Submitted to *INFORMS Journal on Computing*
manuscript JOC-2017-03-OA-043

An MDD-based Lagrangian Approach to the Multi-Commodity Pickup-and-Delivery TSP

Margarita P. Castro
Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario M5S 3G8, Canada, mpcastro@mie.utoronto.ca

Andre A. Cire
Department of Management, University of Toronto Scarborough and Rotman School of Management, Toronto, Ontario M1E 1A4, acire@utsc.utoronto.ca

J. Christopher Beck
Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario M5S 3G8, Canada, jcb@mie.utoronto.ca

We address the one-to-one multi-commodity pickup-and-delivery traveling salesman problem (m-PDTSP), a challenging variant of the traveling salesman problem which includes the transportation of commodities between locations. The goal is to find a minimum-cost tour such that each commodity is delivered to its destination and the maximum capacity of the vehicle is never exceeded. We propose an exact approach that uses a discrete relaxation based on *multivalued decision diagrams* (MDDs) to better represent the combinatorial structure of the problem. We enhance our relaxation by using the MDDs as a subproblem to a Lagrangian relaxation technique, leading to significant improvements both in bound quality and run time performance. Our work extends the use of MDDs for solving routing problems by presenting new construction methods and filtering rules based on capacity restrictions. Experimental results show that our approach outperforms state-of-the-art methodologies, closing 33 open instances from the literature with 27 of those closed by our best variant.

Key words: decision diagrams; Lagrangian duality; vehicle routing; traveling salesman problem

1. Introduction

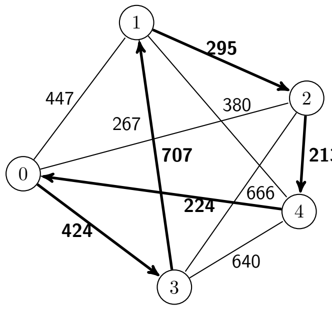
Pickup-and-delivery refers to a large class of optimization problems that is primarily concerned with the transportation of persons or commodities between geographically-dispersed locations. Such problems represent core routing decisions in a wide range of practical applications. Examples include parcel delivery (Holland et al. 2017), dial-a-ride problems (Cordeau and Laporte 2007, Liu et al. 2018), home healthcare (Liu et al. 2013), robotics (Coltin and Veloso 2014), and emergency dispatch (Cordeau et al. 2007), to name a few. The area is associated with a pervasive and rich literature in optimization and scheduling; see, e.g., Savelsbergh and Sol (1995), Parragh et al. (2008).

This work investigates a new exact approach for the *one-to-one multi-commodity pickup and delivery traveling salesman problem* (m-PDTSP), a variant of the classical traveling salesman problem (TSP) that incorporates the delivery of a fixed set of commodities by a capacitated vehicle. Specifically, the problem is defined over a directed graph $\mathscr{G}$, where nodes represent locations and arcs are associated with non-negative travel costs. We are also given a set of commodities, each having a weight, a pickup location, and a delivery location. A solution to the m-PDTSP is a minimum-cost Hamiltonian tour on $\mathscr{G}$ where each commodity's pickup location must be visited prior to its corresponding delivery location, and the total weight carried by the vehicle never exceeds its capacity. Figure 1 depicts an example with 5 locations, two commodities and a vehicle with capacity 5, where an optimal tour starting and finishing at the depot 0 is presented in bold.

The m-PDTSP was introduced by Hernández-Pérez and Salazar-González (2009) and can be viewed as an important subproblem in vehicle routing applications, for instance when routes must be optimized for freight delivery (Holland et al. 2017). In contrast to classical pickup-and-delivery problems (Parragh et al. 2008), the locations in the m-PDTSP can be both a source and a destination for multiple commodities at the same time. The problem thereby generalizes several related single-vehicle variants, such as the *pickup-and-delivery TSP* (Dumitrescu et al. 2008), where each location is either the source or destination of at most one commodity; the *sequential ordering problem* (Ascheuer et al. 2000), where the vehicle is uncapacitated; and the *one-commodity pickup-and-delivery TSP* (1-PDTSP) (Hernández-Pérez and Salazar-González 2004), where all commodities are equivalent.

We propose a novel exact approach for the m-PDTSP that applies Lagrangian duality to combine a linear and a discrete relaxation of the problem. In particular, the discrete relaxation is encoded as a multivalued decision diagram (MDD), a graphical structure that compactly represents a set of solutions to a problem. Our methodology considers *relaxed* MDDs, which are diagrams of parametrized size that approximate the feasible solution set. In this work, we leverage the underlying network representation of an MDD to better exploit the combinatorial structure of the m-PDTSP while also incorporating dual information from a linear programming relaxation of the problem.

The resulting approach provides a flexible relaxation that yields bounds on the optimal solution value of the m-PDTSP and can be embedded, e.g., in any branching search.



Capacity (C): 5

Commodity	Weight (w_k)	Pickup Location (p_k)	Delivery Location (d_k)
1	3	1	4
2	2	3	2

Figure 1 **Example of an** m-PDTSP **instance. Bold arcs represent the optimal tour.**

Computational experiments using a constraint programming solver indicate that the resulting MDD relaxation can enhance solution times by orders of magnitude in a number of instances, in particular when capacities are small. We also find provably optimal solutions to 33 instances in the literature for the first time, 27 of those with our best parametrization.

Main contributions. Our first contribution is to introduce an MDD-based discrete relaxation for the m-PDTSP. Its main purpose is to provide valid bounds on the optimal solution of the problem. Our data structure, inspired by a mixed-integer linear programming (MILP) formulation introduced by Gouveia and Ruthmair (2015), compactly encodes all feasible solutions as paths in a directed acyclic graph where edges represent positions in a tour. We present structural results and strategies for constructing relaxed MDDs that take into account both tour constraints and vehicle capacities, extending previous work on MDDs for sequencing problems. Specifically, our capacity-based construction guarantees the satisfaction of the capacity constraint for all solutions represented in the relaxed MDD.

Our second key contribution is a Lagrangian technique that significantly strengthens the existing bounds for the m-PDTSP based on the concepts introduced by Bergman et al. (2015). Namely, we incorporate Lagrange multipliers that penalize solutions of the MDD which do not represent valid Hamiltonian tours or violate precedence and capacity constraints. The key advantage of this framework is that it exploits the discrete representation of the m-PDTSP while still taking advantage of linear programming (LP) relaxation information, thereby benefiting from both MDD and LP methodologies.

Finally, we present an extensive numerical study that evaluates our MDD construction strategies and the performance of distinct MDD-based Lagrangian relaxations for the m-PDTSP. To this end, we incorporate our relaxation mechanism into a constraint programming model and evaluate the quality of our bounds and the solution performance with respect to state-of-the-art techniques. We show that our integrated methodology can provide significant improvements over the existing dataset, which is more pronounced when the instance is associated with a small vehicle capacity relative to the commodity weights.

Outline of the paper. The paper is organized as follows. Section 2 presents a review of the previous related works. We formalize the m-PDTSP and introduce notation in Section 3. Specifically, Section 3.1 presents a mathematical formulation of the problem, while Section 3.2 describes an MDD representation for the problem. Section 4 introduces our MDD-based Lagrangian relaxation and establishes the combinatorial structure that we exploit. Section 5 provides the associated MDD construction techniques to expose such structure. The overall approach is presented in Section 6. Numerical experiments are presented in Section 7, which includes a comparison with the state-of-the-art approaches. Finally, Section 8 provides a discussion on the work presented, the main results obtained, and final remarks.

2. Literature Review

The m-PDTSP is a capacitated version of the sequential ordering problem (SOP), a variant of the asymmetric TSP where tours are subject to additional precedence constraints between locations. Several mathematical formulations and heuristic techniques have been previously investigated for the SOP; see, e.g., Balas et al. (1995), Ascheuer et al. (2000), Hernádvölgyi (2004), Gouveia and Pesneau (2006). These works typically form the basis upon which the existing m-PDTSP mathematical models are built.

The m-PDTSP remains a challenging problem with few exact approaches in the literature. Existing solution methods are primarily based on MILPs that exploit polyhedral structure and decomposition strategies. The first models were investigated by Hernández-Pérez and Salazar-González (2009), who propose a multi-commodity flow and a path-based formulation. Both are solved using Benders decomposition strategies that iteratively add vehicle capacity constraints to a relaxed MILP model, solving instances with up to 47 locations in less than two hours. Nonetheless, these techniques are not able to optimally solve smaller instances when tighter vehicle capacities were considered. Letchford and Salazar-González (2016) later extended this formulation by introducing valid inequalities for the original multi-commodity flow model. These inequalities significantly improve the LP relaxation bound at the root node, albeit negatively impacting their solution times due to computationally expensive separation routines.

The state-of-the-art solution methods for the m-PDTSP are branch-and-cut algorithms proposed by Gouveia and Ruthmair (2015). The authors show that the problem can be reduced to an 1-PDTSP by considering additional precedence constraints, and propose new MILP formulations based on *layered graph models*. These models formulate tours as paths in an expanded graph, where each layer corresponds to a position in the tour. The models either combine flow and capacity constraints or are restricted to enforcing precedence relations. The resulting MILPs are then solved by a branch-and-cut algorithm that combines preprocessing methods, primal heuristics based on a variable neighborhood descent, and separation routines. The authors significantly improve upon the existing run times, solving several open instances to optimality.

Despite the significant solution time improvements, the state-of-the-art method by Gouveia and Ruthmair (2015) reports instances with 19 locations and 10 commodities that are still left unsolved within a reasonable amount of time. Such instances typically involve small vehicle capacities relative to the commodity weights, leading to a combinatorial structure that is not well captured by current LP relaxations. Our model builds on the ideas by Gouveia and Ruthmair (2015), but our approach exploits a different type of approximation and focuses on operating directly on the MDD graphical structure.

As for heuristic methods, Rodríguez-Martín and Salazar-González (2012) propose two heuristics that combine a greedy randomized search with a variable neighborhood descent. Their best algorithm computed high-quality solutions for instances with up to 300 locations and 600 commodities, improving upon the best known solution for existing open instances.

A decision diagram is a pervasive data structure in computer science for representing Boolean functions (Bryant 1992). An MDD is a variant where the function arguments can take more than two values. A *relaxed* MDD, first introduced by Andersen et al. (2007), is a diagram of limited size that approximates a set of solutions to a discrete problem. It has been largely applied to mathematical programming and discrete optimization, in particular for obtaining optimization bounds (Hoda et al. 2010). A survey on the use of MDDs for optimization is presented by Bergman et al. (2016).

The use of MDDs for routing problems was initially investigated by Cire and van Hoeve (2013) and Kinable et al. (2017). Cire and van Hoeve (2013) propose an MDD representation for sequencing problems that is similar to a layered graph model, in that each layer of the diagram corresponds to a position of the tour. However, such a model is not incorporated into an MILP formulation, but modified by a combinatorial algorithm during search to more accurately represent the feasible solution space of the problem. The authors investigate structural results of the diagram and improve upon existing bounds for open instances of the TSPLIB. Kinable et al. (2017) extend these ideas for *time-dependent* TSPs, proposing an additive bounding procedure (Fischetti and Toth 1989) that uses reduced-cost information from an LP relaxation to strengthen an MDD bound. We provide structural results showing that our methodology always produces bounds that are at least as strong as the ones obtained from an MDD-based additive bounding technique.

