A Comparison of Column-Generation Approaches to the Synchronized Pickup and Delivery Problem

Timo Gschwind[a]

[a] *Chair of Logistics Management, Johannes Gutenberg University Mainz, Jakob-Welder-Weg 9, D-55128 Mainz, Germany.*

Abstract

In the Synchronized Pickup and Delivery Problem (SPDP), user-specified transportation requests from origin to destination points have to be serviced by a fleet of homogeneous vehicles. The task is to find a set of minimum-cost routes satisfying pairing and precedence, capacities, and time windows. Additionally, temporal synchronization constraints couple the service times at the pickup and delivery locations of the customer requests in the following way: A request has to be delivered within prespecified minimum and maximum time lags (called ride times) after it has been picked up. The presence of these ride-time constraints severely complicates the subproblem of the natural column-generation formulation of the SPDP so that it is not clear if their integration into the subproblem pays off in an integer column-generation approach. Therefore, we develop four branch-and-cut-and-price algorithms for the SPDP based on column-generation formulations that use different subproblems. Two of these subproblems have not been studied before. We derive new dominance rules and labeling algorithms for their effective solution. Extensive computational results indicate that integrating either both types of ride-time constraints or only the maximum ride-time constraints into the subproblem results in the strongest overall approach.

Key words: vehicle routing, pickup and delivery, temporal synchronization, labeling algorithm, branch-and-cut-and-price

1. Introduction

In the family of one-to-one *Pickup-and-Delivery Problems* (PDPs), customer requests consist of transporting goods or people between paired origin and destination points: for each request a specific good or person has to be picked up at one location and to be transported to the corresponding delivery location. Typically, the task is to design a set of minimum-cost routes satisfying all customer requests subject to pairing and precedence, and other problem-specific constraints. For details on different PDP-variants we refer to the recent surveys (Berbeglia *et al.*, 2007; Cordeau *et al.*, 2008; Parragh *et al.*, 2008).

A well-studied one-to-one PDP is the *Pickup-and-Delivery Problem with Time Windows* (PDPTW) (e.g., Dumas *et al.*, 1991; Ropke and Cordeau, 2009; Baldacci *et al.*, 2011) in which vehicle routes must respect pairing and precedence, capacities, and time windows. In this article, we introduce the *Synchronized Pickup and Delivery Problem* (SPDP). It extends the PDPTW by imposing additional constraints that couple the service times at the pickup and delivery locations of the customer requests in the following way: A delivery node has to be serviced within prespecified *minimum* and *maximum* time lags (called *ride times*) after the service at the corresponding pickup node has been completed. Because both pickup and delivery are performed by the same vehicle, these additional constraints are temporal *intra-route synchronization* constraints. As a generalization of the PDPTW the SPDP is clearly $\mathcal{NP}$-hard.

As pointed out, e.g., by Dohn *et al.* (2011) or Drexl (2012), synchronization aspects are highly relevant in routing practice and there is a growing interest on *Vehicle Routing Problems* (VRPs) with synchronization

Email address: gschwind@uni-mainz.de (Timo Gschwind)

constraints in the research community. We see the SPDP as the prototypical VRP with temporal intra-route synchronization in the sense that synchronization takes place only within disjunctive pairs of nodes and that there are no other non-standard constraints present. In this respect, the development of an effective algorithm for solving the SPDP constitutes a central building block for the solution of richer VRPs with synchronization constraints.

A special case of the SPDP is the so-called *Dial-a-Ride Problem* (DARP) in which only a maximum ride time is specified for each pickup-and-delivery pair. The DARP mainly arises in door-to-door transportation services for school children, handicapped persons, or the elderly and disabled (see, e.g., Russell and Morrel, 1986; Madsen *et al.*, 1995; Toth and Vigo, 1997; Borndörfer *et al.*, 1997). In this context, maximum ride times are used to guarantee a certain service level by limiting the time a passenger is on board of the vehicle. A similar service-related use of maximum ride-time constraints is described by Plum *et al.* (2014) in the context of liner shipping service design. When there is a limit on the total working hours of drivers (Ceselli *et al.*, 2009) or when transporting perishable goods (Azi *et al.*, 2010), the time a vehicle is away from the depot has to be restricted. This can be modeled by imposing a maximum ride-time constraint on a dummy request originating and destinating at the depot. Similarly, one might want to have a limit on both the minimum and maximum duration of the routes in order to achieve an even work-distribution of the drivers.

Other applications of temporal intra-route synchronization in which minimum and maximum ride times are relevant include the planning of security guards where locations have to be inspected repeatedly within given time intervals (Bredström and Rönnqvist, 2008). There, no actual pickup at one location followed by a delivery at another location takes place. Instead, just a pairing and precedence relation between the services at the nodes forming a customer request is given. Similar planning problems arise in home health care, e.g., when patients have to be monitored by a nurse several times a day (Eveborn *et al.*, 2006; Rasmussen *et al.*, 2012). Note that in many health care applications, including those considered in Eveborn *et al.* (2006) and Rasmussen *et al.* (2012), it is not mandatory that the patients are always treated by the same nurse, i.e., these problems are of a more general nature than the one considered in this paper. The temporal aspects of this more general synchronization constraints are considered in (Dohn *et al.*, 2011). However, personnel consistency often plays an important role in health care problems (Rasmussen *et al.*, 2012; Kovacs *et al.*, 2014) so that it may be reasonable to have specific patients monitored by a single staff member only and, hence, to require pairing and precedence of the corresponding services.

The contributions of this paper are the following: First, we introduce the SPDP as the prototypical VRP with temporal intra-route synchronization. This problem has to the best of our knowledge not been considered before. Second, we develop four exact solution approaches to the SPDP based on column-generation formulations whose master programs are formulated on different sets of variables implying different subproblems. Two of these subproblems are considered for the first time in the literature. One of them is the natural subproblem of the SPDP, in which time windows as well as temporal intra-route synchronization with both minimum and maximum ride times have to be dealt with. In the other one, maximum ride-times are relaxed. We derive new dominance rules and labeling algorithms for their solution. The other subproblems are solved with algorithms proposed by Dumas *et al.* (1991) and Gschwind and Irnich (2014), respectively. Finally, to compare the strength of the different solution approaches, we report extensive computational results over a large number of test instances with different characteristics regarding the number of customer requests and the tightness of capacity, time-window, and minimum and maximum ride-time constraints. The analysis shows that integrating either both types of ride-time constraints or only the maximum ride-time constraints into the subproblem results in the strongest overall approach regarding the number of optimal solutions, computation times, and remaining integrality gap.

Integer column-generation methods have proven to be very successful in solving many VRP-variants including PDPs (e.g., Dumas *et al.*, 1991; Ropke and Cordeau, 2009; Baldacci *et al.*, 2011). The column-generation master programs of such approaches typically are extended set-partitioning models formulated on variables representing feasible routes for the problem at hand. These formulations provide stronger bounds compared to other formulations like, e.g., arc-flow formulations or extended set-partitioning models formulated on a relaxed set of variables, if the respective subproblem does not possess the integrality property (Lübbecke and Desrosiers, 2005). This is the case for many VRPs where the subproblems are typically Elementary Shortest-Path Problems with Resource Constraints (ESPPRC, Desaulniers *et al.*, 1998). However, the overall success of an integer column-generation approach for VRP-variants relies not only on strong bounds but also on the effective solution of the subproblem.

This is the main challenge when synchronization comes into play (Drexl, 2012). In the case of inter-route synchronization, additional constraints have to be included in the master programs (Desaulniers *et al.*, 1998). Because of the dual variables associated with these constraints, the resulting subproblems are highly complex (e.g., Christiansen and Nygreen, 1998; Ioachim *et al.*, 1999; Dohn *et al.*, 2011) and cannot be solved by standard dynamic-programming labeling algorithms. This is also true for intra-route synchronization where no additional linking constraints are necessary. There, the increased complexity of the subproblems is not caused by additional duals but by the synchronization constraints themselves, which may be hard to incorporate into the subproblem. For the DARP, e.g., Hunsaker and Savelsbergh (2002) have demonstrated that in the presence of time windows and maximum ride times checking the feasibility of a given route is intricate. Clearly, the effective generation of such routes within a column-generation approach is even more challenging.

In the case of intra-route synchronization, the complexity of the subproblems can be reduced by relaxing one or more types of constraints in the subproblem and handling them in the master programs instead (see, e.g., Ropke and Cordeau (2005) for the DARP or Cherkesly *et al.* (2014) for the PDPTW with LIFO Loading). The resulting easier-to-solve subproblems come at the cost of weaker lower bounds and, thus, larger branch-and-bound trees. Often, it is a priori not clear what is the best compromise between the strength of the CG formulation and the hardness of the subproblem.

The recent work of Gschwind and Irnich (2014) provides insights regarding this trade-off for the DARP: They proposed a branch-and-cut-and-price algorithm that handles all route constraints of the DARP in the subproblem which is solved by means of an effective labeling algorithm. In a computational study, they compared the strength of their approach to the branch-and-cut-and-price algorithm of Ropke and Cordeau (2005) that uses a subproblem in which the maximum ride-time constraints are relaxed. The results indicated that their approach significantly outperforms the algorithm of Ropke and Cordeau (2005) in terms of computation times and number of solved instances. However, they also tested their approach with a different labeling algorithm that uses a weaker dominance rule and observed that in this case the approach with the relaxed subproblem of Ropke and Cordeau (2005) shows the better overall performance. Decisive for the success of the approach using the stronger formulation, thus, is the availability of an effective pricing procedure for the harder subproblem.

Compared to the DARP, the additional presence of minimum ride times significantly complicates the natural subproblem of the SPDP. As a result, the dominance rule that we are able to derive for its solution is much weaker compared to those that can be used for the subproblems in which one or both types of ride-time constraints are relaxed. Therefore, we propose and compare the efficiency of four column-generation algorithms for the SPDP. Each algorithm uses a different subproblem: One that handles all route constraints of the SPDP, one that relaxes the minimum ride times, one that relaxes the maximum ride times, and one that relaxes both types of ride-time constraints.

The remainder of the paper is organized as follows. Section 2 defines the SPDP and presents column-generation formulations of it. The dominance rules and labeling algorithms we use for solving the different subproblems are detailed in Section 3. In Section 4, we briefly describe our basic branch-and-cut-and-price algorithm and report extensive computational results. The paper ends with a short conclusion.

2. Problem definition and column-generation formulations

In this section, we give a formal definition of the SPDP and describe different column-generation formulations of it.

2.1. Definition of the SPDP

The SPDP is defined on a directed graph $G = (N, A)$ with node set $N = P \cup D \cup \{0, 2n + 1\}$ and arc set A. The subsets $P = \{1, ..., n\}$ and $D = \{n + 1, ..., 2n\}$ contain the pickup and delivery nodes of n transportation requests, respectively. Node 0 denotes the origin depot and node $2n + 1$ the destination depot. For each request $i = 1, ..., n$, a minimum ride time $\underline{L}_i$ and a maximum ride time $\overline{L}_i$ are specified, coupling the service times at the pickup node i and the delivery node $i + n$.

With each node $i \in N$, a non-negative service duration s_i and a demand d_i such that $d_i = -d_{i+n}$ for all $i = 1, ..., n$ are associated. We assume $d_0 = d_{2n+1} = 0$. A time window $[a_i, b_i]$ in which the service has to be started is associated with each node $i \in N$. When arriving at node i prior to a_i, the vehicle has to wait until time a_i before starting its service. Furthermore, it is allowed to delay the start of service voluntarily at any node and any time. We assume that there is no restriction on the length of the waiting times. The possibility of delaying the start of service at some nodes is crucial for the feasibility of routes in the presence of ride times and time windows (see Hunsaker and Savelsbergh, 2002; Gschwind and Irnich, 2014).

With each arc $(i, j) \in A$, a routing cost c_{ij} and a travel time t_{ij} are associated. We assume that both routing costs and travel times are non-negative and satisfy the triangle inequality. To serve the n transportation requests, a fleet K of identical vehicles with capacity C is located at the depot 0.

The SPDP consists in finding $|K|$ vehicle routes starting and ending at the depot nodes 0 and $2n + 1$, respectively, such that each request is served exactly once and the total routing costs are minimal. Thereby, the routes have to satisfy the following conditions:

Pairing and precedence: For each request i, pickup node i and delivery node $i + n$ are visited on the same route, and the pickup node i is visited first.

Capacity: The load of the vehicle must not exceed C at any time.

Time windows: For each node i, the start of service must lie within the time window $[a_i, b_i]$.

Ride times: The service at a delivery node $i+n$ has to start at least $\underline{L}_i$ and at most $\overline{L}_i$ units of time after the service at the corresponding pickup node i has been completed.

Note that it is not straightforward to decide on the feasibility of a route in the SPDP sense due to the presence of the different types of potentially contrasting temporal constraints. More precisely, to verify the feasibility of a given route $r = (h_1, ..., h_q)$ with $h_1 = 0$ and $h_q = 2n + 1$ one has to find a time schedule $T_r = (\tau_1, ..., \tau_q)$ satisfying

$$\tau_i + s_{h_i} + t_{h_i h_{i+1}} \leq \tau_{i+1} \quad \forall i = 1, ..., q - 1, \tag{1}$$

$$a_{h_i} \leq \tau_i \leq b_{h_i} \quad \forall i = 1, ..., q, \tag{2}$$

$$\tau_i + s_{h_i} + \underline{L}_{h_i} \leq \tau_j \quad \text{if } h_i + n = h_j, \tag{3}$$

$$\tau_i + s_{h_i} + \overline{L}_{h_i} \geq \tau_j \quad \text{if } h_i + n = h_j, \tag{4}$$

where τ_i denotes the start of service at node h_i. Constraints (1) ensure consistency of the service times along the route. Inequalities (2) impose time windows, while (3) and (4) are minimum and maximum ride-time constraints, respectively. A schedule satisfying (1)–(4) is called feasible. We denote by $\mathscr{T}_r$ the set of all feasible schedules for a route r. Furthermore, let $\mathscr{T}_r(t) = \{T_r \in \mathscr{T}_r : \tau_q \leq t\}$ be the set of feasible schedules with start of service τ_q at the last node h_q not later than t. All definitions and notations for routes and schedules are also used for partial routes, i.e., with $h_q \neq 2n + 1$, and corresponding partial schedules. Note that for partial routes it is possible to visit only the pickup node of a request i. In this case, no ride-time constraints (3) or (4) have to be respected for this request in a partial schedule.

2.2. Column-generation formulations of the SPDP

To formulate the SPDP as a set-partitioning problem, let Ω be the set of all SPDP-feasible routes. The cost of a route $r \in \Omega$ is denoted by c_r. Moreover, for each route r and each request $i \in P$ denote by $a_{ir} \in \mathbb{Z}$ the number of times request i is performed by route r. Let λ_r be binary variables indicating if route r is used in the solution. The SPDP can then be formulated as follows:

$$(IMP) \quad \min \sum_{r \in \Omega} c_r \lambda_r \tag{5}$$

$$\text{s.t.} \sum_{r \in \Omega} a_{ir} \lambda_r = 1 \quad \forall i \in P, \tag{6}$$

$$\sum_{r \in \Omega} \lambda_r = |K|, \tag{7}$$

$$\lambda_r \in \{0, 1\} \quad \forall r \in \Omega. \tag{8}$$

The objective function (5) minimizes the total routing costs. Partitioning constraints (6) ensure that all requests are served exactly once. Equality (7) imposes the number of routes in the solution, while (8) are binary conditions for route variables.

Typically, the number $|\Omega|$ of feasible routes is very large so that model *IMP* cannot be solved directly. We, therefore, use an integer column-generation approach to solve it. The linear relaxation of the so-called integer master program *IMP* is initialized with a proper subset of routes and missing routes with negative reduced cost are added dynamically. Integrality is ensured by integrating this process into a branch-and-bound algorithm (Lübbecke and Desrosiers, 2005).

To identify negative reduced-cost routes the column-generation subproblem has to be solved. Let $\pi_i, i \in P$ and μ be the dual variables associated with constraints (6) and (7), respectively. The reduced cost of arc $(i, j) \in A$ is defined as

$$\tilde{c}_{ij} = \begin{cases} c_{ij} - \pi_i & \text{if } i \in P, \\ c_{ij} & \text{otherwise.} \end{cases} \tag{9}$$

The reduced cost $\tilde{c}_r$ of a route $r \in \Omega$ is $\tilde{c}_r = \sum_{(i,j) \in A(r)} \tilde{c}_{ij} - \mu$ where $A(r)$ denotes the sequence of arcs traversed by route r. The subproblem is then given by

$$\min_{r \in \Omega} \{\tilde{c}_r\}. \tag{10}$$

The set-partitioning model *IMP* is the most natural formulation for column-generation based approaches to VRP-variants including the SPDP in the following sense: The variable set Ω consists of all routes that are feasible for the problem at hand, i.e., the subproblem takes care of all constraints relating to single routes, while the master program comprises only coupling constraints. Decisive for the success of approaches based on such a formulation is that an effective solution procedure for generating feasible routes with negative reduced cost is available. For the SPDP, this means being able to simultaneously handle time-window and ride-time constraints, among others, that together impose a complex scheduling problem. The key difficulty lies in the trade-off between servicing nodes as early as possible, which is the best strategy for time windows and minimum ride times, and servicing them as late as possible, which is preferable for maximum ride times. Integrating this trade-off into a solution procedure is highly intricate and, therefore, the natural subproblem of the SPDP is significantly more complex than the natural subproblem of many related VRPs (see also Section 3.4).

An alternative approach is to formulate the master program in relaxed routing variables $r \in \Omega' \supseteq \Omega$ that may violate one or several types of constraints relating to a single route. This can be promising when generating routes $r \in \Omega$ is complex, while working with the relaxed set Ω' results in a well-solvable subproblem. A relaxation that is used in many column-generation approaches to VRPs is to drop the elementarity condition of routes, i.e., to allow multiple visits at the same node. This has the advantage that the resulting subproblems are solvable in pseudo-polynomial time (Desrochers *et al.*, 1992) while the original elementary versions are $\mathcal{NP}$-hard in the strong sense (Dror, 1994). In this case, the partitioning constraints (6) ensure that non-elementary routes can never be part of an integer solution. Thus, *IMP* with variable set Ω' has the same set of optimal solutions as *IMP* with variable set Ω.

This property, however, does not hold for all relaxations Ω' of Ω and a route $r \in \Omega' \setminus \Omega$ that is infeasible for the original problem might then be part of an integer solution. Consequently, the constraints that have been relaxed in the subproblem must be enforced in the master program to ensure feasibility of the solutions. Adding infeasible path elimination constraints (IPEC) is one way of doing this. Let $\mathcal{I}$ be the set of all paths that are infeasible with respect to the constraints that have been relaxed in the subproblem. Moreover, for any infeasible path $I \in \mathcal{I}$ and any route r let b_{Ir} be the number of times route r traverses arcs of path I. The IPEC can then be written as

$$\sum_{r \in \Omega'} b_{Ir} \lambda_r \leq |I| - 1 \quad \forall I \in \mathcal{I}, \tag{11}$$

where $|I|$ denotes the length of the infeasible path I, i.e., the number of its arcs. We denote by *IMP-I* a master program that incorporates the set-partitioning model (5)–(8) formulated on a relaxed variable set Ω' together with the IPEC (11) to handle the remaining route constraints.

Obviously, approaches based on formulation *IMP* benefit from stronger LP-bounds compared to approaches using formulation *IMP-I*. The reason is that SPDP-infeasible routes, which are excluded in the former, may be convex-combined to form routes that do not violate the IPEC in the latter. Typically, the tighter LP-bounds lead to smaller search trees for *IMP*-based approaches. This comes at the cost of a harder to solve subproblem and it is a priori not clear which formulation enables the overall strongest algorithm.

In the following sections, we consider branch-and-price algorithms for the SPDP based on four different column-generation formulations. We denote by IMP^{max}_{min} the approach working on variable set Ω in the master program. The addition of IPEC is not necessary in this case and the master program comprises only coupling constraints. The corresponding subproblem SP^{max}_{min} has to generate SPDP-feasible routes.

The other approaches formulate their master programs in routing variables $r \in \Omega'$ that relax either the minimum ride times, the maximum ride times, or both. By *IMP-I* we denote the algorithm that ignores both minimum and maximum ride times in the subproblem (denoted *SP*) and handles them using IPEC in the master program. *SP* is the natural subproblem of the PDPTW and generates routes that respect pairing and precedence, capacity, and time-window constraints, i.e., a time schedule satisfying constraints (1) and (2) exists for such routes.

The approach that handles only the maximum ride times in the subproblem and uses routing variables where the minimum ride times have been relaxed is denoted by $IMP\text{-}I^{max}$. The corresponding subproblem is SP^{max}. It is the natural subproblem of the DARP. Routes generated by SP^{max} satisfy pairing and precedence, and capacity constraints. Moreover, these routes can be assigned a time schedule respecting constraints (1), (2), and (4). $IMP\text{-}I_{min}$ and SP_{min} are the analog to $IMP\text{-}I^{max}$ and SP^{max} where minimum ride rimes are handled in the subproblem.

3. Column-generation subproblems

In this section, we describe solution algorithms for the different subproblems *SP*, SP^{max}, SP_{min}, and SP^{max}_{min}. All four subproblems are ESPPRC which are typically solved using dynamic-programming labeling algorithms (Irnich and Desaulniers, 2005). In a labeling algorithm, partial paths are gradually extended in a graph G seeking to find a minimum-cost path from the source node to the sink node. The partial paths are represented by labels that store the accumulated cost and resource consumption along the path. We denote by $\mathcal{P}_\ell$ the partial path corresponding to label ℓ. Decisive for the effectiveness of a labeling algorithm is the use of strong dominance rules to eliminate unpromising labels. A more detailed discussion on ESPPRC and labeling algorithms can be found, e.g., in (Irnich and Desaulniers, 2005).

Note that for the rest of this paper we consider the non-elementary versions of the four subproblems for the following two reasons: First, preliminary computational results indicated that the linear-relaxation lower bounds of the master programs obtained by subproblems with the elementarity conditions were rarely stronger compared to the corresponding non-elementary subproblems. This resulted in slightly weaker overall algorithms for the former. Second, the extension of all dominance rules and labeling algorithms to the elementary case is straightforward (see Ropke and Cordeau, 2009; Gschwind and Irnich, 2014). In the presence of pairing and precedence, non-elementarity means that a request can be picked up again, after it has been picked up and delivered. Hence, several pickup-and-delivery pairs of the same request can be present in a path. For ease of notation, however, we assume for the rest of the paper that all partial paths are elementary. Furthermore, we assume that the service duration is zero for all nodes. All proofs and arguments are analog when considering non-elementary partial paths and non-zero service durations.

Also, we assume that the reduced-cost matrix satisfies $\tilde{c}_{ij} \leq \tilde{c}_{ik} + \tilde{c}_{kj}$ for all $(i, j) \in A, k \in D$. Ropke and Cordeau (2009) call this property the *delivery triangle inequality* (DTI). It enables the use of stronger dominance rules for all considered subproblems. Roughly speaking, the DTI ensures that visiting an additional delivery node is never beneficial. Ropke and Cordeau (2009) also show how to transform a reduced-cost matrix that does not satisfy the DTI into one that does, while maintaining the cost of each route unchanged. Hence, working with this assumption is no restriction.

