513 in around 500 seconds, while LPX fails to find a solution in 1000 seconds, and LP gives 505 in 8 seconds. On the other hand, for model 110000 LP and LPX give similar solutions in around 8 seconds, while MIP and LPZ need 164 and 149 seconds to find slightly better solutions. Here one would prefer MIP before LPZ, and probably LP before LPX.

Finally, for vikingstad-a-3-te, LPX in some cases fails, while the others succeed in around 5 seconds. Here we see that there is something unstable in LPX. Furthermore MIP is

		Obj	Time	Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$		$q = 4$	
askeby-te	000000	214.000	6.275	142.667	7.876	107.000	8.224
askeby-te	100000	428.000	6.261	356.667	7.910	336.000	8.370
askeby-te	110000	1152.854	7.761	844.392	8.903	747.994	8.874
askeby-te	111000	1152.854	7.710	844.392	9.166	747.994	9.111
askeby-te	111100	1154.354	7.972	844.392	10.428	747.994	9.698
askeby-te	111110	1154.354	192.052	844.392	186.801	749.670	35.126
askeby-te	111111	1154.354	477.285	844.392	486.274	749.670	137.693
bankekind-te	000000	108.500	5.567	72.333	10.552	54.250	9.347
bankekind-te	100000	217.000	5.232	190.500	7.999	190.500	9.350
bankekind-te	110000	1049.269	5.849	752.406	8.785	606.168	9.883
bankekind-te	111000	1049.269	5.945	752.406	8.819	606.168	9.951
bankekind-te	111100	1049.269	5.983	752.406	9.748	606.168	10.501
bankekind-te	111110	1049.269	29.395	752.406	69.413	608.725	31.668
bankekind-te	111111	1049.269	59.032	752.406	534.466	608.725	63.741
rimforsa-a-1-te	000000	240.500	16.596	160.333	23.657	120.250	19.976
rimforsa-a-1-te	100000	481.000	16.987	400.833	24.558	386.500	20.238
rimforsa-a-1-te	110000	3746.697	22.878	2598.175	30.552	2142.287	22.511
rimforsa-a-1-te	111000	3746.697	24.001	2598.175	31.473	2142.287	23.045
rimforsa-a-1-te	111100	3746.697	24.367	2598.175	40.454	2142.287	26.133
rimforsa-a-2-te	000000	106.500	5.448	71.000	6.973	53.250	7.555
rimforsa-a-2-te	100000	213.000	5.480	177.500	7.008	159.750	7.670
rimforsa-a-2-te	110000	917.235	6.403	640.354	7.616	505.287	7.975
rimforsa-a-2-te	111000	917.235	6.609	640.354	7.686	505.287	8.067
rimforsa-a-2-te	111100	922.530	6.631	641.705	8.362	505.287	8.333
rimforsa-a-2-te	111110	922.530	66.053	641.837	51.621	511.100	15.911
rimforsa-a-2-te	111111	922.530	105.296	641.837	379.277	511.100	26.415
ull1-te	000000	157.000	4.054	104.667	6.081	78.500	5.762
ull1-te	100000	314.000	4.120	273.000	5.938	273.000	5.758
ull1-te	110000	693.050	4.903	523.600	6.763	444.600	6.100
ull1-te	111000	693.050	5.047	523.600	6.816	444.600	6.208
ull1-te	111100	693.050	5.183	523.600	8.046	444.600	6.659
ull1-te	111110	693.050	34.298	523.600	67.717	447.100	20.512
ull1-te	111111	693.050	82.317	523.600	119.830	447.100	20.359
ull2-te	000000	158.000	8.057	105.333	10.567	79.000	10.973
ull2-te	100000	316.000	8.218	263.333	10.879	237.000	11.157
ull2-te	110000	1737.417	9.857	1205.435	12.316	941.222	11.989
ull2-te	111000	1737.417	10.304	1205.435	12.474	941.222	11.979
ull2-te	111100	1740.687	10.571	1205.435	15.329	941.222	12.930
ull2-te	111110	1740.687	153.811	1205.435	522.798	941.222	33.162
ull2-te	111111	1740.687	926.966	-	-	941.222	228.929

Table 19: Solution of the LP-relaxation of several models for the city-instances 1a.

		Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$	
grebo-te	000000	247.000	2.611	164.667	2.826
grebo-te	100000	509.000	2.557	509.000	2.783
grebo-te	110000	628.300	2.999	567.100	3.058
grebo-te	111000	628.300	2.992	567.100	3.095
grebo-te	111100	628.300	3.083	567.100	3.247
grebo-te	111110	628.300	20.451	567.100	10.920
grebo-te	111111	628.300	35.737	567.100	14.322
ryde-te	000000	235.000	2.112	156.667	3.064
ryde-te	100000	615.000	2.060	615.000	3.057
ryde-te	110000	706.512	2.387	666.214	3.485
ryde-te	111000	706.512	2.404	666.214	3.559
ryde-te	111100	706.513	2.439	666.214	3.676
ryde-te	111110	706.513	10.480	667.667	15.582
ryde-te	111111	706.512	11.832	667.667	18.794
sturefors-te	000000	541.500	23.728	361.000	34.659
sturefors-te	100000	1083.000	24.972	959.500	35.850
sturefors-te	110000	3821.486	41.069	2812.567	48.486
sturefors-te	111000	3821.486	38.863	2812.567	50.071
sturefors-te	111100	3821.486	44.461	2812.567	71.835

Table 20: Solution of the LP-relaxation of several models for the city-instances 1b.

		Obj	Time	Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$		$q = 4$	
valla-a-te	000000	132.500	14.066	88.333	18.020	66.250	20.067
valla-a-te	100000	265.000	14.463	220.833	18.234	198.750	20.567
valla-a-te	110000	4162.131	17.945	2806.247	20.669	2132.453	21.956
valla-a-te	111000	4162.131	18.615	2806.247	21.035	2132.453	22.192
valla-a-te	111100	4166.377	19.425	2806.249	27.850	2132.453	24.760
valla-a-te	111110	4166.377	314.109	2806.249	241.185	2132.453	75.421
vikingstad-a-1-te	000000	110.500	5.553	73.667	7.494	55.250	9.806
vikingstad-a-1-te	100000	221.000	5.673	184.167	7.579	165.750	9.789
vikingstad-a-1-te	110000	1368.930	6.389	933.515	8.132	717.242	10.452
vikingstad-a-1-te	111000	1368.930	6.418	933.515	8.300	717.242	10.526
vikingstad-a-1-te	111100	1371.733	6.595	936.343	8.775	720.179	11.039
vikingstad-a-1-te	111110	1371.733	26.138	936.343	27.479	720.179	24.745
vikingstad-a-1-te	111111	1371.733	140.313	936.343	110.209	720.179	85.138
vikingstad-a-3b-te	000000	28.000	2.464	18.667	3.364	14.000	4.319
vikingstad-a-3b-te	100000	56.000	2.473	46.667	3.369	42.000	4.229
vikingstad-a-3b-te	110000	319.440	2.552	219.192	3.425	169.600	4.339
vikingstad-a-3b-te	111000	319.440	2.601	219.192	3.471	169.600	4.393
vikingstad-a-3b-te	111100	321.640	2.598	220.942	3.507	171.100	4.436
vikingstad-a-3b-te	111110	321.640	4.196	220.942	4.390	175.650	4.903
vikingstad-a-3b-te	111111	321.640	5.504	220.942	5.230	175.650	5.621
vikingstad-a-3-te	000000	53.000	2.925	35.333	3.921	26.500	4.823
vikingstad-a-3-te	100000	106.000	2.813	88.333	3.562	79.500	4.844
vikingstad-a-3-te	110000	384.975	3.119	268.750	3.747	214.400	4.953
vikingstad-a-3-te	111000	384.975	3.101	268.750	3.761	214.400	5.045
vikingstad-a-3-te	111100	386.855	3.177	269.479	3.895	214.400	5.193
vikingstad-a-3-te	111110	386.855	8.232	269.479	6.230	218.400	7.483
vikingstad-a-3-te	111111	386.855	11.653	269.479	6.888	218.400	8.179

Table 21: Solution of the LP-relaxation of several models for the city-instances 1c.