Lagrangian duality is a bounding technique extensively investigated in mathematical programming; see, e.g., the review by Fisher (2004). The methodology, as described by Geoffrion (1974), consists of dropping constraints and penalizing their violation in the objective function, thereby leading to a relaxed problem that is easier and possibly decomposed. Such constraint violations, in particular, are weighted by *Lagrange multipliers*. The problem of finding the set of multipliers that provide the best possible bound defines a maximization problem with a piecewise linear concave objective known as the *Lagrangian dual*, initially tackled by Held and Karp (1971) using subgradient methods. In this paper, we consider the methodology by Frangioni et al. (Frangioni 2002), which is a generalized version of the Bundle method introduced by Lemaréchal (1975).

The use of MDDs for Lagrangian duality was first proposed by Bergman et al. (2015), who introduce the general concept and report preliminary results for the TSP with time windows. The concept has roots in a method proposed by Beasley and Cao (1998) that combines dynamic programming (DP) with Lagrangian techniques for airline crew scheduling. In contrast, the MDD methodology operates directly on a graphical representation of the state-transition graph as opposed to the DP model.

Our MDD approach is related to the constrained shortest path problem (CSPP), in that finding the optimal solution to the m-PDTSP is equivalent to solving the CSPP over the relaxed MDD graph where the side constraints and costs are given by the m-PDTSP parameters (Section 4). Our Lagrangian approach and filtering procedure share the same underlying ideas as the techniques used to tackle the resource variant of the CSPP (Beasley and Christofides 1989, Carlyle et al. 2008). Nonetheless, we generalize such techniques by leveraging the graphical structure of an MDD through refinement techniques that exploit the pickup-and-delivery constraint structure.

The proposed methodology can also be viewed as a type of *state-space* relaxation similar to what was considered by Baldacci et al. (2012) for the TSPTW. Nonetheless, while the latter work used dual information to strengthen a specific DP state-space relaxation (the *ng*-route), our diagram can be compiled to better exploit the m-PDTSP structure.

3. Problem Definition and Formulations

In this section, we present the m-PDTSP and the notation used throughout the text. We also introduce the mathematical programming formulation and the decision diagram model that will be combined in our MDD-based Lagrangian dual methodology.

The m-PDTSP is defined on a complete directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where $\mathscr{V} := \{0, \ldots, n\}$ is a set of $n+1$ locations with 0 as the *depot*. Each arc $(i, j) \in \mathscr{E}$ is associated with a travel cost $c_{i,j} \in \mathbb{R}_{\geq 0}$, where potentially $c_{i,j} \neq c_{j,i}$ (i.e., costs may be asymmetric). We are also given a set of commodities $\mathscr{K} = \{1, \ldots, m\}$. A commodity $k \in \mathscr{K}$ has an integral positive weight $w_k \in \mathbb{Z}_{>0}$, a pickup location $p_k \in \mathscr{V} \setminus \{0\}$, and a delivery location $d_k \in \mathscr{V} \setminus \{0\}$. Finally, we assume a single vehicle with capacity C.

A solution is a minimum-cost Hamiltonian tour on $\mathscr{G}$ that observes the vehicle capacity and the pickup-and-delivery order established for each commodity. This can be formally represented using the 1-PDTSP reduction by Gouveia and Ruthmair (2015). Namely, let

$$\Delta q_i = \sum_{k \in \mathscr{K}, p_k = i} w_k - \sum_{k \in \mathscr{K}, d_k = i} w_k.$$

be the net weight of pickup and deliveries at a location $i \in \mathscr{V}$. A *feasible tour* to the m-PDTSP is a sequence of locations $\pi := (\pi_0, \pi_1, \ldots, \pi_n, \pi_{n+1})$ which satisfies (i)-(iii):

(i) The sequence starts and ends at the depot, $\pi_0 = \pi_{n+1} = 0$, and $(\pi_1, \ldots, \pi_n)$ is a permutation of $\mathscr{V} \setminus \{0\}$.

(ii) The accumulated net weight in every position of the sequence never exceeds the vehicle capacity, i.e., $\sum_{t=1}^{t'} \Delta q_{\pi_t} \in \{0, \ldots, C\}$ for all $t' \in \{1, \ldots, n\}$; and

(iii) For every commodity $k \in \mathscr{K}$, its pickup location is visited prior to its delivery location, i.e., $\pi_t = p_k$ and $\pi_{t'} = d_k$ implies $t < t'$.

The cost $c(\pi)$ of a feasible tour π is the total travel cost starting at the depot, visiting each location in $\mathscr{V}$ in the order defined by π, and returning to the depot. That is,

$$c(\pi) = \sum_{t=0}^{n} c_{\pi_t, \pi_{t+1}}.$$

The m-PDTSP requires a feasible tour π that minimizes the tour cost $c(\pi)$. The optimal tour cost is denoted by ν^*.

Example 1. Figure 1 depicts an instance of the m-PDTSP used as a running example in the text. The underlying graph has 5 locations (with 0 as the depot), where travel costs are represented as arc labels. The vehicle capacity is $C = 5$ and two commodities must be considered, with weights and pickup-and-delivery locations described in the table in the figure. Note that $\Delta q_1 = 3$, $\Delta q_2 = -2$, $\Delta q_3 = 2$, and $\Delta q_4 = -3$.

The optimal tour is $\pi = (0,3,1,2,4,0)$ with a cost of $\nu^* =$1,863, as represented by the bold arcs in Figure 1. The tour picks up commodity 2 at location 3, then picks commodity 1 at location 1, and finally delivers commodities 2 and 1 at locations 2 and 4, respectively. The net weight in π is always between 0 and the vehicle capacity $C = 5$. Furthermore, the delivery location for each commodity succeeds its corresponding pickup location. □

3.1. Mathematical Programming Formulation

We now formalize the m-PDTSP in terms of an integer linear program (ILP) that will be leveraged in our MDD-based Lagrangian framework. The m-PDTSP has a large array of formulations proposed in the literature (Letchford and Salazar-González 2016). We consider a well-known time-indexed formulation for the TSP (see, e.g., Dash et al. 2012) augmented with precedence constraints.

Let x_{ij}^t be a binary variable indicating if location j follows location i at position t in the tour, for $i,j \in \mathscr{V}$ and $t \in \{0,\ldots,n\}$. Also, let $y_{i,t} \in \{0,1\}$ be a binary variable that indicates if location i is visited in position t of a feasible tour. Model $\mathscr{P}$ formulates the m-PDTSP:

$$\min_{x,y} \quad \sum_{t=0}^{n}\sum_{i=0}^{n}\sum_{j=0}^{n} c_{i,j}\, x_{i,j}^t \tag{$\mathscr{P}$}$$

$$\text{s.t.} \quad \sum_{i=0}^{n} y_{i,t} = 1 \qquad t = 0,\ldots,n+1, \tag{1}$$

$$\sum_{t=1}^{n} y_{i,t} = 1 \qquad i \in \mathscr{V}\setminus\{0\}, \tag{2}$$

$$\sum_{t=1}^{t'}\sum_{i=1}^{n} y_{i,t}\cdot \Delta q_i \geq 0 \qquad t' = 1,\ldots,n, \tag{3}$$

$$\sum_{t=1}^{t'}\sum_{i=1}^{n} y_{i,t}\cdot \Delta q_i \leq C \qquad t' = 1,\ldots,n, \tag{4}$$

$$\sum_{t=1}^{n} t \cdot y_{p_k,t} - \sum_{t=1}^{n} t \cdot y_{d_k,t} \leq -1 \qquad k \in \mathscr{K}, \tag{5}$$

$$y_{0,0} = y_{0,n+1} = 1,\ y_{0,t} = 0 \qquad t = 1, \ldots, n, \tag{6}$$

$$y_{i,t} - \sum_{j=0}^{n} x_{i,j}^{t} = 0 \qquad i \in \mathscr{V}, t = 0, \ldots, n, \tag{7}$$

$$y_{j,t} - \sum_{i=0}^{n} x_{i,j}^{t-1} = 0 \qquad j \in \mathscr{V},\ t = 1, \ldots, n+1, \tag{8}$$

$$y_{i,t} \in \{0,1\} \qquad i \in \mathscr{V},\ t = 0, \ldots, n+1,$$

$$x_{i,j}^{t} \in \{0,1\} \qquad i, j \in \mathscr{V},\ t = 0, \ldots, n.$$

The objective function is a reformulation of the tour cost in terms of x. Equalities (1) and (2) are matching constraints enforcing that tour positions are assigned to exactly one location and that each location must be visited exactly once, respectively. Inequalities (3) and (4) state the vehicle capacity limitation. Inequalities (5) impose precedence constraints, i.e., each pickup location is visited prior to its delivery location. Such precedence inequalities are typical, e.g., in time-indexed formulations of resource-constrained project scheduling problems (Artigues 2017). The equalities in (6) indicate that a tour should start and end at the depot. Lastly, (7) and (8) establish the connection between x and y.

A feasible solution y' to $\mathscr{P}$ defines a feasible tour π' such that $\pi'_t = \sum_{i=0}^{n} i\, y'_{i,t}$ for every $t \in \{0, \ldots, n+1\}$. Every tour can be converted to a feasible solution to $\mathscr{P}$ analogously. Also, there is an one-to-one mapping between feasible binary vectors y and x based on (7)-(8).

While other m-PDTSP formulations are also applicable in our framework, model $\mathscr{P}$ has two advantages that we exploit. First, $\mathscr{P}$ has a polynomial number of linear inequalities, which when relaxed leads to a tractable number of Lagrange multipliers that can be efficiently optimized. Second, the time-indexed variables y have a direct translation to the layered network representation of a decision diagram, as described in Sections 3.2 and 4.

3.2. Multivalued Decision Diagram Formulation

In this section, we introduce an MDD model for the m-PDTSP based on the sequencing representation by Cire and van Hoeve (2013) and Kinable et al. (2017). The model is a graphical representation of the set of feasible tours of an m-PDTSP instance, which can be limited in size to provide valid bounds for the m-PDTSP.