All notation previously introduced for (partial) routes is also used for (partial) paths in the following. Moreover, we use the same notation for all subproblems. The meaning should be clear from the context. The set of feasible schedules $\mathscr{T}_{\mathcal{P}}$ for a path $\mathcal{P}$, e.g., always refers to feasibility regarding the temporal constraints that are present in the considered subproblem.

3.1. *SP - subproblem without ride-time constraints*

SP is an elementary shortest-path problem with pairing and precedence, capacities, and time windows. It is the natural subproblem of the PDPTW. In this context, it has been subject to prior research (Dumas *et al.*, 1991; Ropke and Cordeau, 2009; Baldacci *et al.*, 2011) and strong dominance rules exist for its solution by a labeling algorithm.

In what follows, we summarize the main concepts of Dumas *et al.* (1991) and Ropke and Cordeau (2009) for solving *SP*. Both the dominance rule and the labeling strategy for *SP* also serve as basis for the solution approaches to the other subproblems in Sections 3.2 - 3.4. Table 1 summarizes all resources that are needed in the solution algorithms for the different subproblems and indicates which resource is relevant for which subproblem.

Dominance rule for SP. Within each label ℓ, the following information has to be stored: the node η_ℓ the label belongs to, its reduced cost $\tilde{c}_\ell$, the earliest start of service t_ℓ at node η_ℓ, and the set of open requests O_ℓ (requests that have been picked up but not yet delivered). Then, the following dominance rule is valid for *SP* (Dumas *et al.*, 1991):

Proposition 1. *(Dom-SP) A feasible label ℓ_1 dominates a label ℓ_2 if*

$$\eta_{\ell_1} = \eta_{\ell_2}, \quad \tilde{c}_{\ell_1} \leq \tilde{c}_{\ell_2}, \quad t_{\ell_1} \leq t_{\ell_2}, \quad O_{\ell_1} \subseteq O_{\ell_2}. \tag{12}$$

Labeling algorithm for SP. We now briefly describe the labeling algorithm of Dumas *et al.* (1991) for solving *SP*. In addition to the resources needed for dominance in Proposition 1, they store at each label ℓ the load l_ℓ of the vehicle when leaving η_ℓ, enabling a fast consistency check regarding capacity. The extension of a label ℓ along arc $(\eta_\ell, x) \in A$ is only allowed if either $x \notin O_\ell$ if $x \in P$, or $x - n \in O_\ell$ if $x \in D$, or $O_\ell = \varnothing$ if $x = 2n + 1$ holds. Otherwise, pairing and precedence are not satisfied resulting in an infeasible label. Furthermore, consistency with respect to time-window and capacity constraints is ensured by requiring $t_\ell + t_{\eta_\ell, x} \leq b_x$ and $l_\ell + d_x \leq C$, respectively.

If extending label ℓ along arc $(\eta_\ell, x) \in A$ is feasible, a new label ℓ' is created. Its resources are determined according to the following resource extension functions (REFs):

$$\eta_{\ell'} = x, \quad \tilde{c}_{\ell'} = \tilde{c}_\ell + \tilde{c}_{\eta_\ell, x}, \quad t_{\ell'} = \max\{a_x, t_\ell + t_{\eta_\ell, x}\}, \quad l_{\ell'} = l_\ell + d_x, \tag{13}$$

$$O_{\ell'} = \begin{cases} O_\ell \cup \{x\} & \text{if } x \in P, \\ O_\ell \setminus \{x - n\} & \text{if } x \in D. \end{cases} \tag{14}$$

To reduce the number of labels that have to be processed in the algorithm, unpromising labels are eliminated using dominance rule *Dom-SP*. Moreover, labels that cannot be feasibly completed to node

Resource	Description	SP	SP^{max}	SP_{min}	SP^{max}_{min}
η_ℓ	The node of the label	•	•	•	•
$\tilde{c}_\ell$	The reduced cost	•	•	•	•
t_ℓ	The earliest start of service at the node η_ℓ	•	•	•	•
l_ℓ	The current load	•	•	•	•
O_ℓ	The set of open requests	•	•	•	•
$\tilde{b}_\ell$	The latest feasible start of service at node η_ℓ		•		•
$ld^i_\ell(t)$	The latest possible delivery time of request $i \in O_\ell$		•		•
B^i_ℓ	The point of time when $ld^i_\ell(t)$ becomes constant		•		•
$\underline{ld}^i_\ell(t)$	The latest possible delivery time of request $i \in O_\ell$ such that all other open requests $j \in O_\ell \setminus \{i\}$ are picked up as early as possible				•
$\underline{B}^i_\ell$	The point of time when $\underline{ld}^i_\ell(t)$ becomes constant				•
ed^i_ℓ	The earliest feasible start of service at node η_ℓ			•	•
$\overline{ed}^i_\ell(t)$	The earliest possible delivery time of request $i \in O_\ell$ such that all other open requests $j \in O_\ell \setminus \{i\}$ are picked up as late as possible				•
$\overline{B}^i_\ell$	The point of time when $\overline{ed}^i_\ell(t)$ becomes constant				•

Table 1: Resources of a label ℓ. A bullet indicates that the resource is relevant for the respective subproblem.

$2n+1$ can be discarded. For *SP*, pairing constraints require that each feasible completion to a label ℓ must visit the delivery nodes $i+n$ of all open requests $i \in O_\ell$ and thereby obey all time-window constraints (see Dumas *et al.*, 1991, for details).

3.2. SP^{max} - subproblem with maximum ride-time constraints

In SP^{max}, the natural subproblem of the DARP, paths have to respect pairing and precedence, capacities, time windows, and maximum ride times. The latter two impose that for each feasible path a time schedule satisfying inequalities (1), (2), and (4) must exist. The main difficulty for solution approaches to SP^{max} is to deal with these partially contrasting temporal constraints. In fact, they impose a trade-off between servicing all nodes as early as possible and servicing pickup nodes as late as possible. The implication for labeling algorithms is as follows. Considering only the earliest start of service (as in *Dom-SP*) is not sufficient to guarantee dominance with respect to the temporal constraints of SP^{max} (see Example 1 of Gschwind and Irnich, 2014). Thus, one either has to include additional time-related resources in a dominance rule based on *Dom-SP* or come up with a different strategy to deal with the temporal constraints of SP^{max}.

Recently, Gschwind and Irnich (2014) proposed an effective labeling algorithm for solving SP^{max} that uses an extended version of *Dom-SP* as dominance rule. The basic idea is the following: Let ℓ be a label with $O_\ell \neq \varnothing$. For each open request $i \in O_\ell$, the corresponding maximum ride-time constraint (4) imposes an upper bound on the start of service at delivery node $i+n$ restricting the set of feasible completions to ℓ. Clearly, a larger value for this bound is preferable. As a result, dominance between two labels is only possible if for each of its open requests the dominating label has a larger upper bound value for the start of service at the respective delivery node. Determining these bounds, however, is not straightforward. They obviously depend on the actual service times at the corresponding pickup nodes within the path $\mathcal{P}_\ell$. Thereby, the possibility to delay the start of service at some nodes has to be incorporated.

Dominance rule for SP^{max}. To formalize their approach in a dominance rule, Gschwind and Irnich (2014) first define the *latest possible delivery time* ld^i_ℓ, i.e., the latest feasible start of service at the delivery node, of an open request $i \in O_\ell$ as a function in the start of service $t \geq t_\ell$ at the current node η_ℓ. Let $\mathcal{P}_\ell =$

$(h_1, ..., h_q = \eta_\ell)$ be the path corresponding to label ℓ. Then, $ld_\ell^{h_i}(t)$ with $t \geq t_\ell$ and $h_i \in O_\ell$ is given by

$$ld_\ell^{h_i}(t) = \min\{b_{h_i+n}, \overline{\tau}_i(t) + \overline{L}_{h_i}\}, \tag{15}$$

where $\overline{\tau}_i(t) = \max_{T_{\mathcal{P}_\ell} \in \mathscr{T}_{\mathcal{P}_\ell}(t)}\{\tau_i\}$ is the latest feasible start of service at the pickup node h_i while $\tau_q \leq t$.

Moreover, two important properties of $ld_\ell^{h_i}(t)$ are proven. First, they show that all open requests and the associated latest delivery times can be treated independently in the dominance criterion. Second, they show that the functions $ld_\ell^{h_i}(t)$ are of the form $ld_\ell^{h_i}(t) = \min\{k_1^i + t, k_2^i\}$ with constants k_1^i and k_2^i. With this property, the comparison of two such functions can be simplified to comparing them at two distinct points of time.

Let B_ℓ^i be the point of time when $ld_\ell^i(t)$ becomes constant. The following proposition describes a valid dominance rule for SP^{max} (Gschwind and Irnich, 2014):

Proposition 2. *(Dom-SP^{max}) A feasible label ℓ_1 dominates a label ℓ_2 if*

$$\eta_{\ell_1} = \eta_{\ell_2}, \quad \tilde{c}_{\ell_1} \leq \tilde{c}_{\ell_2}, \quad t_{\ell_1} \leq t_{\ell_2}, \quad O_{\ell_1} \subseteq O_{\ell_2}, \text{ and} \tag{16}$$

$$ld_{\ell_1}^i(t_{\ell_1}) + (t_{\ell_2} - t_{\ell_1}) \geq ld_{\ell_2}^i(t_{\ell_2}) \quad \text{and} \quad ld_{\ell_1}^i(B_{\ell_1}^i) \geq ld_{\ell_2}^i(B_{\ell_2}^i) \quad \forall i \in O_{\ell_1}. \tag{17}$$

Labeling algorithm for SP^{max}. The labeling algorithm of Gschwind and Irnich (2014) for solving SP^{max} is analog to that of Dumas *et al.* (1991) for *SP* sketched in Section 3.1. The additional presence of maximum ride times and the use of dominance rule *Dom-SP^{max}* involve only some minor modifications (see Gschwind and Irnich, 2014, for details). A key factor for the effectiveness of the labeling algorithm is that the information on the latest possible delivery times $ld_\ell^i, i \in O_\ell$ significantly enhances the elimination of labels that cannot be feasibly completed to node $2n + 1$.

3.3. SP_{min} - subproblem with minimum ride-time constraints

Subproblem SP_{min} is an elementary shortest path problem with pairing and precedence, capacity, time-window, and minimum ride-time constraints. To the best of our knowledge, SP_{min} has not been considered before and an effective labeling algorithm for its solution is presented here for the first time.

Similar to SP^{max}, different types of temporal constraints are present in SP_{min}. More precisely, a schedule satisfying inequalities (1)–(3) must be assignable to each feasible path. The main task for a labeling approach to SP_{min} based on *Dom-SP* is to ensure consistency of the dominance rule with these constraints.

In contrast to SP^{max}, however, the temporal constraint system of SP_{min} is rather straightforward to handle in a labeling algorithm. Both types of constraints that couple the service times at two different nodes are less or equal constraints (from front to back of the path). Consequently, the optimal strategy regarding time-window constraints, i.e., servicing all nodes as early as possible, is also an optimal strategy in the additional presence of minimum ride-time constraints. This implies that waiting and delaying the service at some node is never beneficial and the possibility to do so can be neglected. Still, inequalities (3) induce that a time schedule in SP_{min} is linked not only between consecutive nodes. Thus, for a label ℓ not only the service time at the current node η_ℓ, but also the service times at the pickup nodes of all open requests $i \in O_\ell$ are important.

Dominance rule for SP_{min}. To obtain a formal dominance criterion for SP_{min}, we follow the approach of Gschwind and Irnich (2014) for SP^{max}. For each open request $i \in O_\ell$ of label ℓ, the minimum ride-time constraints impose a lower bound on the start of service at the delivery node $i + n$. We define this *earliest possible delivery time* $ed_\ell^{h_i}$ for request $h_i \in O_\ell$ as

$$ed_\ell^{h_i} = \max\{a_{h_i+n}, t_\ell + t_{\eta_\ell, h_i+n}, \underline{\tau}_i + \underline{L}_{h_i}\}, \tag{18}$$

where $\underline{\tau}_i = \min_{T_{\mathcal{P}_\ell} \in \mathscr{T}_{\mathcal{P}_\ell}}\{\tau_i\}$ with $\mathcal{P}_\ell = (h_1, ..., h_q = \eta_\ell)$ is the earliest feasible start of service at the pickup node h_i. Regarding the set of feasible completions to ℓ, a small value $ed_\ell^{h_i}$ is obviously less restrictive than a larger one. Furthermore, the following lemma shows that for each feasible path $\mathcal{P}$ the time schedule $\underline{T}_{\mathcal{P}}$ which assigns each node its earliest feasible start of service is feasible. Thus, the values $ed_\ell^{h_i}$ can be treated independently in a dominance rule.

Lemma 1. *Let $\mathcal{P} = (h_1, ..., h_q)$ be a feasible partial path. Then, $\underline{T}_{\mathcal{P}} = (\underline{\tau}_1, ..., \underline{\tau}_q) \in \mathscr{T}_{\mathcal{P}}$.*

The proofs of all lemmas and propositions are presented in Section A of the appendix.

Using the values ed_ℓ^i, we obtain the following extension to *Dom-SP* that is a valid dominance criterion for SP_{min}:

Proposition 3. *(Dom-SP_{min}) A feasible label ℓ_1 dominates a label ℓ_2 if*

$$\eta_{\ell_1} = \eta_{\ell_2}, \quad \tilde{c}_{\ell_1} \leq \tilde{c}_{\ell_2}, \quad t_{\ell_1} \leq t_{\ell_2}, \quad O_{\ell_1} \subseteq O_{\ell_2}, \quad \textit{and} \quad ed_{\ell_1}^i \leq ed_{\ell_2}^i \ \forall i \in O_{\ell_1}. \tag{19}$$

Labeling algorithm for SP_{min}. SP_{min} can be solved using the labeling algorithm of Section 3.1 for solving *SP*. Let ℓ' be the label resulting from the extension of label ℓ along arc (η_ℓ, x). The REFs for the additional resources $ed_{\ell'}^i, i \in O_{\ell'}$ are

$$ed_{\ell'}^i = \begin{cases} \max\{a_{x+n}, t_{\ell'} + t_{x,x+n}, t_{\ell'} + \underline{L}_x\} & \text{if } i = x, \\ \max\{ed_\ell^i, t_{\ell'} + t_{x,i+n}\} & \text{otherwise.} \end{cases} \tag{20}$$

Moreover, the REF (13) for the earliest start of service t_ℓ has to be replaced by

$$t_{\ell'} = \begin{cases} \max\{ed_\ell^{x-n}, t_\ell + t_{\eta_\ell,x}\} & \text{if } x \in D, \\ \max\{a_x, t_\ell + t_{\eta_\ell,x}\} & \text{otherwise.} \end{cases} \tag{21}$$

Again, the information ed_ℓ^i is used for eliminating labels that cannot be completed to feasible $0-(2n+1)$-paths.

3.4. SP_{min}^{max} - subproblem with minimum and maximum ride-time constraints

Subproblem SP_{min}^{max} is the natural subproblem of the SPDP in which generated paths represent SPDP-feasible routes, i.e., they have to respect pairing and precedence, capacities, time windows, and minimum and maximum ride times. The implied scheduling problem is (1)–(4). It simultaneously includes both minimum and maximum ride times which significantly complicates SP_{min}^{max} compared to SP^{max} and SP_{min}. The key problem is the interference of different types of ride-time constraints of different requests so that a straightforward combination of the approaches of Sections 3.2 and 3.3 is not possible. We demonstrate this in more detail in the following.

Generalizing the basic idea of Gschwind and Irnich (2014), the minimum and maximum ride times of an open request $i \in O_\ell$ impose a lower bound (ed_ℓ^i) and an upper bound (ld_ℓ^i) on the start of service at the delivery node $i+n$. Again, a small value ed_ℓ^i and a large value ld_ℓ^i are preferable. The implied optimal strategies for the start of service at the pickup node i, i.e., an early-as-possible service to minimize ed_ℓ^i and a late-as-possible service to maximize ld_ℓ^i, are clearly opposing. Even more, different strategies for the pickup times of different open requests may interfere. More precisely, servicing one node late may imply that another one cannot be serviced early, and vice versa. As a result, there is generally no feasible time schedule that minimizes the values ed_ℓ^i for some $i \in O_\ell$ and at the same time maximizes the values ld_ℓ^j for some other $j \in O_\ell$. Thus in SP_{min}^{max}, the open requests and the associated earliest and latest delivery times cannot be treated independently in a dominance rule. Table 2 gives a small example to illustrate this.

Let ℓ_1 and ℓ_2 be two labels representing the paths $\mathcal{P}_{\ell_1} = (0, i, j, k)$ and $\mathcal{P}_{\ell_2} = (0, j, i, k)$. Assume identical travel times of 10 between all nodes. Furthermore, let the minimum and maximum ride times for all requests be 40 and 50, respectively. The time windows of nodes 0, i, j, and k are specified in Table 2, while the time windows at the corresponding delivery nodes are assumed to be not binding ($[0, \infty]$). Then, the earliest possible delivery times of requests i and j for label ℓ_1 (as defined in Section 3.3) are $ed_{\ell_1}^i = \max\{0, 50+10, 10+40\} = 60$ and $ed_{\ell_1}^j = \max\{0, 50+10, 20+40\} = 60$. The latest possible delivery times $ld_{\ell_1}^i$ and $ld_{\ell_1}^j$ as defined in Section 3.2 are, in general, functions in the start of service at the current node $\eta_{\ell_1} = k$. Here, the only feasible start of service at node k is at time 50. Thus, we only need to consider

	Nodes in $\mathcal{P}(\ell_1)$	0	i	j	k
Label ℓ_1 for path $(0,i,j,k)$	Time window $[a_., b_.]$	$[0,100]$	$[0,30]$	$[20,50]$	$[50,50]$
	Earliest start of service t_{ℓ_1}	0	10	20	50
	Earliest possible delivery $ed^{\cdot}_{\ell_1}$	–	60	60	90
	Latest possible delivery $ld^{\cdot}_{\ell_1}$	–	80	90	100
	Nodes in $\mathcal{P}(\ell_2)$	0	j	i	k
Label ℓ_2 for path $(0,j,i,k)$	Time window $[a_., b_.]$	$[0,100]$	$[20,50]$	$[0,30]$	$[50,50]$
	Earliest start of service t_{ℓ_2}	0	20	30	50
	Earliest possible delivery $ed^{\cdot}_{\ell_2}$	–	60	70	90
	Latest possible delivery $ld^{\cdot}_{\ell_2}$	–	70	80	100

Table 2: Label ℓ_1 dominates label ℓ_2 in the sense of $Dom-SP^{max}$ and $Dom-SP_{min}$. This does not imply a valid dominance relation for SP^{max}_{min}.

the values $ld^i_{\ell_1}(50) = \min\{\infty, 30+50\} = 80$ and $ld^i_{\ell_1}(50) = \min\{\infty, 40+50\} = 90$, which imply delaying the start of service at node i until time 30 and at node j until time 40. It is easy to see, however, that there is no feasible time schedule $T_{\mathcal{P}_{\ell_1}}$ that at the same time allows a latest possible delivery time of 80 for request i and an earliest possible delivery time of 60 for request j. Note that the former implies a service time not smaller than 30 at node i while the latter implies a service time not larger than 20 at node j.

As a result, simply combining the relations for ed^i_ℓ and ld^i_ℓ of dominance rules *Dom-*SP_{min} and *Dom-*SP^{max} does not lead to a valid dominance criterion for SP^{max}_{min}. The completion $Q = (j+n, i+n, k+n, 2n+1)$, e.g., is feasible for label ℓ_2 but infeasible for ℓ_1, although $ed^x_{\ell_1} < ed^x_{\ell_2}$ and $ld^x_{\ell_1}(50) > ld^x_{\ell_2}(50)$ hold for $x = i, j, k$ (see Table 2).

Dominance rule for SP^{max}_{min}. The example above has shown that the interdependence of different open requests has to be incorporated when trying to dominate labels in SP^{max}_{min}. Roughly speaking, this means that one has to be careful when determining the earliest and latest delivery times of the open requests of a label.

On the one hand, there are completions where one or more requests can only be delivered at the earliest or latest possible time. Consequently, for a dominated label ℓ we have to consider the best possible values for feasible delivery times of all open requests $i \in O_\ell$, i.e., we consider ed^i_ℓ and $ld^i_\ell(t), t \geq t_\ell$ as defined in Sections 3.3 and 3.2, respectively. Note again that here $\mathscr{T}_{\mathcal{P}_\ell}$ refers to the set of all schedules satisfying constraint system (1)–(4).

On the other hand, a completion might generally require picking up some open requests early and some other open requests late. For a dominating label ℓ with $\mathcal{P}_\ell = (h_1, ..., h_q = \eta_\ell)$ we, therefore, determine two bounds on the service time of an open request $h_i \in O_\ell$ that are independent of the service times at the other open requests $h_j \in O_\ell \setminus \{h_i\}$.

First, we use the following upper bound $\underline{\tau}^O_i(t)$ for an early-as-possible service at a pickup node h_i within path $\mathcal{P}_\ell$. Practically speaking, $\underline{\tau}^O_i(t)$ gives the earliest service time at h_i that can be attained without restricting the pickup times of the other open requests. Or from the opposite perspective, when scheduling all other open requests $h_j \in O_\ell \setminus \{h_i\}$ in the most unfavorable way for picking up h_i early, i.e., as late as possible, then the earliest possible service time at h_i that is still feasible is $\underline{\tau}^O_i(t)$. Formally, $\underline{\tau}^O_i(t) = \min_{T_{\mathcal{P}_\ell} \in \underline{\mathscr{T}}^i_{\mathcal{P}_\ell}(t)} \{\tau_i\}$ with $\underline{\mathscr{T}}^i_{\mathcal{P}_\ell}(t) = \{T_{\mathcal{P}_\ell} \in \mathscr{T}_{\mathcal{P}_\ell}(t) : \tau_j \geq \overline{\tau}_j(t)\ \forall h_j \in O_\ell \setminus \{h_i\}\}$. Note that $\underline{\tau}^O_i(t)$ is a function in t, as the times $\overline{\tau}_j(t)$ depend on t.

Second and analog to $\underline{\tau}^O_i(t)$, a lower bound for a late-as-possible service at node h_i is denoted by $\overline{\tau}^O_i(t)$. It gives the latest feasible start of service at the pickup node h_i such that the start of service at all other open requests $h_j \in O_\ell \setminus \{h_i\}$ takes its minimal value $\underline{\tau}_j$ and $\tau_q \leq t$, i.e., $\overline{\tau}^O_i(t) = \max_{T_{\mathcal{P}_\ell} \in \overline{\mathscr{T}}^i_{\mathcal{P}_\ell}(t)} \{\tau_i\}$ with $\overline{\mathscr{T}}^i_{\mathcal{P}_\ell}(t) = \{T_{\mathcal{P}_\ell} \in \mathscr{T}_{\mathcal{P}_\ell}(t) : \tau_j \leq \underline{\tau}_j\ \forall h_j \in O_\ell \setminus \{h_i\}\}$. Maximizing the service at h_i may delay the start of service τ_q at the current node η_ℓ. Thus, $\overline{\tau}^O_i(t)$ is also a function in t.

