An Overview of Nested Decomposition for Multi-Level Optimization Problems

Stephen J. Maher[1] and İbrahim Muter[2]

[1]College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, United Kingdom
[2]Amazon Web Services, re:Invent Building, Seattle, WA 98121, USA

Abstract

Nested multi-level structures are frequently encountered in many real-world optimization problems. Decomposition techniques are a commonly applied approach used to handle nested multi-level structures; however, the typical problem-specific focus of such techniques has led to numerous specialized formulations and solution methods. This lack of generalized results for nested multi-level optimization problems is addressed in this paper with the proposal of a theoretical framework for their formulation and the application of decomposition techniques. The developed theoretical framework will be used to highlight the prevalence of general multi-level structures within a wide range of application areas. Further, state-of-the-art solution methods for nested multi-level optimization problems will be described in the context of the proposed framework. The discussion in this paper will highlight the broad applicability of the general formulation and solution methodologies developed for this important class of real-world optimization problems.
Key words: two-level decomposition; multi-level decomposition; column generation; branch-and-price; integer programming.

1 Introduction

Structure, either mathematical or physical, is a common characteristic of large-scale optimization problems. This is due in part to the desire of optimization experts to introduce structure into mathematical models of large-scale systems, but also due to real-world problems typically exhibiting a physical relationship between the components of the system being modeled. Classical examples of structure in large-scale optimization problems are the routes in vehicle routing problems (VRP) that are independent for each vehicle but are linked for planning at the depot level or the independence of the production schedules for each generator in the unit commitment problem that must satisfy the demand at a network level. Aggregation is a common feature of such problems where the individual base units are combined to form an operational structure. The modeling approaches and solution methodologies for handling such aggregation are the primary concerns of this paper.

Consider a mathematical optimization problem given by

$$\begin{aligned} \text{minimize} \quad & cx, && \text{(1a)} \\ \text{subject to} \quad & \hat{A}x \geq \hat{b}, && \text{(1b)} \\ & x \in \{0,1\}^n. && \text{(1c)} \end{aligned}$$

We consider the case of problem (1) where c is positive and the constraint matrix $\hat{A}$ exhibits some structure. Further, we assume that this structure can be exploited through the application of decomposition techniques, such as Dantzig-Wolfe reformulation, Benders' decomposition or Lagrangian relaxation. While problems can exhibit a structure where many different decomposition techniques can be applied, we will focus primarily on problem structures that are amenable to the application of Dantzig-Wolfe reformulation.

Formally, the structure that we wish to exploit is the following: Consider n sets denoted by I^i for $i \in \{1, 2, \dots, n\}$. We describe the set I^1 as the first level, I^2 the second level and so on. The elements in these sets are binary vectors of length $|I^{i-1}|$. The basic units that are aggregated in levels I^i for $i > 1$ are contained in I^1. The sets can be defined as $I^i \subseteq \{x \,|\, \mathcal{C}^i(x) = 1,\, x \neq \mathbf{0},\, x \in \{0,1\}^{|I^{i-1}|}\}$ for $i \in \{2, 3, \dots, n\}$, where $\mathcal{C}^i(x) = 1$ indicates that x satisfies all the conditions defined by $\mathcal{C}^i$ in level i. In the mathematical programming context $\mathcal{C}^i$ can be a set of inequalities. For $x \in I^i, i > 1$, $x_j = 1$ if the element indexed by j in I^{i-1} is included in the aggregation given by x. This leads to the definition of a nested multi-level optimization problem:

Definition 1 *A* nested multi-level optimization problem *is a problem in which for any $N \in \{2, \dots, n\}$ there exists a formulation where each level $i < N$ is a subproblem for level $i+1$. The goal of this problem is to find a selection of elements $\hat{I} \subseteq I^N$ that minimizes $\sum_{k \in \hat{I}} c_k x_k$.*

The nested multi-level structure is exemplified in Figure 1. This is an example of a feasible solution of a multi-level optimization problem. The first level, I^1 are the base elements of the optimization problem, such as customer locations in the VRP or flights in the airline crew pairing problem. In the second level, I^2, these base elements are aggregated to form an ordered set satisfying the conditions $\mathcal{C}^2$. These aggregations could be vehicle routes visiting multiple customers or crew duties comprised of a sequence of flights. The second level elements are then aggregated further in the third level, I^3.

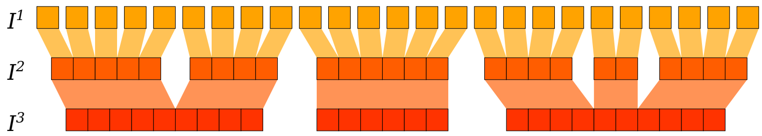

Figure 1: In a feasible solution, the levels can be represented as aggregation of elements from the preceding level.

This final aggregation could represent multiple routes in the multi-trip VRP or crew pairings—made up of multiple duties—in the crew pairing problem. Without loss of generality it is assumed that in an optimal solution of a nested multi-level problem each element from level i is contained in at most one element in level $i + 1$.

In practical examples of multi-level optimization problems, N is commonly at most 3. Problems where $N = 2$ are classical problems where Dantzig-Wolfe reformulation is typically applied, which includes bin packing, cutting stock and vertex coloring problems. A more complex nested relationship between the levels is modeled in problems where $N = 3$, which is observed in the multi-trip vehicle routing problem and integrated order batching and scheduling problems. There are very few problems that natively have a nested structure with more than three levels. One classical example is the multi-stage cutting stock problem; however, formulations exist to restrict the number of levels to three.

This paper presents nested multi-level optimization problems as an important class of problems that arises in many application domains. While there is a broad range of application types, there has been little effort to unify the advancements in solving these problems across domains. The goal of this paper is to pose the nested multi-level optimization problem as a fundamental problem class and to highlight the key formulations and solution methods that have been developed. In achieving this goal, the contributions of this paper are: We (a) present the theoretical basis of the nested multi-level optimization problem. This theoretical basis is used to (b) develop three primary problem formulations for multi-level optimization problems. Decomposition plays an important role in the solution of nested multi-level optimization problems. Thus, this paper will primarily focus on the application of decomposition methods and the associated challenges in solving these problems to integer optimality. We (c) highlight key applications of multi-level optimization problems arising from the vehicle routing, scheduling, staff rostering and cutting stock domains. (d) An overview of the most effective solution methods drawn from successful applications is presented. Finally, we (e) demonstrate that decomposition and solution techniques can be drawn for various applications domains to drive advancement in the solution of real-world multi-level optimization problems.

This paper is organized as follows: Section 2 presents the theoretical basis of multi-level optimization problems proposed in this paper and a generic approach to modeling such problems. Importantly, various methods for decomposing nested multi-level optimization problems are presented in Section 2. The multi-trip vehicle routing problem (MTVRP) and multi-stage cutting stock problem (MSCS) will be used as examples in Section 2.5 to illustrate the key aspects of the modeling and decomposition of nested multi-level optimization problems. In Section 3, we review the literature on problems that are amenable to multi-level decomposition. Various solution methodologies used to solve the practical applications described in Section 3 will be analyzed and discussed in Section 4. Section 5 will present some final remarks and directions for future research.