Formally, we define an MDD for the m-PDTSP as a directed acyclic layered graph $\mathscr{M} = (\mathscr{N}, \mathscr{A})$, where the set of nodes $\mathscr{N}$ is partitioned into $n+3$ layers $\mathscr{N}_0, \ldots, \mathscr{N}_{n+2}$. The layer of a node u is denoted by $\ell(u)$, i.e., $\ell(u) = t$ if $u \in \mathscr{N}_t$. The first and last layers, $\mathscr{N}_0$ and $\mathscr{N}_{n+2}$, are singletons containing a *root node* $\mathbf{r}$ and a *terminal node* $\mathbf{t}$, respectively. For $t \in \{0, \ldots, n+1\}$, an arc $a := (u, v) \in \mathscr{A}$ emanating from $u \in \mathscr{N}_t$ is always directed to a node $v \in \mathscr{N}_{t+1}$ in the next layer. Moreover, for each arc $a = (u, v)$, $\ell(u) = t$, we associate a label $\theta(a) \in \mathscr{V}$ that represents the location assigned to the t-th position of a tour; i.e., paths traversing a are such that $\pi_t = \theta(a)$. Thus, the set of tours encoded by $\mathscr{M}$ is:

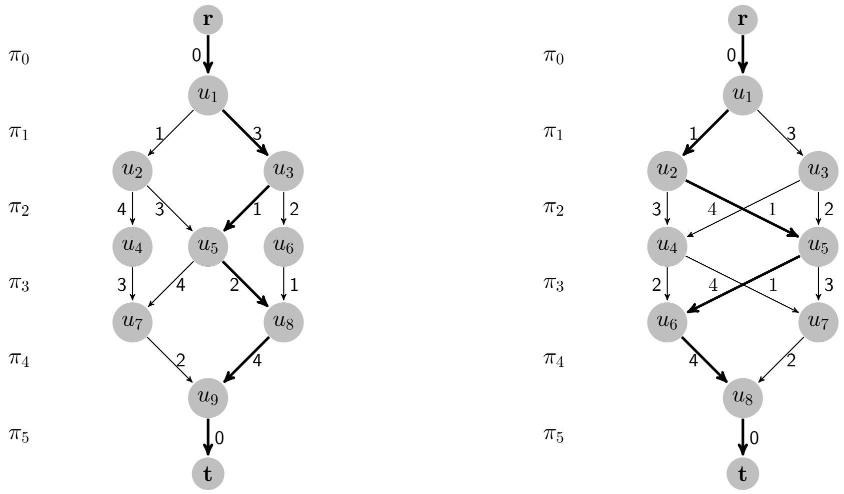
$$\mathrm{Sol}(\mathscr{M}) = \{(\theta(a_0), \theta(a_1), \theta(a_2), \ldots, \theta(a_{n+1})) : (a_0, a_1, a_2, \ldots, a_{n+1}) \text{ is an } \mathbf{r}-\mathbf{t} \text{ path in } \mathscr{M}\}.$$

An MDD is *exact* if there is an one-to-one correspondence between $\mathrm{Sol}(\mathscr{M})$ and the feasible tours of the m-PDTSP instance. Alternatively, an MDD is *relaxed* if $\mathrm{Sol}(\mathscr{M})$ over-approximates the set of feasible tours, i.e., every feasible tour is encoded in some path of $\mathscr{M}$, but not all paths in $\mathrm{Sol}(\mathscr{M})$ are necessarily feasible tours. Specifically, infeasible tours in $\mathscr{M}$ may either represent invalid permutations, violate the vehicle capacity, or fail to observe pickup-and-delivery precedence constraints. While exact MDDs are exponential in size in general, relaxed MDDs can be built with arbitrary size (Andersen et al. 2007).

Example 2. An exact MDD for the instance in Example 1 is depicted in Figure 2a. Each $\mathbf{r}-\mathbf{t}$ path encodes a feasible tour and equivalently every tour is encoded by exactly one $\mathbf{r}-\mathbf{t}$ path. In particular, path $(\mathbf{r}, u_1, u_3, u_5, u_8, u_9, \mathbf{t})$, in bold, encodes the optimal tour $\pi = (0, 3, 1, 2, 4, 0)$.

Figure 2b depicts a relaxed MDD for the same example. Every path in the exact MDD has an associated path in the relaxed MDD. Nonetheless, the relaxed MDD contains the infeasible tour $(0, 1, 4, 1, 4, 0)$ given by the path $(\mathbf{r}, u_1, u_2, u_5, u_6, u_8, \mathbf{t})$. □

For notation purposes, let $\delta^{in}(u)$ and $\delta^{out}(u)$ be the set of incoming and outgoing arcs at a node $u \in \mathscr{N}$, respectively. An arc $a := (t(a), h(a))$ emanates from a tail $t(a) \in \mathscr{N}$ and points at a head $h(a) \in \mathscr{N}$. The layer $\ell(a)$ of an arc a is the layer of its tail node, i.e., $\ell(a) = \ell(t(a))$. The width $\omega(\mathscr{M})$ of an MDD $\mathscr{M}$ is the maximum number of nodes in any layer, i.e., $\omega(\mathscr{M}) := \max_{t=1,\ldots,n+1} |\mathscr{N}_t|$.



(a) MDD representation. (b) 2-width relaxed MDD

Figure 2 Exact and Relaxed MDDs for Example 1.

Optimizing Travel Costs over $\mathscr{M}$. If $\mathscr{M}$ is an exact MDD encoding Hamiltonian paths in a graph, Kinable et al. (2017) show that the minimum total travel cost, i.e., $\min\{c(\pi) : \pi \in \mathrm{Sol}(\mathscr{M})\}$, can be found in polynomial time in the size of $\mathscr{M}$. To this end, the authors equip $\mathscr{M}$ with a more general travel cost matrix ζ, where $\zeta^t_{i,j}$ represents the cost of traveling from location i to location j when i is assigned to the t-th position of the tour, for $i, j \in \mathscr{V}$ and $t \in \{0, \dots, n\}$. With such a matrix ζ, the cost $c(\pi)$ of a tour π becomes

$$c(\pi) = \sum_{t=0}^{n} \zeta^t_{\pi_t, \pi_{t+1}}. \tag{9}$$

Note that $\zeta^t_{i,j} = c_{i,j}$ for all $t \in \{0, \dots, n\}$ in any given m-PDTSP instance, i.e., we could drop the additional index t. Nonetheless, we maintain this general cost representation when optimizing over $\mathscr{M}$, as it will be later directly applied to our Lagrangian dual in Section 4.2, where travel costs are also position dependent.

Let $\tau(a)$ be the minimum cost of all partial tours encoded by paths starting at the root $\mathbf{r}$ and ending at an arc $a \in \mathscr{A}$. Such values are obtained using the recurrence

$$\tau(a) := \begin{cases} 0, & \text{if } \ell(a) = 0, \\ \min_{a' \in \delta^{in}(t(a))} \left\{ \tau(a') + \zeta^{\ell(a')}_{\theta(a'),\theta(a)} \right\}, & \text{otherwise.} \end{cases} \tag{10}$$

That is, for an arc a in an layer $\ell(a) > 1$, $\tau(a)$ is the minimum cost among its possible predecessor locations, i.e., all locations $\theta(a')$ such that $a' \in \delta^{in}(t(a))$, plus the cost to travel from the predecessor to the arc's location $\theta(a)$. The optimal tour cost ν^* is, by definition,

$$\tau(\mathscr{M}) := \min_{a \in \mathscr{A}: \ell(a) = n+1} \tau(a). \tag{11}$$

A proof of the validity of (10) is presented by Kinable et al. (2017) in the context of time-dependent sequencing. If $\mathscr{M}$ has a width of $\omega(\mathscr{M})$, all such values can be computed with a breadth-first search traversal in $\mathscr{O}(n|\mathscr{A}|\omega(\mathscr{M}))$.

If $\mathscr{M}$ is a relaxed MDD of arbitrary size, the value obtained in (11) provides instead a lower bound to the optimal solution value of the m-PDTSP. This follows since $\mathrm{Sol}(\mathscr{M})$ over-approximates the set of feasible tours. We further investigate relaxed MDD construction techniques for the m-PDTSP in Section 5, based on our Lagrangian dual framework established in Section 4.

Example 3. Consider the exact MDD presented in Figure 2a. We use (10) to compute the arc costs $\tau((\mathbf{r}, u_1)) = 0$, $\tau((u_1, u_2)) = \tau((\mathbf{r}, u_1)) + \zeta^0_{\theta((\mathbf{r},u_1)),\theta((u_1,u_2))} = c_{0,1} = 447$, and analogously for the remaining arcs. In particular, the cost of arc (u_5, u_8) is given by

$$\begin{aligned}\tau((u_5,u_8)) &= \min\{\tau((u_2,u_5)) + \zeta^1_{\theta((u_2,u_5)),\theta((u_5,u_8))},\ \tau((u_3,u_5)) + \zeta^1_{\theta((u_3,u_5)),\theta((u_5,u_8))}\} \\ &= \min\{\tau((u_2,u_5)) + c_{3,2},\ \tau((u_3,u_5)) + c_{1,2}\} = \min\{1154 + 666,\ 1131 + 295\} = 1426.\end{aligned}$$

Similarly, we apply recursion (10) to compute the arc costs of the relaxed MDD shown in Figure 2b. In this case, the optimal tour cost (11) is given by $\tau(\mathscr{M}) = \min\{\tau((u_8, \mathbf{t}))\} = 1811$, which encodes the infeasible tour $\pi = (0, 1, 4, 1, 4, 0)$. □

4. An MDD-based Lagrangian Dual for the m-PDTSP

This section introduces the Lagrangian dual for the m-PDTSP that combines the ILP and the MDD formulations described in Section 3. Specifically, our main purpose is to provide a new mechanism to obtain valid lower bounds to the problem. Such bounds can be used, e.g., to certify the quality of a feasible solution or to enhance a branch-and-bound search.

Our approach assumes that we are given a relaxed MDD $\mathscr{M}$ for an m-PDTSP instance (e.g., the one proposed in Section 5). Because $\mathscr{M}$ and a linear relaxation of $\mathscr{P}$ may be complementary in terms of the combinatorial structure each encodes, we wish to combine them into a single model that leverages the strengths of both formulations. To this end, we propose a Lagrangian dual that incorporates information from the LP relaxation of $\mathscr{P}$ as costs into $\mathscr{M}$, building on earlier works integrating dynamic programming and MDDs with Lagrangian relaxation (Beasley and Cao 1998, Bergman et al. 2015).

In the remainder of this section, we first describe a model that integrates both relaxations to enhance bounds. Next, we show how such model can be addressed by solving its Lagrangian dual, which yields a subproblem that is polynomially solvable in $\mathscr{M}$.

4.1. Hybrid ILP-MDD Relaxation for the m-PDTSP

For ease of notation, let $A \in \mathbb{R}^{r\times|\mathscr{V}|\times(n+2)}$ and $b \in \mathbb{R}^r$ represent the matrix coefficients and right-hand side vector of the inequalities in $\mathscr{P}$, respectively, that only involve y. Specifically,

$$\left\{y \in \mathbb{R}^{|\mathscr{V}|\times(n+2)} : Ay \le b\right\} = \left\{y \in \mathbb{R}^{|\mathscr{V}|\times(n+2)} : y \text{ satisfies (1)-(5)}\right\},$$

assuming an appropriate dimension $r > 0$ encoding the number of constraints. In particular, an element $a_{l,i,t}$ of A is the coefficient of the variable $y_{i,t}$ in the l-th constraint of $\mathscr{P}$. Notice that $Ay \le b$ models the matching, capacity, and precedence constraints of $\mathscr{P}$.