Using $\underline{\tau}_i^O(t)$ and $\overline{\tau}_i^O(t)$ we have the following upper and lower bounds for the earliest and latest delivery times of an open request $h_i \in O$, respectively:

$$\overline{ed}_\ell^{h_i}(t) = \max\{a_{h_i+n}, t_\ell + t_{\eta_\ell, h_i+n}, \underline{\tau}_i^O(t) + \underline{L}_{h_i}\}, \tag{22}$$

$$\underline{ld}_\ell^{h_i}(t) = \min\{b_{h_i+n}, \overline{\tau}_i^O(t) + \overline{L}_{h_i}\}. \tag{23}$$

Denote by $\tilde{b}_\ell$ the latest feasible start of service at node η_ℓ for a label ℓ whose parent label is ℓ'. The value $\tilde{b}_\ell$ is given by $\min\{b_{\eta_\ell}, ld_\ell^{\eta_\ell - n}(\tilde{b}_{\ell'})\}$ if $\eta_\ell \in D$ and b_{η_ℓ} otherwise. Then, a valid dominance rule for SP_{min}^{max} is as follows:

Proposition 4. *(Dom^*-SP_{min}^{max}) A feasible label ℓ_1 dominates a label ℓ_2 if*

$$\eta_{\ell_1} = \eta_{\ell_2}, \quad \tilde{c}_{\ell_1} \le \tilde{c}_{\ell_2}, \quad t_{\ell_1} \le t_{\ell_2}, \quad O_{\ell_1} \subseteq O_{\ell_2}, \text{ and} \tag{24}$$

$$\overline{ed}_{\ell_1}^i(t) \le ed_{\ell_2}^i \quad \text{and} \quad \underline{ld}_{\ell_1}^i(t) \ge ld_{\ell_2}^i(t) \quad \forall i \in O_{\ell_1}, t \in [t_{\ell_2}, \tilde{b}_{\ell_2}]. \tag{25}$$

Applying dominance rule Dom^*-SP_{min}^{max} requires the comparison of different functions in inequalities (25) which is clearly not practicable for general functions within a labeling algorithm. The following lemma characterizes the shape of the functions $\overline{ed}_\ell^i(t)$, $ld_\ell^i(t)$, and $\underline{ld}_\ell^i(t)$ allowing for a simplified version of Dom^*-SP_{min}^{max}.

Lemma 2. *Let $\mathcal{X}$ be the set of all $X = (x_1, ..., x_q) \in \mathbb{R}^q$ satisfying*

$$a_i \le x_i \le b_i \quad \forall i = 1, ..., q, \tag{26}$$

$$x_i + c_{ij} \le x_j \quad \forall j = 2, ..., q; i < j, \tag{27}$$

$$x_i + d_{ij} \ge x_j \quad \forall j = 2, ..., q; i < j, \tag{28}$$

with real-valued constants a_i, b_i, c_{ij}, and d_{ij}. Denote $\mathcal{X}(t) = \{X \in \mathcal{X} : x_q \le t\}, t \in \mathbb{R}$. Let also be $\overline{x}_i = \max_{X \in \mathcal{X}}\{x_i\}$, $\underline{x}_i = \min_{X \in \mathcal{X}}\{x_i\}$, $\overline{x}_i(t) = \max_{X \in \mathcal{X}(t)}\{x_i\}$, and $\underline{x}_i(t) = \min_{X \in \mathcal{X}(t)}\{x_i\}$. Furthermore, define by $\mathcal{X}^{\underline{S}}(t) = \{X \in \mathcal{X}(t) : x_i \le \underline{x}_i(t) \ \forall i \in \underline{S}\}$ and by $\mathcal{X}^{\overline{S}}(t) = \{X \in \mathcal{X}(t) : x_i \ge \overline{x}_i(t) \ \forall i \in \overline{S}\}$ with $\underline{S}, \overline{S} \subseteq \{1, ..., q\}$. Denote $\overline{x}_i^{\underline{S}}(t) = \max_{X \in \mathcal{X}^{\underline{S}}(t)}\{x_i\}$ and $\underline{x}_i^{\overline{S}}(t) = \min_{X \in \mathcal{X}^{\overline{S}}(t)}\{x_i\}$. Finally, let t^* be the smallest t with $\mathcal{X}(t) \ne \varnothing$. Then, the following properties hold:

1. *$\overline{x}_i(t) = \min\{k_i^1, k_i^2 + t\}$ for all $i = 1, ..., q, t \ge t^*$ with constants k_i^1 and k_i^2.*
2. *$\underline{x}_i(t) = \underline{x}_i$ for all $i = 1, ..., q, t \ge t^*$.*
3. *$\overline{x}_i^{\underline{S}}(t) = \min\{k_i^1, k_i^2 + t\}$ for all $i = 1, ..., q, t \ge t^*$ with constants k_i^1 and k_i^2.*
4. *$\underline{x}_i^{\overline{S}}(t) = \max\{k_i^1, \min\{k_i^2, k_i^3 + t\}\}$ for all $i = 1, ..., q, t \ge t^*$ with constants k_i^1, k_i^2, and k_i^3.*

Clearly, the scheduling problem (1)–(4) of SP_{min}^{max} is a special case of the constraint system (26)–(28) considered in Lemma 2. Thus, the functions $\overline{ed}_\ell^i(t)$, $ld_\ell^i(t)$, and $\underline{ld}_\ell^i(t)$ are of the forms $\overline{ed}_\ell^i(t) = \max\{k_i^1, \min\{k_i^2, k_i^3 + t\}\}$, $ld_\ell^i(t) = \min\{k_i^4, k_i^5 + t\}$, and $\underline{ld}_\ell^i(t) = \min\{k_i^6, k_i^7 + t\}$, where all $k_i^\cdot$ are constants. Herewith, the comparison of the functions in Dom^*-SP_{min}^{max} can be reduced to comparing them at two distinct points of time. Denote by $\underline{B}_\ell^i$ and $\overline{B}_\ell^i$ the points of time such that $\underline{ld}_\ell^i(t) = \underline{ld}_\ell^i(\underline{B}_\ell^i)$ and $\overline{ed}_\ell^i(t') = \overline{ed}_\ell^i(\overline{B}_\ell^i)$ holds for all $t \ge \underline{B}_\ell^i$ and $t' \ge \overline{B}_\ell^i$, respectively. Then, the following dominance rule for SP_{min}^{max} results:

Proposition 5. *(Dom-SP_{min}^{max}) A feasible label ℓ_1 dominates a label ℓ_2 if*

$$\eta_{\ell_1} = \eta_{\ell_2}, \quad \tilde{c}_{\ell_1} \le \tilde{c}_{\ell_2}, \quad t_{\ell_1} \le t_{\ell_2}, \quad O_{\ell_1} \subseteq O_{\ell_2}, \tag{29}$$

$$\overline{ed}_{\ell_1}^i(t_{\ell_1}) \le ed_{\ell_2}^i \quad \text{and} \quad \overline{ed}_{\ell_1}^i(\overline{B}_{\ell_1}^i) - \max\{0, \overline{B}_{\ell_1}^i - \tilde{b}_{\ell_2}\} \le ed_{\ell_2}^i \quad \forall i \in O_{\ell_1}, \text{ and} \tag{30}$$

$$\underline{ld}_{\ell_1}^i(t_{\ell_1}) + (t_{\ell_2} - t_{\ell_1}) \ge ld_{\ell_2}^i(t_{\ell_2}) \quad \text{and} \quad \underline{ld}_{\ell_2}^i(\underline{B}_{\ell_1}^i) \ge ld_{\ell_2}^i(B_{\ell_2}^i) \quad \forall i \in O_{\ell_1}. \tag{31}$$

Note that for determining valid bounds $\overline{ed}_\ell^{h_i}$ and $\underline{ld}_\ell^{h_i}$ on the earliest and latest delivery times of request h_i that do not restrict the pickup times of other open requests $h_j \in O_\ell \setminus \{h_i\}$, it is generally not necessary to consider the maximum and minimum values $\overline{\tau}_j$ and $\underline{\tau}_j$ for the starts of service at the nodes h_j. Instead, it is sufficient to ensure that all h_j can be delivered at their earliest or latest possible delivery times $ed_\ell^{h_j}$ and $ld_\ell^{h_j}$. These times induce starting the service not later than $ed_\ell^{h_j} - \underline{L}_{h_j} \geq \underline{\tau}_j$ and not earlier than $ld_\ell^{h_j} - \underline{L}_{h_j} \leq \overline{\tau}_j$. Hence, bounds that are stronger than $\overline{ed}_\ell^i$ and $\underline{ld}_\ell^i$ can be obtained enabling more dominance when used in *Dom-SP*$_{min}^{max}$. For simplicity of notation and exposition this has been disregarded in the derivation of *Dom-SP*$_{min}^{max}$. All proofs, however, are analog.

Dom-SP$_{min}^{max}$ can further be strengthened by using a concept proposed by Gschwind and Irnich (2014) for *Dom-SP*^{max}. Let ℓ be the parent label of ℓ'. The information ed_ℓ^i and ld_ℓ^i on the feasible delivery times of open requests $i \in O_\ell$ can be used to determine an upper bound on the start of service at node $\eta_{\ell'}$ for which $\mathcal{P}_{\ell'}$ can be completed to a feasible $0-(2n+1)$-path. This bound, which is generally smaller than $\tilde{b}_{\ell'}$, strengthens the dominance relation in *Dom-SP*$_{min}^{max}$ (see Section 4.6 in Gschwind and Irnich, 2014, for details).

Labeling algorithm for SP$_{min}^{max}$*.* The basic course of our labeling algorithm with *Dom-SP*$_{min}^{max}$ for solving *SP*$_{min}^{max}$ is identical to those in Sections 3.1-3.3. When creating a new label ℓ', the resources $\eta_{\ell'}$, $\tilde{c}_{\ell'}$, $l_{\ell'}$, and $O_{\ell'}$ are updated using the REFs (13) and (14). The earliest start of service $t_{\ell'}$ is set according to the adapted REF (21).

Determining the values of the resources related to feasible delivery times of open requests $i \in O_{\ell'}$ is intricate. Because of the simultaneous handling of minimum and maximum ride-time constraints, the information on the earliest and latest delivery is interdependent. As a consequence, the determination of these values is much more complex than in the isolated cases in *SP*$_{min}$ and *SP*^{max}. The key problems are the following: When creating a new label ℓ', the implied scheduling problem has additional constraints compared to the scheduling problem implied by the parent label ℓ. These constraints impose bounds on the start of service at the current node $\eta_{\ell'}$ that may restrict other service times within the schedule. The impact on these service times may further propagate throughout the constraint system (see proofs of Proposition 4 and Lemma 2) so that their effect on the earliest and latest pickup times at the open requests $i \in O_{\ell'}$ is non-trivial to identify. Moreover, if the extended label ℓ' ends at a delivery node $\eta_{\ell'} \in D$, then the corresponding request $\eta_\ell - n$ is no longer open. For all open requests $h_i \in O_{\ell'}$, this reduces the set of requests whose latest and earliest service times have to be taken into account when determining the bounds $\underline{\tau}_i^O$ and $\overline{\tau}_i^O$, respectively. Thus, the relation between the resource values $\overline{ed}_{\ell'}^i(t_{\ell'})$, $\overline{ed}_{\ell'}^i(\overline{B}_{\ell'}^i)$, $\underline{ld}_{\ell'}^i(t_{\ell'})$, and $\underline{ld}_{\ell'}^i(\underline{B}_{\ell'}^i)$ with $i \in O_{\ell'}$ and the corresponding values $\overline{ed}_\ell^i(t_\ell)$, $\overline{ed}_\ell^i(\overline{B}_\ell^i)$, $\underline{ld}_\ell^i(t_\ell)$, and $\underline{ld}_\ell^i(\underline{B}_\ell^i)$ of the parent label ℓ is highly complex.

As a result, we were not able to derive simple update formulas for the resources related to feasible delivery times of open requests. We suspect that if there are REFs for these resources carrying along several auxiliary resources needed for the calculations is necessary. It seems also mandatory for the computation of these resources to know the actual node sequence $\mathcal{P}_{\ell'}$ represented by label ℓ'.

In our algorithm, the earliest and latest delivery times are computed from scratch within each label. To do so, we use a generalized version of the feasibility test of Tang *et al.* (2010) for the DARPto obtain a feasible schedule with early-as-possible service times for all nodes which provides the values $ed_{\ell'}^i, i \in O_{\ell'}$. Starting from this schedule, we repeatedly delay the service at distinct nodes to obtain the remaining delivery times. This is done using the concept of forward time slack originally introduced by Savelsbergh (1992) for the TSPTW. Note that it is not necessary to consider the complete path $\mathcal{P}_{\ell'}$ for these computations. Instead, it is sufficient to take into account the subpath between the node at which the vehicle was empty for the last time and the current node $\eta_{\ell'}$. For technical details on the adapted feasibility test of Tang *et al.* (2010) and the adapted version of the forward time slack we refer to a companion paper (Gschwind, 2015).

The elimination of labels with no feasible completion to node $2n+1$ makes use of both the earliest possible delivery times ed_ℓ^i and the latest possible delivery times ld_ℓ^i. With this information, the label elimination strategy is very effective.

4. Computational results

This section summarizes the computational experiments that we have conducted to compare the performance of the four different branch-and-cut-and-price approaches to the SPDP.

4.1. Branch-and-cut-and-price algorithm

In the following, we briefly describe the main components of the basic branch-and-cut-and-price algorithm. Based on this algorithm, we devise our four different integer column-generation approaches to the SPDP. Each of the approaches formulates the master problem on a different variable set implying subproblem SP, SP_{min}, SP^{max}, or SP^{max}_{min}. The subproblems are solved using the respective labeling algorithms of Sections 3.1 - 3.4.

Preprocessing. Time-window tightening and arc elimination is performed according to the rules proposed by Desrochers *et al.* (1992), Dumas *et al.* (1991), and Cordeau (2006) for the VRPTW or tailored to the PDPTW or the DARP. The integration of minimum and maximum ride-time constraints into these rules is straightforward.

Pricing problem heuristics. To speed up the column-generation process, heuristics can be used to identify negative reduced-cost columns fast. When the heuristics are unable to find additional columns, one has to resort to an exact method to solve the subproblem. In our algorithms, we tried two straightforward pricing problem heuristics. The first is to solve a more relaxed subproblem, e.g., solving SP when actually having to solve SP^{max}, and to drop all routes that are infeasible for the actual subproblem. The other is to solve the subproblem on a reduced network only. Preliminary computational tests indicated that the benefits from using these heuristics were rather limited for all algorithms.

Cutting Planes. In our branch-and-cut-and-price algorithms, we use the following types of valid inequalities: 2-path inequalities (Kohl *et al.*, 1999), rounded capacity inequalities in a form proposed by Ropke and Cordeau (2009) for the PDPTW, fork inequalities (Ropke *et al.*, 2007), and two different liftings of IPEC introduced by Ascheuer *et al.* (2000) for the TSPTW and Cordeau (2006) for the DARP. Heuristic separation procedures proposed by Ropke and Cordeau (2009) are used to separate 2-path inequalities, rounded capacity inequalities, and fork inequalities. For the exact separation of the lifted IPEC we use a straightforward enumeration procedure (see Ascheuer *et al.*, 2000). SPDP-feasibility of an integer solution obtained by approaches using a relaxed variable set Ω' is, thus, guaranteed by the lifted IPEC.

Branching strategy and node selection. A hierarchical branching scheme is used to obtain integer solutions in our algorithms. We first branch on the number of vehicles, if fractional. We then branch on the outflow of a node set of cardinality two. Both branching rules are enforced by adding a single linear constraint to the master problem. The structure of the subproblems remains unchanged. The branch-and-bound tree is explored with a best-first strategy and no upper bounds are given to the algorithm.

All algorithms described in this paper were implemented in C++ using CPLEX 12.2 as LP-solver. Arc costs and travel times are computed with double precision. The experiments were performed on a standard PC with an Intel(R) Core(TM) i7-2600 at 3.4 GHz with 16.0 GB main memory using a single thread only. The time limit was set to one hour.

4.2. Test instances

The test instances used in the computational study are all based on the benchmark set for the DARP originally introduced by Cordeau (2006) and later extended by instances with larger problem sizes by Ropke *et al.* (2007). For a detailed description of these instances and their generation we refer to (Cordeau, 2006).

The DARP benchmark set consists of random Euclidean instances with problem sizes reaching from two vehicles and 16 customer requests to eight vehicles and 96 customer requests. Additional larger instances with up to ten vehicles and 120 customer requests were generated in the same fashion as the DARP instances. All these instances are characterized by small vehicle capacities and narrow time windows. To also consider harder instances in which these constraints are less restrictive, three new instances were constructed from each original instance by enlarging both capacities and time-window lengths by factors of 4/3, 5/3, and 6/3.

In the original benchmark set, there are two subsets of instances (type `a` and type `b`) with different characteristics regarding customer demand and vehicle capacity. Moreover, the maximum ride times are specified by a fixed value $\overline{L}$ for each subset and are identical for all requests and all instances of the given subset. In the way the instances are constructed (the time windows are specified either for the pickup or the delivery node of a request) the maximum ride-time constraint of each request is always restrictive, independent of how small or large $\overline{L}$ is chosen. Specifying the minimum ride times analogously, i.e., setting them to one identical value $\underline{L}$ for each request raises the following issue: If the value $\underline{L}$ is chosen small, the minimum ride-time constraints of many requests are redundant because the direct travel time between pickup and delivery node of the request is already larger than $\underline{L}$. If $\underline{L}$ is chosen large enough so that none of the minimum ride-time constraints is redundant, many instances of the benchmark become infeasible. We, therefore, chose not to use fixed values for the ride times in our test instances. Instead, we modeled both maximum and minimum ride times for each request proportional to the direct travel time between pickup and delivery node. As a result, it is assured that none of the ride-time constraints is redundant, which in our opinion provides the fairest benchmark for the comparison of the different algorithms. Moreover, it allows us in a good way to create instances with different characteristics regarding the tightness of the ride-time constraints. More precisely, in our instances the maximum ride time of a request is equal to the product of the direct travel time and a random number chosen according to a uniform distribution over a given interval. We considered the two intervals $[2.25, 2.75]$ (for more restrictive maximum ride times) and $[2.75, 3.25]$ (for less restrictive maximum ride times). The minimum ride times were specified in a similar fashion using the intervals $[1.75, 2.25]$ and $[1.25, 1.75]$ for generating instances with more restrictive and less restrictive minimum ride times, respectively.

The complete benchmark comprises 864 instances labeled in the form `RT-TW-iK-n`, where `n` denotes the number of requests, `K` denotes the number of vehicles, and $\mathtt{i} \in \{\mathtt{a}, \mathtt{b}\}$ denotes the subset the instance originates from. Moreover, `TW` = `A` refers to the original instances with small vehicle capacities and time-window lengths, while `TW` = `B`, `TW` = `C` and `TW` = `D` denote the instances in which these values have been enlarged by factor 4/3, 5/3, and 6/3, respectively. The characteristics regarding ride times are specified by $\mathtt{RT} \in \{\mathtt{MM}, \mathtt{ML}, \mathtt{LM}, \mathtt{LL}\}$, where `M` and `L` indicate the more restrictive and less restrictive cases, respectively, while the first character refers to minimum ride times and the second character refers to maximum ride times. Note that some of the small instances are infeasible in the presence of minimum and maximum ride-time constraints. We allowed the use of additional vehicles to obtain well-defined instances in these cases. All instances are available at `http://logistik.bwl.uni-mainz.de/Dateien/SPDP.zip`.

4.3. Analysis of results

Table 3 summarizes our results averaged over all benchmark instances. Tables 4 and 5 present averaged results for the subclasses `A`, `B`, `C`, `D` and `MM`, `LM`, `ML`, `MM`, respectively. In each table, we also report the results averaged only over the larger instances with $n > 80$ requests. More detailed results can be found in Tables 6–13 in Section B of the appendix. The columns of the tables have the following meaning:

tree — number of optimal solutions (*opt*), average computation time in seconds ($t\,[\mathrm{s}]$), and average percentage integrality gap (*% gap*) of the respective branch-and-cut-and-price algorithm

root — number of optimal solutions (*opt*) and average percentage integrality gap (*% gap*) in the root node

% time — average percentage time spent for the solution of the subproblem (*sp*), separation (*sep*), and reoptimization of the master program (*lp*)

 # solved — number of solved branch-and-bound nodes (*nd*) and subproblems (*sp*)

 # gen — number of generated columns (*col*) and cuts (*cut*)

The results in Table 3 indicate that algorithms IMP^{max}_{min} and $IMP\text{-}I^{max}$ are clearly superior to $IMP\text{-}I_{min}$ and *IMP-I*. The overall performance of the two stronger approaches IMP^{max}_{min} and $IMP\text{-}I^{max}$ is comparable. In total, IMP^{max}_{min} is able to solve 786 out of the 864 instances to optimality, four more than $IMP\text{-}I^{max}$.

	tree			root		% time			# solved		# gen	
algorithm	*opt*	*t* [s]	*% gap*	*opt*	*% gap*	*sp*	*sep*	*lp*	*nd*	*sp*	*col*	*cut*
All instances (864)												
IMP_{min}^{max}	786	466	0.05	381	0.18	80	13	6	29	2296	160082	11
$IMP\text{-}I^{max}$	782	474	0.05	320	0.25	53	28	20	82	4595	181881	65
$IMP\text{-}I_{min}$	612	1180	0.33	312	0.54	34	20	45	47	2669	95890	280
$IMP\text{-}I$	598	1274	0.38	278	0.60	28	21	51	43	2807	92173	308
Instances with $n > 80$ (192)												
IMP_{min}^{max}	127	1532	0.20	22	0.37	90	6	4	58	5784	474642	21
$IMP\text{-}I^{max}$	127	1545	0.21	14	0.47	67	12	22	174	12544	582031	126
$IMP\text{-}I_{min}$	62	2603	1.02	13	1.20	38	7	55	62	4209	215036	515
$IMP\text{-}I$	58	2725	1.13	10	1.32	32	8	61	63	4533	214094	551

Table 3: Summary results aggregated over all 864 instances

Both approaches solve 127 out of the 192 instances with $n > 80$. Regarding computation times, $IMP\text{-}I_{min}^{max}$ is on average also slightly faster than $IMP\text{-}I^{max}$. Algorithm $IMP\text{-}I_{min}$ is inferior to both of the former approaches regarding number of solved instances, computation times and remaining integrality gap. $IMP\text{-}I$ performs even worse on all these numbers.