2 Two-level Decomposable Optimization Problem

In this section, and throughout this paper we will focus on a three-level nested optimization problem. First, to guide the discussion, we formally define an $N-$*level decomposable optimization problem* as

Definition 2 *An N-*level decomposable optimization problem *is a multi-level optimization problem with at least $N+1$ levels, and this multi-level structure can be exploited by the application of decomposition methods.*

By Definition 2, for a multi-level optimization problem with $n+1$ levels, it is possible to apply decomposition methods to exploit $1, 2, \ldots, n$ levels. The resulting decompositions are described as a one-level, two-level, ..., $n-$level decomposition respectively.

2.1 Exploiting problem structure

The corresponding form of (1) exhibiting the structure amenable to the application of Dantzig-Wolfe reformulation is

$$\begin{aligned} \text{minimize} \quad & cx, && \text{(2a)} \\ \text{subject to} \quad & Ax \geq b, && \text{(2b)} \\ & Bx \leq d, && \text{(2c)} \\ & Ex \leq f, && \text{(2d)} \\ & x \in \{0,1\}^n. && \text{(2e)} \end{aligned}$$

The matrix A is referred to as the *Linking* or *Master Block* and the matrices B and E are commonly referred to as subsystem blocks. It is common for B and E to exhibit a block-diagonal structure, where each system is disjoint and linked through the constraints given by A. However, in a multi-level optimization problem B may also provide a linking between the disjoint blocks of E. The application of Dantzig-Wolfe reformulation results in a master problem consisting of the linking constraints and corresponding variables. The subproblems comprise the variables and constraints of each disjoint block. We will show in this section that the application of Dantzig-Wolfe reformulation results in different formulations based upon the selection of constraints that comprise the master and subproblems.

An important feature of Dantzig-Wolfe reformulation is that the variables in the master problem are a reformulation of the original variables. Problem (2) represents a problem with multiple levels, where $\mathcal{C}^2(x) := Bx \leq d$ and $\mathcal{C}^3(x) := Ex \leq f$. Consider the feasible region given by $I^2 = \{x \in \{0,1\}^n | Bx \leq d\}$, which we assume is a bounded polytope. All feasible solutions in I^2 can be written as an integer combination of points $\{\mathbf{x}^q\}_{q \in Q}$, which is given by

$$x = \sum_{q \in Q} \mathbf{x}^q \lambda_q, \quad \sum_{q \in Q} \lambda_q = 1, \quad \lambda \in \mathbb{Z}_+^{|Q|}, \tag{3}$$

where Q is a finite set. The variables λ_q are implicitly restricted to binary variables and indicate whether the column given by the feasible solution $\mathbf{x}^q$ is selected. Upon substituting (3) into (2), this reformulation provides a separation of the master and subproblems by exploiting the structure exhibited by $\hat{A}$ and subsequently B. This type of reformulation is described as the discretization approach (Vanderbeck 2000). The ability to form the set I^2, which comprises the constraints $Bx \leq d$, is the basis for the resulting reformulation. It is also possible to define $I^{2'} = \{x \in \{0,1\}^n_+ | Bx \leq d, Ex \leq f\}$ and perform a similar reformulation. Using I^2 or $I^{2'}$ to perform the reformulation of (1) results in a different structure of the master and subproblems. Exploiting these different reformulation possibilities is the main driver for the solution approaches presented in this paper.

One may notice that the variable reformulation given by (3) replaces a set of constraints in the master problem with exponentially more variables than in the original problem. The motivation for performing such a reformulation is that the discretization of the feasible region I^2 potentially results in an improved bound on the LP relaxation of (1). A prominent method for handling this significant increase in variables is to solve the master problem using a column generation algorithm. Briefly, a column generation algorithm generates the necessary columns required in the optimal solution of a linear programming (LP) problem by solving a pricing subproblem.

Dantzig-Wolfe formulation and column generation are very popular mathematical programming techniques that have been used to solve many problems arising from practical applications. For more details regarding Dantzig-Wolfe reformulation and column generation, we refer to Desaulniers et al. (2005) and Lübbecke and Desrosiers (2005). Our interest in nested multi-level problems comes from the observation that numerous problems from different application domains are amenable to Dantzig-Wolfe reformulation in more than one level. From the literature it is evident that care must be taken when modeling mathematical programs with nested multi-level structures, particularly when considering the solution methodology. This paper will provide guidance and insight into different formulations and the available solution methodologies for nested multi-level decomposable problems.

2.2 First-level Decomposition

Typically there is no one unique method for applying Dantzig-Wolfe reformulation to problem (2). While different reformulations are possible, the effectiveness of related solution algorithms is affected by the reformulation that is applied. In the above section, a reformulation of the original problem variables is achieved using the feasible solutions contained in I^2. In this case, the binary operator $\mathcal{C}^2(x) = 1$ is equivalent to the system of inequalities $Bx \leq d$. While for such a problem type this is true in general, it is expected that such a reformulation will be most effective if $Bx \leq d$ induces a well-structured set that lends itself to efficient optimization over the polyhedron, such as network flow or knapsack problems.

We define I^2 as above and apply the reformulation given by (3). After the substitutions $c_q = c\mathbf{x}^q$, $e_q = E\mathbf{x}^q$ and $a_q = A\mathbf{x}^q$ for $q \in Q$, the first-level master problem, which we will refer to as MP1, can be written as

$$\begin{aligned} \textbf{(MP1)} \quad \text{minimize} \quad & \sum_{q \in Q} c_q \lambda_q, && \text{(4a)} \\ \text{subject to} \quad & \sum_{q \in Q} a_q \lambda_q \geq b, && \text{(4b)} \\ & \sum_{q \in Q} e_q \lambda_q \leq f, && \text{(4c)} \\ & \sum_{q \in Q} \lambda_q = 1, && \text{(4d)} \\ & \lambda_q \in \mathbb{Z}_+, \quad q \in Q. && \text{(4e)} \end{aligned}$$

Constraints (4b) and (4c) correspond to the constraint sets (2b) and (2d) respectively. These constraints have been reformulated as a result of applying a Dantzig-Wolfe reformulation. The constraint (4d) is the convexity constraint that assures exactly one column satisfying (4b) and (4c) is selected. In many applications of decomposition, this constraint is omitted from the model without affecting the optimal solution.