Furthermore, for a given exact or relaxed MDD $\mathscr{M}$, let $\mathrm{Sol}_\gamma(\mathscr{M})$ be the set of binary solutions (x,y) that can be mapped to a tour encoded by $\mathscr{M}$. That is,

$$\mathrm{Sol}_\gamma(\mathscr{M}) := \Bigg\{(x,y) \text{ binaries} : \exists\, \pi \in \mathrm{Sol}(\mathscr{M}) \text{ s.t. } \sum_{i=0}^{n} i\, y_{i,t} - \pi_t = 0 \text{ for all } t \in \{0,\dots,n+1\}, \text{ and } (x,y) \text{ satisfies (7)-(8)}\Bigg\},$$

where the dimensions of (x,y) are as in $\mathscr{P}$, omitted above for exposition. Note that there is an one-to-one mapping between a vector pair $(x,y) \in \mathrm{Sol}_\gamma(\mathscr{M})$ and a tour $\pi \in \mathrm{Sol}(\mathscr{M})$.

Let $\mathscr{M}$ be a relaxed MDD and denote the convex hull of a set $\mathscr{X}$ by $\mathrm{conv}(\mathscr{X})$. We propose a new bound for the m-PDTSP obtained by solving the following hybrid relaxation $\mathscr{H}$:

$$\begin{aligned} \nu_R := \min_{x,y} \quad & \sum_{t=0}^{n}\sum_{i=0}^{n}\sum_{j=0}^{n} c_{i,j}\, x_{i,j}^t \\ \text{s.t.} \quad & Ay \le b, \\ & (x,y) \in \mathrm{conv}(\mathrm{Sol}_\gamma(\mathscr{M})). \end{aligned} \tag{$\mathscr{H}$}$$

The optimal solution value of $\mathscr{H}$ is a lower bound to the original problem, i.e., $\nu^* \ge \nu_R$, since $\mathscr{H}$ is the intersection of two over-approximations of the feasible tour set.

The choice of formulation $\mathscr{H}$ follows from three key motivations. First, since a convex hull can be equivalently described by a set of linear inequalities, problem $\mathscr{H}$ is a well-defined linear program that captures both the linear relaxation of $\mathscr{P}$ and the relaxation structure of $\mathscr{M}$. Second, using Lagrangian duality, $\mathscr{H}$ can be solved by exploiting an efficient combinatorial algorithm over $\mathscr{M}$. Finally, the bound provided by $\mathscr{H}$ is never worse than the original MDD bound or the LP relaxation of $\mathscr{P}$ when each is considered separately. As indicated by our numerical study, such a bound is often stronger and leads to significant speed-ups in our branch-and-bound search.

4.2. Solving $\mathscr{H}$ by Lagrangian Duality

We address $\mathscr{H}$ by dropping inequalities $Ay \leq b$ and penalizing their violation in the objective function with Lagrange multipliers λ. This yields a Lagrangian dual that observes strong duality with respect to $\mathscr{H}$. Namely, by Conforti et al. (2014), Theorem 8.2, we have that

$$\nu_R = \max_{\lambda} \{\mathscr{L}(\lambda) : \lambda \geq 0\}, \qquad (\mathscr{D})$$

where $\mathscr{L}(\cdot)$ is the *Lagrangian subproblem* defined as

$$\mathscr{L}(\lambda) := \min_{x,y} \left\{ \sum_{t=0}^{n} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{i,j} x_{i,j}^t + \lambda^T (Ay - b) : (x,y) \in \mathrm{Sol}_\gamma(\mathscr{M}) \right\}.$$

Notice that $\mathscr{L}(\cdot)$ optimizes a linear function over $\mathrm{Sol}_\gamma(\mathscr{M})$ as opposed to the convex hull of such set. We now show that $\mathscr{L}(\cdot)$ is also tractable in the size of $\mathscr{M}$.

PROPOSITION 1. *For any $\lambda \geq 0$, the Lagrangian subproblem $\mathscr{L}(\lambda)$ can be solved in polynomial time in the size of $\mathscr{M}$. Specifically,*

$$\mathscr{L}(\lambda) = \tau(\mathscr{M}) + \lambda^T b,$$

where $\tau(\mathscr{M})$ is defined as in (11) and computed using the MDD cost structure

$$\zeta_{i,j}^t = c_{i,j} + \sum_{l=1}^{r} (\lambda_l \, a_{l,i,t})$$

for all $i, j \in \mathscr{V}$ and $t \in \{0, 1, \ldots, n\}$.

Proof. By definition, any $(x,y) \in \mathrm{Sol}_\gamma(\mathscr{M})$ maps to a (unique) $\pi \in \mathrm{Sol}(\mathscr{M})$. For such a triple (x,y,π), recall that $\sum_{t=0}^n \sum_{i=0}^n \sum_{j=0}^n c_{i,j} x_{i,j}^t = c(\pi) = \sum_{t=0}^n c_{\pi_t,\pi_{t+1}}$. Thus,

$$\begin{aligned}
\sum_{t=0}^n \sum_{i=0}^n \sum_{j=0}^n c_{i,j} x_{i,j}^t + \lambda^T(Ay-b) &= \sum_{t=0}^n c_{\pi_t,\pi_{t+1}} + \sum_{l=1}^r \lambda_l \left(\sum_{i=0}^n \sum_{t=0}^n a_{l,i,t} y_{i,t} - b_l \right) \\
&= \sum_{t=0}^n c_{\pi_t,\pi_{t+1}} + \sum_{t=0}^n \sum_{i=0}^n \left(\sum_{l=1}^r \lambda_l a_{l,i,t} \right) y_{i,t} - \lambda^T b \\
&= \sum_{t=0}^n \left(c_{\pi_t\,\pi_{t+1}} + \sum_{l=1}^r \sum_{i=0}^n (\lambda_l a_{l,i,t}) y_{i,t} \right) - \lambda^T b \\
&= \sum_{t=0}^n \left(c_{\pi_t,\pi_{t+1}} + \sum_{l=1}^r (\lambda_l a_{l,\pi_t,t}) \right) - \lambda^T b.
\end{aligned}$$

This implies that $\sum_{t=0}^n \sum_{i=0}^n \sum_{j=0}^n c_{i,j} x_{i,j}^t + \lambda^T Ay = \sum_{t=0}^n \zeta_{\pi_t,\pi_{t+1}}^t$ for costs $\zeta_{\pi_t,\pi_{t+1}}^t$ of the same form (9) as required for $\tau(\mathscr{M})$. ■

4.3. Incorporating the Lagrange Multipliers in $\mathscr{M}$

In this section, we provide further details on how to incorporate the Lagrange multipliers associated with the linear system $Ay \leq b$ into a relaxed MDD $\mathscr{M}$ by means of Proposition 1. Our implementation focuses on the Lagrange multipliers related to the tour equalities (2), the capacity inequalities (3)-(4), and the precedence inequalities (5). Notice that we are not required to consider inequalities (1) and (6) since they are enforced by construction of $\mathscr{M}$. Other valid linear inequalities to the m-PDTSP, however, can be incorporated analogously.

Consider the Lagrange multipliers $\lambda = (\beta, \mu, \sigma)$, where $\beta \in \mathbb{R}^{|\mathscr{V}|}$, $\mu \in \mathbb{R}^{2n}$ $(\mu \geq 0)$, and $\sigma \in \mathbb{R}^{|\mathscr{K}|}$ $(\sigma \geq 0)$ are associated with constraints (2), (3)-(4), and (5), respectively. For this set of multipliers, the cost matrix ζ of Proposition 1 is given by

$$\zeta_{i,j}^t = c_{i,j} + \quad \beta_i \quad + \quad \Delta q_i \sum_{t'=t}^n (\mu_{n+t'} - \mu_{t'}) \quad + \quad t \left(\sum_{\{k \in \mathscr{K}: p_k = i\}} \sigma_k - \sum_{\{k \in \mathscr{K}: d_k = i\}} \sigma_k \right).$$

Notice that the constant $\lambda^T b$ is

$$\lambda^T b = -\sum_{i \in \mathscr{V}} \beta_i - \sum_{t=n+1}^{2n} C\mu_t + \sum_{k \in \mathscr{K}} \sigma_k.$$

The cost matrix ζ is used in recurrence (10)-(11) to compute a new lower bound over $\mathscr{M}$, as illustrated in the example below. Any valid set of multipliers suffice to obtain a valid lower bound for the m-PDTSP. In particular, the strongest bound is at least as strong as the one obtained from the original relaxed MDD $\mathscr{M}$ (Fisher 2004).

EXAMPLE 4. Consider our running example and the relaxed MDD $\mathscr{M}$ shown in Figure 2b. Suppose, for illustration purposes, that we incorporate only equalities (2) and let $\beta \in \mathbb{R}^{|\mathscr{V}|}$ be the vector of Lagrange multipliers associated with (2). If we set $\beta_1 = \beta_4 = 100$ and $\beta_i = 0$ for all $i \in \mathscr{V} \setminus \{1,4\}$, we obtain $\mathscr{L}(\lambda) = \tau(\mathscr{M}) + \lambda^T b = 1863$. This solution corresponds to the optimal tour $\pi = (0,3,1,2,4,0)$ and improves the original MDD bound of 1811. □

4.4. Solution Method for the Lagrangian Dual

Problem $\mathscr{D}$ is the *Lagrangian dual* problem, maximization problem with a piecewise linear concave objective (Fisher 2004). It can be solved iteratively by computing $\mathscr{L}(\lambda_0)$ for some $\lambda_0 \geq 0$, obtaining a new set of Lagrange multipliers λ_1 based on the solution of $\mathscr{L}(\lambda_0)$, and repeating until some termination criteria is reached. The function $\mathscr{L}(\cdot)$ can be computed efficiently in $\mathscr{O}(n|\mathscr{A}|\omega(\mathscr{M}))$ as described in Sections 4.2 and 4.3.

For the update of the Lagrange multipliers, we apply Bundle methods (Lemaréchal 1975) that have a relatively fast convergence rate in comparison to other methods; i.e., typically bounded by $\mathscr{O}(1/\epsilon^3)$ for a given precision ϵ. A Bundle method is a variant of the cutting plane method, which adds a quadratic stabilizer to improve convergence. The method iteratively solves $\mathscr{L}(\lambda)$ until it converges to the optimal set of multipliers λ^*. Each optimal solution (x', y') of $\mathscr{L}(\lambda)$ has an associated subgradient, $Ay' - b$, that is used to generate a cutting plane that is valid to the function $\mathscr{L}(\cdot)$.