		Obj	Time	Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$		$q = 4$	
atvid-a-1-te	000000	326.500	36.561	217.667	53.034	163.250	1.023
atvid-a-1-te	100000	653.000	39.722	544.167	57.012	489.750	5.530
atvid-a-1-te	110000	16797.858	58.633	11278.043	75.063	8524.917	1.272
atvid-a-1-te	111000	16797.858	68.358	11278.043	85.669	8524.917	4.747
atvid-a-1-te	111100	16812.259	77.222	11291.582	139.499	8531.811	42.145
atvid-a-2-te	000000	240.000	36.386	160.000	37.514	120.000	43.648
atvid-a-2-te	100000	480.000	39.038	400.000	39.264	360.000	45.487
atvid-a-2-te	110000	13664.329	59.872	9165.859	50.569	6923.628	52.095
atvid-a-2-te	111000	13664.329	64.216	9165.859	54.346	6923.628	54.989
atvid-a-2-te	111100	13672.532	69.703	9172.386	96.013	6929.378	65.304
borensberg-a-te	000000	472.500	75.783	315.000	88.270	236.250	152.140
borensberg-a-te	100000	945.000	82.224	787.500	94.844	708.750	171.382
borensberg-a-te	110000	31560.605	179.346	21159.073	156.077	15960.393	274.973
borensberg-a-te	111000	31560.605	178.076	21159.073	154.081	15960.393	313.308
borensberg-a-te	111100	31574.240	223.468	21169.160	610.559	15967.084	698.517
borensberg-b-te	000000	349.500	40.897	233.000	57.194	174.750	63.532
borensberg-b-te	100000	699.000	44.280	582.500	62.390	524.250	67.892
borensberg-b-te	110000	18890.570	72.272	12683.699	90.834	9587.318	86.047
borensberg-b-te	111000	18890.570	79.335	12683.699	95.050	9587.318	90.631
borensberg-b-te	111100	18901.078	96.032	12690.578	180.501	9590.138	156.392
boxholm-1-te	000000	317.000	47.280	211.333	50.178	158.500	50.094
boxholm-1-te	100000	634.000	50.984	528.333	53.930	475.500	52.991
boxholm-1-te	110000	17259.662	85.283	11576.534	72.495	8744.925	64.942
boxholm-1-te	111000	17259.662	99.141	11576.534	82.620	8744.925	65.697
boxholm-1-te	111100	17265.792	120.532	11579.042	161.205	8746.389	108.895
brokind-1-te	000000	328.500	20.785	219.000	31.142	164.250	34.085
brokind-1-te	100000	657.000	21.230	547.500	32.149	492.750	35.308
brokind-1-te	110000	6764.951	29.177	4612.341	40.476	3545.275	41.445
brokind-1-te	111000	6764.951	30.101	4612.341	41.368	3545.275	42.772
brokind-1-te	111100	6765.053	33.041	4612.341	57.693	3545.275	54.630
brokind-te	000000	606.500	49.916	404.333	80.187	303.250	59.746
brokind-te	100000	1213.000	54.251	1010.833	86.974	909.750	63.784
brokind-te	110000	14105.057	90.648	9714.416	138.102	7589.734	80.007
brokind-te	111000	14105.057	104.408	9714.416	142.131	7589.734	81.903
brokind-te	111100	14105.321	116.583	9714.416	264.672	7589.734	126.491
ekangen-te	000000	485.000	37.999	323.333	48.289	242.500	54.880
ekangen-te	100000	970.000	40.501	808.333	51.237	727.500	58.701
ekangen-te	110000	12069.268	60.916	8202.900	68.560	6287.993	72.375
ekangen-te	111000	12069.268	65.796	8202.900	71.713	6287.993	77.496
ekangen-te	111100	12082.203	74.910	8211.470	123.356	6290.991	84.451
liu-te	000000	374.500	32.144	249.667	47.672	187.250	46.165
liu-te	100000	749.000	34.441	624.167	50.780	561.750	48.912
liu-te	110000	15276.746	52.626	10371.083	67.656	7988.442	58.613
liu-te	111000	15276.746	57.014	10371.083	72.747	7988.442	60.249
liu-te	111100	15276.746	59.756	10371.083	114.275	7988.442	100.889
mantorp-a-te	000000	461.000	62.196	307.333	76.315	230.500	67.648
mantorp-a-te	100000	922.000	69.579	768.333	84.347	691.500	72.769
mantorp-a-te	110000	22652.726	138.439	15231.342	115.944	11531.157	90.020
mantorp-a-te	111000	22652.726	160.301	15231.342	138.801	11531.157	94.131
mantorp-a-te	111100	22654.241	203.272	15231.359	306.325	11531.157	145.395

Table 22: Solution of the LP-relaxation of several models for the city-instances 2a.