2.3 Second-level Decomposition

Suppose that $\sum_{q \in Q} e_q \lambda_q \leq f$ also has a structure that stimulates decomposition for the second time. This further decomposition will be referred to as the second-level decomposition. The solution set that satisfies the constraints (4c) and (4e) can be represented as $I^3 = \{\lambda \in \mathbb{Z}^+ | \sum_{q \in Q} e_q \lambda_q \leq f\}$. Once again, we assume that the polytope I^3 is bounded. The index set of all feasible solutions contained in I^3 is denoted by S and $\lambda^s = [\lambda_1^s \ \lambda_2^s \ .. \ \lambda_{|Q|}^s]^T$ denotes the associated solutions. Let the variable y_s equal 1 to indicate whether the associated solution λ^s is selected, and 0 otherwise. After the substitutions $c_s = \sum_{q \in Q} c_q \lambda_q^s$, $a_s = \sum_{q \in Q} a_q \lambda_q^s$ for $s \in S$, the second-level master problem, which we will refer to as MP2, can be written as

$$\textbf{(MP2)} \quad \text{minimize} \quad \sum_{s \in S} c_s y_s, \tag{5a}$$

$$\text{subject to} \quad \sum_{s \in S} a_s y_s \geq b, \tag{5b}$$

$$\sum_{s \in S} y_s = 1, \tag{5c}$$

$$y_s \in \mathbb{Z}_+, \quad s \in S. \tag{5d}$$

Following the application of Dantzig-Wolfe reformulation to MP2, many similarities can be observed when comparing with MP1. In particular, the large cardinality of S makes solving MP2 using a general purpose solver impractical. Column generation is then employed to dynamically generate variables by first forming a restricted master problem by replacing S with $\bar{S} \subseteq S$. However, because a two-level decomposition has been performed, the classical column generation approaches may not be computationally effective. A discussion of variants to the classical column generation approach used to solve MP2 will be presented in Section 4.

As explained in the previous section, the first-level decomposition must perform an aggregation of I^1 into elements of I^2. Thus, the set I^3 would not be achieved in the first-level decomposition due to the hierarchy between the solution sets I^1 and I^2. Hence, it is not a matter of choice in the first level to apply decomposition on $Bx \leq b$ or $Ex \leq f$, i.e., the levels of the decomposition are not interchangeable. The illustrative examples in Section 2.5 will demonstrate the order dependency in the application of decomposition techniques on the original and master problems.

On the other hand, the constraint set (2d) can be included in the first-level decomposition by defining $I^{2'} = \{x \in \{0,1\}^n | Bx \leq d, Ex \leq f\}$. This approach simultaneously decomposes both constraints sets that were decomposed in the first- and second-level decomposition. Though this reformulation is not as prevalent as applying both levels of decomposition, it will pave the way for a specific solution method.

2.4 Combining First- and Second-level Decompositions

The descriptions of the first- and second-level decomposition presented in Sections 2.2 and 2.3 suggest that the level structure can be completely encapsulated within each preceding level. There is an alternative formulation of the multi-level problems that combines the variables sets from the first and second levels, namely λ_q, $q \in Q$ and y_s, $s \in S$. The integration of the first and second levels may deliver benefits to the solution methodology, which will be explained in Section 4.

This integration between the first and second levels can be achieved through a set of constraints that link λ and y. Such a model, which we will refer to as MP1-2, can be written as:

$$\textbf{(MP1-2)} \quad \text{minimize} \quad \sum_{s \in S} c_s y_s, \tag{6a}$$

$$\text{subject to} \quad \sum_{q \in Q} a_q \lambda_q \geq b, \tag{6b}$$

$$\sum_{s \in S} h_{qs} y_s \geq \lambda_q, \quad q \in Q, \tag{6c}$$

$$\sum_{s \in S} y_s = 1, \tag{6d}$$

$$\lambda_q \in \mathbb{Z}_+, \quad q \in Q, \tag{6e}$$

$$y_s \in \mathbb{Z}_+, \quad s \in S. \tag{6f}$$

Without loss of generality, we disregard the convexity constraints associated with the first-level decomposition, which is imposed on $\lambda-$variables, as (6c) serve the same purpose in most applications. The linking between the second and third levels of the nested structure is established by constraints (6c). The variables y_p (λ_q) equal 1 to indicate whether $s \in S$ ($q \in Q$) is selected in the solution of (6). In constraints (6c), the parameter h_{qs} indicates the number of times the element $q \in Q$, which corresponds to an element in I^2, is used in $s \in S$, an element in I^3. The interpretation of constraints (6c) is: Selecting element q indicates that at least one element s that is an aggregation of elements from I^2 including q must be selected. If q is not selected, there is no restriction on the selection of s; however, since c_s is positive, y_s will be set to zero.

The interesting feature of MP1-2 is that when applying column generation to the variable set Q, each additional column q introduces an additional constraint to (6c). Thus, this problem grows both column- and row-wise. This is a phenomenon that has been observed in problems with column-dependent rows, and MP1-2 has a structure that complies with the properties of this class of problems, for examples, see Muter et al. (2013) and Maher (2016). This structure and the column-and-row generation algorithm it entails will be discussed in Section 4.

As mentioned previously, an advantage of applying Dantzig-Wolfe reformulation is that it can result in an improved bound on the LP relaxation of (1). Since problem MP1-2 convexifies both the I^2 and I^3 feasible regions, there is no additional gain in the bound for the LP relaxation compared to MP2. Thus, the choice to apply the decomposition of the form MP2 or MP1-2 is dependent on the problem structure and the expectation of solution algorithm effectiveness.

2.5 Illustrative Examples

We exemplify the steps of the two-level decomposition and the resulting models using MTVRP and MSCS. The MTVRP is a variant of the classical vehicle routing problem where each vehicle is allowed to perform more than one route during the workday. In this problem, there exists a set of identical vehicles, indexed by $k \in K$, located at a single depot. Each vehicle k has a capacity D. The vehicles are used to satisfy the demands of a set of customers, indexed by $i \in I^1$. A vehicle k has a maximum driving time T, and by that time it should end its schedule at the depot.

The MTVRP can be represented on a complete directed graph $G = (N, A)$, where $N = \{0\} \cup I^1$ is the set of nodes comprising the depot indexed as 0 and the set of customers I^1. There exists a demand of size d_i associated with each customer $i \in I^1$. An arc $(i, j) \in A$ is defined between each pair of nodes with an associated cost c_{ij}, which corresponds to time. A vehicle route is given by a path formed by a set of arcs that originates from the depot, visits a set of nodes and returns back to the depot. Additionally, a route is only valid if the total demand of the customers to be visited does not exceed the capacity of the vehicle. The time required to complete all routes assigned to a vehicle must not exceed T. A set of routes R is associated with each vehicle $k \in K$, and the variables x_{ij}^{rk} are defined to equal 1 if arc (i, j) is traversed in route $r \in R$ by vehicle k. The objective of the MTVRP is to find a set of routes and its assignment to a number of vehicles such that the total travel time is minimized. The MTVRP can be formulated as

$$\text{minimize} \quad \sum_{k\in K}\sum_{r\in R}\sum_{(i,j)\in A} c_{ij}x_{ij}^{rk}, \tag{7a}$$

$$\text{subject to} \quad \sum_{k\in K}\sum_{r\in R}\sum_{j\in N} x_{ij}^{rk} = 1, \quad i \in I^1, \tag{7b}$$

$$\sum_{r\in R}\sum_{(i,j)\in A} c_{ij}x_{ij}^{rk} \le T, \quad k \in K, \tag{7c}$$