For the specific case of the m-PDTSP, consider the k-th iteration of the procedure with λ^k as the current set of multipliers associated with the system $Ay \leq b$. Our implementation solves $\mathscr{L}(\lambda^k)$ using the recursive arc cost procedure ((10)-(11)) over $\mathscr{M}$ with the cost structure shown in Proposition 1. Solution $\pi^k = \arg\max\{\tau(\mathscr{M})\}$ is mapped to (x^k, y^k) for the subgradient computation, $Ay^k - b$. The method then solves the following quadratic problem to generate a new set of multipliers λ_{k+1},

$$\lambda^{k+1} = \arg\max_{\lambda} \left\{ z + \frac{1}{2t}||\lambda - \lambda^k|| : z \leq \mathscr{L}(\lambda^s) + (Ay^s - b)^T(\lambda - \lambda^s),\ \forall s = 0, \ldots, k, z \in \mathbb{R},\ \lambda \in \mathbb{R}^r_{\geq 0} \right\}.$$

In the problem above, z is a variable that over approximates the Lagrangian dual bound, and λ corresponds to the set of Lagrange multipliers. The objective function includes a quadratic stabilizer $\frac{1}{2t}||\lambda - \lambda^k||$ $(t < 1)$ to improve convergence (Lemaréchal 1975). In each iteration of the procedure, the set of constraints increases by one, where each new constraint is a cutting plane derived from the subgradients in the previous iterations.

5. Relaxed MDD Construction

In this section, we describe the relaxed MDD construction procedure that we use as an input to the hybrid model $\mathscr{H}$. Our methodology is a variant of the incremental refinement framework by Andersen et al. (2007). Specifically, we generate a sequence of relaxed MDDs $\mathscr{M}_0, \mathscr{M}_1, \mathscr{M}_2, \ldots$ that finitely converge to the exact MDD representation of an instance. In this framework, an iterate $\mathscr{M}_{t+1}$ is constructed by expanding nodes of the previous diagram $\mathscr{M}_t$ to rule out infeasible solutions based on the constraint set of the problem.

In the context of the m-PDTSP, an infeasible solution is a tour that violates at least one of the three possible constraints: (i) the ones imposing that each location must be visited exactly once (i.e., the tour constraints); (ii) the ones which enforce that the vehicle capacity must be observed at all visits; or (iii) the precedence relations that are implied by the pickup-and-delivery conditions. While such conditions are approximated by inequalities (1)-(5) in $\mathscr{H}$, we can leverage the flexibility of MDDs to better encode the constraints that are not tightly relaxed by these inequalities alone.

We propose a relaxed MDD construction that enforces each of the three constraint classes one at a time, until either they are satisfied by all tours in the MDD, or a maximum allowed size of the network is reached. To this end, we start with a small valid relaxed MDD and rule out the solutions violating a particular constraint by expanding nodes and removing arcs accordingly. Once a constraint type has been fully observed by the paths of the MDD, we repeat this procedure iteratively with the remaining constraints. We give priority to capacity constraints, which are well-known to be challenging in integer programming formulations for the m-PDTSP (Letchford and Salazar-González 2016), while conversely MDD relaxations may be weak when enforcing tour constraints (Cire and van Hoeve 2013) and are better represented by linear assignment constraints.

The overall construction procedure is depicted in Algorithm 1. We start with a width-one trivial relaxed MDD, as depicted in Figure 3a. Next, for each layer, we expand nodes to first satisfy the vehicle capacity constraints (procedure ExpandNodesCapacity), and second to satisfy the tour constraints and precedence relations (procedure ExpandNodesTour), while ensuring that the width $\omega(\mathscr{M})$ of the network does not exceed $\mathscr{W} > 0$. We note that the order of the expansion procedures can be inverted without loss of generality.

Algorithm 1 Relaxed MDD construction for the m-PDTSP

procedure ConstructMDD(**Input:** m-PDTSP instance I, max. width $\mathscr{W}$)
  Construct an width-one relaxed MDD $\mathscr{M}$ for I
  for each layer $t = 1, \dots, n+1$ **do**
    ExpandNodesCapacity($\mathscr{N}_t$, $\mathscr{W}$)
    ExpandNodesTour($\mathscr{N}_t$, $\mathscr{W}$)
  return $\mathscr{M}$.

5.1. Capacity Constraints

In this section, we develop the ExpandNodesCapacity procedure for Algorithm 1. Its main purpose is to modify a given relaxed MDD so that the vehicle capacity constraints are satisfied by the paths in $\mathrm{Sol}(\mathscr{M})$.

Let $\mathscr{M} = (\mathscr{N}, \mathscr{A})$ be a relaxed MDD. With each node $u \in \mathscr{N}$, we associate two labels, $Q^{\min}(u), Q^{\max}(u) \in \mathbb{R}$, representing the minimum and maximum accumulated net weights, respectively, of all partial tours starting at $\mathbf{r}$ and ending at u. That is,

$$Q^{\min}(u) := \min \left\{ \sum_{t=0}^{\ell(u)-1} \Delta q_{\theta(a_t)} : (a_0, a_1, \dots, a_{\ell(u)-1}) \text{ is an } \mathbf{r} - u \text{ path in } \mathscr{M} \right\},$$

and analogously for $Q^{\max}(u)$ with "min" replaced by "max". Such labels can be efficiently computed by a breadth-first search on $\mathscr{M}$ by fixing $Q^{\min}(\mathbf{r}) = Q^{\max}(\mathbf{r}) = 0$ and, for any node $u \neq \mathbf{r}$, computing the following recurrence during a top-down search:

$$Q^{\min}(u) = \min_{a \in \delta^{in}(u)} \left\{ Q^{\min}(t(a)) + \Delta q_{\theta(a)} \right\} \text{ and} \tag{12}$$

$$Q^{\max}(u) = \max_{a \in \delta^{in}(u)} \left\{ Q^{\max}(t(a)) + \Delta q_{\theta(a)} \right\}. \tag{13}$$

The labels above provide a mechanism to measure the degree of infeasibility of $\mathscr{M}$ with respect to the capacity constraints. This is formalized in the result below.

Proposition 2. *For all $u \in \mathscr{N}$ and some $\epsilon \geq 0$, suppose that (i) $Q^{\min}(u) \leq C$, $Q^{\max}(u) \geq 0$; and (ii) $Q^{\max}(u) - Q^{\min}(u) \leq \epsilon$. Then, for all $\pi \in \mathrm{Sol}(\mathscr{M})$,*

$$-\epsilon \leq \sum_{t=0}^{t'} \Delta q_{\pi_t} \leq C + \epsilon, \qquad t' = 0, \dots, n+1. \tag{14}$$

That is, tours in $\mathscr{M}$ violate the capacity constraints by at most ϵ.

Proof. Assume (i) and (ii) hold for some $\epsilon \geq 0$ as defined in the proposition statement. For any partial path π from the root $\mathbf{r}$ to a node $u \in \mathscr{N}_t$, $t \geq 1$,

$$\sum_{t=0}^{\ell(u)-1} \Delta q_{\pi_t} \leq Q^{\max}(u) \leq Q^{\min}(u) + \epsilon \leq C + \epsilon.$$

The first inequality follows from the definition of $Q^{\max}(\cdot)$. The second follows from condition (ii), and the last is from (i). An equivalent reasoning can be used for $\sum_{t=1}^{t'} \Delta q_{\pi_t} \geq -\epsilon$. ■

We can assume condition (i) from Proposition 2 always holds, since otherwise we can simply remove the violating nodes from $\mathscr{M}$ as they only encode infeasible paths. This leads directly to the following corollary.

COROLLARY 1. *The tours in* $\mathrm{Sol}(\mathscr{M})$ *satisfy the vehicle capacity constraints if*

$$0 \leq Q^{\min}(u) = Q^{\max}(u) \leq C, \qquad \text{for all } u \in \mathscr{N}_t,\ t = 0, \ldots, n+2.$$

We can now state our expansion procedure, which is formalized in Algorithm 2. Given the nodes $\mathscr{N}_t$ in a layer t, we first compute labels $Q^{\min}(\cdot)$ and $Q^{\max}(\cdot)$ using (12) and (13), respectively. Next, we define the expansion set $E^C(\mathscr{N}_t)$ of $\mathscr{N}_t$ as the set of nodes in $\mathscr{N}_t$ from which all parents satisfy the conditions of Corollary 1, i.e.,

$$E^C(\mathscr{N}_t) := \left\{ u \in \mathscr{N}_t : Q^{\min}(t(a)) = Q^{\max}(t(a)) \text{ for all } a \in \delta^{in}(u) \right\}.$$

Notice that, by definition of the labels, a node $\overline{u} \notin E^C(\mathscr{N}_t)$ can never be expanded to satisfy the corollary conditions, since the minimum net weight at $\overline{u}$ will always be strictly lower than its maximum net weight regardless on how its incoming arcs are partitioned.

If all $u \in E^C(\mathscr{N}_t)$ are such that $Q^{\min}(u) = Q^{\max}(u)$, then, by Corollary 1, no more expansion is needed. Otherwise, consider $u \in E^C(\mathscr{N}_t)$ such that $Q^{\min}(u) < Q^{\max}(u)$. Furthermore, define $\mathscr{A}^{\min}(u)$ as the set of incoming arcs at u that certify the label $Q^{\min}(u)$, i.e.,

$$\mathscr{A}^{\min}(u) := \left\{ a \in \delta^{in}(u) : Q^{\min}(t(a)) + \Delta q_{\theta(a)} = Q^{\min}(u) \right\}. \tag{15}$$

If the limit $\mathscr{W}$ on the maximum width is not met, we create a new node v and redirect the tail of the arcs in $a \in \mathscr{A}^{\min}(u)$ to v. This redirection will ensure, by construction, that $Q^{\min}(v) = Q^{\max}(v)$ and will increase $Q^{\min}(u)$ as shown in Proposition 3. We also copy the arcs emanating from u and assign them to emanate from v to ensure that the paths originally crossing arcs in $\mathscr{A}^{\min}(u)$ are preserved. Finally, we update the labels $Q^{\min}(\cdot)$ and $Q^{\max}(\cdot)$ accordingly and repeat the procedure until either the maximum width is met or Corollary 1 is satisfied for the nodes in that layer.

Algorithm 2 Expanding nodes based on capacity.

1: **procedure** ExpandNodesCapacity(**Input:** MDD node set $\mathscr{N}_t$, max. width $\mathscr{W}$)
2: Compute labels $Q^{\min}(u)$, $Q^{\max}(u)$ for each $u \in E^C(\mathscr{N}_t)$.
3: **while** $|\mathscr{N}_t| < \mathscr{W}$ and $\exists u \in E^C(\mathscr{N}_t)$ such that $Q^{\max}(u) - Q^{\min}(u) > 0$ **do**
4: Create a new node v, add it to $\mathscr{N}_t$.
5: Set $t(a^*) = v$ for all $a^* \in \mathscr{A}^{\min}(u)$, as defined in (15).
6: For every arc $a \in \delta^{out}(u)$, create a new arc $a' = (v, h(a'))$.
7: Update labels $Q^{\min}(u), Q^{\max}(u), Q^{\min}(v), Q^{\max}(v)$.