		Obj	Time	Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$		$q = 4$	
mantorp-b-te	000000	247.000	26.248	164.667	32.892	123.500	30.450
mantorp-b-te	100000	494.000	27.614	411.667	34.637	370.500	30.357
mantorp-b-te	110000	6731.980	42.282	4565.394	45.281	3484.988	34.388
mantorp-b-te	111000	6731.980	43.366	4565.394	46.538	3484.988	35.080
mantorp-b-te	111100	6732.431	46.646	4565.394	66.514	3484.988	44.058
mantorp-te	000000	608.000	88.745	405.333	89.005	304.000	92.373
mantorp-te	100000	1216.000	98.202	1013.333	98.148	912.000	102.441
mantorp-te	110000	28864.265	191.646	19463.149	155.109	14772.203	134.857
mantorp-te	111000	28864.265	207.309	19463.149	173.924	14772.203	197.327
mantorp-te	111100	28864.265	296.063	19463.149	323.681	14772.203	288.413
norsholm-te	000000	206.500	24.408	137.667	34.092	103.250	30.469
norsholm-te	100000	413.000	25.439	344.167	34.953	309.750	30.673
norsholm-te	110000	8064.621	35.417	5445.294	44.308	4150.195	33.836
norsholm-te	111000	8064.621	37.660	5445.294	48.127	4150.195	34.205
norsholm-te	111100	8068.919	40.469	5446.820	68.018	4150.195	40.452
rimforsa-a-te	000000	340.500	23.412	227.000	35.068	170.250	35.039
rimforsa-a-te	100000	681.000	24.117	567.500	37.315	510.750	35.809
rimforsa-a-te	110000	8403.568	35.290	5708.263	49.759	4369.174	41.174
rimforsa-a-te	111000	8403.568	34.903	5708.263	50.261	4369.174	42.173
rimforsa-a-te	111100	8403.895	39.007	5708.263	68.254	4369.174	55.076
rimforsa-te	000000	699.500	87.889	466.333	95.977	349.750	96.375
rimforsa-te	100000	1399.000	99.666	1165.833	109.438	1049.250	107.126
rimforsa-te	110000	31208.213	184.680	21020.719	169.615	15986.714	136.978
rimforsa-te	111000	31208.213	205.907	21020.719	171.942	15986.714	149.176
rimforsa-te	111100	31219.374	263.673	21025.594	513.988	15988.239	324.892
skanninge-a-1-te	000000	248.000	20.597	165.333	26.940	124.000	29.725
skanninge-a-1-te	100000	496.000	21.400	413.333	28.077	384.000	30.657
skanninge-a-1-te	110000	6871.909	28.602	4683.025	33.419	3721.734	34.701
skanninge-a-1-te	111000	6871.909	30.333	4683.025	34.470	3721.734	34.830
skanninge-a-1-te	111100	6871.909	34.155	4683.025	50.393	3721.734	42.230
skanninge-a-2-te	000000	139.500	15.895	93.000	21.715	69.750	21.435
skanninge-a-2-te	100000	279.000	15.920	232.500	22.222	209.250	21.847
skanninge-a-2-te	110000	4765.695	19.349	3209.957	25.595	2433.838	23.393
skanninge-a-2-te	111000	4765.695	21.467	3209.957	27.043	2433.838	23.582
skanninge-a-2-te	111100	4772.279	22.354	3216.262	35.025	2439.206	26.690
skanninge-a-te	000000	384.000	44.398	256.000	63.580	192.000	66.931
skanninge-a-te	100000	768.000	48.943	640.000	68.885	576.000	70.425
skanninge-a-te	110000	21886.431	88.212	14713.579	96.290	11136.916	88.588
skanninge-a-te	111000	21886.431	87.723	14713.579	103.025	11136.916	91.276
skanninge-a-te	111100	21886.431	144.641	14713.579	200.294	11136.916	165.566
skanninge-te	000000	434.500	60.192	289.667	78.622	217.250	71.106
skanninge-te	100000	869.000	66.334	724.167	86.811	651.750	77.177
skanninge-te	110000	23242.661	115.980	15614.769	124.784	11821.803	97.711
skanninge-te	111000	23242.661	151.027	15614.769	133.998	11821.803	101.813
skanninge-te	111100	23252.975	176.336	15614.769	414.686	11821.803	176.404
valla-te	000000	243.000	30.668	162.000	27.982	121.500	28.174
valla-te	100000	486.000	32.127	405.000	28.975	364.500	28.821
valla-te	110000	6258.864	52.823	4266.026	36.042	3292.059	32.306
valla-te	111000	6258.864	53.890	4266.026	36.004	3292.059	32.689
valla-te	111100	6263.658	55.546	4266.135	56.923	3292.059	40.680

Table 23: Solution of the LP-relaxation of several models for the mini-instances 2b.

		Obj	Time	Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$		$q = 4$	
vikingstad-a-2-te	000000	218.500	24.137	145.667	31.621	109.250	37.471
vikingstad-a-2-te	100000	437.000	25.234	364.167	32.878	327.750	38.957
vikingstad-a-2-te	110000	10383.269	33.665	6968.333	39.633	5266.970	44.223
vikingstad-a-2-te	111000	10383.269	38.612	6968.333	41.735	5266.970	44.889
vikingstad-a-2-te	111100	10387.527	41.311	6971.823	60.573	5270.378	57.849
vikingstad-a-te	000000	382.500	56.267	255.000	67.548	191.250	89.454
vikingstad-a-te	100000	765.000	62.335	637.500	74.010	573.750	75.041
vikingstad-a-te	110000	24495.216	108.854	16487.220	105.581	12525.144	107.507
vikingstad-a-te	111000	24495.216	123.187	16487.220	113.528	12525.144	103.657
vikingstad-a-te	111100	24501.701	125.531	16492.275	327.680	12530.188	158.146
vikingstad-te	000000	593.500	81.097	395.667	94.086	296.750	82.235
vikingstad-te	100000	1187.000	91.374	989.167	103.171	981.500	89.564
vikingstad-te	110000	25334.807	182.782	17234.345	161.394	15411.660	114.575
vikingstad-te	111000	25334.807	179.711	17234.345	170.966	15411.660	119.616
vikingstad-te	111100	25335.106	202.152	17234.345	403.570	15411.660	192.630

Table 24: Solution of the LP-relaxation of several models for the city-instances 2c.

		Obj	Time	Obj	Time
Network	Model	$q = 3$		$q = 4$	
atvid-a-te	000000	450.333	188.656	337.750	187.019
atvid-a-te	100000	1125.833	224.550	1013.250	218.578
atvid-a-te	110000	64246.588	588.783	48316.611	441.415
atvid-te	000000	516.333	187.774	387.250	492.094
atvid-te	100000	1290.833	228.150	1161.750	623.681
atvid-te	110000	64912.937	581.612	-	-
borensberg-te	000000	390.667	145.945	293.000	115.080
borensberg-te	100000	976.667	167.710	879.000	128.368
borensberg-te	110000	25111.171	294.185	18961.938	203.479
boxholm-te	000000	459.667	144.687	344.750	120.243
boxholm-te	100000	1149.167	162.000	1034.250	134.828
boxholm-te	110000	30201.938	257.857	22800.201	190.713

Table 25: Solution of the LP-relaxation of several models for the city-instances 3.

	MIP		LPZ		LPX		LP	
Model	Obj	Time	Obj	Time	Obj	Time	Obj	Time
000000	84.0	9.226	54.2	9.047	84.0	9.482	54.250	9.028
100000	192.5	9.147	190.5	9.132	192.5	9.444	190.500	9.080
010000	84.0	10.084	54.2	9.917	84.0	9.953	54.250	9.335
001000	84.0	9.095	54.2	9.134	84.0	9.545	54.250	9.013
000100	192.9	9.400	192.9	9.358	192.9	9.549	190.500	9.110
000010	84.0	13.846	54.2	12.989	84.0	13.827	54.250	13.441
000001	84.0	11.017	54.2	10.973	84.0	11.056	54.250	11.000
011000	84.0	10.119	54.2	9.975	84.0	9.854	54.250	9.429
010100	-	>1000	-	>1000	-	>1000	606.168	10.060
011100	-	>1000	-	>1000	-	>1000	606.168	10.154
110000	-	>1000	-	>1000	606.2	9.926	606.168	9.682
111000	-	>1000	-	>1000	606.2	10.179	606.168	9.745
111100	-	>1000	-	>1000	-	>1000	606.168	10.197
111110	-	>1000	-	>1000	-	>1000	608.725	31.251
111111	-	>1000	-	>1000	-	>1000	608.725	61.387

Table 26: Relaxations of integrality models for bankekind-te with $q = 4$.