$$\sum_{j\in N} x_{ij}^{rk} = \sum_{j\in N} x_{ji}^{rk}, \quad i \in N, k \in K, r \in R, \tag{7d}$$

$$\sum_{i\in I^1}\sum_{j\in N} d_i x_{ji}^{rk} \le D, \quad k \in K, r \in R, \tag{7e}$$

$$\sum_{i\in \bar{I}}\sum_{j\in \bar{I}} x_{ij}^{rk} \le |\bar{I}| - 1, \quad k \in K, r \in R, \bar{I} \subset I^1, \tag{7f}$$

$$x_{ij}^{rk} \in \{0,1\}, \quad (i,j) \in A, k \in K, r \in R. \tag{7g}$$

The objective to minimize the time required to visit all customers is given by (7a). The constraints (7b) ensure that every customer is visited exactly once across the complete set of routes for all vehicles. The total duration of the routes assigned to vehicle k, which is computed by summing the transit time of the arcs assigned to that vehicle, is restricted to T by constraints (7c). Constraints (7d) are the flow conservation constraints and the capacity of each vehicle is imposed by constraints (7e). Finally, the subtour elimination constraints are given by (7f).

The MTVRP is an example of the nested multi-level problem class given by problem (2). The constraints (7b) correspond to the Master block constraints $Ax \ge b$. The constraint set $Bx \le d$ is equivalent to constraints (7d)–(7f) in the MTVRP. Finally, constraints (7c) is the counterpart of $Ex \le f$. Given this correspondence between (2) and (7), it is possible to perform the reformulations that are discussed in Section 2. After applying the first-level decomposition on the constraints (7d)-(7f), which forms the set of routes Q, the MP1 formulation becomes

$$\text{minimize} \quad \sum_{k\in K}\sum_{q\in Q} c_q\lambda_q^k, \tag{8a}$$

$$\text{subject to} \quad \sum_{k\in K}\sum_{q\in Q} a_{iq}\lambda_q^k = 1, \quad i \in I^1, \tag{8b}$$

$$\sum_{q\in Q} c_q\lambda_q^k \le T, \quad k \in K, \tag{8c}$$

$$\lambda_q^k \in \mathbb{Z}_+, \quad q \in Q, k \in K. \tag{8d}$$

The variable λ_q^k is equal to 1 if route q is performed by vehicle k, and 0, otherwise. Note that the MP1 formulation of (8) is not a direct application of Dantzig-Wolfe reformulation described in Section 2.1. This can be seen by the elimination of the r index and the omission of the convexity constraint. While it is possible to include these two aspects in the formulation, they are redundant.

An interesting feature of problem (8) is the knapsack structure in (8c), which can be further exploited through a second-level decomposition. A combination of vehicle routes satisfying this constraint set, which is separable for each vehicle k, constitutes a vehicle schedule. Observe that (7c) exhibited this block diagonal structure also in the original formulation. However, from the two-level decomposition perspective, it is impossible to represent solutions satisfying constraints associated with the schedules before the route set Q is identified. Hence, the decomposition of constraints (7c) is deferred to the second level of the decomposition. The second-level decomposition results in the MP2 formulation as follows:

$$\text{minimize} \quad \sum_{s\in S} c_s y_s, \tag{9a}$$

$$\text{subject to} \quad \sum_{s\in S} a_{is} y_s \geq 1, \quad i \in I^1, \tag{9b}$$

$$\sum_{s\in S} y_s \leq |K|, \tag{9c}$$

$$y_s \in \{0,1\}, \quad s \in S, \tag{9d}$$

where S is the set of schedules, which are the aggregation of routes $q \in Q$ subject to the constraints imposed in the nested structure. The binary variable y_s is defined to equal to 1 only if schedule s is selected. Since vehicles are identical and a vehicle is not required to perform a schedule, the convexity constraints take the form of (9c). This model could be achieved by defining $I^{2'}$ as the set of solutions satisfying (7c)-(7g) in the first level of decomposition applied to problem (7). However, as mentioned previously, even though applying two-level decomposition and defining $I^{2'}$ in a one level decomposition leads to the same formulation of the master problem, the solution methods and pricing subproblems will differ due to the number of levels and constraints involved in the decomposition process.

An alternative model can be obtained by combining the first- and second-level decomposition to form MP1-2, as explained in Section 2.4. Due to the resemblance between problem (6) and this alternative model for solving the MTVRP, we refrain from repetition, and explain the analogy as follows: Constraint set (6b) in MTVRP ensures that each customer $i \in I^1$ is visited in exactly one route. Binary parameter h_{qs}, as defined in MP1-2, equals 1 only if route $q \in Q$ is performed as part of schedule $s \in S$. Constraint set (6c) in this problem imposes that if a route $q \in Q$ is selected, it must be performed in at least one vehicle schedule. This formulation bears an algorithmic and computational burden in terms of the number of constraints in

(6c) when column generation is applied to two-level decomposable problems. Hence, MP1-2 is not a prevalent formulation in most application areas. An exception is the MSCS, which exemplifies the case where tackling MP1-2 can be preferable over MP2. The benefits of the MP1-2 formulation and its relationship with MP2 will be addressed and generalized based on the MSCS in Section 4. In the following, we present the problem description of the MSCS and its MP1-2 formulation.

Unlike the conventional cutting stock problem, MSCS arises when the stock roll cannot be cut directly into the finished rolls $i \in I^1$, each identified by width c_i. In this setting, a stock roll is first cut into intermediate rolls whose width is restricted to be within the interval $[t^{min}, t^{max}]$. These intermediate rolls are then cut into finished rolls with a certain width, which is based on demand. The first level decomposition forms the finished roll cutting pattern set I^2, which are cut from intermediate rolls $j \in J$. In the second level, intermediate roll cutting patterns that are cut from the stock roll are defined in set I^3. Adhering to the set notation and definitions given in the previous section, we define Q and S as the sets of finished roll and intermediate roll patterns. The parameters a_{iq} denote the number of times finished roll i is cut in finished roll pattern $q \in Q$. Similarly, the parameters h_{js} denote the number of times intermediate roll j is cut in intermediate roll pattern $s \in S$. The MP1-2 formulation of the MSCS is given by

$$\text{minimize} \quad \sum_{s \in S} y_s, \tag{10a}$$

$$\text{subject to} \quad \sum_{q \in Q} a_{iq}\lambda_q \geq b_i, \quad i \in I^1, \tag{10b}$$

$$\sum_{s \in S} h_{js} y_s \geq \sum_{q \in Q^j} \lambda_q, \quad j \in J, \tag{10c}$$

$$y_s \in \mathbb{Z}_+, \quad s \in S, \tag{10d}$$

$$\lambda_q \in \mathbb{Z}_+, \quad q \in Q. \tag{10e}$$

Constraints (10b) ensures that the demand on finished rolls is satisfied. Constraints (10c) imposes that the number of intermediate roll $j \in J$ produced through cutting stock rolls is at least the number of finished roll cutting patterns cut from this intermediate roll, where Q^j denotes the set of finished roll patterns that are cut from intermediate roll $j \in J$. We point out here that (10c) is not defined over $q \in Q$, as in constraint (6c) of MP1-2 for the MTVRP, but over a set of intermediate rolls J, each associated with a set of finished roll patterns in Q. Hence, there is a one-to-many relationship between Q and J as many finished roll cutting patterns can be cut from each intermediate roll width. The size of J is generally prohibitive for enumeration so that only a subset of this set, denoted by $\bar{J}$, is maintained during the course of column generation, prompting both a column- and row-wise increase in the size of the problem.