Proposition 3. *For a sufficiently large $\mathscr{W}$, the procedure* ExpandNodesCapacity *ensures that, for every node $u \in \mathscr{N}_t$, $Q^{\min}(u) = Q^{\max}(u)$.*

Proof. It suffices to show that the procedure ends for any arbitrarily large $\mathscr{W}$. Assume all previous layers satisfy the condition of the statement and consider an iteration that chooses a certain u such that $Q^{\min}(u) < Q^{\max}(u)$. The new node v satisfies $Q^{\min}(v) = Q^{\max}(v)$ because of (15). Moreover, for all arcs $a' \in \delta^{in}(u) \setminus \mathscr{A}^{\min}(u)$,

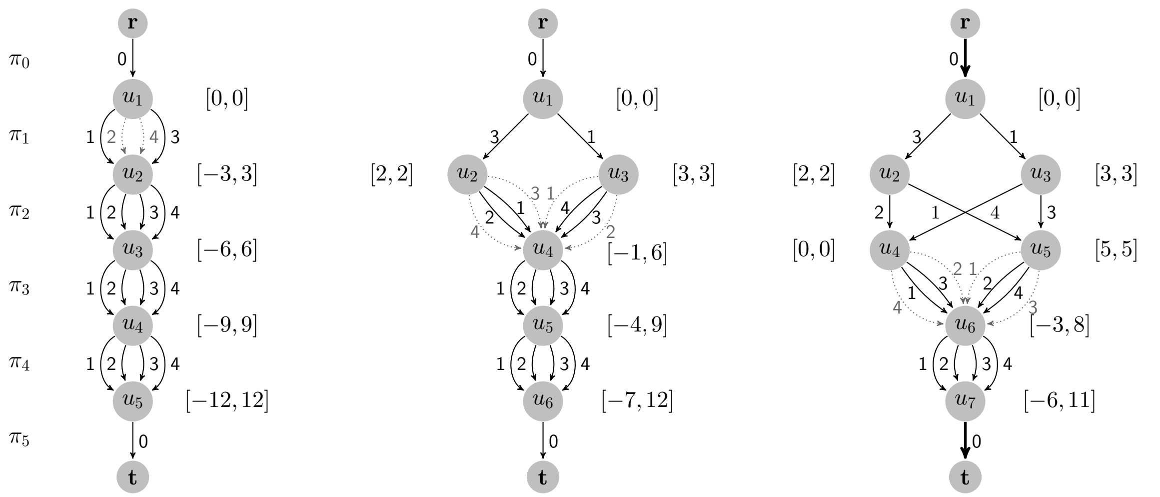
$$Q^{\min}(t(a')) + \Delta q_{\theta(a')} > Q^{\min}(u)$$

and therefore the updated label $Q^{\min}(u)$ strictly increases. Since $Q^{\max}(u)$ is finite and remains constant for the next iterations that pick the same node u, the result follows. ■

Proposition 3 and Corollary 1 ensure that, for a sufficiently large width $\mathscr{W}$, all tours in $\mathscr{M}$ satisfy the capacity constraints. Note that, if the input node set has a cardinality of one (as in our width-one MDD case), then the maximum width required for Proposition 3 is $C+1$, since no two nodes will have the same labels. Second, the choice of node u in line 3 of Algorithm 2 can be done in a systematic fashion. For instance, if we choose the node with maximum $\epsilon := Q^{\max}(u) - Q^{\min}(u)$, we move towards decreasing the total violation ϵ of paths, based on Proposition 2. In our numerical experiments, we choose $\mathscr{W}$ to be sufficiently large (i.e., $C+1$) to ensure all paths do not violate vehicle capacities.

Example 5. Figure 3 shows three relaxed MDDs that illustrate the ExpandNodesCapacity procedure for our running example. Notice that dotted lines indicate infeasible arcs identified by our filtering algorithms (Section 5.3).

Starting with the width-one MDD in Figure 3a, the procedure expands layer $\mathscr{N}_2$ that has a single node u_2 with $Q^{\min}(u_2) = -3$ and $Q^{\max}(u_2) = 3$. The resulting MDD is shown in Figure 3b and has two nodes satisfying Corollary 1. Similarly, Figure 3c illustrate the ExpandNodesCapacity procedure when it is applied to $\mathscr{N}_3$. □



(a) Width-one relaxed MDD (b) Capacity node expansion in $\mathscr{N}_2$ (c) Capacity node expansion in $\mathscr{N}_3$

Figure 3 Construction procedure. Depicts ExpandNodesCapacity procedure and filtering rules.

5.2. Tour and Precedence Constraints

In this section, we develop the ExpandNodesTour procedure for Algorithm 1 to impose that tours in a relaxed MDD $\mathscr{M}$ do not violate precedence and tour constraints. We proceed analogously to the capacity case and associate a label to each node, which will be used to identify when an expansion is necessary. Such labels have been previously investigated by Andersen et al. (2007) and Cire and van Hoeve (2013), who derive them from the classical dynamic programming state representation for the traveling salesperson problem (see, e.g., Christofides et al. 1981). In the context of the m-PDTSP, we consider a simpler variant of such labels that suffices to impose our desired conditions.

Let $\mathscr{M} = (\mathscr{N}, \mathscr{A})$ be a relaxed MDD. For this expansion procedure, we associate one label $L(u) \subseteq \mathscr{V}$ to each MDD node $u \in \mathscr{N}$. The label $L(u)$ represents the subset of locations that are visited by all $\mathbf{r} - u$ paths in $\mathscr{M}$. That is,

$$L(u) := \bigcap \left\{ \bigcup_{t=0}^{\ell(u)-1} \theta(a_t) : (a_0, a_1, \ldots, a_{\ell(u)-1}) \text{ is an } \mathbf{r} - u \text{ path in } \mathscr{M} \right\}.$$

Such a label can be efficiently computed by a breadth-first search on $\mathscr{M}$ by fixing $L(\mathbf{r}) = \emptyset$ and, for any node $u \neq \mathbf{r}$,

$$L(u) = \bigcap_{a \in \delta^{in}(u)} \{L(t(a)) \cup \{\theta(a)\}\}. \tag{16}$$

The tour and precedence feasibility can be verified by the following proposition.

PROPOSITION 4. *For all $u \in \mathcal{N}_t$, $t = 0, \dots, n+1$, suppose that*

(i) $|L(u)| = \ell(u)$*; and*

(ii) For all arcs $a \in \delta^{out}(u)$ emanating from u, the deliveries in $\theta(a)$ are preceded by their associated pickups in $L(u)$, i.e., if $d_k = \theta(a)$ for some commodity $k \in \mathcal{K}$, then $p_k \in L(u)$.

If (i) and (ii) hold, the tours $\pi \in \mathrm{Sol}(\mathcal{M})$ satisfy the tour and precedence constraints.

Proof. We show by induction on t that, for any node $u \in \mathcal{N}_t$, the tours associated with $\mathbf{r} - u$ paths satisfy the tour and precedence constraints. This is trivially valid for the basis case $t = 0$. Assume now this statement holds for $t = 0, \dots, t'$ for some $t' \geq 1$.

Pick any node $v \in \mathcal{N}_{t'+1}$. For an arc $a \in \delta^{in}(v)$ and a tour $\pi' = (\pi_0, \pi_1, \dots, \pi_{t'-1})$ encoded by an $\mathbf{r} - t(a)$ path, the recursion (16) implies that $|L(v)| = \ell(v)$ is true only if $\theta(a) \neq \pi_t$ for all $t = 0, \dots, t'-1$; i.e., the extended tour $\pi := (\pi', \theta(a))$ satisfies the tour constraints. Moreover, assumption (ii) directly implies that π also satisfies precedence constraints. ∎

Algorithm 3 states the tour expansion procedure. Given the nodes $\mathcal{N}_t$, we first compute the labels $L(\cdot)$ using (16). Next, we define the expansion set $E^L(\mathcal{N}_t)$ of $\mathcal{N}_t$ as the set of nodes in $\mathcal{N}_t$ from which all parents satisfy the assumption (i) of Proposition 4:

$$E^L(\mathcal{N}_t) := \left\{ u \in \mathcal{N}_t : L(t(a)) = |\ell(t(a))| \text{ for all } a \in \delta^{in}(u) \right\}.$$

Notice that, by definition of the labels, a node $\overline{u} \notin E^L(\mathcal{N}_t)$ can never be expanded to satisfy the required assumption, since the cardinality of L can increase by at most one per layer.

If all nodes $u \in E^L(\mathcal{N}_t)$ are such that $|L(u)| = \ell(u)$, then, by Proposition 4, no more expansion is needed and we can stop. Otherwise, let $u \in E^L(\mathcal{N}_t)$ be a node such that $|L(u)| < \ell(u)$. Furthermore, for any $\hat{a} \in \delta^{in}(u)$, define $\mathcal{A}^L(v, \hat{a})$ as the set of incoming arcs at u that lead to the same label $L(\cdot)$ as when applying $\hat{a}$, i.e.,

$$\mathcal{A}^L(u, \hat{a}) := \left\{ a \in \delta^{in}(u) : L(t(a)) \cup \{\theta(a)\} = L(t(\hat{a})) \cup \{\theta(\hat{a})\} \right\}. \tag{17}$$

If the limit $\mathcal{W}$ on the maximum width is not met, we select any $\hat{a} \in \delta^{in}(u)$, create a new node v, and redirect the tail of the arcs in $a \in \mathcal{A}^L(u, \hat{a})$ to v, which imposes assumption (i) from Proposition 4. We also copy the arcs emanating from u to also emanate from v to ensure that the paths originally crossing arcs in $\mathcal{A}^L(u, \hat{a})$ are preserved, update the labels

Algorithm 3 Expanding nodes based on tour and precedence.

procedure ExpandNodesTour(**Input:** MDD node set $\mathscr{N}_t$, max. width $\mathscr{W}$)
  Compute label $L(u)$ for each $u \in \mathscr{N}_t$.
  while $|\mathscr{N}_t| < \mathscr{W}$ and $\exists u \in E^L(\mathscr{N}_t)$ such that $|L(u)| < \ell(u)$ **do**
    Create a new node v, add it to $\mathscr{N}_t$.
    Select any arc $\hat{a} \in \delta^{in}(u)$.
    Set $t(a) = v$ for all $a \in \mathscr{A}^L(u, \hat{a})$, as defined in (17).
    For every arc $a \in \delta^{out}(u)$, create a new arc $a' = (v, h(a))$.
    Update labels $L(u), L(v)$.
  for $u \in \mathscr{N}_t$ **do**
    Remove all arcs $a \in \delta^{out}(u)$ such that $\theta(a) \in L(u)$.
    Remove all arcs $a \in \delta^{out}(u)$ such that $\exists k \in \mathscr{K}$ with $d_k = \theta(a)$, $p_k \notin L(u)$.

accordingly and repeat the procedure. Finally, we include an extra step to remove arcs that violate condition (ii) of Proposition 4.

We state the following result, with proof analogous to Proposition 3.

Proposition 5. *For a sufficiently large $\mathscr{W}$, the procedure* ExpandNodesTour *ensures that, for every node $u \in \mathscr{N}_t$, assumptions (i) and (ii) of Proposition 4 are satisfied.*

The minimum width required for Proposition 5 is $\mathscr{O}(2^n)$, since it requires to enumerate all subsets of $\mathscr{V} \setminus \{0\}$. Notice that it may be significantly larger than the pseudo-polynomial size for the capacity constraints in the m-PDTSP case. The choice of $\hat{a}$ and u in Algorithm 3 can also be done in a systematic way, as discussed by Cire and van Hoeve (2013).