	MIP		LPZ		LPX		LP	
Model	Obj	Time	Obj	Time	Obj	Time	Obj	Time
000000	54.0	7.560	53.2	7.499	54.0	7.441	53.250	7.400
100000	160.5	7.616	159.8	7.498	160.5	7.479	159.750	7.459
010000	54.0	8.180	53.2	8.016	54.0	7.825	53.250	7.573
001000	54.0	7.477	53.2	7.507	54.0	7.520	53.250	7.428
000100	161.8	7.491	161.8	7.500	161.8	7.511	159.750	7.480
000010	54.0	9.413	53.2	8.976	54.0	9.357	53.250	9.232
000001	54.0	8.697	53.2	8.696	54.0	8.601	53.250	8.568
011000	54.0	8.095	53.2	7.948	54.0	7.916	53.250	7.650
010100	513.8	489.005	513.8	500.115	-	>1000	505.287	8.011
011100	-	>1000	1770.3	1007.547	-	>1000	505.287	8.063
110000	514.2	164.516	513.7	148.997	506.6	8.244	505.287	7.781
111000	513.7	482.501	513.4	362.057	505.3	8.453	505.287	7.815
111100	513.6	778.358	513.6	791.189	-	>1000	505.287	8.184
111110	-	>1000	-	>1000	-	>1000	511.100	15.850
111111	-	>1000	-	>1000	-	>1000	511.100	25.411

Table 27: Relaxations of integrality in models for rimforsa-a-2-te with $q = 4$.

	MIP		LPZ		LPX		LP	
Model	Obj	Time	Obj	Time	Obj	Time	Obj	Time
000000	28.0	4.722	26.5	4.753	28.0	4.749	26.500	4.720
100000	81.0	4.870	79.5	4.729	81.0	4.885	79.500	4.683
010000	28.0	5.134	26.5	4.956	-	>1000	26.500	4.791
001000	28.0	4.722	26.5	4.706	28.0	4.745	26.500	4.687
000100	81.3	5.252	81.3	4.996	81.3	5.149	79.500	4.713
000010	28.0	5.680	26.5	5.501	28.0	5.752	26.500	5.584
000001	28.0	5.398	26.5	5.372	28.0	5.395	26.500	5.310
011000	28.0	5.233	26.5	5.006	-	>1000	26.500	4.800
010100	217.3	10.496	217.3	10.438	217.3	489.644	214.400	4.928
011100	217.3	55.863	217.3	55.887	217.3	215.201	214.400	4.984
110000	217.9	21.852	217.3	21.196	214.4	5.192	214.400	4.842
111000	217.3	10.575	217.3	12.008	214.4	5.131	214.400	4.877
111100	217.3	33.001	217.3	33.022	217.3	852.337	214.400	4.995
111110	222.4	71.285	222.4	71.062	220.8	721.813	218.400	7.644
111111	225.0	85.056	224.8	117.484	-	>1000	218.400	8.683

Table 28: Relaxations of integrality in models for vikingstad-a-3-te with $q = 4$.

in principle better, or not much worse, than LPZ. In some cases LPX gives something better than LP, in comparable time. But in other cases it simply takes too long.

A general conclusion is that if the complete MIP is too difficult, and the complete LP-relaxation is too weak, LPX can be considered. It can give better objective function value than LP and can be solved quicker than MIP. But it might also fail.

An alternative to LP-relaxation is to simply ignore some (more) constraints, but still solve the MIP. Therefore we need to investigate these possibilities.

Therefore, in the following tables, we solve various models (as MIP) of all the instances. Here we do not include the most difficult models, because we know that we will not be able to solve them in 1000 seconds. Our goal is to find a suitable subproblem in the following tables. Tables 29 - 32 contain the mini-instances, while tables 33 - 37 contain the easier city-instances.

In the future, we plan to use Lagrangian relaxation and subgradient optimization. Then we would need a rather easy subproblem, since it should be solved many times. Therefore, in tables 38 and 39, we give the results for solving the simplest subproblem, namely model 000000.

The conclusion of these tests is that it is perfectly feasible to solve this subproblem for all our instances (once). However, to do 100 iterations would for some instances be rather time consuming. Also, as noted earlier, the objective function values are rather low.

4.7 Summation of the results

We have found that the exact model M10 is too difficult to solve exactly for almost all instances. Therefore we must consider different relaxations. One possibility is to use LP-relaxation of all or some of the variables. Another possibility is to remove certain sets of constraints. There are many possibilities. The goal of course is to affect the objective function value as little as possible. (A third option is to use Lagrangian relaxation on different sets of constraints. This will be investigated in a forthcoming paper.)

We recall that our main goal is to get good estimations of the optimal objective function value, in reasonable time. We are here mainly interested in lower bounds, since we have not obtained such bounds in our previous work. Upper bounds are of course also interesting. However, we have in several previous papers described heuristics with the aim of obtaining good feasible solution, which yields good upper bounds. So this is not our main focus in this paper.

Our basic model is M14. We can also add the whole of model M10, with constraints connecting the x and I variables to the z variables. This model we call M1014.

Both M10 and M1014 are exact models, containing all aspects we need, and thus give the exact optimal objective function value. (We should however mention that we use

		Obj	Time	Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$		$q = 4$	
mini-05a	000000	14.0	0.185	10.0	0.084	8.0	0.126
mini-05a	100000	28.0	0.046	24.0	0.054	22.0	0.059
mini-05a	010000	14.0	0.065	10.0	0.075	8.0	0.082
mini-05a	001000	14.0	0.045	10.0	0.055	8.0	0.059
mini-05a	000100	28.0	0.046	24.0	0.058	22.2	0.062
mini-05a	000010	14.0	0.051	10.0	0.064	8.0	0.065
mini-05a	000001	14.0	0.069	10.0	0.083	8.0	0.083
mini-05a	011000	14.0	0.072	10.0	0.082	8.0	0.084
mini-05a	010100	70.7	0.163	53.7	0.484	45.9	0.185
mini-05a	011100	71.2	0.116	53.7	0.291	45.9	0.220
mini-05b	000000	18.0	0.112	12.0	0.112	10.0	0.164
mini-05b	100000	33.5	0.072	27.5	0.110	25.5	0.092
mini-05b	010000	18.0	0.100	12.0	0.153	10.0	0.138
mini-05b	001000	18.0	0.075	12.0	0.104	10.0	0.115
mini-05b	000100	33.8	0.083	27.8	0.105	26.0	0.113
mini-05b	000010	18.0	0.091	12.0	0.134	10.0	0.131
mini-05b	000001	18.0	0.109	12.0	0.163	10.0	0.170
mini-05b	011000	18.0	0.104	12.0	0.162	10.0	0.144
mini-05b	010100	87.0	0.603	64.2	1.211	53.4	0.491
mini-05b	011100	87.0	0.642	64.2	1.326	53.4	1.121
mini-05c	000000	20.0	0.195	14.0	0.112	10.0	0.198
mini-05c	100000	38.0	0.118	32.0	0.113	28.0	0.190
mini-05c	010000	20.0	0.170	14.0	0.159	10.0	0.270
mini-05c	001000	20.0	0.117	14.0	0.116	10.0	0.184
mini-05c	000100	38.2	0.150	32.2	0.124	28.2	0.194
mini-05c	000010	20.0	0.187	14.0	0.141	10.0	0.272
mini-05c	000001	20.0	0.193	14.0	0.179	10.0	0.317
mini-05c	011000	20.0	0.177	14.0	0.169	10.0	0.309
mini-05c	010100	109.4	1.622	78.4	1.691	63.4	1.899
mini-05c	011100	109.4	1.187	78.4	2.045	63.4	5.135
mini-05d	000000	20.0	0.106	14.0	0.250	10.0	0.191
mini-05d	100000	38.0	0.107	32.0	0.206	28.0	0.159
mini-05d	010000	20.0	0.146	14.0	0.275	10.0	0.212
mini-05d	001000	20.0	0.102	14.0	0.203	10.0	0.166
mini-05d	000100	38.2	0.142	32.2	0.218	28.2	0.171
mini-05d	000010	20.0	0.142	14.0	0.302	10.0	0.186
mini-05d	000001	20.0	0.163	14.0	0.325	10.0	0.238
mini-05d	011000	20.0	0.149	14.0	0.296	10.0	0.203
mini-05d	010100	109.4	0.708	78.4	3.844	63.4	1.982
mini-05d	011100	109.4	1.018	78.4	2.866	63.4	1.022

Table 29: MIP solution of several models for the mini-instances 1.