On the other hand, in the MP2 formulation, which is given by,

$$\text{minimize} \quad \sum_{s \in S} y_s, \tag{11a}$$

$$\text{subject to} \quad \sum_{s \in S} a_{is} y_s \geq b_i, \quad i \in I^1, \tag{11b}$$

$$y_s \in \mathbb{Z}_+, \quad s \in S, \tag{11c}$$

the intermediate rolls J do not explicitly exist in the master problem formulation. As such, it appears that the finished rolls are cut directly from the stock roll through the parameter a_{is}, which denotes the number of finished rolls cut in pattern $s \in S$. While not explicitly modeled, the generation of the intermediate rolls has been relegated to the pricing subproblem of MP2. Thus, their consideration is still necessary in the development solution algorithms to solve the MSCS.

3 Applications

The general modeling framework presented in Section 2 provides a platform for comparing multi-level optimization problems arising from different application domains. A discussion of various applications from the literature and how they fit within the proposed framework is presented in this section. An overview of the literature and application domains is presented in Table 1. Since a broad overview of application domains is more relevant for highlighting the prevalence of multi-level structures and the various decomposition approaches used, we refrain from presenting an exhaustive literature survey. For a detailed review of nested decomposition literature the reader is referred to Tilk et al. (2019).

We confine the scope in this section to the general aspects of the considered problems and modeling approaches. The related solution methods will be discussed in Section 4.

Table 1: List of references for nested programs

Problem	References	I^1	I^2	I^3	I^4
MTVRP	Mingozzi et al. (2013)	Customers	Routes	Schedules	
MTVRPTW	Azi et al. (2010), Macedo et al. (2011), Hernandez et al. (2014), Hernandez et al. (2016)	Customers	Routes	Schedules	
VRPIF	Akca (2010), Muter et al. (2014), Desaulniers et al. (2016)	Customers	Inter-Facility Route	Schedules	
PTVRP	Karabuk (2009)	Customers	Mini-routes	Routes	
VRP with multiple resource interdependencies	Tilk et al. (2019)	Customers	Paths	Route	
Crew Pairing	Desaulniers et al. (1997), Vance et al. (1997)	Flights	Duty	Pairing	Duty Period Set
Crew Pairing & Assignment	Zeighami and Soumis (2019)	Flights	Pairings	Rosters	
Staff Rostering	Dohn and Mason (2013)	Shifts	On-Stretch	Work-Stretch	Roster-Line
Routing & scheduling with split P&D	Hennig et al. (2012)	Locations	Routes	Pattern	
Parallel Batch Machine Scheduling (PBPM)	Muter (2020), Wang and Tang (2010)	Jobs	Batches	Schedules	
Order Batching and Picker Scheduling (OBPS)	Muter and Öncan (2021)	Orders	Order Batches	Schedules	
Multi-Dimension/Stage Cutting Stock	Song (2009), Vanderbeck (2001), Zak (2002) Muter and Sezer (2018), Valério de Carvalho and Rodrigues (1995)	Finished Parts	Intermediate Parts	Cutting Pattern	
Recursive Ring Packing	Gleixner et al. (2020)	Individual rings	Telescoping rings within rings	Packed rings in rectangles	

3.1 Vehicle Routing Problem Variants

The application of two-level decomposition to VRP variants typically yields a set of routes I^2 in the first level, which starts from and ends at the same or different locations as stipulated by $\mathcal{C}^1$, and a set of schedules in the second level, which consists of an aggregation of routes satisfying $\mathcal{C}^2$. Mingozzi et al. (2013) present the MP1 and MP2 formulations of MTVRP. They prove that the LP relaxation bound of MP2 is greater than or equal to that of MP1. In fact, the application of the second level decomposition to MP1 over the vehicle set attests to the superiority of the LP relaxation bound of MP2. An extension of this problem in which time feasibility constraints replace the driving time constraint is the MTVRP with time-windows (MTVRPTW). This extension first proposed by Azi et al. (2010) includes limits on the route duration, which keep the number of routes to a manageable size. The authors formulate the MP2 model in which the variable set corresponds to feasible vehicle schedules. Macedo et al. (2011) and Hernandez et al. (2014) formulate the MP1 model based on the pre-enumerated route set I^2. Alternatively, Hernandez et al. (2016) relax the route duration constraint, which proliferates the number of routes, and compare the algorithmic performance for solving corresponding MP1 and MP2 formulations.

The VRP with intermediate facilities (VRPIF) is an extension of MTVRP where intermediate facilities provide replenishment opportunities (Bard et al. (1998)). The VRPIF has several extensions that are considered from the two-level decomposition perspective. The multi-depot extension of VRPIF—referred to as the multi-depot VRP with inter-depot routes (MDVRPI)—is investigated by Crevier et al. (2007) where depots act as intermediate facilities. Crevier et al. (2007) model the MDVRPI with a formulation corresponding to MP1 in which the first-level decomposition forms the set of inter-depot routes as I^2. Muter et al. (2014) present an MP2 formulation of this problem where the vehicle schedules, which are combinations of the inter-depot routes satisfying a set of connectivity constraints, are defined as I^3.

A problem similar to the VRPIF in which electric vehicles are utilized is the electric VRP with time-windows (EVRPTW). A major difference between the VRPIF and EVRPTW is that in the latter two different capacities must be considered—the vehicle and battery capacity. The battery capacity is replenished at charging (intermediate) facilities; however, the vehicle capacity is not. Desaulniers et al. (2016) present a formulation for the EVRPTW corresponding to MP2. This formulation results from decomposing $I^{2'}$, which contains vehicle routes satisfying all time-windows as well as the battery and capacity constraints.

The integration of location decisions to the multi-depot version of MTVRP is called the integrated location, routing and scheduling problem (ILRSP). Akca (2010) present two different formulations—graph-based and set partitioning-based—that are both tantamount to MP2 where I^3 corresponds to the set of vehicle schedules as defined in MTVRP.

The paratransit VRP (PTVRP) is a multi-depot pickup and delivery problem with time windows and side constraints. A typical modeling approach for the PTVRP is to cluster compatible requests. An example using mini-clusters is presented by Ioachim et al. (1995). Karabuk (2009) present an MP2 formulation, where the mini-clusters and routes that are formed of these clusters are handled in the first- and second-level, respectively.