5.3. Filtering

Given a relaxed MDD $\mathscr{M}$, filtering consists of identifying and removing arcs a for which all paths traversing a only encode infeasible tours. Andersen et al. (2007) presents a number of filtering conditions for relaxed MDDs representing Hamiltonian paths that can be directly applied to our case. Here, we describe a new simple necessary condition that removes tours in $\mathrm{Sol}(\mathscr{M})$ violating the m-PDTSP capacity constraints.

With each node $u \in \mathscr{N}$, we introduce new labels $Q_{\uparrow}^{\min}(u), Q_{\uparrow}^{\max}(u) \in \mathbb{R}$ that are symmetric versions of $Q^{\min}(u)$ and $Q^{\max}(u)$. Namely, the labels $Q_{\uparrow}^{\min}(u)$ and $Q_{\uparrow}^{\max}(u)$ represent the minimum and maximum accumulated net weights, respectively, of all partial tours starting at u and ending at the terminal node $\mathbf{t}$. Thus, analogously as in the previous case,

$$Q_{\uparrow}^{\min}(u) = \min_{a \in \delta^{out}(u)} \left\{ Q_{\uparrow}^{\min}(h(a)) + \Delta q_{\theta(a)} \right\} \text{ and } \tag{18}$$

$$Q_{\uparrow}^{\max}(u) = \max_{a \in \delta^{out}(u)} \left\{ Q_{\uparrow}^{\max}(h(a)) + \Delta q_{\theta(a)} \right\}. \tag{19}$$

PROPOSITION 6. *The arc $a = (u, v)$ can be removed from $\mathscr{M}$ if:*

$$Q^{\min}(u) + \Delta q_{\theta(a)} + Q_{\uparrow}^{\min}(v) > C, \quad \text{or} \quad Q^{\max}(u) + \Delta q_{\theta(a)} + Q_{\uparrow}^{\max}(v) < 0. \tag{20}$$

Proof. By the definition of each label, it follows that any tour $\pi \in \mathrm{Sol}(\mathscr{M})$ from a path which includes a satisfies

$$Q^{\min}(u) + \Delta q_{\theta(a)} + Q_{\uparrow}^{\min}(v) \le \sum_{t=0}^{n+1} \Delta q_{\pi_t} \le Q^{\max}(u) + \Delta q_{\theta(a)} + Q_{\uparrow}^{\max}(v),$$

and therefore π is infeasible if (20) are satisfied. ■

Other similar rules that combine information from both $\mathbf{r} - u$ and $u - \mathbf{t}$ are analogously defined for the tour constraints and costs, if an upper bound to the objective function is given. We refer to Bergman et al. (2016), Chapter 11, for a comprehensive list of filtering rules for general sequencing problems.

6. Overall Solution Approach

Our complete solution approach to the m-PDTSP uses the Lagrangian dual $\mathscr{D}$ as a bounding mechanism in a branch-and-bound procedure. The approach exploits the graphical structure of $\mathscr{M}$ to branch in sequential order with respect to the layers in $\mathscr{M}$.

We first build a relaxed MDD of maximum width $\mathscr{W}$ using the construction procedure described in Section 5. The Lagrangian dual is then solved to optimality (Section 4). We then perform a depth-first search by branching on the π variables in this sequence. Each branching decision fixes to either $\pi_t = i$ or $\pi_t \neq i$, which is equivalent to a binary branching over y. Given a variable π_t to branch on, we choose the location i that is part of a shortest $\mathbf{r} - \mathbf{t}$ path (10) (ties are broken according to a lexicographic order).

When fixing $\pi_t = i$, we update $\mathscr{M}$ by removing infeasible arcs, re-applying the expansion method, and recomputing the Lagrangian dual objective function using the optimal multipliers from the root node (i.e., we never resolve $\mathscr{D}$ to optimality). This provides us a new lower bound that is used to prune nodes according to the best feasible solution found during search. We do not consider any additional primal heuristics.

We implement two variants of this methodology. The first, denoted by $\mathscr{M}^{\mathscr{C}}$, builds a relaxed MDD using ExpandNodesCapacity (Section 5.1) first. If the capacity constraints can be represented exactly with a smaller width (e.g., when $\mathscr{W} > C+1$), we apply the ExpandNodesTour until the maximum width is met. The second implementation, denoted by $\mathscr{M}^{\mathscr{T}}$, inverts the order of the expansion procedures, i.e., ExpandNodesTour is performed before ExpandNodesCapacity. We will investigate the impact of dualizing the different inequalities (2)-(5) over $\mathscr{M}^{\mathscr{T}}$ and $\mathscr{M}^{\mathscr{C}}$.

7. Numerical Study

This section describes the experimental details and results of our comparison of the proposed Lagrangian approach with current state-of-the-art methods. We use the benchmark of 1,178 instances developed by Hernández-Pérez and Salazar-González (2009), divided into three classes. Class 1 is a set of modified SOP problems from Ascheuer et al. (2000), where each precedence relation in the original instance is associated with a commodity. The class is divided into two groups that differ on the commodity weights: *max1* for unitary weights (i.e., $w_k = 1$ for all $k \in \mathscr{K}$), and *max5* for discrete weights up to 5 units (i.e., $w_k \in \{1, ..., 5\}$ for all $k \in \mathscr{K}$). Classes 2 and 3 are generated by placing locations on a grid uniformly at random and considering the Euclidean distance between locations as traveling costs. The weights in these two classes are also generated uniformly at random from the set $\{1, \ldots, 5\}$. Instances of Class 2 have no restriction on the number of commodities that each location can supply or demand. Instances of Class 3 have $m = n/2$ commodities and each location is either a pickup or a delivery spot for exactly one commodity.

7.1. Experimental Set-up

All experiments use $\mathscr{W} = 1{,}024$ and solve the Lagrangian dual $\mathscr{D}$ using the proximal bundle method implemented by Frangioni (2002) in C++, kindly provided by the author, using a specialized single-thread quadratic programming solver. The MDD construction was implemented within the constraint solver ILOG CP Optimizer 12.7 (IBM 2017), which was used only for the purpose of handling the depth-first search bookkeeping, i.e., we disabled all constraint propagation and additional features of the solver. [1]

[1] The code is available as an Online Supplement Material and can also be found in the authors personal webpages.

For comparison purposes, we also present the results of using ILOG CP Optimizer 12.7 modeling language and its complete set of features, except that computation was limited to a single core (parameter `Workers` = 1). The constraint programming (CP) model applied is presented in the Online Supplemental Material, referred to as $\mathscr{CP}$.

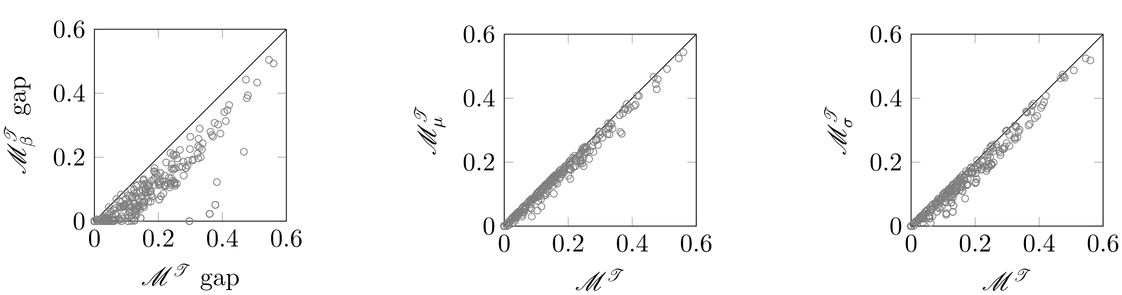
The experiments were run on an Intel(R) Xeon(R) CPU E5-2640 v3 at 2.60GHz and 8 GB RAM considering a time limit of 2 hours (7,200 seconds). The MDD-related times always account for both the Lagrangian dual solution times and the actual search time.

7.2. Experimental Results

Relaxation Analysis. We first investigate incorporating different subsets of inequalities from $\mathscr{P}$. As introduced in Section 4.3, we use β, μ, and σ to represent the Lagrange multipliers related to the tour (2), capacity (3)-(4), and precedence (5) constraints, respectively; e.g., $\mathscr{M}_\beta$ corresponds to relaxing inequality (2). We note that, in all cases, solving the Lagrangian dual takes less than 2 minutes. The experiments here only consider the tour-based MDD, $\mathscr{M}^\mathscr{T}$, as similar results were obtained with the capacity-based MDD.

We start analyzing the bound quality at the root node, i.e., the solution to the Lagrangian dual problem. To this end, we compute the optimality gap for each relaxation as $gap = (opt - LB)/opt$, where LB is the lower bound and opt the optimal value. We then compare the gap obtained by each Lagrangian relaxation with the gap produced by $\mathscr{M}^\mathscr{T}$ with no Lagrange multipliers.

Figures 4a, 4b and 4c show the gap improvement when constraints (2), (3)-(4), and (5) are considered in the Lagrangian dual problem, respectively. For each graph, a point represents an instance, its x-coordinate the gap computed by $\mathscr{M}^\mathscr{T}$, and the y-coordinate the gap obtained by solving the Lagrangian dual problem. Points below the diagonal are instances where the Lagrangian gap is smaller.



(a) Tour multipliers (b) Capacity multipliers (c) Precedence multipliers

Figure 4 Optimality gap comparison at the root node

The graphs show that relaxations based on the tour constraint obtain the greatest optimality gap reduction. Moreover, this result directly correlates with a significant increase in the number of instances solved (Table 1). In contrast, $\mathscr{M}^{\mathscr{T}}_{\mu}$ and $\mathscr{M}^{\mathscr{T}}_{\sigma}$ slightly improve the bound quality and solve no more than a couple of additional instances.

Table 1 Number of instances solved for different Lagrangian relaxations

Approach	$\mathscr{M}^{\mathscr{T}}$	$\mathscr{M}^{\mathscr{T}}_{\beta}$	$\mathscr{M}^{\mathscr{T}}_{\mu}$	$\mathscr{M}^{\mathscr{T}}_{\sigma}$	$\mathscr{M}^{\mathscr{T}}_{\beta+\mu}$	$\mathscr{M}^{\mathscr{T}}_{\beta+\sigma}$
Instances solved	503	**518**	505	504	**518**	**518**

Other combinations of multipliers were tested and have not shown substantial improvements. For instance, relaxations including tour-based multipliers with any of the two others had only a small optimality gap reduction (Table 1).

Comparison with state of the art. We now compare with previous methodologies in the literature. Detailed results for all instances are included in the Online Supplement.