The superiority of IMP_{min}^{max} and $IMP\text{-}I^{max}$ over $IMP\text{-}I_{min}$ and $IMP\text{-}I$ can be attributed to the following reasons: First, the root node lower bounds of IMP_{min}^{max} and $IMP\text{-}I^{max}$ are significantly stronger resulting in smaller search trees for these approaches. Second, for $IMP\text{-}I_{min}$ and $IMP\text{-}I$ substantially more cuts are added to the master programs severely complicating their reoptimization. This is also the reason why the average number of solved nodes is smaller for approaches $IMP\text{-}I_{min}$ and $IMP\text{-}I$ compared to approach $IMP\text{-}I^{max}$. While $IMP\text{-}I^{max}$ explores a huge number of nodes when solving difficult instances, algorithms $IMP\text{-}I_{min}$ and $IMP\text{-}I$ spend a lot of time reoptimizing the master programs and, thus, can solve only few nodes within the time limit. When comparing instances solved by all approaches, the number of explored nodes is indeed much higher for approaches $IMP\text{-}I_{min}$ and $IMP\text{-}I$.

The more disaggregated results in Tables 4 and 5 indicate that all findings from the overall results regarding the performance of the different approaches do also hold for all subclasses of instances. This means that the characteristics of the ride-time constraints have only limited influence on the relation of the strengths of the considered algorithms. This is also true for vehicle capacity, customer demands, and time-window lengths.

Another interesting result of our experiments is that handling the maximum ride-time constraints in the subproblem seems to be more important than integrating the minimum ride times into the subproblem. Our interpretation is that the minimum ride-time constraints are often satisfied without explicitly considering them for the following reasons: When the time windows are narrow, many customer requests are picked up at their origin node i at time a_i. In these cases, the time-window tightening rules ensure that the minimum ride times are respected. With wide time windows, on the other hand, several other customer nodes are often visited in between the pickup and delivery of a request. This increases the ride times of the respective request compared to the direct travel times so that the minimum ride-time constraints might already be satisfied.

5. Conclusion

In this paper, we introduced the Synchronized Pickup and Delivery Problem (SPDP) as the prototypical VRP with temporal intra-route synchronization. In the SPDP, vehicle routes have to satisfy pairing and precedence, capacities, and time windows. Additionally, temporal synchronization constraints couple the service times at the pickup and delivery locations of the customer requests in the following way: A delivery

		tree			*root*		*% time*			*# solved*		*# gen*	
	algorithm	*opt*	*t* [s]	*% gap*	*opt*	*% gap*	*sp*	*sep*	*lp*	*nd*	*sp*	*col*	*cut*
All instances (216 each)													
A (orig.)	IMP_{min}^{max}	212	167	0.00	128	0.07	77	14	9	19	2211	136508	7
	$IMP\text{-}I^{max}$	212	167	0.00	114	0.10	50	31	19	40	2987	105862	32
	$IMP\text{-}I_{min}$	198	389	0.01	120	0.11	40	27	32	29	2045	73045	106
	$IMP\text{-}I$	196	425	0.02	108	0.13	33	30	37	29	1999	64006	122
B (4/3)	IMP_{min}^{max}	206	242	0.00	108	0.09	78	15	7	20	1690	113264	9
	$IMP\text{-}I^{max}$	208	242	0.01	89	0.12	51	32	17	44	3169	127318	46
	$IMP\text{-}I_{min}$	179	735	0.08	84	0.22	34	24	42	37	2386	87613	207
	$IMP\text{-}I$	175	858	0.10	74	0.25	27	24	49	38	2633	85351	234
C (5/3)	IMP_{min}^{max}	193	560	0.04	84	0.20	81	14	5	31	2911	243361	13
	$IMP\text{-}I^{max}$	191	574	0.05	69	0.28	53	27	20	98	5297	210325	79
	$IMP\text{-}I_{min}$	138	1479	0.34	63	0.60	30	17	53	60	2971	105573	342
	$IMP\text{-}I$	134	1628	0.40	54	0.66	23	18	60	53	3213	101596	377
D (6/3)	IMP_{min}^{max}	175	893	0.16	61	0.36	86	10	4	47	2370	147196	16
	$IMP\text{-}I^{max}$	171	913	0.16	48	0.51	57	22	21	145	6925	284020	103
	$IMP\text{-}I_{min}$	97	2115	0.90	45	1.22	33	13	54	62	3273	117329	464
	$IMP\text{-}I$	93	2187	1.01	42	1.36	27	13	59	54	3382	117739	497
Instances with $n > 80$ (48 each)													
A (orig.)	IMP_{min}^{max}	44	623	0.01	11	0.13	86	6	8	52	7001	498632	12
	$IMP\text{-}I^{max}$	45	621	0.00	10	0.18	63	18	19	121	9660	383002	63
	$IMP\text{-}I_{min}$	35	1308	0.04	9	0.21	40	12	48	73	5695	215803	225
	$IMP\text{-}I$	33	1421	0.06	8	0.24	33	14	53	68	5775	206167	249
B (4/3)	IMP_{min}^{max}	38	963	0.01	6	0.15	86	10	5	56	5815	444128	19
	$IMP\text{-}I^{max}$	40	946	0.02	3	0.22	63	17	20	130	10828	481830	100
	$IMP\text{-}I_{min}$	22	2142	0.28	3	0.48	34	8	59	80	5608	257743	439
	$IMP\text{-}I$	21	2402	0.33	2	0.53	24	6	70	76	5940	244192	481
C (5/3)	IMP_{min}^{max}	29	1805	0.18	5	0.37	93	4	3	60	5634	540175	22
	$IMP\text{-}I^{max}$	28	1867	0.19	1	0.49	69	8	23	204	13924	645650	151
	$IMP\text{-}I_{min}$	5	3360	1.11	1	1.34	29	4	67	56	3485	204250	650
	$IMP\text{-}I$	4	3476	1.26	0	1.45	24	5	71	54	3544	192288	707
D (6/3)	IMP_{min}^{max}	16	2737	0.59	0	0.80	96	2	2	65	4684	415632	32
	$IMP\text{-}I^{max}$	14	2744	0.62	0	1.01	72	4	24	240	15763	817641	191
	$IMP\text{-}I_{min}$	0	3600	2.63	0	2.77	48	6	46	38	2046	182347	744
	$IMP\text{-}I$	0	3600	2.88	0	3.05	45	6	49	53	2873	213731	767

Table 4: Aggregated results for subclasses A, B, C, and D

	algorithm	tree			root		% time			# solved		# gen	
		opt	t [s]	% gap	opt	% gap	sp	sep	lp	nd	sp	col	cut
All instances (216 each)													
MM	IMP^{max}_{min}	209	252	0.01	116	0.11	78	14	8	24	1689	112534	10
	$IMP\text{-}I^{max}$	204	312	0.02	92	0.17	46	30	24	59	3265	109589	73
	$IMP\text{-}I_{min}$	166	986	0.21	97	0.34	31	20	48	35	1887	64692	296
	$IMP\text{-}I$	162	1068	0.25	84	0.39	23	21	56	22	1776	58741	331
LM	IMP^{max}_{min}	201	364	0.02	99	0.13	78	14	8	31	2109	160862	11
	$IMP\text{-}I^{max}$	202	316	0.02	91	0.17	52	29	19	62	3497	154340	51
	$IMP\text{-}I_{min}$	158	1100	0.29	82	0.43	30	20	51	19	1648	67254	325
	$IMP\text{-}I$	156	1169	0.32	78	0.48	25	21	54	22	1712	59723	343
ML	IMP^{max}_{min}	191	566	0.10	80	0.26	82	13	5	28	1910	113225	11
	$IMP\text{-}I^{max}$	184	695	0.12	64	0.39	54	26	20	113	5989	211589	85
	$IMP\text{-}I_{min}$	147	1288	0.41	65	0.67	36	22	42	57	3098	104723	257
	$IMP\text{-}I$	143	1417	0.49	56	0.78	28	22	50	57	3421	103122	290
LL	IMP^{max}_{min}	185	681	0.07	86	0.23	83	12	5	34	3474	253708	12
	$IMP\text{-}I^{max}$	192	574	0.06	73	0.29	60	25	15	93	5627	252007	50
	$IMP\text{-}I_{min}$	141	1345	0.42	68	0.71	40	20	40	76	4043	146891	242
	$IMP\text{-}I$	137	1443	0.47	60	0.74	34	21	45	72	4319	147106	266
Instances with $n > 80$ (48 each)													
MM	IMP^{max}_{min}	41	955	0.05	8	0.22	88	7	5	55	5404	426134	19
	$IMP\text{-}I^{max}$	37	1086	0.08	6	0.31	57	13	30	117	9240	365199	143
	$IMP\text{-}I_{min}$	21	2344	0.67	5	0.77	29	9	62	40	3367	163060	610
	$IMP\text{-}I$	17	2548	0.82	4	0.91	22	9	69	26	3079	140107	666
LM	IMP^{max}_{min}	35	1308	0.08	4	0.24	89	6	5	59	6031	586758	22
	$IMP\text{-}I^{max}$	37	1075	0.07	3	0.28	66	13	21	130	10206	534659	105
	$IMP\text{-}I_{min}$	16	2604	0.87	3	0.95	28	8	64	20	2185	132582	663
	$IMP\text{-}I$	16	2570	0.92	2	1.02	22	8	69	21	2183	122956	679
ML	IMP^{max}_{min}	26	1851	0.41	5	0.57	93	4	3	54	4244	295062	22
	$IMP\text{-}I^{max}$	24	2132	0.47	2	0.79	71	8	22	230	14829	628445	160
	$IMP\text{-}I_{min}$	13	2693	1.31	2	1.56	45	6	49	80	4707	230725	413
	$IMP\text{-}I$	13	2887	1.50	2	1.77	39	6	55	92	5701	247585	446
LL	IMP^{max}_{min}	25	2014	0.25	5	0.43	92	5	3	66	7456	590613	22
	$IMP\text{-}I^{max}$	29	1886	0.23	3	0.51	74	13	13	217	15901	799820	96
	$IMP\text{-}I_{min}$	12	2770	1.22	3	1.52	50	6	44	108	6576	333776	372
	$IMP\text{-}I$	12	2894	1.29	2	1.58	44	7	49	112	7169	345731	413

Table 5: Aggregated results for subclasses MM, LM, ML, and LL

node has to be serviced within prespecified minimum and maximum time lags (called ride times) after the service at the corresponding pickup node has been completed.

The minimum and maximum ride-time constraints severely complicate the subproblem of the natural column-generation formulation of the SPDP and it is not clear if their integration into the subproblem pays off in an integer column-generation approach. We, therefore, developed four solution approaches to the SPDP based on column-generation formulations with differing subproblems. Two of these subproblems, the natural subproblem of the SPDP that integrates all constraints relating to single routes and the one in which the maximum ride-time constraints are relaxed, were considered for the first time in this paper. New dominance rules and labeling algorithms for their solution have been derived. Extensive computational experiments demonstrate the applicability of these labeling algorithms in the sense that they are capable of solving subproblems arising in state-of-the-art benchmark instances in reasonable time.

The computational results also indicate a clear ranking of the four presented algorithms for solving the SPDP. The strongest approaches are the approach based on the natural column-generation formulation and the one that handles only the maximum ride times in the subproblem. They performed comparably well and were consistently significantly stronger than the approach with only the minimum ride times in the subproblem. This was in turn slightly stronger than the approach in which both ride-time constraints are relaxed in the subproblem. We conclude that the integration of temporal intra-route synchronization constraints into the column-generation subproblem is beneficial for the SPDP and that it is particularly rewarding for the maximum ride-times.

References

Ascheuer, N., Fischetti, M., and Grötschel, M. (2000). A polyhedral study of the asymmetric traveling salesman problem with time windows. *Networks*, **36**(2), 69–79.

Azi, N., Gendreau, M., and Potvin, J.-Y. (2010). An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, **202**(3), 756–763.

Baldacci, R., Bartolini, E., and Mingozzi, A. (2011). An exact algorithm for the pickup and delivery problem with time windows. *Operations Research*, **59**(2), 414–426.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *TOP*, **15**(1), 1–31.

Borndörfer, R., Klostermeier, F., Grötschel, M., and Küttner, C. (1997). Telebus Berlin: vehicle scheduling in a dial-a-ride system. Technical report SC 97-23, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Germany.

Bredström, D. and Rönnqvist, M. (2008). Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operational Research*, **191**, 19–29.

Ceselli, A., Righini, G., and Salani, M. (2009). A column generation algorithm for a rich vehicle-routing problem. *Transportation Science*, **43**(1), 56–69.

Cherkesly, M., Desaulniers, G., and Laporte, G. (2014). Branch-price-and-cut algorithms for the pickup and delivery problem with time windows and last-in-first-out loading. *Transportation Science*. Forthcoming.

Christiansen, M. and Nygreen, B. (1998). Modelling path flows for a combined ship routing and inventory management problem. *Annals of Operations Research*, **82**, 391–413.

Cordeau, J.-F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, **54**(3), 573–586.

Cordeau, J.-F., Laporte, G., and Ropke, S. (2008). Recent models and algorithms for one-to-one pickup and delivery problems. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 327–357. Springer US, Boston and MA.

Desaulniers, G., Desroisiers, J., Ioachim, I., Solomon, M., Soumis, F., and Villeneuve, D. (1998). A unified framework for deterministic time constraint vehicle routing and crew scheduling problems. In T. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 57–93. Kluwer, Norwell and MA.

Desrochers, M., Desrosiers, J., and Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, **40**(2), 342–354.

Dohn, A., Rasmussen, M. S., and Larsen, J. (2011). The vehicle routing problem with time windows and temporal dependencies. *Networks*, **58**(4), 273–289.

Drexl, M. (2012). Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints. *Transportation Science*, **46**(3), 297–316.

Dror, M. (1994). Note on the complexity of the shortest path models for column generation in vrptw. *Operations Research*, **42**(5), 977–978.

Dumas, Y., Desrosiers, J., and Soumis, F. (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research*, **54**(1), 7–22.

Eveborn, P., Flisberg, P., and Rönnqvist, M. (2006). Laps care – an operational system for staff planning of home care. *European Journal of Operational Research*, **171**(3), 962 – 976.

Gschwind, T. (2015). Route feasibility testing and forward time slack for the synchronized pickup and delivery problem. Technical Report LM-2015-01, Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Mainz, Germany.

Gschwind, T. and Irnich, S. (2014). Effective handling of dynamic time windows and its application to solving the dial-a-ride problem. *Transportation Science*. Forthcoming.

Hunsaker, B. and Savelsbergh, M. (2002). Efficient feasibility testing for dial-a-ride problems. *Operations Research Letters*, **30**(3), 169–173.

Ioachim, I., Desrosiers, J., Soumis, F., and Bélanger, N. (1999). Fleet assignment and routing with schedule synchronization constraints. *European Journal of Operational Research*, **119**(1), 75–90.

Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 33–65. Springer, New York.

Kohl, N., Desrosiers, J., Madsen, O. B. G., Solomon, M. M., and Soumis, F. (1999). 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, **33**(1), 101–116.

Kovacs, A. A., Golden, B. L., Hartl, R. F., and Parragh, S. N. (2014). Vehicle routing problems in which consistency considerations are important: A survey. *Networks*, **64**(3), 192–213.

Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, **53**(6), 1007–1023.

Madsen, O., Ravn, H., and Rygaard, J. (1995). A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research*, **60**, 193–208.

Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2008). A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, **58**(2), 81–117.

Plum, C. E., Pisinger, D., Salazar-González, J.-J., and Sigurd, M. M. (2014). Single liner shipping service design. *Computers & Operations Research*, **45**, 1–6.

Rasmussen, M. S., Justesen, T., Dohn, A., and Larsen, J. (2012). The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, **219**(3), 598–610.

Ropke, S. and Cordeau, J.-F. (2005). Branch and cut and price for the pickup and delivery problem with time windows. Chapter 9 of S. Ropke's cumulative Ph.D. dissertation, Heuristic and exact algorithms for vehicle routing problems, University of Copenhagen, Copenhagen, Denmark.

Ropke, S. and Cordeau, J.-F. (2009). Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, **43**(3), 267–286.

Ropke, S., Cordeau, J.-F., and Laporte, G. (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, **49**(4), 258–272.

Russell, R. A. and Morrel, R. B. (1986). Routing special-education school buses. *Interfaces*, **16**(5), 56–64.

Savelsbergh, M. W. P. (1992). The vehicle routing problem with time windows: Minimizing route duration. *INFORMS Journal on Computing*, **4**(2), 146–154.

Tang, J., Kong, Y., Lau, H., and Ip, A. W. (2010). A note on "efficient feasibility testing for dial-a-ride problems". *Operations Research Letters*, **38**(5), 405–407.

Toth, P. and Vigo, D. (1997). Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science*, **31**(1), 60–71.

Appendix

A. Proofs

For the presentation of the proofs some additional notation is necessary. For any set of numbers M and any number n, let $n+M=\{n+m : m \in M\}$. Moreover, given a sequence $\mathcal{P}=(h_1,...,h_q)$ (path or schedule) and a set of numbers M, let $\mathcal{P}\setminus M$ be the sub-sequence of $\mathcal{P}$ where h_i is removed if $h_i=m$ for all $m \in M$.

We first sketch the proof of Proposition 1 (see Dumas *et al.*, 1991) which is the basis for Propositions 3 and 4.

Proof of Proposition 1: The basic idea of the proof is as follows. Let Q_2 be a feasible completion to ℓ_2, i.e., Q_2 is a extension of $\mathcal{P}_{\ell_2}$ to node $2n+1$ such that the path $\mathcal{P}_2=(\mathcal{P}_{\ell_2},Q_2)$ is feasible. Consider the path $\mathcal{P}_1=(\mathcal{P}_{\ell_1},Q_1)$ where $Q_1=Q_2\setminus\{n+(O_{\ell_2}\setminus O_{\ell_1})\}$ is the completion to ℓ_1 resulting from Q_2 by skipping the delivery nodes of all additional open requests of ℓ_2. Clearly, pairing and precedence constraints are then satisfied by $\mathcal{P}_1$. Using that $\mathcal{P}_2$ is feasible, the relations in Proposition 1 also ensure that $\mathcal{P}_1$ is feasible with respect to capacity and time windows. Furthermore, when the DTI holds the cost of $\mathcal{P}_1$ is always not higher than that of $\mathcal{P}_2$. □

Proof of Lemma 1: $\underline{T}_{\mathcal{P}}$ is feasible if it satisfies inequalities (1)–(3). By definition of $\underline{\tau}_i$, there exists for each $i=1,...,q$ a feasible schedule $T^i_{\mathcal{P}}=(\tau^i_1,...,\underline{\tau}_i,...,\tau^i_q)$ with $\tau^i_j \geq \underline{\tau}_j$ for all $j \neq i$. It follows immediately that $\underline{T}_{\mathcal{P}}$ satisfies (2) for all $i=1,...,q$. Using $\tau^{i+1}_i + t_{h_i h_{i+1}} \leq \underline{\tau}_{i+1}$ and $\underline{\tau}_i \leq \tau^{i+1}_i$ it follows that $\underline{T}_{\mathcal{P}}$ satisfies (1) for all $i=1,...,q-1$. Also, using $\tau^j_i + \underline{L}_{h_i} \leq \underline{\tau}_j$ and $\underline{\tau}_i \leq \tau^j_i$ it follows that $\underline{T}_{\mathcal{P}}$ satisfies (3) for all $h_i = h_j - n$. □

Proof of Proposition 3: The proof is similar to the proof of Proposition 1. Following the same argumentation and using the same notation, it remains to show that $\mathcal{P}_1$ also respects minimum ride-time constraints, i.e., we have to show that there exists a feasible time schedule $T_{\mathcal{P}_1}$ for $\mathcal{P}_1$.

Let $T_{\mathcal{P}_2}=(T_{\mathcal{P}_{\ell_2}},T_{Q_2})$ be a feasible schedule for $\mathcal{P}_2$ with $T_{Q_2}=(\tau_{q+1},...,\tau_r)$. Denote by $\underline{T}_{\mathcal{P}_{\ell_1}}=(\underline{\tau}_1,...,\underline{\tau}_q)\in\mathscr{T}_{\mathcal{P}_{\ell_1}}$ the time schedule for $\mathcal{P}_{\ell_1}$ that minimizes the start of service at all nodes and by $T_{Q_1}=T_{Q_2}\setminus\{\tau_i : h_i-n\in O_{\ell_2}\setminus O_{\ell_1}\}$ the schedule for Q_1 that assigns each node h_i of Q_1 the same start of service τ_i as in T_{Q_2}. Then, using that $\underline{T}_{\mathcal{P}_{\ell_1}}$ and $T_{\mathcal{P}_2}$ are feasible, $\underline{\tau}_q=t_{\ell_1}\leq t_{\ell_2}$, and $ed^{h_i}_{\ell_1}\leq ed^{h_i}_{\ell_2}\leq\tau_i$ for all nodes h_i of Q_1 with $h_i-n\in O_{\ell_1}$ it follows that the schedule $T_{\mathcal{P}_1}=(\underline{T}_{\mathcal{P}_{\ell_1}},T_{Q_1})$ is feasible. □

Proof of Proposition 4: The basic course of the proof is similar to that in the proof of Proposition 3. With the same argumentation and notation, it again remains to show that there exists a feasible time schedule $T_{\mathcal{P}_1}$ for $\mathcal{P}_1$.

Let $\tau^{\ell_2}_q$ be the start of service at the current node $h_q=\eta_{\ell_1}$ within the schedule $T_{\mathcal{P}_2}$. Denote by $\overline{\tau}'_i(\tau^{\ell_2}_q)=\max_{T_{\mathcal{P}_{\ell_1}}\in\mathscr{T}_{\mathcal{P}_{\ell_1}}(\tau^{\ell_2}_q),\tau_j=\tau'_j\,\forall h_j\in O_{\ell_1}\setminus\{h_i\}}\{\tau_i\}$ the latest feasible start of service at node $h_i\in O_{\ell_1}$ given a feasible choice τ'_j for the values $\tau_j, h_j\in O_{\ell_1}\setminus\{h_i\}$, i.e., a choice such that a feasible schedule $T_{\mathcal{P}_{\ell_1}}=(\tau_1,...,\tau_q)\in\mathscr{T}_{\mathcal{P}_{\ell_1}}(\tau^{\ell_2}_q)$ with $\tau_j=\tau'_j$ for all $h_j\in O_{\ell_1}\setminus\{h_i\}$ exists. We first show that $\overline{\tau}'_i(\tau^{\ell_2}_q)\geq\overline{\tau}^O_i(\tau^{\ell_2}_q)$ holds.