A VRP extension that involves the trade-off between cost, time and load is the VRP with multiple resource interdependencies. Tilk et al. (2019) present a variant of this problem with time windows, soft demand and synchronized visits constraints. An MP2 formulation is proposed that distinguishes between route set I^3 with a certain time schedule and distribution plan and path set I^2 that satisfies the time-window and capacity constraints.

A nested multi-level problem that involves routing and scheduling in logistics is the crude oil tanker routing and scheduling problem with split pickup and split delivery (Hennig et al. 2012). The modeling of this problem by Hennig et al. (2012) incorporates many real-life complexities, such as a heterogeneous fleet, multiple commodities, many-to-many relations for pickup and delivery of each commodity, sequence dependent vehicle capacities, and cargo quantity dependent pickup and delivery times. The problem is handled in two levels such that the master problem is analogous to MP2, where the first level decomposition handles the route set as I^2, the second-level decomposition is based on the cargo pattern set I^3 that are assigned to ships.

3.2 Nested Scheduling Problem Variants

The order batching and picker scheduling problem (OBPS)—one of the prominent problems in the warehouse order picking process—has a similar objective and constraint set to the MTVRP. However, the MTVRP and OBPS diverge from each other in the way the routes and batches, respectively, are constructed. Muter and Öncan (2021) present formulations for the OBPS in the form of MP1 and MP2 and devised a bounding procedure that exploits the relationship between them.

The parallel batch processing machine scheduling problem (PBPM) aims to group a set of jobs into batches, where each is processed simultaneously on one of the several machines operating in parallel. Wang and Tang (2010) present a formulation for the PBPM corresponding to MP1-2. Muter (2020) adopt the makespan objective on this problem and presents the corresponding MP2 formulation. The formulation proposed by Muter (2020) for the PBPM defines I^2 and I^3 as job batches and the machine schedules formed of batches, respectively. This definition of I^2 and I^3 is analogous to the MTVRP where the former corresponds to routes and the latter corresponds to schedules. Thus, many solution approaches are similar between the two problem types.

3.3 Staff Rostering and Scheduling

A large class of optimization problems that exhibit a multi-level structure is staff rostering. The multi-level structure comes from the rosters being an aggregation of shifts and rest periods. Dohn and Mason (2013) present a formulation that features four sets of entities. The first is the shifts for staff that cover prespecified demand, which are contained in I^1. The second, denoted by I^2, are the sequences of shifts that form on-stretches and non-working periods that form off-stretches. The set I^3 contains the work-stretches, which are the combination of on- and off-stretches. Finally, a staff roster-line set, denoted by I^4, is at the top of the hierarchy containing sequences of work-stretches.

One well-known staff scheduling problem is the airline crew pairing problem. Like staff rosters, crew pairings exhibit multiple levels as they are formed as an aggregation of duties, which themselves are an aggregation of flights. Following the application of Dantzig-Wolfe reformulation, Desaulniers et al. (1997) present an MP2 formulation; however, the aggregation of flights is not explicitly considered. Vance et al. (1997) propose two models for the crew pairing problem—MP1-2 and MP2-3 formulations. In the former model, the duty periods, corresponding to I^2, are combined to form pairings, which correspond to the elements of I^3. In their latter model, after partitioning the flights into duty periods, an additional set I^4, referred to as the duty set, is defined to comprise duties that cover these disjoint flight sets. Subsequently, Vance et al. (1997) combines I^3 with I^4, i.e., pairings are to be matched with duty sets. This leads to the MP2-3 formulation.

A further example of the crew scheduling problem, described as the crew pairing and assignment problem, is presented by Zeighami and Soumis (2019). While similar to Vance et al. (1997), Zeighami and Soumis (2019) formulate a model where the crew assignments are made up from crew pairings. A problem of the form MP1-2 is used and then Benders' decomposition is applied. Three levels are defined for this MP1-2 formulation, where I^1 comprises the flights of the airline schedule that need to be covered by a pilot and co-pilot, I^2 comprises the crew pairings and I^3 comprises the crew schedules.

3.4 Cutting Stock Problem Variants

The common characteristic of the multi-level structures in the cutting stock domain is the physical entities that form each level such that the pieces in level I^i, $i > 1$ are cut into those of I^{i-1}, as given by cutting patterns. Vanderbeck (2001) presents the MP3 formulation for the three-stage two-dimensional cutting stock problem and developed a column generation approach to solve it. The levels of the decomposition handle the following steps in the given order: i) the ordered pieces are combined into horizontal combinations, ii) the horizontal combinations are used to generate the sections, and iii) sections are used to generate the cutting patterns. In the approach proposed by Vanderbeck (2001), the set of horizontal combinations are enumerated a priori.

Zak (2002) presents the MP1-2 formulation of the MSCS given in (10). The decomposition approach devised in Zak (2002), which is implicitly reflected by their solution methodology, results from defining $I^{2'}$ using the constraints associated with both intermediate roll and finished roll cutting patterns. The same MP1-2 formulation is also utilized in Muter and Sezer (2018) where the steps of the two-level decomposition are followed. A special version of this problem is studied by Valério de Carvalho and Rodrigues (1995) in which each intermediate roll is cut into a single finished roll width. The authors propose an MP2 formulation for this problem.

In a similar setting to the MSCS problem, Gleixner et al. (2020) present a recursive ring packing problem that is derived from the transportation of fixed length pipes. A set of rings can be packing directly into the rectangle or into other rings already contained inside the rectangle. This nested multi-level optimization problem has up to n-levels, where n is the number of different ring types. Gleixner et al. (2020) first formulate this problem in the form of MP1 after defining the set $I^{2'}$; however, this is too difficult to solve. This leads to Gleixner et al. (2020) proposing a more computationally effective MP1-2 formulation.

4 Solution Methods

The large cardinality of I^2 and I^3 induce variable sets that typically render the reformulated problem intractable for general purpose solvers. Thus, many variations of column generation algorithms have been developed to solve multi-level optimization problems. Our focal point is to investigate the distinctive characteristics of problems with nested structure and their respective models, and provide a guideline of algorithms for their solution.

4.1 Bounds Based On MP1

Multi-level optimization problems commonly include a set of constraints that link entities pertaining to specific levels. Of particular interest is when the linking constraints take the form of knapsack constraints. For example, in the MTVRP knapsack constraints link the trips performed by a single vehicle by imposing a limit on the total travel time. Knapsack constraints also provide the link between levels in the MSCS, OBPS and PBPM problems. Relaxing these linking constraints in MP1 can eliminate levels in the multi-level optimization problem and lead to a more easily solvable problem, such as the VRP in the context of the MTVRP. Specifically, when (4c) are knapsack constraints, relaxing these constraints results in a relaxation of MP1 with the second level of the nested structure removed. This relaxation, which is denoted as MP1-R, provides a valid lower bound on the optimal solution to the original problem as a result of the removal of conditions $\mathcal{C}^3$ that guide the aggregation of I^2 into I^3.