We consider the two best-performing MDD variants: the capacity-based MDD $\mathscr{M}^{\mathscr{C}}_{\beta}$ and the tour-based MDD variant $\mathscr{M}^{\mathscr{T}}_{\beta}$, both which are strengthened using the linear inequalities (2) within our Lagrangian dual. These two versions are compared to the CP model $\mathscr{CP}$, to the Benders decomposition $\mathscr{BE}$ by Hernández-Pérez and Salazar-González (2009), and to the branch-and-cut algorithm by Gouveia and Ruthmair (2015), $CUTR^*$, which we will refer to as $\mathscr{CU}$. Due to the lack of results presented in previous papers, we restrict our comparison to a subset of 527 feasible instances. Notice that the results for $\mathscr{BE}$ were obtained using CPLEX 10.2 and a personal computer with Intel Pentium 3.0 Ghz, while the $\mathscr{CU}$ results used CPLEX 12.6 with an an Intel Xeon E5540 machine with 2.53 GHz.

To evaluate the robustness of the methodologies, Table 2 presents a summary of the number of instances solved for each class. The table includes the results for $\mathscr{M}^{\mathscr{T}}$ and $\mathscr{M}^{\mathscr{C}}$ (i.e., both construction procedures without Lagrange multipliers) to illustrate the performance of a pure discrete relaxation approach. Both MDD-based Lagrangian techniques solve the same instances as the MILP-based approaches, in addition to several open instances in the literature. $\mathscr{M}^{\mathscr{C}}_{\beta}$ closes 27 open instances, while $\mathscr{M}^{\mathscr{T}}_{\beta}$ closes 26. If we consider the total number of instances solved by $\mathscr{M}^{\mathscr{C}}_{\beta}$ and $\mathscr{M}^{\mathscr{T}}_{\beta}$ together, we were able to prove optimality for 33 open instances for the first time. $\mathscr{CP}$ has the weakest performance solving only 387 instances to optimality. However, it should be noted that $\mathscr{CP}$ is the most basic approach

Table 2 **Total number of instances solved per class**

	# instances	$\mathscr{BE}$	$\mathscr{CU}$	$\mathscr{CP}$	$\mathscr{M}^{\mathscr{T}}$	$\mathscr{M}^{\mathscr{C}}$	$\mathscr{M}^{\mathscr{T}}_{\beta}$	$\mathscr{M}^{\mathscr{C}}_{\beta}$	$\mathscr{M}^{\mathscr{T}}_{\beta}$ & $\mathscr{M}^{\mathscr{C}}_{\beta}$
Class 1	36	24	26	18	22	22	**35**	30	35
Class 2	341	315	330	280	**341**	**341**	**341**	**341**	341
Class 3	150	134	136	89	140	137	142	**148**	149
Total	527	473	492	387	503	500	518	**519**	**525**

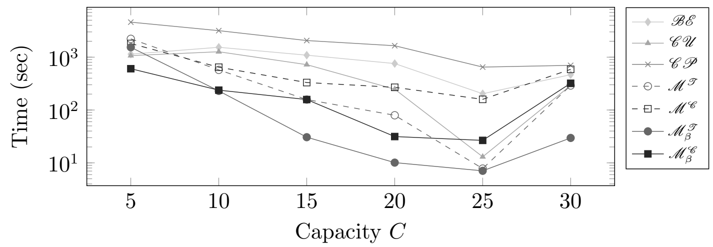


Figure 5 **Average run time comparison**

tested, an "out-of-the-box" model without the substantial reformulation and algorithmic effort that has gone into the other models.

Figure 5 compares the average run times for all techniques. The instances are divided according to the capacity restriction C as follows: $\{C \leq 5, C = 10, C = 15, C = 20, C = 25, C \geq 30\}$. $\mathscr{M}^{\mathscr{C}}_{\beta}$ has the lowest average run time when the capacity is small, but $\mathscr{M}^{\mathscr{T}}_{\beta}$ is stronger on instances with looser capacity restrictions.

8. Conclusions

We presented a novel approach to tackle the m-PDTSP, a challenging problem from the vehicle routing literature. The approach considers a discrete relaxation, encoded as a relaxed MDD, to better represent the combinatorial structure of the problem. We used Lagrangian duality to combine the discrete relaxation with a linear representation of the problem. Overall, the technique closes 33 instances in the literature, whereas our best implementation closes 27 of those instances.

The work emphasizes the value of exploiting a discrete relaxation for problems with a complex combinatorial structure, such as the m-PDTSP, alongside valid linear relaxations. This extends the use of MDDs to solve sequencing problems with capacity restrictions by presenting new construction and filtering strategies. Possible extensions of this work include single-commodity and time windows pickup-and-delivery problems, which can be naturally incorporated into this framework.

Acknowledgments

We would like to thank the anonymous reviewers whose valuable feedback helped improve the paper. The authors gratefully acknowledge funding from the Natural Sciences and Engineering Research Council of Canada (NSERC) and CONICYT (Becas Chile).

References

Andersen HR, Hadzic T, Hooker JN, Tiedemann P (2007) A constraint store based on multivalued decision diagrams. Bessière C, ed., *Principles and Practice of Constraint Programming–CP 2007*, 118–132 (Berlin, Heidelberg: Springer).

Artigues C (2017) On the strength of time-indexed formulations for the resource-constrained project scheduling problem. *Operations Research Letters* 45(2):154 – 159.

Ascheuer N, Jünger M, Reinelt G (2000) A branch & cut algorithm for the asymmetric traveling salesman problem with precedence constraints. *Computational Optimization and Applications* 17(1):61–84.

Balas E, Fischetti M, Pulleyblank WR (1995) The precedence-constrained asymmetric traveling salesman polytope. *Mathematical Programming* 68(1):241–265.

Baldacci R, Mingozzi A, Roberti R (2012) New state-space relaxations for solving the traveling salesman problem with time windows. *INFORMS Journal on Computing* 24(3):356–371.

Beasley J, Cao B (1998) A dynamic programming based algorithm for the crew scheduling problem. *Computers & Operations Research* 25(78):567 – 582.

Beasley JE, Christofides N (1989) An algorithm for the resource constrained shortest path problem. *Networks* 19(4):379–394.

Bergman D, Cire AA, van Hoeve WJ (2015) Lagrangian bounds from decision diagrams. *Constraints* 20(3):346–361.

Bergman D, Cire AA, van Hoeve WJ, Hooker J (2016) *Decision Diagrams for Optimization* (Cham: Springer International Publishing).

Bryant RE (1992) Symbolic boolean manipulation with ordered binary decision diagrams. *ACM Computing Surveys* 24:293– 318.

Carlyle WM, Royset JO, Kevin Wood R (2008) Lagrangian relaxation and enumeration for solving constrained shortest-path problems. *Networks: An International Journal* 52(4):256–270.

Christofides N, Mingozzi A, Toth P (1981) State-space relaxation procedures for the computation of bounds to routing problems. *Networks* 11(2):145–164.

Cire AA, van Hoeve WJ (2013) Multivalued decision diagrams for sequencing problems. *Operations Research* 61(6):1411–1428.

Coltin B, Veloso M (2014) Online pickup and delivery planning with transfers for mobile robots. Parker L, ed., *IEEE International Conference on Robotics and Automation*, 5786–5791 (Hong Kong, China: IEEE).

Conforti M, Cornujols G, Zambelli G (2014) *Integer Programming* (Cham: Springer International Publishing).

Cordeau JF, Laporte G (2007) The dial-a-ride problem: models and algorithms. *Ann.Oper.Res.* 153(1):29–46.

Cordeau JF, Laporte G, Potvin JY, Savelsbergh MW (2007) Transportation on demand. Barnhart C, Laporte G, eds., *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, 429 – 466 (Amsterdam, The Netherlands: Elsevier).

Dash S, Günlük O, Lodi A, Tramontani A (2012) A time bucket formulation for the traveling salesman problem with time windows. *INFORMS Journal on Computing* 24(1):132–147.

Dumitrescu I, Ropke S, Cordeau JF, Laporte G (2008) The traveling salesman problem with pickup and delivery: polyhedral results and a branch-and-cut algorithm. *Mathematical Programming* 121(2):269.

Fischetti M, Toth P (1989) An additive bounding procedure for combinatorial optimization problems. *Operations Research* 37(2):319–328.

Fisher ML (2004) The lagrangian relaxation method for solving integer programming problems. *Management science* 50(12):1861–1871.

Frangioni A (2002) Generalized bundle methods. *SIAM Journal on Optimization* 13(1):117–156.

Geoffrion AM (1974) *Lagrangean relaxation for integer programming*, 82–114 (Berlin, Heidelberg: Springer).

Gouveia L, Pesneau P (2006) On extended formulations for the precedence constrained asymmetric traveling salesman problem. *Networks* 48(2):77–89.

Gouveia L, Ruthmair M (2015) Load-dependent and precedence-based models for pickup and delivery problems. *Computers & Operations Research* 63(1):56–71.

Held M, Karp RM (1971) The traveling-salesman problem and minimum spanning trees: Part ii. *Mathematical programming* 1(1):6–25.

Hernádvölgyi IT (2004) Solving the sequential ordering problem with automatically generated lower bounds. Ahr D, Fahrion R, Oswald M, Reinelt G, eds., *Operations Research Proceedings 2003*, 355–362 (Berlin, Heidelberg: Springer).

Hernández-Pérez H, Salazar-González JJ (2004) A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. *Discrete Applied Mathematics* 145(1):126–139.

Hernández-Pérez H, Salazar-González JJ (2009) The multi-commodity one-to-one pickup-and-delivery traveling salesman problem. *European Journal of Operational Research* 196(3):987–995.

Hoda S, Van Hoeve WJ, Hooker JN (2010) A systematic approach to MDD-based constraint programming. Cohen D, ed., *Principles and Practice of Constraint Programming–CP 2010*, 266–280 (Berlin, Heidelberg: Springer).

Holland C, Levis J, Nuggehalli R, Santilli B, Winters J (2017) UPS optimizes delivery routes. *Interfaces* 47(1):8 – 23.

IBM (2017) Ilog cplex studio 12.7 manual. URL https://www.nature.com/nature/.

Kinable J, Cire AA, van Hoeve WJ (2017) Hybrid optimization methods for time-dependent sequencing problems. *European Journal of Operational Research* 259(3):887 – 897.

Lemaréchal C (1975) *An extension of davidon methods to non differentiable problems*, 95–109 (Berlin, Heidelberg: Springer Berlin Heidelberg).

Letchford AN, Salazar-González JJ (2016) Stronger multi-commodity flow formulations of the (capacitated) sequential ordering problem. *European Journal of Operational Research* 251(1):74–84.

Liu C, Aleman DM, Beck JC (2018) Modelling and solving the senior transportation problem. van Hoeve WJ, ed., *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, 412–428 (Cham: Springer).

Liu R, Xie X, Augusto V, Rodriguez C (2013) Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research* 230(3):475 – 486.

Parragh SN, Doerner KF, Hartl RF (2008) A survey on pickup and delivery problems. *Journal für Betriebswirtschaft* 58(1):21–51.

Rodríguez-Martín I, Salazar-González JJ (2012) A hybrid heuristic approach for the multi-commodity one-to-one pickup-and-delivery traveling salesman problem. *Journal of Heuristics* 18(6):849–867.

Savelsbergh MW, Sol M (1995) The general pickup and delivery problem. *Transportation science* 29(1):17–29.