		Obj	Time	Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$		$q = 4$	
mini-06a	000000	36.0	0.342	24.0	0.473	18.0	0.372
mini-06a	100000	69.0	0.290	57.0	0.323	51.0	0.316
mini-06a	010000	36.0	0.438	24.0	0.446	18.0	0.453
mini-06a	001000	36.0	0.290	24.0	0.320	18.0	0.319
mini-06a	000100	69.6	0.296	57.6	0.335	51.6	0.337
mini-06a	000010	36.0	0.475	24.0	0.462	18.0	0.418
mini-06a	000001	36.0	0.484	24.0	0.525	18.0	0.504
mini-06a	011000	36.0	0.409	24.0	0.451	18.0	0.439
mini-06a	010100	324.2	84.747	226.4	32.593	180.1	29.655
mini-06a	011100	324.2	124.723	226.9	31.790	180.1	28.304
mini-06b	000000	34.0	0.307	23.0	0.355	18.0	0.347
mini-06b	100000	65.0	0.230	54.0	0.309	49.0	0.312
mini-06b	010000	34.0	0.327	23.0	0.452	18.0	0.411
mini-06b	001000	34.0	0.225	23.0	0.306	18.0	0.291
mini-06b	000100	65.5	0.228	54.5	0.323	49.8	0.310
mini-06b	000010	34.0	0.328	23.0	0.428	18.0	0.394
mini-06b	000001	34.0	0.360	23.0	0.502	18.0	0.494
mini-06b	011000	34.0	0.317	23.0	0.453	18.0	0.431
mini-06b	010100	288.9	11.861	204.1	14.662	163.1	46.570
mini-06b	011100	288.9	5.563	204.1	9.289	163.1	46.139
mini-06c	000000	38.0	0.440	25.0	0.619	19.0	0.406
mini-06c	100000	73.0	0.423	60.0	0.587	54.0	0.422
mini-06c	010000	38.0	0.593	25.0	0.964	19.0	0.512
mini-06c	001000	38.0	0.429	25.0	0.580	19.0	0.351
mini-06c	000100	73.5	0.417	60.5	0.642	54.5	0.364
mini-06c	000010	38.0	0.804	25.0	1.082	19.0	0.484
mini-06c	000001	38.0	0.726	25.0	0.972	19.0	0.586
mini-06c	011000	38.0	0.629	25.0	0.991	19.0	0.494
mini-06c	010100	357.2	200.444	246.6	132.045	195.6	411.663
mini-06c	011100	357.2	131.743	246.6	175.798	195.6	441.898

Table 30: MIP solution of several models for the mini-instances 2.

		Obj	Time	Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$		$q = 4$	
mini-08a	000000	37.0	0.456	25.0	0.522	19.0	0.580
mini-08a	100000	72.5	0.438	60.5	0.445	54.5	0.573
mini-08a	010000	37.0	0.617	25.0	0.608	19.0	0.820
mini-08a	001000	37.0	0.428	25.0	0.428	19.0	0.567
mini-08a	000100	72.7	0.482	60.7	0.518	54.7	0.627
mini-08a	000010	37.0	0.786	25.0	0.642	19.0	0.839
mini-08a	000001	37.0	0.731	25.0	0.728	19.0	0.940
mini-08a	011000	37.0	0.601	25.0	0.646	19.0	0.906
mini-08a	010100	425.9	10.512	291.9	186.915	226.5	243.029
mini-08a	011100	424.9	40.810	291.9	273.158	226.5	477.751
mini-08b	000000	32.0	0.288	21.0	0.379	16.0	0.399
mini-08b	100000	62.5	0.261	51.5	0.348	46.5	0.382
mini-08b	010000	32.0	0.350	21.0	0.436	16.0	0.487
mini-08b	001000	32.0	0.255	21.0	0.342	16.0	0.374
mini-08b	000100	62.7	0.267	51.7	0.368	46.8	0.420
mini-08b	000010	32.0	0.347	21.0	0.431	16.0	0.495
mini-08b	000001	32.0	0.407	21.0	0.531	16.0	0.605
mini-08b	011000	32.0	0.355	21.0	0.468	16.0	0.510
mini-08b	010100	324.8	6.860	225.7	4.944	180.2	10.346
mini-08b	011100	325.8	6.866	225.5	9.907	180.2	11.392
mini-08c	000000	34.0	0.325	23.0	0.382	17.0	0.479
mini-08c	100000	65.5	0.284	54.5	0.366	48.5	0.440
mini-08c	010000	34.0	0.393	23.0	0.488	17.0	0.576
mini-08c	001000	34.0	0.273	23.0	0.355	17.0	0.434
mini-08c	000100	65.9	0.304	54.9	0.430	48.9	0.473
mini-08c	000010	34.0	0.383	23.0	0.466	17.0	0.557
mini-08c	000001	34.0	0.437	23.0	0.565	17.0	0.671
mini-08c	011000	34.0	0.391	23.0	0.541	17.0	0.585
mini-08c	010100	344.7	7.815	239.2	17.413	190.9	110.099
mini-08c	011100	344.7	20.921	239.4	39.416	190.9	141.882

Table 31: MIP solution of several models for the mini-instances 3.

		Obj	Time	Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$		$q = 4$	
mini-09a	000000	66.0	1.848	44.0	1.856	33.0	1.745
mini-09a	100000	128.0	1.848	106.0	1.855	95.0	1.946
mini-09a	010000	66.0	3.085	44.0	2.495	33.0	2.339
mini-09a	001000	66.0	1.825	44.0	1.838	33.0	1.724
mini-09a	000100	128.7	1.842	106.7	2.033	95.8	1.716
mini-09a	000010	66.0	5.175	44.0	4.004	33.0	3.157
mini-09a	000001	66.0	3.009	44.0	3.061	33.0	2.999
mini-09a	011000	66.0	3.051	44.0	2.644	33.0	2.382
mini-09a	010100	-	>1000	-	>1000	-	>1000
mini-09a	011100	-	>1000	-	>1000	-	>1000
mini-09b	000000	51.0	0.579	34.0	0.712	26.0	0.895
mini-09b	100000	103.0	0.535	85.0	0.693	77.0	0.857
mini-09b	010000	51.0	0.696	34.0	0.870	26.0	1.031
mini-09b	001000	51.0	0.528	34.0	0.680	26.0	0.868
mini-09b	000100	103.0	0.535	85.0	0.690	77.2	0.871
mini-09b	000010	51.0	0.707	34.0	0.869	26.0	1.057
mini-09b	000001	51.0	0.811	34.0	1.016	26.0	1.273
mini-09b	011000	51.0	0.723	34.0	0.908	26.0	1.098
mini-09b	010100	-	>1000	594.7	162.885	466.7	100.812
mini-09b	011100	-	>1000	594.8	530.566	466.7	64.590
mini-09c	000000	71.0	2.202	47.0	2.510	36.0	2.873
mini-09c	100000	138.0	2.192	114.0	2.512	103.0	2.861
mini-09c	010000	71.0	3.324	47.0	3.269	36.0	3.915
mini-09c	001000	71.0	2.156	47.0	2.508	36.0	2.856
mini-09c	000100	138.7	2.238	114.7	2.627	103.7	3.455
mini-09c	000010	71.0	7.189	47.0	6.940	36.0	7.009
mini-09c	000001	71.0	3.704	47.0	4.295	36.0	4.759
mini-09c	011000	71.0	3.555	47.0	3.347	36.0	4.465
mini-09c	010100	-	>1000	-	>1000	-	>1000
mini-09c	011100	-	>1000	-	>1000	-	>1000

Table 32: MIP solution of several models for the mini-instances 4.