The bound provided by this relaxation has been shown empirically to be very tight, regardless of the form of (4c). However, if constraint set (4c) exhibits a knapsack structure the LP bound is not weakened as a result of the relaxation. A proof of this result is presented by Mingozzi et al. (2013)—focusing specifically on the MTVRP. Importantly, this result can also be applied to other problems with a knapsack structure in (4c).

If the solution of MP1-R satisfies (4c), this solution is optimal for MP1. For problems that exhibit the knapsack structure, the feasibility of the solution to MP1-R can be checked by solving an optimization problem. Consider an optimal solution to MP1-R and let Q^+ denote the set of variables that are non-zero in this solution. The problem solved to check the feasibility of the solution of MP1-R is as follows:

$$\bar{C}_{max} = \text{minimize } C_{max}, \tag{12a}$$

$$\text{subject to } \sum_{k \in K} \lambda_q^k = 1, \quad q \in Q^+, \tag{12b}$$

$$\sum_{q \in Q^+} c_q \lambda_q^k \leq C_{max}, \quad k \in K, \tag{12c}$$

$$\lambda_q^k \in \{0,1\}, \quad q \in Q^+, k \in K. \tag{12d}$$

Note that this problem is the parallel machine scheduling problem with makespan minimization. If a solution of (12) satisfies $\bar{C}_{max} \leq T$, where T is the right-hand side of the knapsack constraint, then the assignments $\hat{\lambda}_q^k$ constitute the optimal solution of MP1. The computational results reported in Mingozzi et al. (2013), Muter (2020) and Muter and Öncan (2021) for the MTVRP, OBPS and PBPM, respectively, show that in most of the large instances considered—namely those with at least 50 customers, orders or jobs—the relaxation MP1-R provides solutions that are optimal for MP1. In cases where the solution of the relaxation is not feasible, the above studies revert to solving the MP2 formulation.

4.2 Solving MP2

Since the size of the column set S in MP2 can be prohibitively large, column generation is a viable method to solve its LP relaxation to optimality. Thus, a restricted form of the second-level decomposition master problem MP2, which we refer to as RMP2, is constructed by replacing S by its subset $\bar{S}$ and relaxing the integrality constraints (5d). Letting $\pi \geq 0$ and $\zeta \in \mathbb{R}$ denote the dual variables associated with constraints (5b) and (5c), respectively, the pricing subproblem of the second-level master problem (PSP2) is

$$\min_{s \in S} c_s - \pi a_s - \zeta. \tag{13}$$

Many of the details regarding column generation approaches for MP2 are hidden in the definition of the set S. Since this set corresponds to I^3, the solution approach for PSP2 must take into account the nested structure and the constraints that define the aggregation in levels 2 and 3. We will present three methods to solve this PSP2: branch-and-price, the two-phase approach, and the one-level method.

4.2.1 Branch-and-Price.

The PSP2 given in (13) can also be written as

$$\textbf{(PSP2)} \quad \text{minimize} \quad \sum_{q \in Q} (c_q - \pi a_q) \lambda_q - \zeta, \tag{14a}$$

$$\text{subject to} \quad \sum_{q \in Q} e_q \lambda_q \leq f, \tag{14b}$$

$$\lambda_q \in \{0, 1\}, \quad q \in Q, \tag{14c}$$

where the substitutions $\sum_{q \in Q} c_q \lambda_q^s = c_s$ and $\sum_{q \in Q} a_q \lambda_q^s = a_s$ performed in the second-level decomposition have been reversed. Constraints (14b) and (14c) induce the integer points corresponding to the solution set S. In the formulation of MP2, the PSP2 contains the constraints that define the aggregation of I^2 into I^3. As a result, the set Q is maintained in PSP2.

The large cardinality of Q can make the explicit enumeration of this set cumbersome. However, for small to medium sized instances problem structure may facilitate the enumeration of this set. For example, Azi et al. (2010) show for the MTVRPTW that the route duration limit and time windows makes route enumeration possible. This pre-enumeration enables branch-and-price to be performed only on one-level. In the algorithm presented by Azi et al. (2010), each node in the graph for PSP2 denotes a pre-enumerated feasible route. This algorithmic structure conveniently eliminates the need to employ branch-and-price when solving PSP2. For the MTVRP, Mingozzi et al. (2013) develop a bounding procedure based upon MP1-R to enumerate a subset of routes in Q that are necessary in the solution of MP2. Using these routes, the authors then perform an additional enumeration step to generate the set of schedules in MP2 before solving it by an off-the-shelf solver. In the context of staff rostering, Dohn and Mason (2013) present an MP2 formulation that can be solved to integer optimality by only employing a one-level branching scheme. The subproblem PSP2, while also solved by column generation, doesn't require branching to find integer optimal solutions. This is the result of formulating PSP2 such that all optimal solutions to the LP relaxation satisfy the integrality requirements. A column generator described by Mason and Smith (1998) is used to effectively solve PSP2.

When enumeration is not possible, column generation can be applied to solve PSP2. If the optimal solutions to the LP relaxation of PSP2 do not satisfy the integrality requirements, then branch-and-price is required. This leads to a nested column generation algorithm and the branching schemes are described as *two-level branching*. Letting $\theta \leq 0$ be the vector of dual variables associated with (14b), the restricted pricing subproblem of the PSP2, denoted by PSP1, is given by

$$\min_{q \in Q} c_q - \pi a_q - \theta e_q. \tag{15}$$

Since Q results from the first-level decomposition, we obtain the following formulation by reversing the substitutions given in (3):

$$\begin{aligned} \text{minimize} \quad & (c - \pi A - \theta E)x, & \text{(16a)} \\ \text{subject to} \quad & Bx \geq d, & \text{(16b)} \\ & x \in \mathbb{Z}^+. & \text{(16c)} \end{aligned}$$

Example 4.1 *PSP2 for the MTVRP can be written as*

$$\begin{aligned} \textit{minimize} \quad & \sum_{q \in Q} \bar{c}_q \lambda_q - \zeta, & \text{(17a)} \\ \textit{subject to} \quad & \sum_{q \in Q} c_q \lambda_q \leq T, & \text{(17b)} \\ & \sum_{q \in Q} a_{iq} \lambda_q \leq 1, \quad i \in I^1, & \text{(17c)} \\ & \lambda_q \in \{0, 1\}, \quad q \in Q. & \text{(17d)} \end{aligned}$$

The above problem is a set-packing problem with a knapsack constraint. This form of subproblem also arises as PSP2 in other applications, such as OBPS and PBPM. An alternative form of this subproblem is a knapsack problem with conflicts. The conflicts can be defined between pairs of members of Q, $(q_r, q_p) : q_r, q_p \in Q$, for which $\exists i \in I^1 : a_{iq_r} = a_{iq_p} = 1$ (Bettinelli et al. (2017)). However, regardless of the formulation, PSP2 for the MTVRP tends to be a difficult problem to solve—even for moderately-sized instances.