The proof is constructive. Denote by $\overline{\tau}'_k(\tau^{\ell_2}_q)$ and $\overline{\tau}^O_k(\tau^{\ell_2}_q)$ also the latest start of service at all other nodes $h_k\neq h_i$ subject to the choices τ'_j and $\underline{\tau}_j$ for all $h_j\in O_{\ell_1}\setminus\{h_i\}$, respectively. Moreover for ease of notation, the explicit functional dependence on the start of service at the current node is omitted in the following, as all considered values relate to schedules with $\tau_q\leq\tau^{\ell_2}_q$. We start from the schedule $\overline{T}'_{\mathcal{P}_{\ell_1}}=(\overline{\tau}'_1,...,\overline{\tau}'_q)\in\mathscr{T}_{\mathcal{P}_{\ell_1}}$. Feasibility of $\overline{T}'_{\mathcal{P}_{\ell_1}}$ follows analog to the proof of Lemma 1. Thus, $\overline{T}'_{\mathcal{P}_{\ell_1}}$ satisfies the constraint system (1)–(4), $\tau_q\leq\tau^{\ell_2}_q$, and $\tau_j=\tau'_j, h_j\in O_{\ell_1}\setminus\{h_i\}$ of the maximization problems of $\overline{\tau}'_k, k=1,...,q$. Replacing equalities $\tau_j=\tau'_j$ by $\tau_j=\underline{\tau}_j$ for all $h_j\in O_{\ell_1}\setminus\{h_i\}$ leads to the constraints system for the maximization problems of $\overline{\tau}^O_i, k=1,...,q$. The modified equalities directly impose stronger bounds on all values τ_{j-1} because of inequalities (1). The case $h_{j-1}\in O_{\ell_1}\setminus\{h_i\}$ can be neglected, as the value of τ_{j-1} is then fixed to $\underline{\tau}_{j-1}$ and consistency of $\underline{\tau}_{j-1}$ and $\underline{\tau}_j$ follows directly from feasibility of the choice $\underline{\tau}_k, h_k \in O_{\ell_1}$ which can again be shown analog to the proof of Lemma 1.

In the case $h_{j-1} \notin O_{\ell_1} \setminus \{h_i\}$, the maximal value at position $j-1$ decreases from $\overline{\tau}'_{j-1}$ to $\overline{\tau}^{O_{\ell_1}}_{j-1} = \min\{\overline{\tau}'_{j-1}, \underline{\tau}_j - t_{h_{j-1}h_j}\} = \overline{\tau}'_{j-1} + \Delta_{j-1}$ with $\Delta_{j-1} \leq 0$. A reduced value $\overline{\tau}^{O_{\ell_1}}_{j-1}$ compared to $\overline{\tau}'_{j-1}$ may in turn influence the maximal values at positions $j-2$, k with $h_k = h_{j-1} - n$, and l with $h_l = h_{j-1} + n$ because of inequalities (1), (3), and (4), respectively. W.l.o.g. consider the impact on the maximal value at position $j-2$ which decreases from $\overline{\tau}'_{j-2}$ to

$$\begin{aligned}
\overline{\tau}^{O_{\ell_1}}_{j-2} &= \min\{\overline{\tau}'_{j-2}, \overline{\tau}^{O_{\ell_1}}_{j-1} - t_{h_{j-2}h_{j-1}}\} \\
&= \overline{\tau}'_{j-2} + \min\{0, \overline{\tau}^{O_{\ell_1}}_{j-1} - t_{h_{j-2}h_{j-1}} - \overline{\tau}'_{j-2}\} \\
&= \overline{\tau}'_{j-2} + \min\{0, \overline{\tau}'_{j-1} + \Delta_{j-1} - t_{h_{j-2}h_{j-1}} - \overline{\tau}'_{j-2}\} \\
&= \overline{\tau}'_{j-2} + \Delta_{j-2},
\end{aligned}$$

with $\Delta_{j-2} \leq 0$. As $\overline{\tau}'_{j-1} - t_{h_{j-2}h_{j-1}} - \overline{\tau}'_{j-2} \geq 0$ it follows that $0 \leq \Delta_{j-2} \leq \Delta_{j-1}$. Thus, the value by which $\overline{\tau}^{O_{\ell_1}}_{j-2}$ is decreased compared to $\overline{\tau}'_{j-2}$ is strictly not larger than the value by which $\overline{\tau}^{O_{\ell_1}}_{j-1}$ is decreased compared to $\overline{\tau}'_{j-1}$. The same result is obtained for the maximal values at positions k with $h_k = h_{j-1} - n$ and l with $h_l = h_{j-1} + n$.

Decreased values $\overline{\tau}^{O_{\ell_1}}_{j-2}$, $\overline{\tau}^{O_{\ell_1}}_k$ with $h_k = h_{j-1} - n$, and $\overline{\tau}^{O_{\ell_1}}_l$ with $h_l = h_{j-1} + n$ in turn constrain the maximal values at several other positions m within the path in an analog fashion as $\overline{\tau}^{O_{\ell_1}}_{j-1}$ constrained themselves as shown above. As a result, we get $\overline{\tau}^{O_{\ell_1}}_m = \overline{\tau}'_m + \Delta_m$ with $\Delta_m \leq 0$. Moreover, $0 \leq \Delta_m \leq \Delta_{j-2}, \Delta_k, \Delta_l$ also holds, i.e., again the decreasing effects on values $\overline{\tau}^{O_{\ell_1}}_m$ compared to $\overline{\tau}'_m$ are strictly not larger than the decreasing effect on $\overline{\tau}^{O_{\ell_1}}_{j-2}$, $\overline{\tau}^{O_{\ell_1}}_k$, and $\overline{\tau}^{O_{\ell_1}}_l$ compared to $\overline{\tau}'_{j-2}$, $\overline{\tau}'_k$, and $\overline{\tau}'_l$, respectively.

Iteratively propagating the stronger conditions resulting from decreased values $\overline{\tau}^{O_{\ell_1}}_k, k = 1, ..., q$ compared to $\overline{\tau}'_k$ in the constraint system allows to construct $\overline{T}^{O_{\ell_1}} = (\overline{\tau}^{O_{\ell_1}}_1, ..., \overline{\tau}^{O_{\ell_1}}_q)$ with $\overline{\tau}^{O_{\ell_1}}_k = \overline{\tau}'_k + \Delta_k$ and $\Delta_k \leq 0$ for all $k = 1, ..., q$. If there are more than just one decreasing effects that have not been propagated yet, they are processed in non-increasing absolute values. Using the property that the decreasing effects Δ_k are non-increasing from propagation step to propagation step, there exists a unique value by which the maximal start of service at each position is decreased and no cycle effects can occur. As a result, $\overline{\tau}'_i(\tau^{\ell_2}_q) \geq \overline{\tau}^{O_{\ell_1}}_i(\tau^{\ell_2}_q)$ holds for any feasible choice τ'_j of the values $\tau_j, h_j \in O_{\ell_1} \setminus \{h_i\}$.

In an analog fashion, we can show that $\underline{\tau}'_i(\tau^{\ell_2}_q) \leq \underline{\tau}^O_i(\tau^{\ell_2}_q)$ holds. Clearly, we also have that $\underline{\tau}'_i(\tau^{\ell_2}_q) \leq \overline{\tau}'_i(\tau^{\ell_2}_q)$. Moreover, it is easy to see that any convex combination of two feasible schedules is also a feasible schedule. Thus, given any feasible choice τ'_j for the values $\tau_j, h_j \in O_{\ell_1} \setminus \{h_i\}$ there exists for each $\tau^*_i \in \left[\underline{\tau}'_i(\tau^{\ell_2}_q), \overline{\tau}'_i(\tau^{\ell_2}_q)\right]$ with $\underline{\tau}'_i(\tau^{\ell_2}_q) \leq \underline{\tau}^O_i(\tau^{\ell_2}_q)$ and $\overline{\tau}'_i(\tau^{\ell_2}_q) \geq \overline{\tau}^{O_{\ell_1}}_i(\tau^{\ell_2}_q)$ a feasible schedule with start of service τ^*_i at node $h_i \in O_{\ell_1}$.

Using this property, we can construct a feasible schedule $T_{\mathcal{P}_1} = (T^*_{\mathcal{P}_{\ell_1}}, T_{Q_1})$ as follows. As defined in the proof of Proposition 3, fix $T_{Q_1} = T_{Q_2} \setminus \{\tau_i : h_i - n \in O_{\ell_2} \setminus O_{\ell_1}\}$ to the schedule for Q_1 that assigns each node h_i of Q_1 the same start of service τ_i as in T_{Q_2}. Let $T_{\mathcal{P}_{\ell_1}} = (\tau_1, ..., \tau_q) \in \mathscr{T}_{\mathcal{P}_{\ell_1}}(\tau^{\ell_2}_q)$. For $T^*_{\mathcal{P}_{\ell_1}} = T_{\mathcal{P}_{\ell_1}}$, schedule $T_{\mathcal{P}_1}$ clearly satisfies inequalities (1) and (2). Also, it satisfies inequalities (3) and (4) for each request $h_i \notin O_{\ell_1}$.

If $T_{\mathcal{P}_1}$ also respects inequalities (3) and (4) for each request $h_i \in O_{\ell_1}$, $T_{\mathcal{P}_1}$ is feasible and the proof is complete. Otherwise, consider any request h_i for which either constraint (3) or (4) is violated. Denote by τ_k the start of service at the delivery node $h_k = h_i + n$ as fixed in schedule T_{Q_1}. To satisfy inequalities (3) and (4), $\tau_i \leq \tau_k - \underline{L}_{h_i}$ and $\tau_i \geq \tau_k - \overline{L}_{h_i}$ must hold, respectively, for the start of service τ_i at pickup node h_i within schedule $T^*_{\mathcal{P}_{\ell_1}}$. Let $\underline{\tau}'_i(\tau^{\ell_2}_q)$ and $\overline{\tau}'_i(\tau^{\ell_2}_q)$ be the minimal and maximal starts of service at node h_i, respectively, given the values $\tau_j, h_j \in O_{\ell_1} \setminus \{h_i\}$ as fixed within schedule $T_{\mathcal{P}_{\ell_1}}$. Then, $\tau_k - \underline{L}_{h_i} \geq ed^{h_i}_{\ell_2} - \underline{L}_{h_i} \geq \overline{ed}^{h_i}_{\ell_1}(\tau^{\ell_2}_q) - \underline{L}_{h_i} \geq \underline{\tau}^O_i(\tau^{\ell_2}_q) \geq \underline{\tau}'_i(\tau^{\ell_2}_q)$ and $\tau_k - \overline{L}_{h_i} \leq ld^{h_i}_{\ell_2}(\tau^{\ell_2}_q) - \overline{L}_{h_i} \leq \underline{ld}^{h_i}_{\ell_1}(\tau^{\ell_2}_q) - \overline{L}_{h_i} \leq \overline{\tau}^{O_{\ell_1}}_i(\tau^{\ell_2}_q) \leq \overline{\tau}'_i(\tau^{\ell_2}_q)$

hold. Using the property shown above, there exists a feasible schedule $T'_{\mathcal{P}_{\ell_1}} = (\tau'_1, ..., \tau'_q)$ with $\tau'_j = \tau_j$ for all $h_j \in O_{\ell_1} \setminus \{h_i\}$ and any $\tau'_i = \tau^*_i \in \left[\underline{\tau}'_i(\tau_q^{\ell_2}), \overline{\tau}'_i(\tau_q^{\ell_2})\right]$. Thus for $T^*_{\mathcal{P}_{\ell_1}} = T'_{\mathcal{P}_{\ell_1}}$, $T_{\mathcal{P}_1}$ satisfies constraints (3) and (4) for request h_i. Moreover, all other constraints that have been respected in the case $T^*_{\mathcal{P}_{\ell_1}} = T_{\mathcal{P}_{\ell_1}}$ are still respected.

If for $T_{\mathcal{P}_1}$ inequalities (3) and (4) are still violated for some requests, we can iteratively repeat the same procedure just described. In each iteration one constraint that was previously violated gets satisfied. Thus, a feasible schedule $T_{\mathcal{P}_1}$ for path $\mathcal{P}_1$ eventually results which completes the proof. $\square$

Proof of Lemma 2: Note first that $\mathcal{X}(t) \notin \varnothing$ for all $t \geq t^*$. Also note that $\overline{X} = (\overline{x}_1, ..., \overline{x}_q) \in \mathcal{X}$, $\underline{X} = (\underline{x}_1, ..., \underline{x}_q) \in \mathcal{X}$, $\overline{X}(t) = (\overline{x}_1(t), ..., \overline{x}_q(t)) \in \mathcal{X}(t)$, $\underline{X}(t) = (\underline{x}_1(t), ..., \underline{x}_q(t)) \in \mathcal{X}(t)$, $\overline{X}^{\underline{S}}(t) = (\overline{x}_1^{\underline{S}}(t), ..., \overline{x}_q^{\underline{S}}(t)) \in \mathcal{X}^{\underline{S}}(t)$, and $\underline{X}^{\overline{S}}(t) = (\underline{x}_1^{\overline{S}}(t), ..., \underline{x}_q^{\overline{S}}(t)) \in \mathcal{X}^{\overline{S}}(t)$. The proofs are analog to the proof of Lemma 1.

1. The proof is constructive. We start from $\overline{X} \in \mathcal{X}$, i.e., the q-tuple with maximal values for all $\overline{x}_i, i = 1, ..., q$ satisfying (26)–(28). When considering $\mathcal{X}(t), t \geq t^*$ instead of $\mathcal{X}$, we have the additional constraint $x_q \leq t$. The case $\overline{x}_q \leq t$ is trivial. If $\overline{x}_q > t$ for some $t \geq t^*$, the maximal value at position q decreases to $\overline{x}_q(t) = \min\{\overline{x}_q, t\} = \overline{x}_q + \min\{0, t - \overline{x}_q\} = \overline{x}_q + \Delta_q(t)$ with $\Delta_q(t) = \min\{k_q^1, k_q^2 + t\} \leq 0$. If inequalities (27) are satisfied for the decreased value $\overline{x}_q(t)$ and all $\overline{x}_i, i < q$, then clearly $\overline{x}_j(t) = \overline{x}_j = \min\{k_i^1, k_i^2 + t\}$ for all $j = 1, ..., q$ with constants k_i^1 and k_i^2.

   If $\overline{x}_i + c_{iq} > \overline{x}_q(t)$ holds for some $i < q$ and $t \geq t^*$, then $\overline{x}_i(t) = \min\{\overline{x}_i, \overline{x}_q(t) - c_{iq}\}$ which can be rewritten in the forms

   $$\begin{aligned}\overline{x}_i(t) &= \min\{\overline{x}_i, \overline{x}_q(t) - c_{iq}\} \\ &= \min\{k_i^1, k_i^2 + t\} \\ &= \overline{x}_i + \min\{0, \overline{x}_q(t) - c_{iq} - \overline{x}_i\} \\ &= \overline{x}_i + \min\{0, \overline{x}_q - c_{iq} - \overline{x}_i + \Delta_q(t)\} \\ &= \overline{x}_i + \Delta_i(t),\end{aligned}$$

   where $\Delta_i(t) = \min\{\tilde{k}_i^1, \tilde{k}_i^2 + t\} \leq 0$ with constants $\tilde{k}_i^1$ and $\tilde{k}_i^2$. As $\overline{x}_q - c_{iq} - \overline{x}_i \geq 0$, it follows that $0 \leq \Delta_i(t) \leq \Delta_q(t)$. Thus, the value by which $\overline{x}_i(t)$ is decreased compared to $\overline{x}_i$ is strictly not larger than the value by which $\overline{x}_q(t)$ is decreased compared to $\overline{x}_q$.

   Decreased values $\overline{x}_i(t)$ in turn impose stronger bounds compared to $\overline{x}_i$ on the maximal values at all positions $j < i$ and $k > i$ because of inequalities (27) and (28), respectively. Thus, the decreasing effects propagate through the constraint system in the same fashion as in the proof of Proposition 4. A schedule $\overline{X}(t) = (\overline{x}_1(t), ..., \overline{x}_q(t)) \in \mathcal{X}(t)$ with $\overline{x}_i(t) = \max\{k_j^1, k_j^2 + t\}\}$ for all $i = 1, ..., q, t \geq t^*$ can then be constructed analog to there.

2. It is straightforward to verify that $\underline{X} \in \mathcal{X}(t)$ and $\mathcal{X}(t) \subseteq \mathcal{X}$ for all $t \geq t^*$. Thus, $\underline{x}_i(t) = \underline{x}_i$ for all $i = 1, ..., q, t \geq t^*$.

3. $\overline{x}_i^{\underline{S}}(t)$ is given by $\max\{x_i\}$, s.t. (26)–(28), $x_q \leq t$, and $x_j \leq \underline{x}_j, j \in \underline{S}$. Obviously, the latter maximization problem is of the same structure as the one for $\overline{x}_i(t)$ and, thus, $\overline{x}_i^{\underline{S}}(t) = \max\{k_i^1, k_i^2 + t\}$ with constants k_i^1 and k_i^2 for all $i = 1, ..., q, t \geq t^*$.

4. Comparing the minimization problems for $\underline{x}_i(t)$ and $\underline{x}_i^{\overline{S}}(t)$, there are additional constraints $x_j \geq \overline{x}_j(t), j \in \overline{S}$ in the latter. When constructing $\underline{X}^{\overline{S}}(t) = (\underline{x}_1^{\overline{S}}(t), ..., \underline{x}_q^{\overline{S}}(t)) \in \mathcal{X}^{\overline{S}}(t)$ starting from $\underline{X}(t) = (\underline{x}_1, ..., \underline{x}_q) \in \mathcal{X}(t)$, these constraints may force $\underline{x}_j^{\overline{S}}(t)$ to be increased compared

to $\underline{x}_j$ for $j \in \overline{S}$ and we have

$$\begin{aligned}\underline{x}_j^{\overline{S}}(t) &= \max\{\underline{x}_j, \overline{x}_j(t)\} \\ &= \max\{\underline{x}_j, \min\{k_j^1, k_j^2 + t\}\} \\ &= \underline{x}_j + \max\{0, \min\{k_j^1, k_j^2 + t\} - \underline{x}_j\} \\ &= \underline{x}_j + \Delta_j(t),\end{aligned}$$

with $\Delta_j(t) = \max\{\tilde{k}_j^1, \min\{\tilde{k}_j^2, \tilde{k}_j^3 + t\}\} \geq 0$. Increased values $\underline{x}_j^{\overline{S}}(t)$ compared to $\underline{x}_j$ might in turn influence other values and the increasing effects propagate throughout the constraint system.

A schedule $\underline{X}^{\overline{S}}(t) = (\underline{x}_1^{\overline{S}}(t), ..., \underline{x}_q^{\overline{S}}(t)) \in \mathcal{X}^{\overline{S}}(t)$ with $\underline{x}_i^{\overline{S}}(t) = \max\{k_j^1, \min\{k_j^2, k_j^3 + t\}\}$ for all $i = 1, ..., q, t \geq t^*$ can then be constructed in the same fashion as $\overline{X}(t)$ was constructed in the proof of the first property. $\square$

Proof of Proposition 5: We need to show that (30) implies $\overline{ed}_{\ell_1}^i(t) \leq ed_{\ell_2}^i, t \in [t_{\ell_2}, \tilde{b}_{\ell_2}]$ and that (31) implies $\underline{ld}_{\ell_1}^i(t) \geq ld_{\ell_2}^i(t), t \in [t_{\ell_2}, \tilde{b}_{\ell_2}]$. The latter follows directly from Proposition 5 of Gschwind and Irnich (2014). For the former, recall that $\overline{ed}_{\ell_1}^i(t) = \max\{k_i^1, \min\{k_i^2, k_i^3 + t\}\}$ and note that $\overline{ed}_{\ell_1}^i(t)$ is clearly non-decreasing in t.

Consider first the case $\overline{B}_{\ell_1}^i \leq \tilde{b}_{\ell_2}$. Using that $\overline{ed}_{\ell_1}^i(t)$ is constant for all $t \geq \overline{B}_{\ell_1}^i$, we have that $\overline{ed}_{\ell_1}^i(t) = \overline{ed}_{\ell_1}^i(\overline{B}_{\ell_1}^i) \leq ed_{\ell_2}^i$ for all $\overline{B}_{\ell_1}^i \leq t \leq \tilde{b}_{\ell_2}$. For all $t_{\ell_2} \leq t \leq \overline{B}_{\ell_1}^i$, clearly $\overline{ed}_{\ell_1}^i(t) \leq \overline{ed}_{\ell_1}^i(\overline{B}_{\ell_1}^i) \leq ed_{\ell_2}^i$ holds.