		Obj	Time	Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$		$q = 4$	
askeby-te	000000	217.0	6.211	144.0	7.943	124.0	8.220
askeby-te	100000	432.0	6.216	359.0	7.987	340.0	8.238
askeby-te	010000	216.0	7.838	144.0	9.159	124.0	8.915
askeby-te	001000	217.0	6.120	144.0	7.881	124.0	8.120
askeby-te	000100	432.6	6.199	359.5	7.912	339.5	8.117
askeby-te	000010	217.0	23.145	144.0	20.832	124.0	12.737
askeby-te	000001	217.0	8.461	144.0	10.817	124.0	9.844
askeby-te	011000	216.0	8.152	144.0	9.396	124.0	9.045
askeby-te	010100	-	>1000	-	>1000	-	>1000
askeby-te	011100	-	>1000	-	>1000	-	>1000
bankekind-te	000000	111.0	5.269	84.0	7.874	84.0	9.172
bankekind-te	100000	219.5	5.081	192.5	7.868	192.5	9.156
bankekind-te	010000	111.0	6.256	84.0	9.024	84.0	10.062
bankekind-te	001000	111.0	5.055	84.0	7.932	84.0	9.119
bankekind-te	000100	219.9	5.094	192.9	8.226	192.9	9.339
bankekind-te	000010	111.0	9.757	84.0	15.788	84.0	13.756
bankekind-te	000001	111.0	6.398	84.0	9.919	84.0	11.063
bankekind-te	011000	111.0	5.931	84.0	8.981	84.0	10.152
bankekind-te	010100	-	>1000	-	>1000	-	>1000
bankekind-te	011100	-	>1000	-	>1000	-	>1000
rimforsa-a-1-te	000000	246.0	17.468	164.0	25.671	148.0	20.271
rimforsa-a-1-te	100000	486.5	18.074	404.5	26.314	388.5	20.581
rimforsa-a-1-te	010000	245.0	21.706	164.0	31.852	148.0	24.639
rimforsa-a-1-te	001000	246.0	17.484	164.0	24.926	148.0	20.616
rimforsa-a-1-te	000100	487.6	17.416	405.5	24.927	390.5	20.687
rimforsa-a-1-te	000010	246.0	102.971	164.0	144.466	148.0	40.933
rimforsa-a-1-te	000001	246.0	25.049	164.0	36.800	148.0	26.000
rimforsa-a-1-te	011000	245.0	22.618	164.0	33.121	148.0	23.626
rimforsa-a-1-te	010100	-	>1000	-	>1000	-	>1000
rimforsa-a-1-te	011100	-	>1000	-	>1000	-	>1000
rimforsa-a-2-te	000000	108.0	5.526	72.0	7.010	54.0	7.721
rimforsa-a-2-te	100000	214.5	5.574	179.5	7.041	160.5	7.779
rimforsa-a-2-te	010000	108.0	7.312	72.0	7.845	54.0	8.426
rimforsa-a-2-te	001000	108.0	5.751	72.0	7.057	54.0	7.743
rimforsa-a-2-te	000100	215.8	5.518	178.8	6.984	161.8	7.709
rimforsa-a-2-te	000010	108.0	14.653	72.0	12.867	54.0	9.697
rimforsa-a-2-te	000001	108.0	7.393	72.0	8.867	54.0	8.952
rimforsa-a-2-te	011000	108.0	7.074	72.0	7.898	54.0	8.344
rimforsa-a-2-te	010100	-	>1000	-	>1000	513.8	548.519
rimforsa-a-2-te	011100	-	>1000	-	>1000	-	>1000

Table 33: MIP solution of several models for the easier city-instances 1.

		Obj	Time	Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$		$q = 4$	
ull1-te	000000	158.0	4.319	118.0	6.065	118.0	5.675
ull1-te	100000	317.0	4.076	275.0	6.179	275.0	5.658
ull1-te	010000	158.0	5.155	118.0	7.344	118.0	6.289
ull1-te	001000	159.0	4.074	118.0	6.108	118.0	5.647
ull1-te	000100	316.2	4.091	275.6	6.123	275.3	5.691
ull1-te	000010	158.0	14.590	118.0	20.271	118.0	9.664
ull1-te	000001	158.0	5.545	118.0	8.350	118.0	6.985
ull1-te	011000	158.0	5.114	118.0	7.769	118.0	6.467
ull1-te	010100	-	>1000	-	>1000	452.9	261.183
ull1-te	011100	-	>1000	-	>1000	453.4	41.780
ull2-te	000000	161.0	8.108	107.0	11.056	81.0	11.358
ull2-te	100000	321.0	8.152	267.0	10.928	240.0	11.371
ull2-te	010000	161.0	10.774	107.0	12.577	81.0	12.460
ull2-te	001000	161.0	8.144	107.0	10.992	81.0	11.353
ull2-te	000100	320.7	8.100	267.8	10.884	240.8	11.372
ull2-te	000010	161.0	25.140	107.0	25.954	81.0	16.349
ull2-te	000001	161.0	10.781	107.0	14.097	81.0	13.463
ull2-te	011000	161.0	10.131	107.0	12.709	81.0	12.643
ull2-te	010100	-	>1000	-	>1000	-	>1000
ull2-te	011100	-	>1000	-	>1000	-	>1000
valla-a-te	000000	140.0	14.715	93.0	19.256	70.0	21.492
valla-a-te	100000	272.5	15.132	225.5	18.762	202.5	20.172
valla-a-te	010000	139.0	19.373	93.0	22.461	70.0	23.048
valla-a-te	001000	140.0	14.590	93.0	18.970	70.0	21.229
valla-a-te	000100	273.9	14.771	226.8	19.932	203.9	21.276
valla-a-te	000010	140.0	43.333	93.0	38.186	70.0	29.235
valla-a-te	000001	140.0	19.544	93.0	24.218	70.0	25.053
valla-a-te	011000	139.0	19.669	93.0	22.869	70.0	23.243
valla-a-te	010100	-	>1000	-	>1000	-	>1000
valla-a-te	011100	-	>1000	-	>1000	-	>1000
vikingstad-a-1-te	000000	114.0	5.598	76.0	7.622	57.0	10.829
vikingstad-a-1-te	100000	224.5	5.527	186.5	7.663	168.5	10.132
vikingstad-a-1-te	010000	114.0	6.366	76.0	8.219	57.0	10.927
vikingstad-a-1-te	001000	114.0	5.544	76.0	7.566	57.0	10.776
vikingstad-a-1-te	000100	225.1	5.807	187.1	7.634	169.1	10.178
vikingstad-a-1-te	000010	114.0	10.060	76.0	10.812	57.0	13.330
vikingstad-a-1-te	000001	114.0	6.886	76.0	9.017	57.0	11.967
vikingstad-a-1-te	011000	114.0	6.350	76.0	8.302	57.0	10.991
vikingstad-a-1-te	010100	-	>1000	-	>1000	-	>1000
vikingstad-a-1-te	011100	-	>1000	-	>1000	-	>1000

Table 34: MIP solution of several models for the easier city-instances 2.