Consider now the case $\overline{B}_{\ell_1}^i > \tilde{b}_{\ell_2}$. If $\overline{ed}_{\ell_1}^i(\tilde{b}_{\ell_2}) \leq \overline{ed}_{\ell_1}^i(\overline{B}_{\ell_1}^i) - (\overline{B}_{\ell_1}^i - \tilde{b}_{\ell_2})$, then $\overline{ed}_{\ell_1}^i(t) \leq \overline{ed}_{\ell_1}^i(\tilde{b}_{\ell_2}) \leq \overline{ed}_{\ell_1}^i(\overline{B}_{\ell_1}^i) - (\overline{B}_{\ell_1}^i - \tilde{b}_{\ell_2}) \leq ed_{\ell_2}^i$ holds for all $t_{\ell_2} \leq t \leq \tilde{b}_{\ell_2}$. If $\overline{ed}_{\ell_1}^i(\tilde{b}_{\ell_2}) > \overline{ed}_{\ell_1}^i(\overline{B}_{\ell_1}^i) - (\overline{B}_{\ell_1}^i - \tilde{b}_{\ell_2})$ we differentiate two cases. First, if $\overline{B}_{\ell_1}^i = t_{\ell_1}$ it follows directly that $\overline{ed}_{\ell_1}^i(t) = \overline{ed}_{\ell_1}^i(\overline{B}_{\ell_1}^i) = \overline{ed}_{\ell_1}^i(t_{\ell_1}) \leq ed_{\ell_2}^i$ for all $t_{\ell_2} \leq t \leq \tilde{b}_{\ell_2}$. Second, if $\overline{B}_{\ell_1}^i > t_{\ell_1}$ we have that $\overline{ed}_{\ell_1}^i(t) < \overline{ed}_{\ell_1}^i(\overline{B}_{\ell_1}^i)$ for all $t < \overline{B}_{\ell_1}^i$ and consequently $\overline{ed}_{\ell_1}^i(\overline{B}_{\ell_1}^i) = k_i^2 = k_i^3 + \overline{B}_{\ell_1}^i$ must hold. Using $\overline{ed}_{\ell_1}^i(\tilde{b}_{\ell_2}) > \overline{ed}_{\ell_1}^i(\overline{B}_{\ell_1}^i) - (\overline{B}_{\ell_1}^i - \tilde{b}_{\ell_2})$ it follows that $\overline{ed}_{\ell_1}^i(\tilde{b}_{\ell_2}) > k_i^3 + \tilde{b}_{\ell_2}$ and, thus, $\overline{ed}_{\ell_1}^i(\tilde{b}_{\ell_2}) = k_i^1$. As a result, we have $\overline{ed}_{\ell_1}^i(t) = \overline{ed}_{\ell_1}^i(\tilde{b}_{\ell_2}) = k_i^1 = \overline{ed}_{\ell_1}^i(t_{\ell_1}) \leq ed_{\ell_2}^i$ for all $t_{\ell_2} \leq t \leq \tilde{b}_{\ell_2}$. $\square$

B. Detailed computational results

Tables 6–9 show aggregated results for all algorithms and subclasses. In each table, we also report the results averaged only over the larger instances with $n > 80$ requests. The columns have the following meaning:

tree	number of optimal solutions (*opt*), average computation time in seconds (t [s]), and average percentage integrality gap (*% gap*) of the respective branch-and-cut-and-price algorithm
root	number of optimal solutions (*opt*) and average percentage integrality gap (*% gap*) in the root node
% time	average percentage time spent for the solution of the subproblem (*sp*), separation (*sep*), and reoptimization of the master program (*lp*)
# solved	number of solved branch-and-bound nodes (*nd*) and subproblems (*sp*)
# gen	number of generated columns (*col*) and cuts (*cut*)

	algorithm	tree			root		% time			# solved		# gen	
	algorithm	*opt*	t [s]	*% gap*	*opt*	*% gap*	*sp*	*sep*	*lp*	*nd*	*sp*	*col*	*cut*
All instances (54 each)													
A (orig.)	IMP^{max}_{min}	54	63	0.00	34	0.07	75	14	11	12	1232	53591	6
	$IMP\text{-}I^{max}$	54	50	0.00	31	0.08	51	30	19	19	1517	49607	29
	$IMP\text{-}I_{min}$	53	226	0.00	32	0.07	40	28	32	18	1561	56796	96
	IMP-I	51	309	0.00	31	0.09	31	30	40	22	2123	77004	115
B (4/3)	IMP^{max}_{min}	53	154	0.00	36	0.06	75	16	9	16	1595	123213	8
	$IMP\text{-}I^{max}$	52	183	0.00	27	0.09	44	36	21	35	2586	83695	53
	$IMP\text{-}I_{min}$	48	530	0.04	28	0.13	35	23	42	25	1497	58034	201
	IMP-I	48	578	0.04	23	0.14	25	23	51	19	1222	37184	238
C (5/3)	IMP^{max}_{min}	52	266	0.01	27	0.12	79	14	7	25	1880	162287	10
	$IMP\text{-}I^{max}$	51	361	0.01	20	0.20	45	28	26	71	3923	127967	87
	$IMP\text{-}I_{min}$	38	1282	0.20	23	0.37	25	17	57	41	2165	74150	356
	IMP-I	35	1496	0.27	18	0.44	20	18	62	30	2194	66324	397
D (6/3)	IMP^{max}_{min}	50	524	0.04	19	0.20	82	12	6	44	2050	111044	16
	$IMP\text{-}I^{max}$	47	655	0.06	14	0.30	44	26	30	110	5035	177086	123
	$IMP\text{-}I_{min}$	27	1905	0.59	14	0.79	24	13	63	56	2326	69789	532
	IMP-I	28	1887	0.70	12	0.89	18	12	70	19	1564	54454	574
Instances with $n > 80$ (12 each)													
A (orig.)	IMP^{max}_{min}	12	270	0.00	2	0.16	88	3	9	44	4761	217909	8
	$IMP\text{-}I^{max}$	12	199	0.00	2	0.19	66	13	21	52	5230	188930	46
	$IMP\text{-}I_{min}$	11	971	0.00	2	0.16	42	8	50	63	5976	227597	206
	IMP-I	9	1254	0.01	2	0.19	32	13	55	54	6529	242554	236
B (4/3)	IMP^{max}_{min}	11	667	0.01	4	0.12	80	14	6	49	5924	518448	18
	$IMP\text{-}I^{max}$	10	787	0.02	3	0.18	52	21	26	102	9254	328495	113
	$IMP\text{-}I_{min}$	7	1660	0.19	2	0.32	30	14	56	68	4351	199817	443
	IMP-I	7	1834	0.19	2	0.29	17	9	74	31	2617	103413	529
C (5/3)	IMP^{max}_{min}	10	1028	0.04	2	0.22	89	6	4	46	5581	622632	20
	$IMP\text{-}I^{max}$	9	1281	0.07	1	0.31	54	12	35	124	9287	370515	182
	$IMP\text{-}I_{min}$	3	3144	0.71	1	0.81	17	5	79	23	2372	134757	775
	IMP-I	1	3503	0.96	0	1.03	15	7	78	16	2222	120613	813
D (6/3)	IMP^{max}_{min}	8	1856	0.17	0	0.39	95	3	2	80	5351	345545	31
	$IMP\text{-}I^{max}$	6	2078	0.24	0	0.56	54	6	39	192	13187	572858	233
	$IMP\text{-}I_{min}$	0	3600	1.79	0	1.79	28	8	64	4	769	90067	1016
	IMP-I	0	3600	2.12	0	2.13	24	7	69	4	949	93847	1087

Table 6: Results aggregated by subclass for MM instances

	algorithm	tree			root		% time			# solved		# gen	
		opt	*t* [s]	*% gap*	*opt*	*% gap*	*sp*	*sep*	*lp*	*nd*	*sp*	*col*	*cut*
All instances (54 each)													
A (orig.)	IMP^{max}_{min}	54	24	0.00	33	0.04	76	15	9	6	519	23733	7
	$IMP\text{-}I^{max}$	54	31	0.00	31	0.06	50	31	20	10	874	33537	23
	$IMP\text{-}I_{min}$	52	264	0.01	30	0.07	39	27	34	10	1523	73134	111
	$IMP\text{-}I$	52	228	0.01	28	0.09	36	29	35	11	877	30538	118
B (4/3)	IMP^{max}_{min}	52	177	0.00	27	0.06	76	16	8	19	1593	112939	10
	$IMP\text{-}I^{max}$	52	167	0.00	24	0.08	51	31	18	33	2406	107659	37
	$IMP\text{-}I_{min}$	46	648	0.06	22	0.15	28	22	50	16	1247	47057	236
	$IMP\text{-}I$	45	722	0.07	21	0.17	25	22	53	18	1404	48272	251
C (5/3)	IMP^{max}_{min}	50	441	0.01	23	0.13	76	16	8	27	2486	231066	12
	$IMP\text{-}I^{max}$	50	366	0.01	22	0.17	52	31	17	65	3999	184217	58
	$IMP\text{-}I_{min}$	34	1475	0.32	17	0.50	26	16	58	23	1874	73298	400
	$IMP\text{-}I$	34	1598	0.35	17	0.54	19	19	63	30	2220	75353	419
D (6/3)	IMP^{max}_{min}	45	813	0.07	16	0.27	83	10	6	73	3838	275707	16
	$IMP\text{-}I^{max}$	46	701	0.06	14	0.35	55	24	22	142	6710	291948	85
	$IMP\text{-}I_{min}$	26	2014	0.77	13	0.99	25	13	62	25	1946	75526	552
	$IMP\text{-}I$	25	2129	0.84	12	1.11	20	14	66	28	2347	84728	585
Instances with $n > 80$ (12 each)													
A (orig.)	IMP^{max}_{min}	12	87	0.00	3	0.07	84	8	8	13	1310	75719	11
	$IMP\text{-}I^{max}$	12	120	0.00	3	0.08	61	22	17	27	2349	111207	48
	$IMP\text{-}I_{min}$	10	1098	0.03	2	0.15	36	11	53	30	2987	137530	247
	$IMP\text{-}I$	10	928	0.03	2	0.17	31	14	56	29	2505	103290	262
B (4/3)	IMP^{max}_{min}	10	758	0.01	1	0.13	84	10	6	59	5899	464365	24
	$IMP\text{-}I^{max}$	10	729	0.00	0	0.17	64	14	22	118	9475	445407	88
	$IMP\text{-}I_{min}$	6	2118	0.25	1	0.35	25	7	68	28	2610	128091	544
	$IMP\text{-}I$	5	2306	0.26	0	0.39	20	5	75	31	3294	143883	540
C (5/3)	IMP^{max}_{min}	8	1827	0.06	0	0.28	92	4	4	78	9136	941338	20
	$IMP\text{-}I^{max}$	9	1265	0.04	0	0.32	73	7	20	158	12899	660848	112
	$IMP\text{-}I_{min}$	0	3600	1.05	0	1.15	17	5	78	20	2289	147984	821
	$IMP\text{-}I$	1	3447	1.10	0	1.20	15	6	79	19	2005	127869	846
D (6/3)	IMP^{max}_{min}	5	2560	0.25	0	0.46	94	3	3	86	7778	865610	33
	$IMP\text{-}I^{max}$	6	2185	0.22	0	0.55	67	8	25	218	16101	921174	170
	$IMP\text{-}I_{min}$	0	3600	2.15	0	2.15	34	8	58	4	855	116722	1042
	$IMP\text{-}I$	0	3600	2.30	0	2.30	24	8	68	4	926	116781	1069

Table 7: Results aggregated by subclass for LM instances

	algorithm	tree			root		% time			# solved		# gen	
		opt	t [s]	% gap	opt	% gap	sp	sep	lp	nd	sp	col	cut
All instances (54 each)													
A (orig.)	IMP^{max}_{min}	52	223	0.00	29	0.09	79	14	7	28	2629	125535	7
	$IMP\text{-}I^{max}$	51	325	0.00	25	0.13	49	31	20	66	4860	159192	45
	$IMP\text{-}I_{min}$	47	500	0.02	29	0.14	39	30	31	41	2383	76904	108
	$IMP\text{-}I$	47	554	0.03	25	0.17	31	32	37	36	2207	63504	134
B (4/3)	IMP^{max}_{min}	51	284	0.01	23	0.10	80	13	6	17	1338	89769	8
	$IMP\text{-}I^{max}$	51	312	0.02	19	0.15	53	30	17	50	3276	120326	56
	$IMP\text{-}I_{min}$	44	795	0.09	18	0.24	35	26	38	40	2375	84908	200
	$IMP\text{-}I$	43	973	0.12	16	0.29	26	28	47	41	2484	74332	233
C (5/3)	IMP^{max}_{min}	48	691	0.05	16	0.25	82	14	3	32	2238	156461	13
	$IMP\text{-}I^{max}$	45	850	0.07	11	0.39	55	24	21	133	6938	247400	107
	$IMP\text{-}I_{min}$	34	1563	0.35	10	0.69	32	19	50	70	3411	109603	314
	$IMP\text{-}I$	33	1731	0.45	7	0.81	23	17	61	67	3988	116236	355
D (6/3)	IMP^{max}_{min}	40	1064	0.36	12	0.58	88	9	3	33	1437	81136	16
	$IMP\text{-}I^{max}$	37	1292	0.40	9	0.87	58	20	22	201	8884	319439	131
	$IMP\text{-}I_{min}$	22	2292	1.17	8	1.60	39	13	48	78	4225	147475	406
	$IMP\text{-}I$	20	2412	1.36	8	1.85	32	13	55	85	5003	158416	439
Instances with $n > 80$ (12 each)													
A (orig.)	IMP^{max}_{min}	10	761	0.01	2	0.14	87	7	7	69	6383	321086	16
	$IMP\text{-}I^{max}$	10	1120	0.01	2	0.21	63	13	24	181	14178	527528	95
	$IMP\text{-}I_{min}$	7	1581	0.05	2	0.24	43	13	45	100	6880	249099	225
	$IMP\text{-}I$	7	1757	0.09	2	0.30	35	13	52	88	6683	217550	258
B (4/3)	IMP^{max}_{min}	9	1104	0.03	1	0.19	89	7	3	41	3791	317222	19
	$IMP\text{-}I^{max}$	9	1169	0.07	0	0.34	69	10	21	148	9688	408005	129
	$IMP\text{-}I_{min}$	5	2227	0.33	0	0.57	34	7	58	83	4983	216713	410
	$IMP\text{-}I$	5	2668	0.44	0	0.70	24	4	72	95	6412	225071	464
C (5/3)	IMP^{max}_{min}	6	2200	0.23	2	0.41	97	1	2	58	4044	326811	23
	$IMP\text{-}I^{max}$	5	2638	0.29	0	0.65	72	6	22	282	17455	735455	192
	$IMP\text{-}I_{min}$	1	3363	1.22	0	1.54	38	2	60	75	4073	218260	524
	$IMP\text{-}I$	1	3523	1.45	0	1.75	31	3	66	80	4590	225975	584
D (6/3)	IMP^{max}_{min}	1	3338	1.36	0	1.56	98	1	1	48	2757	215131	31
	$IMP\text{-}I^{max}$	0	3600	1.50	0	1.97	78	1	20	309	17996	842792	226
	$IMP\text{-}I_{min}$	0	3600	3.64	0	3.88	64	4	32	60	2891	238828	492
	$IMP\text{-}I$	0	3600	4.02	0	4.34	65	3	32	105	5120	321742	477

Table 8: Results aggregated by subclass for ML instances

		tree			*root*		*% time*			*# solved*		*# gen*	
	algorithm	*opt*	*t* [s]	*% gap*	*opt*	*% gap*	*sp*	*sep*	*lp*	*nd*	*sp*	*col*	*cut*
All instances (54 each)													
A (orig.)	IMP^{max}_{min}	52	359	0.00	32	0.09	79	13	9	31	4465	343174	7
	$IMP\text{-}I^{max}$	53	264	0.00	27	0.11	52	30	18	65	4699	181112	29
	$IMP\text{-}I_{min}$	46	566	0.03	29	0.15	44	24	32	44	2714	85345	109
	IMP-I	46	608	0.04	24	0.16	37	28	36	45	2787	84979	122
B (4/3)	IMP^{max}_{min}	50	354	0.00	22	0.14	81	14	6	27	2236	127133	8
	$IMP\text{-}I^{max}$	53	305	0.00	19	0.16	56	30	14	57	4406	197591	36
	$IMP\text{-}I_{min}$	41	968	0.11	16	0.35	37	25	39	65	4426	160455	191
	IMP-I	39	1160	0.15	14	0.38	30	24	46	76	5423	181615	214
C (5/3)	IMP^{max}_{min}	43	843	0.10	18	0.30	85	12	4	39	5042	423629	15
	$IMP\text{-}I^{max}$	45	720	0.09	16	0.36	61	23	16	122	6330	281715	63
	$IMP\text{-}I_{min}$	32	1597	0.48	13	0.83	37	17	46	105	4434	165241	299
	IMP-I	32	1687	0.52	12	0.85	29	18	53	84	4449	148470	339
D (6/3)	IMP^{max}_{min}	40	1170	0.16	14	0.41	89	9	2	38	2153	120897	17
	$IMP\text{-}I^{max}$	41	1005	0.14	11	0.52	70	18	12	127	7072	347608	73
	$IMP\text{-}I_{min}$	22	2248	1.07	10	1.50	43	13	44	88	4597	176525	368
	IMP-I	20	2319	1.16	10	1.58	39	14	47	83	4615	173360	389
Instances with $n > 80$ (12 each)													
A (orig.)	IMP^{max}_{min}	10	1373	0.02	4	0.16	86	7	7	81	15551	1379814	13
	$IMP\text{-}I^{max}$	11	1046	0.00	3	0.22	62	22	16	224	16884	704345	62
	$IMP\text{-}I_{min}$	7	1583	0.08	3	0.28	42	15	43	101	6939	248986	223
	IMP-I	7	1745	0.12	2	0.31	35	16	49	99	7382	261273	243
B (4/3)	IMP^{max}_{min}	8	1323	0.01	0	0.16	90	7	3	76	7646	476477	17
	$IMP\text{-}I^{max}$	11	1098	0.01	0	0.19	66	21	12	150	14896	745414	71
	$IMP\text{-}I_{min}$	4	2564	0.37	0	0.69	45	4	52	141	10489	486350	358
	IMP-I	4	2800	0.44	0	0.75	37	4	59	147	11438	504402	391
C (5/3)	IMP^{max}_{min}	5	2166	0.39	1	0.57	93	5	2	61	3775	269917	26
	$IMP\text{-}I^{max}$	5	2285	0.37	0	0.69	79	8	13	251	16056	815782	118
	$IMP\text{-}I_{min}$	1	3335	1.47	0	1.84	46	2	51	105	5207	315998	481
	IMP-I	1	3429	1.51	0	1.83	36	3	61	101	5359	294695	585
D (6/3)	IMP^{max}_{min}	2	3193	0.59	0	0.81	98	1	1	47	2852	236242	34
	$IMP\text{-}I^{max}$	2	3114	0.54	0	0.96	87	2	11	241	15770	933740	134
	$IMP\text{-}I_{min}$	0	3600	2.94	0	3.27	69	3	29	85	3670	283771	428
	IMP-I	0	3600	3.09	0	3.43	67	5	28	101	4498	322553	435

Table 9: Results aggregated by subclass for LL instances

Tables 10-13 present optimal solution/lower bound values for all instances together with the respective computation times of algorithms IMP_{min}^{max} and $IMP\text{-}I^{max}$. The meaning of the table entries are as follows:

inst	name of the instance
opt	indicates if the instances is solved to proven optimality
lb	value of best known lower bound
IMP_{min}^{max}	computation time in seconds of algorithm IMP_{min}^{max}, *1h* if unable to solve instance within time limit
$IMP\text{-}I^{max}$	computation time in seconds of algorithm $IMP\text{-}I^{max}$, *1h* if unable to solve instance within time limit

MM					LM					ML					LL				
inst	*opt*	*lb*	IMP_{min}^{max}	$IMP\text{-}I^{max}$	*inst*	*opt*	*lb*	IMP_{min}^{max}	$IMP\text{-}I^{max}$	*inst*	*opt*	*lb*	IMP_{min}^{max}	$IMP\text{-}I^{max}$	*inst*	*opt*	*lb*	IMP_{min}^{max}	$IMP\text{-}I^{max}$
MM-A-a2-16	*	325.2	0.0	0.0	LM-A-a2-16	*	296.4	0.0	0.0	ML-A-a2-16	*	293.4	0.0	0.0	LL-A-a2-16	*	291.7	0.1	0.0
MM-A-a2-20	*	362.1	0.1	0.1	LM-A-a2-20	*	360.7	0.1	0.1	ML-A-a2-20	*	334.2	0.1	0.0	LL-A-a2-20	*	332.6	0.1	0.2
MM-A-a2-24	*	423.9	0.2	0.1	LM-A-a2-24	*	423.2	0.2	0.1	ML-A-a2-24	*	414.7	0.5	0.6	LL-A-a2-24	*	410.8	0.4	0.3
MM-A-a3-24	*	325.8	0.1	0.1	LM-A-a3-24	*	318.9	0.3	0.2	ML-A-a3-24	*	319.0	0.4	0.3	LL-A-a3-24	*	313.4	0.1	0.0
MM-A-a3-30	*	510.2	0.8	1.1	LM-A-a3-30	*	489.3	0.9	2.0	ML-A-a3-30	*	465.1	0.3	0.2	LL-A-a3-30	*	453.3	1.9	1.4
MM-A-a3-36	*	558.5	0.6	0.2	LM-A-a3-36	*	558.5	0.5	0.2	ML-A-a3-36	*	542.3	0.5	0.3	LL-A-a3-36	*	537.8	1.0	0.2
MM-A-a4-32	*	485.0	0.2	0.1	LM-A-a4-32	*	477.4	0.2	0.1	ML-A-a4-32	*	477.0	0.2	0.9	LL-A-a4-32	*	474.4	0.3	0.1
MM-A-a4-40	*	590.1	1.0	0.7	LM-A-a4-40	*	568.8	1.4	0.7	ML-A-a4-40	*	571.2	0.7	0.9	LL-A-a4-40	*	560.7	1.3	1.4
MM-A-a4-48	*	677.3	0.9	2.3	LM-A-a4-48	*	657.0	2.2	1.2	ML-A-a4-48	*	641.4	1.6	0.9	LL-A-a4-48	*	635.7	10.1	7.2
MM-A-a5-40	*	506.3	0.4	0.1	LM-A-a5-40	*	497.0	0.3	0.1	ML-A-a5-40	*	494.1	0.9	0.6	LL-A-a5-40	*	494.1	1.0	0.6
MM-A-a5-50	*	673.3	13.6	50.3	LM-A-a5-50	*	664.6	6.7	3.9	ML-A-a5-50	*	646.7	2.5	6.1	LL-A-a5-50	*	641.5	3.8	2.7
MM-A-a5-60	*	817.4	2.3	2.9	LM-A-a5-60	*	783.4	3.1	3.0	ML-A-a5-60	*	754.3	234.3	7.7	LL-A-a5-60	*	752.6	15.6	10.2
MM-A-a6-48	*	637.9	0.6	1.0	LM-A-a6-48	*	621.5	0.7	1.1	ML-A-a6-48	*	611.6	1.1	1.8	LL-A-a6-48	*	601.0	4.2	2.7
MM-A-a6-60	*	786.6	4.9	3.2	LM-A-a6-60	*	778.8	4.1	2.3	ML-A-a6-60	*	750.6	5.2	4.4	LL-A-a6-60	*	748.2	6.7	4.4
MM-A-a6-72	*	930.8	27.4	15.2	LM-A-a6-72	*	918.2	19.3	16.4	ML-A-a6-72	*	908.5	2359.2	*1h*	LL-A-a6-72	*	899.4	1847.8	1161.1
MM-A-a7-56	*	716.9	3.1	2.5	LM-A-a7-56	*	703.5	0.8	0.4	ML-A-a7-56	*	680.0	34.3	31.8	LL-A-a7-56	*	674.0	8.8	7.6
MM-A-a7-70	*	887.5	5.6	8.9	LM-A-a7-70	*	874.7	27.1	18.2	ML-A-a7-70	*	838.0	13.6	15.3	LL-A-a7-70	*	828.5	22.2	9.7
MM-A-a7-84	*	1031.3	684.6	276.1	LM-A-a7-84	*	1016.7	84.1	63.5	ML-A-a7-84	*	985.2	149.2	296.2	LL-A-a7-84	*	981.3	*1h*	2701.7
MM-A-a8-64	*	757.0	5.8	4.4	LM-A-a8-64	*	746.8	5.0	5.8	ML-A-a8-64	*	728.3	30.9	18.4	LL-A-a8-64	*	717.3	4.8	4.8
MM-A-a8-80	*	972.7	7.6	6.7	LM-A-a8-80	*	959.0	9.9	5.6	ML-A-a8-80	*	933.7	4.6	7.8	LL-A-a8-80	*	908.3	57.6	23.0
MM-A-a8-96	*	1187.0	196.6	80.8	LM-A-a8-96	*	1176.4	40.1	33.0	ML-A-a8-96	*	1150.0	*1h*	3430.6	LL-A-a8-96	*	1138.8	1356.4	1884.4
MM-A-a9-72	*	899.7	4.8	21.6	LM-A-a9-72	*	881.9	13.1	9.1	ML-A-a9-72	*	847.4	90.7	282.4	LL-A-a9-72	*	829.6	95.7	154.3
MM-A-a9-90	*	1042.3	551.8	731.9	LM-A-a9-90	*	1029.0	109.0	116.0	ML-A-a9-90	*	996.4	234.5	443.5	LL-A-a9-90	*	985.9	2578.6	3567.2
MM-A-a9-108	*	1188.6	118.6	46.6	LM-A-a9-108	*	1174.7	30.4	16.5	ML-A-a9-108		1135.1	*1h*	*1h*	LL-A-a9-108	*	1120.8	38.5	29.6
MM-A-a10-80	*	931.2	8.8	22.5	LM-A-a10-80	*	919.4	4.9	5.1	ML-A-a10-80	*	888.7	38.7	51.2	LL-A-a10-80	*	892.3	700.1	220.1
MM-A-a10-100	*	1221.2	182.3	47.5	LM-A-a10-100	*	1214.3	100.3	53.5	ML-A-a10-100	*	1162.9	193.2	1163.1	LL-A-a10-100	*	1149.5	345.1	617.0
MM-A-a10-120	*	1344.5	1097.1	464.9	LM-A-a10-120	*	1336.4	483.9	901.8	ML-A-a10-120	*	1283.4	408.6	*1h*	LL-A-a10-120		1280.9	*1h*	*1h*
MM-A-b2-16	*	302.2	0.0	0.1	LM-A-b2-16	*	302.2	0.0	0.1	ML-A-b2-16	*	298.2	0.0	0.0	LL-A-b2-16	*	298.2	0.0	0.0
MM-A-b2-20	*	319.7	0.1	0.1	LM-A-b2-20	*	319.7	0.1	0.1	ML-A-b2-20	*	337.6	0.1	0.1	LL-A-b2-20	*	337.6	0.1	0.0
MM-A-b2-24	*	436.1	0.1	0.1	LM-A-b2-24	*	436.1	0.1	0.0	ML-A-b2-24	*	432.6	0.1	0.1	LL-A-b2-24	*	429.9	0.1	0.0
MM-A-b3-24	*	382.8	0.1	0.0	LM-A-b3-24	*	382.8	0.1	0.0	ML-A-b3-24	*	382.8	0.1	0.0	LL-A-b3-24	*	382.8	0.1	0.0
MM-A-b3-30	*	537.0	0.1	0.1	LM-A-b3-30	*	533.0	0.1	0.1	ML-A-b3-30	*	517.4	0.3	0.2	LL-A-b3-30	*	509.5	0.1	0.1
MM-A-b3-36	*	584.4	0.2	0.1	LM-A-b3-36	*	593.1	0.3	0.1	ML-A-b3-36	*	584.0	0.8	0.6	LL-A-b3-36	*	588.7	0.3	0.1
MM-A-b4-32	*	543.4	0.2	0.3	LM-A-b4-32	*	539.8	0.2	0.1	ML-A-b4-32	*	537.7	0.2	0.1	LL-A-b4-32	*	534.2	0.1	0.1
MM-A-b4-40	*	666.7	0.7	0.6	LM-A-b4-40	*	676.4	2.1	2.9	ML-A-b4-40	*	657.2	0.2	0.2	LL-A-b4-40	*	667.0	0.4	0.2
MM-A-b4-48	*	693.6	1.5	0.9	LM-A-b4-48	*	700.4	2.2	1.5	ML-A-b4-48	*	691.2	1.2	0.8	LL-A-b4-48	*	697.9	2.6	1.5
MM-A-b5-40	*	633.2	2.4	1.6	LM-A-b5-40	*	627.7	1.9	1.2	ML-A-b5-40	*	623.8	1.1	0.8	LL-A-b5-40	*	618.3	0.4	0.2
MM-A-b5-50	*	805.1	0.7	1.2	LM-A-b5-50	*	798.5	1.2	2.5	ML-A-b5-50	*	788.4	1.2	1.6	LL-A-b5-50	*	781.3	1.3	2.2
MM-A-b5-60	*	972.7	5.8	6.1	LM-A-b5-60	*	941.0	14.4	21.4	ML-A-b5-60	*	937.2	9.3	9.0	LL-A-b5-60	*	930.5	10.0	9.3
MM-A-b6-48	*	747.3	0.9	1.0	LM-A-b6-48	*	729.7	0.3	0.3	ML-A-b6-48	*	733.6	1.0	1.0	LL-A-b6-48	*	718.9	0.4	0.1
MM-A-b6-60	*	898.0	1.3	2.2	LM-A-b6-60	*	889.3	1.2	0.4	ML-A-b6-60	*	895.1	9.4	9.6	LL-A-b6-60	*	884.1	1.3	0.5
MM-A-b6-72	*	1015.6	37.1	29.2	LM-A-b6-72	*	1000.4	92.7	100.4	ML-A-b6-72	*	999.6	35.0	31.7	LL-A-b6-72	*	990.8	70.0	29.3
MM-A-b7-56	*	865.1	0.7	0.3	LM-A-b7-56	*	853.3	2.6	1.7	ML-A-b7-56	*	843.6	2.1	2.7	LL-A-b7-56	*	834.6	9.2	8.2
MM-A-b7-70	*	982.6	6.2	9.8	LM-A-b7-70	*	950.5	3.5	5.6	ML-A-b7-70	*	973.1	4.5	8.4	LL-A-b7-70	*	946.4	4.9	11.3
MM-A-b7-84	*	1283.1	67.6	48.5	LM-A-b7-84	*	1272.6	27.7	17.7	ML-A-b7-84	*	1261.9	220.9	75.0	LL-A-b7-84	*	1240.6	2034.5	37.6
MM-A-b8-64	*	888.8	4.1	3.1	LM-A-b8-64	*	869.0	3.7	2.7	ML-A-b8-64	*	881.0	5.1	6.0	LL-A-b8-64	*	861.5	3.6	2.7
MM-A-b8-80	*	1105.4	2.7	7.5	LM-A-b8-80	*	1088.2	5.6	11.5	ML-A-b8-80	*	1090.8	2.8	8.0	LL-A-b8-80	*	1074.1	6.1	8.9
MM-A-b8-96	*	1240.2	14.1	7.9	LM-A-b8-96	*	1237.0	55.1	69.1	ML-A-b8-96	*	1228.0	27.5	20.0	LL-A-b8-96	*	1215.0	10.4	18.3
MM-A-b9-72	*	998.4	4.3	14.2	LM-A-b9-72	*	988.2	4.6	2.9	ML-A-b9-72	*	983.7	3.3	8.4	LL-A-b9-72	*	976.3	6.9	9.7
MM-A-b9-90	*	1254.1	4.5	5.1	LM-A-b9-90	*	1230.1	5.3	8.9	ML-A-b9-90	*	1234.4	14.5	9.1	LL-A-b9-90	*	1217.5	25.6	19.3
MM-A-b9-108	*	1490.3	75.7	186.0	LM-A-b9-108	*	1458.3	34.4	64.8	ML-A-b9-108	*	1433.5	218.4	414.5	LL-A-b9-108	*	1419.4	93.3	29.6
MM-A-b10-80	*	1014.2	5.3	59.8	LM-A-b10-80	*	1004.0	5.3	5.6	ML-A-b10-80	*	977.9	8.5	7.7	LL-A-b10-80	*	974.6	7.2	5.3
MM-A-b10-100	*	1378.7	26.0	26.4	LM-A-b10-100	*	1351.0	51.1	54.7	ML-A-b10-100	*	1342.1	14.3	11.7	LL-A-b10-100	*	1328.1	19.5	12.3
MM-A-b10-120	*	1641.3	216.5	471.0	LM-A-b10-120	*	1613.5	23.4	37.8	ML-A-b10-120	*	1602.5	448.6	375.2	LL-A-b10-120	*	1582.4	2770.4	29.4

Table 10: Results for class A (orig.) instances

MM					LM					ML					LL				
inst	*opt*	*lb*	IMP^{max}_{min}	*IMP*-I^{max}	*inst*	*opt*	*lb*	IMP^{max}_{min}	*IMP*-I^{max}	*inst*	*opt*	*lb*	IMP^{max}_{min}	*IMP*-I^{max}	*inst*	*opt*	*lb*	IMP^{max}_{min}	*IMP*-I^{max}
MM-B-a2-16	*	302.6	0.0	0.0	LM-B-a2-16	*	296.4	0.0	0.0	ML-B-a2-16	*	293.4	0.1	0.0	LL-B-a2-16	*	291.7	0.1	0.0
MM-B-a2-20	*	358.8	0.1	0.0	LM-B-a2-20	*	344.7	0.1	0.2	ML-B-a2-20	*	332.0	0.3	0.2	LL-B-a2-20	*	332.0	0.4	0.3
MM-B-a2-24	*	414.9	0.2	0.3	LM-B-a2-24	*	413.0	0.5	0.4	ML-B-a2-24	*	410.5	0.8	0.8	LL-B-a2-24	*	406.6	0.6	0.5
MM-B-a3-24	*	303.2	0.1	0.1	LM-B-a3-24	*	303.2	0.1	0.1	ML-B-a3-24	*	296.3	0.2	0.1	LL-B-a3-24	*	296.3	0.2	0.1
MM-B-a3-30	*	490.3	1.2	1.0	LM-B-a3-30	*	481.3	0.3	1.0	ML-B-a3-30	*	455.2	4.1	2.4	LL-B-a3-30	*	449.7	4.1	1.8
MM-B-a3-36	*	544.1	1.1	0.7	LM-B-a3-36	*	534.1	1.6	0.8	ML-B-a3-36	*	529.9	1.7	1.5	LL-B-a3-36	*	519.7	2.2	1.0
MM-B-a4-32	*	484.6	0.2	0.9	LM-B-a4-32	*	470.2	0.2	0.1	ML-B-a4-32	*	461.7	0.3	0.2	LL-B-a4-32	*	454.8	1.1	0.9
MM-B-a4-40	*	571.3	0.6	0.2	LM-B-a4-40	*	563.9	4.3	1.9	ML-B-a4-40	*	558.6	1.0	1.6	LL-B-a4-40	*	554.0	22.9	13.7
MM-B-a4-48	*	664.7	14.5	4.6	LM-B-a4-48	*	653.3	18.8	13.1	ML-B-a4-48	*	622.6	2.9	1.2	LL-B-a4-48	*	621.2	15.4	4.8
MM-B-a5-40	*	498.3	1.0	0.8	LM-B-a5-40	*	496.8	1.1	0.5	ML-B-a5-40	*	484.0	1.0	0.7	LL-B-a5-40	*	483.2	2.6	1.7
MM-B-a5-50	*	653.4	2.2	1.6	LM-B-a5-50	*	644.5	5.0	3.1	ML-B-a5-50	*	630.2	7.9	9.0	LL-B-a5-50	*	627.3	54.6	68.0
MM-B-a5-60	*	764.9	5.5	14.6	LM-B-a5-60	*	745.8	9.8	3.3	ML-B-a5-60	*	729.7	39.5	12.7	LL-B-a5-60	*	728.8	135.0	36.8
MM-B-a6-48	*	612.3	1.0	1.3	LM-B-a6-48	*	610.2	2.5	1.0	ML-B-a6-48	*	585.3	4.1	2.3	LL-B-a6-48	*	576.1	2.2	1.9
MM-B-a6-60	*	768.8	7.6	6.6	LM-B-a6-60	*	754.4	7.8	6.0	ML-B-a6-60	*	723.9	6.5	6.2	LL-B-a6-60	*	713.8	13.7	4.8
MM-B-a6-72	*	904.8	9.1	16.0	LM-B-a6-72	*	896.2	13.8	12.0	ML-B-a6-72	*	874.6	1217.0	2247.9	LL-B-a6-72	*	859.1	15.8	20.7
MM-B-a7-56	*	692.1	2.3	3.4	LM-B-a7-56	*	691.1	4.9	8.0	ML-B-a7-56	*	666.7	22.0	24.2	LL-B-a7-56	*	639.7	11.1	5.3
MM-B-a7-70	*	864.9	6.9	11.7	LM-B-a7-70	*	853.0	23.0	11.0	ML-B-a7-70	*	813.8	7.7	8.8	LL-B-a7-70	*	802.4	9.0	12.6
MM-B-a7-84	*	998.5	26.6	26.7	LM-B-a7-84	*	985.5	74.4	79.4	ML-B-a7-84	*	947.1	186.3	115.1	LL-B-a7-84	*	937.9	399.8	621.0
MM-B-a8-64	*	737.2	4.3	4.6	LM-B-a8-64	*	726.4	18.7	13.2	ML-B-a8-64	*	703.7	14.9	10.6	LL-B-a8-64	*	700.0	34.9	20.2
MM-B-a8-80	*	954.5	8.2	7.5	LM-B-a8-80	*	930.0	17.6	13.7	ML-B-a8-80	*	897.9	43.1	204.6	LL-B-a8-80	*	878.8	125.3	89.8
MM-B-a8-96		1159.6	*1h*	*1h*	LM-B-a8-96		1145.9	*1h*	*1h*	ML-B-a8-96	*	1095.8	159.9	961.0	LL-B-a8-96	*	1084.5	189.0	127.9
MM-B-a9-72	*	871.0	3.0	4.1	LM-B-a9-72	*	850.7	223.7	112.6	ML-B-a9-72	*	803.3	19.0	13.5	LL-B-a9-72	*	800.4	288.3	204.3
MM-B-a9-90	*	1008.4	19.3	13.8	LM-B-a9-90	*	1007.6	812.4	637.3	ML-B-a9-90	*	955.2	171.4	161.0	LL-B-a9-90	*	950.7	67.4	61.7
MM-B-a9-108	*	1158.8	2922.1	*1h*	LM-B-a9-108	*	1136.2	67.1	194.3	ML-B-a9-108		1095.9	*1h*	*1h*	LL-B-a9-108	*	1083.3	*1h*	3193.9
MM-B-a10-80	*	919.2	25.5	20.4	LM-B-a10-80	*	899.3	10.9	6.9	ML-B-a10-80	*	869.4	419.3	132.4	LL-B-a10-80	*	863.4	2263.0	2633.1
MM-B-a10-100	*	1166.4	348.5	1134.1	LM-B-a10-100	*	1137.7	533.0	324.4	ML-B-a10-100		1095.4	*1h*	*1h*	LL-B-a10-100	*	1078.6	409.0	101.1
MM-B-a10-120	*	1308.7	45.0	243.8	LM-B-a10-120		1301.0	*1h*	*1h*	ML-B-a10-120		1243.5	*1h*	*1h*	LL-B-a10-120	*	1230.1	*1h*	2726.0
MM-B-b2-16	*	289.2	0.0	0.1	LM-B-b2-16	*	289.2	0.1	0.2	ML-B-b2-16	*	289.2	0.2	0.2	LL-B-b2-16	*	289.2	0.2	0.2
MM-B-b2-20	*	323.7	0.1	0.1	LM-B-b2-20	*	323.7	0.1	0.1	ML-B-b2-20	*	323.7	0.1	0.0	LL-B-b2-20	*	323.7	0.1	0.1
MM-B-b2-24	*	459.0	0.1	0.1	LM-B-b2-24	*	458.7	0.2	0.1	ML-B-b2-24	*	454.6	0.2	0.1	LL-B-b2-24	*	450.2	0.1	0.1
MM-B-b3-24	*	361.1	0.1	0.0	LM-B-b3-24	*	360.2	0.1	0.0	ML-B-b3-24	*	361.1	0.1	0.0	LL-B-b3-24	*	360.2	0.1	0.0
MM-B-b3-30	*	525.8	0.2	0.1	LM-B-b3-30	*	521.4	0.3	0.1	ML-B-b3-30	*	501.3	0.2	0.1	LL-B-b3-30	*	501.3	0.3	0.1
MM-B-b3-36	*	570.2	0.3	0.2	LM-B-b3-36	*	550.9	0.3	0.2	ML-B-b3-36	*	570.2	0.3	0.2	LL-B-b3-36	*	550.9	0.4	0.2
MM-B-b4-32	*	487.5	0.2	0.6	LM-B-b4-32	*	481.4	0.2	0.1	ML-B-b4-32	*	456.5	0.6	0.5	LL-B-b4-32	*	456.3	0.6	0.4
MM-B-b4-40	*	618.1	0.9	1.9	LM-B-b4-40	*	611.2	0.9	1.0	ML-B-b4-40	*	609.7	1.3	1.6	LL-B-b4-40	*	600.2	1.0	0.6
MM-B-b4-48	*	632.1	1.2	0.4	LM-B-b4-48	*	624.1	1.1	1.7	ML-B-b4-48	*	628.5	7.2	3.0	LL-B-b4-48	*	622.2	1.7	0.6
MM-B-b5-40	*	604.0	0.9	0.6	LM-B-b5-40	*	599.9	0.9	0.6	ML-B-b5-40	*	579.8	1.4	1.3	LL-B-b5-40	*	579.4	1.5	1.3
MM-B-b5-50	*	734.2	79.3	160.9	LM-B-b5-50	*	724.6	10.8	6.1	ML-B-b5-50	*	711.9	30.8	17.2	LL-B-b5-50	*	709.1	39.5	79.6
MM-B-b5-60	*	871.9	3.8	14.9	LM-B-b5-60	*	850.0	4.2	9.4	ML-B-b5-60	*	849.1	2.8	6.9	LL-B-b5-60	*	840.7	10.2	10.0
MM-B-b6-48	*	706.5	1.0	2.4	LM-B-b6-48	*	696.2	1.2	1.0	ML-B-b6-48	*	680.9	0.5	1.2	LL-B-b6-48	*	677.5	2.0	2.1
MM-B-b6-60	*	841.9	1.3	4.7	LM-B-b6-60	*	832.4	6.5	5.0	ML-B-b6-60	*	822.5	5.8	4.2	LL-B-b6-60	*	818.0	22.1	10.1
MM-B-b6-72	*	958.2	12.8	35.6	LM-B-b6-72	*	943.8	6.8	8.1	ML-B-b6-72	*	918.9	36.5	24.3	LL-B-b6-72	*	913.9	64.6	39.6
MM-B-b7-56	*	764.6	2.2	2.4	LM-B-b7-56	*	758.3	3.8	2.3	ML-B-b7-56	*	736.9	3.2	2.0	LL-B-b7-56	*	736.8	4.3	2.5
MM-B-b7-70	*	873.9	11.0	6.0	LM-B-b7-70	*	861.7	10.1	5.4	ML-B-b7-70	*	860.8	13.3	16.3	LL-B-b7-70	*	850.4	27.5	17.6
MM-B-b7-84	*	1156.3	13.6	7.1	LM-B-b7-84	*	1149.0	106.8	92.6	ML-B-b7-84	*	1120.7	40.8	27.3	LL-B-b7-84	*	1111.6	19.8	13.7
MM-B-b8-64	*	792.0	10.2	21.0	LM-B-b8-64	*	782.0	15.1	7.2	ML-B-b8-64	*	774.0	2.5	2.6	LL-B-b8-64	*	761.5	9.8	7.6
MM-B-b8-80	*	1009.3	11.4	8.9	LM-B-b8-80	*	995.0	8.8	5.7	ML-B-b8-80	*	995.3	10.5	7.3	LL-B-b8-80	*	986.2	8.6	8.6
MM-B-b8-96	*	1137.7	28.5	34.0	LM-B-b8-96	*	1125.3	74.4	45.5	ML-B-b8-96	*	1105.5	708.5	155.5	LL-B-b8-96		1098.8	*1h*	*1h*
MM-B-b9-72	*	879.8	59.8	82.1	LM-B-b9-72	*	872.7	13.5	16.9	ML-B-b9-72	*	859.0	158.8	47.3	LL-B-b9-72	*	848.6	17.5	11.4
MM-B-b9-90	*	1140.6	875.0	633.8	LM-B-b9-90	*	1113.5	29.9	44.9	ML-B-b9-90	*	1102.5	1017.2	1033.0	LL-B-b9-90	*	1088.7	185.6	52.7
MM-B-b9-108	*	1324.3	49.5	18.8	LM-B-b9-108	*	1299.1	74.1	38.8	ML-B-b9-108	*	1274.2	66.1	110.2	LL-B-b9-108	*	1262.4	*1h*	2529.1
MM-B-b10-80	*	921.3	6.9	3.8	LM-B-b10-80	*	913.1	11.4	6.2	ML-B-b10-80	*	888.8	9.9	19.2	LL-B-b10-80	*	871.0	9.9	4.3
MM-B-b10-100	*	1239.7	18.6	59.0	LM-B-b10-100	*	1219.2	24.2	33.6	ML-B-b10-100	*	1182.6	37.8	87.2	LL-B-b10-100	*	1169.5	54.0	27.6
MM-B-b10-120	*	1476.8	62.1	78.2	LM-B-b10-120	*	1442.3	98.3	53.6	ML-B-b10-120	*	1427.4	65.0	582.7	LL-B-b10-120	*	1400.3	154.5	116.0