		Obj	Time	Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$		$q = 4$	
vikingstad-a-3b-te	000000	30.0	2.643	20.0	3.352	15.0	4.233
vikingstad-a-3b-te	100000	58.0	2.422	48.0	3.271	43.0	4.325
vikingstad-a-3b-te	010000	30.0	2.788	20.0	3.494	15.0	4.424
vikingstad-a-3b-te	001000	30.0	2.412	20.0	3.276	15.0	4.182
vikingstad-a-3b-te	000100	58.3	2.432	48.3	3.287	43.3	4.380
vikingstad-a-3b-te	000010	30.0	2.982	20.0	3.499	15.0	4.374
vikingstad-a-3b-te	000001	30.0	2.782	20.0	3.558	15.0	4.454
vikingstad-a-3b-te	011000	30.0	2.753	20.0	3.502	15.0	4.434
vikingstad-a-3b-te	010100	331.5	48.776	226.7	225.014	175.2	98.687
vikingstad-a-3b-te	011100	330.0	59.017	226.7	161.726	175.2	104.295
vikingstad-a-3-te	000000	55.0	2.800	37.0	3.512	28.0	4.762
vikingstad-a-3-te	100000	108.0	2.782	90.0	3.550	81.0	4.823
vikingstad-a-3-te	010000	55.0	3.350	37.0	3.722	28.0	5.092
vikingstad-a-3-te	001000	55.0	2.778	37.0	3.510	28.0	4.716
vikingstad-a-3-te	000100	108.3	2.797	90.3	4.292	81.3	5.180
vikingstad-a-3-te	000010	55.0	4.442	37.0	4.177	28.0	5.651
vikingstad-a-3-te	000001	55.0	3.403	37.0	3.981	28.0	5.360
vikingstad-a-3-te	011000	55.0	3.263	37.0	3.776	28.0	5.185
vikingstad-a-3-te	010100	396.9	447.965	279.9	724.135	217.3	10.430
vikingstad-a-3-te	011100	396.9	317.768	-	>1000	217.3	55.609

Table 35: MIP solution of several models for the easier city-instances 3.

		Obj	Time	Obj	Time
Network	Model	$q = 2$		$q = 3$	
grebo-te	000000	262.0	2.756	262.0	2.892
grebo-te	100000	509.0	2.733	513.0	2.913
grebo-te	010000	264.0	3.390	262.0	3.332
grebo-te	001000	262.0	2.767	262.0	2.927
grebo-te	000100	509.1	2.771	509.1	2.915
grebo-te	000010	262.0	15.226	262.0	8.116
grebo-te	000001	262.0	3.980	262.0	3.871
grebo-te	011000	264.0	3.605	262.0	3.416
grebo-te	010100	634.6	5.462	568.2	4.394
grebo-te	011100	631.5	6.874	568.2	6.658
ryde-te	000000	380.0	2.138	380.0	3.114
ryde-te	100000	619.0	2.138	621.0	3.112
ryde-te	010000	382.0	2.598	380.0	3.832
ryde-te	001000	380.0	2.111	380.0	3.134
ryde-te	000100	621.2	2.105	615.4	3.130
ryde-te	000010	380.0	7.577	380.0	11.101
ryde-te	000001	380.0	2.789	380.0	4.129
ryde-te	011000	382.0	2.659	380.0	4.011
ryde-te	010100	715.5	5.731	669.8	9.525
ryde-te	011100	714.0	5.561	668.7	9.511

Table 36: MIP solution of several models for the easier city-instances 4.

Network	Model	Obj	Time
sturefors-te	000000	551.0	24.792
sturefors-te	100000	1091.5	26.832
sturefors-te	010000	552.0	36.116
sturefors-te	001000	551.0	25.081
sturefors-te	000100	1094.2	25.308

Table 37: MIP solution of several models for sturefors-te, $q = 2$.

	Obj	Time	Obj	Time
Network	$q = 2$		$q = 3$	
mini-05a	14.0	0.078	-	-
mini-05b	18.0	0.069	-	-
mini-05c	20.0	0.113	-	-
mini-05d	20.0	0.101	-	-
mini-06a	36.0	0.325	-	-
mini-06b	34.0	0.252	-	-
mini-06c	38.0	0.438	-	-
mini-08a	37.0	0.470	25.0	0.454
mini-08b	32.0	0.273	21.0	0.376
mini-08c	34.0	0.294	23.0	0.384
mini-09a	66.0	1.861	44.0	1.945
mini-09b	51.0	0.566	34.0	0.703
mini-09c	71.0	2.173	47.0	2.557

Table 38: Solution of model M14 for the mini-instances.

a tolerance of 1% in Gurobi, which might give small differences in objective function values.)

We find that very few instances can be solved by M10 in less than 2000 seconds. Solving the LP-relaxation is however feasible for these small problems.

Trying to do the same with model M1014, which is larger than M10, we find that interestingly enough M1014 is easier to solve than M10. This does not hold for the LP-relaxation, but the integer problem seems to have a better structure. Probably the z variables give good opportunities for branching. In order to get this effect, however, the integrality should be hard to achieve. If most of the solution time is spent on solving the first LP, the advantage of M1014 over M10 will not appear.

The next question is what parts of the complete model to keep in the problem. In tests with the instances askeby-te and askeby, we find that for askeby-te many models were solved in around 10 seconds, while some were not solved in 1000 seconds. There seems to be a sharp rise in difficulty at a certain stage. The instance askeby, without tree elimination, is more difficult, but shows in principle the same behavior.

The objective function values vary, and since we are aiming for the best lower bound, the highest is best. Several of the models do not give a better value than 000000, even though some of them take more time to solve.

In this test, we find that 100000, i.e. including the demand constraints, yields a much higher objective function value than many of the others. Another similar possibility is 000100, which includes the coupling constraints between x and z. Since the model includes demand constraints for z, coupling z to x gives a similar effect as the demand constraints in x. Including one of the two last constraints sets is for this example not good, since it takes more time without giving any improvement of the objective function values. We also note that 010100 and 011100 are not useful, since they take too long to solve.