Table 11: Results for class B (4/3) instances

MM					LM					ML					LL				
inst	*opt*	*lb*	IMP^{max}_{min}	*IMP*-I^{max}	*inst*	*opt*	*lb*	IMP^{max}_{min}	*IMP*-I^{max}	*inst*	*opt*	*lb*	IMP^{max}_{min}	*IMP*-I^{max}	*inst*	*opt*	*lb*	IMP^{max}_{min}	*IMP*-I^{max}
MM-C-a2-16	*	300.9	0.0	0.0	LM-C-a2-16	*	296.4	0.0	0.0	ML-C-a2-16	*	289.4	0.1	0.1	LL-C-a2-16	*	289.4	0.1	0.1
MM-C-a2-20	*	339.1	0.1	0.2	LM-C-a2-20	*	337.5	0.2	0.2	ML-C-a2-20	*	315.6	0.4	0.4	LL-C-a2-20	*	315.6	0.4	0.3
MM-C-a2-24	*	409.1	1.0	0.8	LM-C-a2-24	*	407.5	1.2	1.0	ML-C-a2-24	*	405.7	2.8	4.0	LL-C-a2-24	*	403.9	3.0	2.7
MM-C-a3-24	*	303.1	0.2	1.2	LM-C-a3-24	*	299.3	0.1	0.3	ML-C-a3-24	*	296.0	0.2	0.8	LL-C-a3-24	*	295.4	0.2	0.4
MM-C-a3-30	*	471.5	0.3	0.5	LM-C-a3-30	*	462.9	1.1	1.6	ML-C-a3-30	*	443.3	2.4	1.3	LL-C-a3-30	*	436.7	0.6	0.8
MM-C-a3-36	*	535.4	1.6	1.1	LM-C-a3-36	*	501.1	1.5	0.9	ML-C-a3-36	*	522.7	1.7	1.8	LL-C-a3-36	*	476.5	1.0	0.3
MM-C-a4-32	*	472.3	0.6	1.3	LM-C-a4-32	*	462.4	0.8	2.8	ML-C-a4-32	*	457.0	2.7	4.9	LL-C-a4-32	*	449.1	1.9	1.9
MM-C-a4-40	*	552.4	1.6	1.4	LM-C-a4-40	*	534.3	1.7	1.3	ML-C-a4-40	*	526.4	29.0	15.0	LL-C-a4-40	*	520.0	4.6	4.0
MM-C-a4-48	*	646.7	2.6	2.4	LM-C-a4-48	*	636.3	4.8	2.8	ML-C-a4-48	*	616.2	93.4	48.7	LL-C-a4-48	*	609.4	274.5	113.0
MM-C-a5-40	*	493.3	0.4	0.2	LM-C-a5-40	*	491.0	1.4	0.9	ML-C-a5-40	*	469.0	1.3	0.9	LL-C-a5-40	*	469.0	1.7	1.0
MM-C-a5-50	*	645.2	6.6	8.0	LM-C-a5-50	*	633.2	5.7	3.9	ML-C-a5-50	*	621.6	97.3	65.4	LL-C-a5-50	*	612.7	119.8	74.1
MM-C-a5-60	*	743.6	6.3	4.8	LM-C-a5-60	*	728.9	4.6	3.8	ML-C-a5-60	*	713.1	3507.9	438.7	LL-C-a5-60	*	711.3	*1h*	231.8
MM-C-a6-48	*	609.3	1.2	2.2	LM-C-a6-48	*	606.4	3.1	1.6	ML-C-a6-48	*	571.7	66.7	13.4	LL-C-a6-48	*	570.1	171.7	22.0
MM-C-a6-60	*	754.5	72.1	71.2	LM-C-a6-60	*	742.3	36.2	53.8	ML-C-a6-60	*	710.3	78.5	100.7	LL-C-a6-60	*	702.0	190.3	65.6
MM-C-a6-72	*	883.9	47.1	21.9	LM-C-a6-72	*	870.9	22.5	33.7	ML-C-a6-72	*	848.5	209.2	*1h*	LL-C-a6-72	*	832.0	145.9	305.4
MM-C-a7-56	*	684.2	37.7	52.2	LM-C-a7-56	*	668.7	5.8	4.5	ML-C-a7-56	*	646.3	12.4	9.1	LL-C-a7-56	*	630.3	22.6	16.2
MM-C-a7-70	*	843.0	254.7	1400.6	LM-C-a7-70	*	827.7	47.8	100.1	ML-C-a7-70	*	797.6	14.2	20.1	LL-C-a7-70	*	786.2	21.3	10.0
MM-C-a7-84	*	971.8	180.2	65.0	LM-C-a7-84	*	954.8	2048.4	228.1	ML-C-a7-84	*	902.4	26.8	37.3	LL-C-a7-84	*	896.0	47.3	36.6
MM-C-a8-64	*	716.4	2.6	14.1	LM-C-a8-64	*	699.3	12.4	6.7	ML-C-a8-64	*	695.3	480.4	156.7	LL-C-a8-64	*	687.6	362.6	63.4
MM-C-a8-80	*	909.4	36.5	54.7	LM-C-a8-80	*	897.8	14.1	19.3	ML-C-a8-80	*	867.7	51.8	122.5	LL-C-a8-80	*	863.5	849.6	898.5
MM-C-a8-96	*	1119.6	320.5	561.7	LM-C-a8-96	*	1109.2	*1h*	2037.6	ML-C-a8-96		1066.7	*1h*	*1h*	LL-C-a8-96	*	1048.5	210.2	69.8
MM-C-a9-72	*	849.7	1027.3	1482.5	LM-C-a9-72	*	829.0	186.9	339.7	ML-C-a9-72	*	790.6	968.4	1968.8	LL-C-a9-72	*	780.8	786.6	1444.0
MM-C-a9-90	*	965.8	30.0	43.4	LM-C-a9-90	*	979.0	67.2	53.3	ML-C-a9-90	*	926.9	1042.9	*1h*	LL-C-a9-90	*	923.9	83.2	625.5
MM-C-a9-108	*	1133.0	355.9	2697.6	LM-C-a9-108		1109.2	*1h*	*1h*	ML-C-a9-108		1066.0	*1h*	*1h*	LL-C-a9-108		1055.5	*1h*	*1h*
MM-C-a10-80	*	893.6	34.0	74.0	LM-C-a10-80	*	875.6	370.0	209.3	ML-C-a10-80	*	848.2	691.4	*1h*	LL-C-a10-80	*	831.1	1348.9	556.1
MM-C-a10-100	*	1133.9	373.7	808.7	LM-C-a10-100		1113.3	*1h*	*1h*	ML-C-a10-100	*	1057.4	2716.9	*1h*	LL-C-a10-100		1046.5	*1h*	*1h*
MM-C-a10-120		1280.0	*1h*	*1h*	LM-C-a10-120		1255.3	*1h*	*1h*	ML-C-a10-120		1209.2	*1h*	*1h*	LL-C-a10-120		1206.8	*1h*	*1h*
MM-C-b2-16	*	279.9	0.2	0.2	LM-C-b2-16	*	279.6	0.3	0.4	ML-C-b2-16	*	278.0	0.3	0.4	LL-C-b2-16	*	274.2	0.1	0.0
MM-C-b2-20	*	315.0	0.1	0.0	LM-C-b2-20	*	302.6	0.1	0.0	ML-C-b2-20	*	315.0	0.2	0.3	LL-C-b2-20	*	300.8	0.3	0.2
MM-C-b2-24	*	417.7	0.2	0.1	LM-C-b2-24	*	417.5	0.2	0.1	ML-C-b2-24	*	410.9	0.4	0.3	LL-C-b2-24	*	408.4	0.2	0.1
MM-C-b3-24	*	353.8	0.2	0.3	LM-C-b3-24	*	351.6	0.1	0.2	ML-C-b3-24	*	353.8	0.2	0.5	LL-C-b3-24	*	351.5	0.1	0.2
MM-C-b3-30	*	466.7	0.4	0.3	LM-C-b3-30	*	466.7	0.5	0.3	ML-C-b3-30	*	456.9	0.3	0.3	LL-C-b3-30	*	455.6	0.3	0.1
MM-C-b3-36	*	549.9	0.4	0.1	LM-C-b3-36	*	542.2	0.4	0.2	ML-C-b3-36	*	533.9	0.4	0.2	LL-C-b3-36	*	525.4	0.5	0.2
MM-C-b4-32	*	466.0	0.5	0.7	LM-C-b4-32	*	466.0	0.7	0.5	ML-C-b4-32	*	450.2	4.4	6.9	LL-C-b4-32	*	449.8	10.4	4.6
MM-C-b4-40	*	592.6	1.0	0.9	LM-C-b4-40	*	580.0	1.3	1.0	ML-C-b4-40	*	566.2	0.7	0.4	LL-C-b4-40	*	555.6	1.5	1.0
MM-C-b4-48	*	610.3	5.3	2.1	LM-C-b4-48	*	604.1	6.0	3.2	ML-C-b4-48	*	606.3	12.1	10.1	LL-C-b4-48	*	591.3	7.5	14.3
MM-C-b5-40	*	563.5	2.0	3.3	LM-C-b5-40	*	560.4	0.6	1.4	ML-C-b5-40	*	552.4	2.5	6.1	LL-C-b5-40	*	550.3	5.5	4.4
MM-C-b5-50	*	674.7	8.7	3.9	LM-C-b5-50	*	668.1	9.9	5.8	ML-C-b5-50	*	659.8	34.1	41.0	LL-C-b5-50	*	654.5	48.4	45.4
MM-C-b5-60	*	800.2	11.8	13.4	LM-C-b5-60	*	794.9	40.2	11.3	ML-C-b5-60	*	760.6	6.8	4.3	LL-C-b5-60	*	747.6	*1h*	5.4
MM-C-b6-48	*	650.9	0.6	1.7	LM-C-b6-48	*	641.5	2.9	2.2	ML-C-b6-48	*	635.4	3.3	10.7	LL-C-b6-48	*	626.1	11.5	6.3
MM-C-b6-60	*	798.1	1.5	7.8	LM-C-b6-60	*	776.5	1.9	1.0	ML-C-b6-60	*	769.1	9.5	10.5	LL-C-b6-60	*	757.3	21.8	15.3
MM-C-b6-72	*	914.6	167.4	286.6	LM-C-b6-72	*	901.6	20.2	16.4	ML-C-b6-72	*	869.4	192.1	54.5	LL-C-b6-72	*	865.9	19.1	14.4
MM-C-b7-56	*	705.3	8.1	4.7	LM-C-b7-56	*	697.0	20.5	9.2	ML-C-b7-56	*	676.1	5.5	5.2	LL-C-b7-56	*	674.8	10.0	8.1
MM-C-b7-70	*	845.4	9.8	140.5	LM-C-b7-70	*	831.5	96.0	35.5	ML-C-b7-70	*	820.5	28.5	38.4	LL-C-b7-70	*	806.2	27.9	24.6
MM-C-b7-84	*	1088.4	229.3	154.0	LM-C-b7-84	*	1080.3	88.3	117.4	ML-C-b7-84	*	1054.3	406.1	1968.0	LL-C-b7-84	*	1044.0	366.7	1407.4
MM-C-b8-64	*	717.7	2.7	6.5	LM-C-b8-64	*	709.7	4.7	3.4	ML-C-b8-64	*	705.9	16.0	25.9	LL-C-b8-64	*	690.0	20.1	20.4
MM-C-b8-80	*	929.0	198.9	282.2	LM-C-b8-80	*	923.5	67.5	41.2	ML-C-b8-80	*	917.5	1087.3	1268.1	LL-C-b8-80	*	908.1	613.7	292.1
MM-C-b8-96	*	1067.8	21.2	34.3	LM-C-b8-96	*	1062.1	50.4	57.8	ML-C-b8-96	*	1028.3	33.9	84.8	LL-C-b8-96	*	1019.9	87.3	81.2
MM-C-b9-72	*	837.4	43.3	125.6	LM-C-b9-72	*	831.1	858.7	*1h*	ML-C-b9-72	*	801.9	3032.2	2195.9	LL-C-b9-72		799.7	*1h*	*1h*
MM-C-b9-90		1065.0	*1h*	*1h*	LM-C-b9-90	*	1029.9	1633.1	407.3	ML-C-b9-90		999.9	*1h*	*1h*	LL-C-b9-90		989.0	*1h*	*1h*
MM-C-b9-108	*	1222.3	30.9	33.2	LM-C-b9-108	*	1206.4	97.2	63.2	ML-C-b9-108	*	1187.6	*1h*	2918.8	LL-C-b9-108		1173.5	*1h*	*1h*
MM-C-b10-80	*	881.9	46.0	23.3	LM-C-b10-80	*	876.1	56.9	41.9	ML-C-b10-80	*	841.4	184.8	399.5	LL-C-b10-80		836.9	*1h*	*1h*
MM-C-b10-100	*	1162.1	47.6	169.8	LM-C-b10-100	*	1137.0	1573.2	792.0	ML-C-b10-100	*	1086.6	570.9	1447.9	LL-C-b10-100		1082.1	*1h*	*1h*
MM-C-b10-120	*	1366.9	3548.5	*1h*	LM-C-b10-120	*	1350.7	1968.1	619.9	ML-C-b10-120		1318.9	*1h*	*1h*	LL-C-b10-120		1300.9	*1h*	*1h*

Table 12: Results for class C (5/3) instances

MM					LM					ML					LL				
inst	*opt*	*lb*	IMP_{min}^{max}	$IMP\text{-}I^{max}$	*inst*	*opt*	*lb*	IMP_{min}^{max}	$IMP\text{-}I^{max}$	*inst*	*opt*	*lb*	IMP_{min}^{max}	$IMP\text{-}I^{max}$	*inst*	*opt*	*lb*	IMP_{min}^{max}	$IMP\text{-}I^{max}$
MM-D-a2-16	*	296.5	0.0	0.0	LM-D-a2-16	*	285.9	0.1	0.0	ML-D-a2-16	*	280.6	0.0	0.1	LL-D-a2-16	*	271.9	0.0	0.1
MM-D-a2-20	*	337.5	0.2	0.2	LM-D-a2-20	*	337.5	1.1	7.3	ML-D-a2-20	*	305.3	0.3	0.4	LL-D-a2-20	*	304.6	0.8	1.6
MM-D-a2-24	*	407.5	1.0	1.6	LM-D-a2-24	*	407.5	2.0	1.3	ML-D-a2-24	*	401.2	3.0	3.0	LL-D-a2-24	*	382.7	2.2	1.7
MM-D-a3-24	*	299.3	0.2	0.5	LM-D-a3-24	*	297.8	0.2	0.3	ML-D-a3-24	*	293.2	1.0	0.8	LL-D-a3-24	*	289.6	1.2	0.5
MM-D-a3-30	*	461.2	2.9	4.5	LM-D-a3-30	*	453.4	4.1	2.6	ML-D-a3-30	*	432.1	1.0	3.6	LL-D-a3-30	*	417.8	1.2	0.3
MM-D-a3-36	*	502.2	1.9	1.2	LM-D-a3-36	*	497.6	1.5	0.9	ML-D-a3-36	*	475.4	3.2	1.4	LL-D-a3-36	*	468.0	1.5	0.4
MM-D-a4-32	*	458.7	2.6	2.3	LM-D-a4-32	*	457.2	2.9	3.6	ML-D-a4-32	*	439.2	3.8	9.7	LL-D-a4-32	*	435.9	2.3	2.6
MM-D-a4-40	*	543.0	3.0	3.1	LM-D-a4-40	*	513.5	6.6	3.1	ML-D-a4-40	*	521.3	35.5	24.2	LL-D-a4-40	*	500.7	10.6	4.4
MM-D-a4-48	*	626.7	4.1	3.9	LM-D-a4-48	*	607.9	6.8	2.7	ML-D-a4-48	*	593.7	52.0	11.9	LL-D-a4-48	*	593.7	618.9	200.0
MM-D-a5-40	*	471.9	1.2	0.8	LM-D-a5-40	*	467.5	0.7	0.2	ML-D-a5-40	*	450.5	7.7	2.7	LL-D-a5-40	*	443.2	1.5	0.3
MM-D-a5-50	*	628.9	20.5	9.3	LM-D-a5-50	*	618.9	59.1	38.0	ML-D-a5-50	*	594.5	20.2	22.8	LL-D-a5-50	*	585.0	3186.8	425.0
MM-D-a5-60	*	731.2	6.5	14.0	LM-D-a5-60	*	721.8	4.5	3.8	ML-D-a5-60	*	692.0	43.5	147.5	LL-D-a5-60	*	681.5	132.1	55.9
MM-D-a6-48	*	589.2	6.1	7.1	LM-D-a6-48	*	575.9	2.1	2.9	ML-D-a6-48	*	559.6	207.7	12.3	LL-D-a6-48	*	558.6	321.4	20.6
MM-D-a6-60	*	744.8	68.6	130.4	LM-D-a6-60	*	728.6	37.2	65.1	ML-D-a6-60	*	696.8	42.4	1457.3	LL-D-a6-60	*	684.1	42.6	23.9
MM-D-a6-72	*	855.8	42.0	18.5	LM-D-a6-72	*	850.8	65.2	25.4	ML-D-a6-72	*	824.7	1066.2	2276.3	LL-D-a6-72	*	809.8	238.4	123.5
MM-D-a7-56	*	664.9	3.8	2.5	LM-D-a7-56	*	659.2	28.8	25.2	ML-D-a7-56	*	632.8	22.9	18.0	LL-D-a7-56	*	619.5	78.6	58.1
MM-D-a7-70	*	821.1	18.8	92.2	LM-D-a7-70	*	804.2	45.5	36.8	ML-D-a7-70	*	782.4	32.6	58.9	LL-D-a7-70	*	759.9	39.0	103.2
MM-D-a7-84	*	935.7	395.0	907.9	LM-D-a7-84		924.3	*1h*	*1h*	ML-D-a7-84		888.9	*1h*	*1h*	LL-D-a7-84		870.1	*1h*	*1h*
MM-D-a8-64	*	707.9	85.6	216.5	LM-D-a8-64	*	693.2	23.2	20.5	ML-D-a8-64	*	685.1	351.6	243.8	LL-D-a8-64	*	677.9	694.1	299.9
MM-D-a8-80	*	897.7	378.3	1841.3	LM-D-a8-80	*	881.6	29.3	27.1	ML-D-a8-80	*	848.6	500.2	1493.0	LL-D-a8-80	*	842.5	1259.4	894.0
MM-D-a8-96	*	1088.8	522.2	161.8	LM-D-a8-96	*	1077.1	1831.1	172.6	ML-D-a8-96		1036.5	*1h*	*1h*	LL-D-a8-96		1021.6	*1h*	*1h*
MM-D-a9-72	*	827.6	2529.9	2303.9	LM-D-a9-72		805.8	*1h*	*1h*	ML-D-a9-72		766.6	*1h*	*1h*	LL-D-a9-72		753.7	*1h*	*1h*
MM-D-a9-90	*	949.1	51.2	68.0	LM-D-a9-90	*	944.1	70.9	50.1	ML-D-a9-90	*	910.8	460.7	*1h*	LL-D-a9-90	*	899.8	582.5	723.6
MM-D-a9-108		1099.4	*1h*	*1h*	LM-D-a9-108		1088.3	*1h*	*1h*	ML-D-a9-108		1028.0	*1h*	*1h*	LL-D-a9-108		1024.0	*1h*	*1h*
MM-D-a10-80	*	865.4	270.8	1014.6	LM-D-a10-80	*	848.6	27.1	221.5	ML-D-a10-80	*	798.7	48.5	26.3	LL-D-a10-80	*	807.2	*1h*	1271.8
MM-D-a10-100	*	1097.5	337.8	1769.9	LM-D-a10-100	*	1089.8	*1h*	3520.2	ML-D-a10-100		1012.2	*1h*	*1h*	LL-D-a10-100		1008.8	*1h*	*1h*
MM-D-a10-120		1253.5	*1h*	*1h*	LM-D-a10-120		1234.1	*1h*	*1h*	ML-D-a10-120		1191.3	*1h*	*1h*	LL-D-a10-120		1173.0	*1h*	*1h*
MM-D-b2-16	*	279.9	0.2	0.3	LM-D-b2-16	*	274.5	0.1	0.1	ML-D-b2-16	*	274.5	0.1	0.0	LL-D-b2-16	*	271.0	0.1	0.0
MM-D-b2-20	*	302.6	0.1	0.1	LM-D-b2-20	*	299.8	0.3	0.7	ML-D-b2-20	*	298.8	0.4	0.4	LL-D-b2-20	*	291.7	0.1	0.3
MM-D-b2-24	*	406.8	0.1	0.5	LM-D-b2-24	*	404.9	0.2	0.1	ML-D-b2-24	*	394.7	1.6	7.1	LL-D-b2-24	*	383.3	0.3	0.2
MM-D-b3-24	*	334.9	0.1	0.0	LM-D-b3-24	*	329.8	0.1	0.2	ML-D-b3-24	*	329.7	0.3	0.2	LL-D-b3-24	*	328.5	0.3	0.2
MM-D-b3-30	*	465.1	0.3	0.8	LM-D-b3-30	*	464.6	0.3	0.2	ML-D-b3-30	*	446.8	1.2	1.0	LL-D-b3-30	*	446.8	1.7	1.3
MM-D-b3-36	*	540.0	0.8	0.5	LM-D-b3-36	*	540.0	0.8	0.4	ML-D-b3-36	*	511.5	0.8	1.4	LL-D-b3-36	*	510.2	0.9	1.4
MM-D-b4-32	*	439.7	0.2	0.5	LM-D-b4-32	*	435.9	1.0	0.6	ML-D-b4-32	*	419.3	0.9	1.4	LL-D-b4-32	*	419.2	1.7	2.0
MM-D-b4-40	*	556.1	1.6	2.1	LM-D-b4-40	*	514.9	1.4	0.9	ML-D-b4-40	*	520.7	2.3	1.9	LL-D-b4-40	*	497.5	3.7	2.2
MM-D-b4-48	*	605.0	5.5	11.6	LM-D-b4-48	*	600.2	5.4	20.8	ML-D-b4-48	*	576.5	23.8	243.5	LL-D-b4-48	*	566.5	5.1	5.3
MM-D-b5-40	*	554.1	20.1	9.6	LM-D-b5-40	*	549.6	3.5	2.5	ML-D-b5-40	*	534.3	20.9	14.2	LL-D-b5-40	*	530.7	9.0	8.2
MM-D-b5-50	*	648.4	12.1	10.4	LM-D-b5-50	*	647.1	102.4	75.7	ML-D-b5-50	*	633.1	997.8	*1h*	LL-D-b5-50	*	620.4	324.7	140.8
MM-D-b5-60	*	764.0	5.4	4.5	LM-D-b5-60	*	761.9	7.5	6.0	ML-D-b5-60	*	723.4	237.7	341.7	LL-D-b5-60	*	719.8	822.1	1061.2
MM-D-b6-48	*	627.3	14.5	20.6	LM-D-b6-48	*	616.9	11.6	7.1	ML-D-b6-48	*	606.2	93.6	1710.9	LL-D-b6-48	*	590.1	47.5	28.0
MM-D-b6-60	*	740.5	2.9	36.5	LM-D-b6-60	*	734.9	86.2	31.6	ML-D-b6-60	*	719.5	10.6	41.7	LL-D-b6-60	*	716.8	1454.3	196.3
MM-D-b6-72	*	871.0	76.9	92.5	LM-D-b6-72	*	862.2	633.3	404.8	ML-D-b6-72	*	830.5	35.3	26.7	LL-D-b6-72	*	814.1	219.9	60.9
MM-D-b7-56	*	682.4	8.5	16.5	LM-D-b7-56	*	674.8	103.5	26.8	ML-D-b7-56	*	635.2	163.4	45.5	LL-D-b7-56	*	630.3	70.2	28.1
MM-D-b7-70	*	816.3	53.0	396.5	LM-D-b7-70	*	800.6	226.2	153.4	ML-D-b7-70	*	770.5	53.6	31.5	LL-D-b7-70	*	757.1	40.5	22.7
MM-D-b7-84	*	1033.7	370.3	143.9	LM-D-b7-84	*	1019.6	40.9	39.2	ML-D-b7-84		991.4	*1h*	*1h*	LL-D-b7-84		983.8	*1h*	*1h*
MM-D-b8-64	*	704.7	46.4	88.4	LM-D-b8-64	*	695.7	3548.0	2627.0	ML-D-b8-64	*	683.4	378.7	295.5	LL-D-b8-64	*	669.1	541.5	193.0
MM-D-b8-80	*	889.4	33.0	165.6	LM-D-b8-80	*	882.2	713.9	478.6	ML-D-b8-80	*	858.3	2143.9	*1h*	LL-D-b8-80	*	845.8	280.5	837.1
MM-D-b8-96	*	1051.4	310.0	282.0	LM-D-b8-96	*	1040.6	725.9	257.4	ML-D-b8-96		1003.4	*1h*	*1h*	LL-D-b8-96	*	985.4	1727.7	650.0
MM-D-b9-72	*	815.0	2290.9	*1h*	LM-D-b9-72		798.2	*1h*	*1h*	ML-D-b9-72		758.0	*1h*	*1h*	LL-D-b9-72		747.9	*1h*	*1h*
MM-D-b9-90		1011.0	*1h*	*1h*	LM-D-b9-90		998.8	*1h*	*1h*	ML-D-b9-90		954.3	*1h*	*1h*	LL-D-b9-90		949.3	*1h*	*1h*
MM-D-b9-108		1175.0	*1h*	*1h*	LM-D-b9-108	*	1166.0	2850.7	578.9	ML-D-b9-108		1120.1	*1h*	*1h*	LL-D-b9-108		1112.2	*1h*	*1h*
MM-D-b10-80	*	842.5	15.6	292.4	LM-D-b10-80	*	830.9	177.7	99.6	ML-D-b10-80		802.1	*1h*	*1h*	LL-D-b10-80		790.4	*1h*	*1h*
MM-D-b10-100	*	1116.6	2693.4	*1h*	LM-D-b10-100		1078.6	*1h*	*1h*	ML-D-b10-100		1034.3	*1h*	*1h*	LL-D-b10-100		1015.5	*1h*	*1h*
MM-D-b10-120	*	1289.3	3194.7	*1h*	LM-D-b10-120		1274.0	*1h*	*1h*	ML-D-b10-120		1239.9	*1h*	*1h*	LL-D-b10-120		1229.0	*1h*	*1h*

Table 13: Results for class D (6/3) instances