We also did similar, but more extensive tests with the instance bankekind-te. We find that there is a large difference in optimal objective function values. 000000 and many

	Obj	Time	Obj	Time	Obj	Time
Network	$q = 2$		$q = 3$		$q = 4$	
askeby-te	217.0	6.147	144.0	7.762	124.0	8.140
atvid-a-1-te	336.0	36.573	224.0	54.801	168.0	64.089
atvid-a-2-te	251.0	37.029	168.0	39.556	126.0	46.474
atvid-a-3-te	117.0	13.870	78.0	17.312	58.0	19.668
atvid-a-te	707.0	175.613	470.0	195.568	352.0	203.711
atvid-te	808.0	191.850	536.0	201.663	402.0	516.804
bankekind-te	111.0	5.149	84.0	7.897	84.0	9.330
borensberg-a-te	488.0	73.466	326.0	89.242	245.0	161.708
borensberg-b-te	362.0	41.965	242.0	61.413	181.0	70.927
borensberg-te	607.0	86.964	404.0	156.974	303.0	129.410
boxholm-1-te	328.0	47.046	218.0	52.422	164.0	55.491
boxholm-te	710.0	124.591	472.0	142.997	354.0	132.964
brokind-1-te	336.0	20.424	223.0	30.907	168.0	35.020
brokind-te	620.0	50.540	413.0	80.665	309.0	60.415
ekangen-1-te	281.0	12.792	188.0	16.417	141.0	18.926
ekangen-te	494.0	38.222	329.0	48.196	247.0	56.153
grebo-te	262.0	2.648	262.0	2.800	-	-
liu-te	382.0	32.386	255.0	47.794	191.0	46.070
mantorp-a-te	474.0	62.553	316.0	75.123	237.0	68.088
mantorp-b-te	252.0	25.502	169.0	31.724	126.0	29.569
mantorp-te	625.0	86.694	415.0	90.034	312.0	101.935
norsholm-te	216.0	24.753	144.0	35.598	108.0	30.357
rimforsa-a-1-te	246.0	16.327	164.0	23.661	148.0	19.712
rimforsa-a-2-te	108.0	5.388	72.0	6.830	54.0	7.557
rimforsa-a-te	349.0	22.687	232.0	34.896	175.0	34.495
rimforsa-te	715.0	87.938	476.0	96.879	357.0	98.159
ryde-te	380.0	2.101	380.0	3.001	-	-
skanninge-a-1-te	255.0	20.918	170.0	27.660	138.0	29.485
skanninge-a-2-te	146.0	15.333	97.0	22.056	73.0	22.374
skanninge-a-te	397.0	45.206	265.0	65.049	199.0	73.957
skanninge-te	449.0	59.559	299.0	82.826	224.0	77.865
sturefors-te	551.0	23.970	420.0	34.762	-	-
ull1-te	158.0	4.065	118.0	5.819	118.0	5.648
ull2-te	161.0	8.102	107.0	10.585	81.0	10.979
valla-a-te	140.0	13.958	93.0	18.626	70.0	21.524
valla-te	253.0	30.356	169.0	27.910	127.0	30.104
vikingstad-a-1-te	114.0	5.598	76.0	7.657	57.0	10.735
vikingstad-a-2-te	229.0	23.690	153.0	32.386	115.0	40.593
vikingstad-a-3b-te	30.0	2.499	20.0	3.306	15.0	4.258
vikingstad-a-3-te	55.0	2.857	37.0	3.545	28.0	4.818
vikingstad-a-te	399.0	56.006	266.0	70.393	200.0	76.074
vikingstad-te	610.0	85.978	407.0	101.521	390.0	84.692

Table 39: Solution of model 14 for the city-instances 3.

others yield 84, while 100000 and 000100 yield 192, and the best yield 580.

For this easy instance, one might aim for the value 580, which can be obtained in 15 seconds with 001100 or in 12 seconds with 101100. We may also get 563 in less than 10 seconds by 101000. On the other hand, several of the models were not solved after 1000 seconds.

Next we solve the LP-relaxation for different models for all instances. A general conclusion is that both solution time and objective function value differs very much. One should carefully choose the model, depending on the amount of time that is available, and depending on the size of the instance.

Taking one step up in difficulty, we then obtain MIP solutions of the various models, omitting the larger instances.

On the other hand, we also solve the easiest subproblem, 000000, i.e. M14, for all instances. We use previously found feasible solutions to get upper bounds on the number of time periods needed, which are used in the dimensioning of arrays. A few instances had no such information for $q = 4$, and were omitted.

Next we consider LP-relaxation in parts, and compare the following four cases: MIP, the integer problem, LPZ, where the z variables were treated as continuous, LPX, where the x variables were treated as continuous, and LP, where all variables are continuous. This was done for instances bankekind-t, rimforsa-a-2-te and vikingstad-a-3-te.

Obviously we expected LPZ and LPX to lie somewhere between MIP and LP. However, in some cases LPX became much harder than the others. We cannot really explain this, but one theory is that it is beneficial to branch on the x-variables in some cases.

Our general question was which of LPX and LPZ would be best, and if they give advantages compared to the pure MIP and LP. For bankekind-te, LPX performed better, i.e. gave the same objective function value as MIP, but in a slightly shorter time. This objective function value was much better than LP and LPZ. This difference, but less pronounced, were observed for some of the instances of rimforsa-a-2-te. Here LPZ gave results very close to MIP. For the instance vikingstad-a-3-te, LPX for models failed completely, i.e. took much longer time than the others. There seems to occur some kind of instability when only branching on the z variables.

Trying to make a general choice of what subproblem to chose, we found the following. For easy instances, we recommend 010100 or 110000. For somewhat harder instances, we recommend 100000 or 000100, and for the hardest 000000.

In practice, one might expect that one operator would be working with the same few areas from year to year. Then one could tune the parameters for these areas.

5 Conclusions and future research

We have described several different ways of obtaining bounds on the optimal objective function value for the urban snow removal problem, based on the MIP formulation. Both LP-relaxation of variables, and removal of constraint sets, are possible to use. Some alternatives were found to be better than others.

The results of this paper will be used in a forthcoming paper on Lagrangian relaxation and subgradient search. We also combine the solutions in this paper with primal heuristics that find better feasible solutions. Finally, all of this could be inserted in a branch-and-bound framework.

Bibliography

[1] Hajizadeh, R. and Holmberg, K., "A branch-and-dive heuristic for single vehicle snow removal", *Networks* 76/4 (2020) 509–521.

[2] Hajizadeh, R. and Holmberg, K., "Coordination of vehicles in urban snow removal", Research Report LiTH-MAT-R–2021/06–SE, Department of Mathematics, Linköping University, Sweden 2021.

[3] Hajizadeh, R. and Holmberg, K., "Urban snow removal: Tree elimination", Research Report LiTH-MAT-R–2022/01–SE, Department of Mathematics, Linköping University, Sweden 2022.

[4] Holmberg, K., "Urban snow removal: Modeling and relaxations", Research Report LiTH-MAT-R–2014/08–SE, Department of Mathematics, Linköping University, Sweden 2014.

[5] Holmberg, K., "Map matching by optimization", Research Report LiTH-MAT-R–2015/01–SE, Department of Mathematics, Linköping University, Sweden 2015.

[6] Holmberg, K., "On using OpenStreetMap and GPS for optimization", Research Report LiTH-MAT-R–2015/15–SE, Department of Mathematics, Linköping University, Sweden 2015.

[7] Holmberg, K., "Prepared test instances extracted from OpenStreetMap data using different network reductions", Research Report LiTH-MAT-R–2018/04–SE, Department of Mathematics, Linköping University, Sweden 2018.

[8] Holmberg, K., "The (over) zealous snow remover problem", *Transportation Science* 53 (2019) 867–881.

[9] Holmberg, K. "Networks extracted from OpenStreetMap" 2020. `http://www.kajh.se/vineopt/problemdata/osm/`.