Letting $\theta^1 \in \mathbb{R}_-$ and $\theta^2 \in \mathbb{R}_-^{|J|}$ be the dual variables associated with constraints (17b) *and* (17c)*, respectively, PSP1 is*

$$\min_{q \in Q} c_q - \sum_{i \in I^1} \pi_i a_{iq} - \sum_{i \in I^1} \theta_i^2 a_{iq} - \theta^1 c_q.$$

The routes in Q are characterized by elementarity and resource constraints so that PSP1 manifests itself as the elementary shortest path problem with resource constraints (ESPPRC).

For the MP2 formulation of the MSCS problem in (11)*, PSP2 can be defined as*

$$\begin{aligned} \textit{maximize} \quad & \sum_{q \in Q} \bar{c}_q \lambda_q - 1, && \text{(18a)} \\ \textit{subject to} \quad & \sum_{q \in Q} c_q \lambda_q \leq T, && \text{(18b)} \\ & \lambda_q \in \mathbb{Z}_+, \quad q \in Q, && \text{(18c)} \end{aligned}$$

where T for this problem denotes the width of the stock roll, $c_q = \sum_{i \in I^1} a_{iq} c_i$ *and* $\bar{c}_q = \sum_{i \in I^1} a_{iq} \pi_i$ *are the total width and reduced cost of the finished roll pattern* $q \in Q$*, respectively. PSP2 is an unbounded knapsack problem whose variables are not known explicitly unless Q is enumerated.*

Letting θ *denote the dual variable associated with* (18b)*, PSP1 can be modeled as the following unbounded knapsack problem:*

$$\begin{aligned} \textit{maximize} \quad & \sum_{i \in I^1} \pi_i a_i - \theta, && \text{(19a)} \\ \textit{subject to} \quad & t^{min} \leq \sum_{i \in I^1} c_i a_i \leq t^{max}, && \text{(19b)} \\ & a_i \in \mathbb{Z}_+, \quad i \in I^1. && \text{(19c)} \end{aligned}$$

In regards to the MSCS, nested column generation has been employed in many studies tackling nested structures (Vanderbeck 2001, Song 2009, Muter and Sezer 2018). Song (2009) present various approaches to solve PSP2, which are based on a set partitioning heuristic. Vanderbeck (2001) enumerates the set I^2 that is used in PSP1 to generate I^3 dynamically by applying column generation in the first level. The generation of cutting patterns I^4 in PSP2 is achieved using a branch-and-bound algorithm where the lower bounds are obtained by Lagrangian relaxation and the upper bounds are found by executing a greedy heuristic.

In Muter and Sezer (2018), the application of branch-and-price to (18), which generates intermediate roll cutting patterns, is hindered by a well-known phenomenon of dichotomous branching in the cutting stock problems. The authors address this issue using a modification of the lexicographic algorithm (Gilmore and Gomory 1963) to solve PSP2 to optimality. At the termination of column generation for MP2, an off-the-shelf solver is then used to solve RMP2 as a MIP to produce an upper bound—in many instances this bound corresponds to the optimal solution.

There have been many applications of nested column generation to solve variants of the VRP. Tilk et al. (2019) apply a nested column generation algorithm to solve the MP2 formulation of the VRP with multiple resource interdependencies. Since PSP2 is formed of a large set of paths, column generation is applied to solve PSP1, which is formulated as an ESPPRC. Two types of branching schemes are used by Tilk et al. (2019) to solve PSP2. The first is based on the original variables and the second is a constraint-based branching. In a similar manner, branch-and-price is employed in both levels by Hennig et al. (2012) to solve the maritime scheduling problem. In the algorithm presented by Hennig et al. (2012), a constraint branching scheme and heuristic pricing are employed.

For the PTVRP, Karabuk (2009) propose a nested column generation algorithm in which the PSP2 generates routes that are formed of a set of mini-clusters. The network underlying PSP2 specifies the nodes to represent requests and the arcs correspond to mini-clusters. For the mini-cluster generation in PSP1, the author presents a dynamic programming algorithm similar to that proposed in Ioachim et al. (1995). Since PSP2 exhibits the total unimodularity property, only one-level branchings are required to solve the PTVRP presented by Karabuk (2009).

4.2.2 Two-phase Approach.

Solving PSP2 by branch-and-price can be both burdensome in terms of the effort invested in coding and unstable in terms of performance. On the other hand, the pre-enumeration of Q and solving PSP2 at every iteration of the column generation algorithm is not always a viable option due to the size of this set. Rather than explicit enumeration, the column set can be implicitly enumerated using dominance relations that are derived from the constraints of PSP2, namely $E\lambda \leq f$. The implicit enumeration that generates the columns needed in the solution of PSP2 is called the first phase of the two-phase approach. After the non-dominated columns are generated, PSP2 is solved directly as an integer program in the second phase. The following propositions specify the general characteristics of the columns generated in the first phase that are used in the second phase.

PROPOSITION 4.1 *For two variables $\lambda_q, \lambda_{q'} \in \mathbb{Z}_+$, $\lambda_{q'} = 0$ in all optimal solutions if $e_q \leq e_{q'}$ and $c_q - \pi a_q \leq c_{q'} - \pi a_{q'}$.*

PROPOSITION 4.2 *For two variables $\lambda_q, \lambda_{q'} \in \{0, 1\}$, $\lambda_q = 0$ in all optimal solutions if $e_q \leq e_{q'}$, $c_q - \pi a_q \leq c_{q'} - \pi a_{q'}$ and there exists a row k in E such that $e_{kq} + e_{kq'} > f_k$.*

PROPOSITION 4.3 *A column λ_q can exist in the solution of PSP2 if and only if $c_q - \pi a^q < 0$.*

Since the values of the dual variables π change at every solution of RMP2, the enumeration taking place in the first phase must be executed at every call of PSP2, which would result in a distinct set of columns denoted by $Q' = \{q \in Q : c_q - \pi a_q < 0\}$.

An example of the complete enumeration of Q at the outset of the column generation algorithm to solve MP2 is presented by Azi et al. (2010). This pre-enumeration of Q can inflict a computational burden on the second phase where each route in Q corresponds to a node in the shortest path problem with resource constraints on a route graph. The approach of Azi et al. (2010) can be enhanced by reducing Q to $Q' = \{q \in Q : c_q - \pi a_q < 0\}$ in the first phase by making use of the dual variables π so that the second phase is applied to a smaller set at each iteration of the column generation algorithm.

The two-phase approach has also been employed in Akca (2010) and Muter et al. (2014). This approach, which involves the enumeration of Q' at each iteration of the column generation algorithm, potentially causes scalability issues in PSP2 when the cardinality of Q' is large. This is particularly evident in the early iterations of the column generation algorithm—before the dual solutions have stabilized. Rather than enumerating Q' at every iteration, a heuristic enumeration scheme can be applied that generates only a subset of Q' to be utilized in the second phase to solve PSP2. The heuristic enumeration continues while a negative reduced cost column is found. Otherwise, exact enumeration is required. The exact enumeration of Q' is typically only performed few times towards the end of the column generation algorithm.

Two-phase approach has been applied to solve the MSCS (Muter and Sezer (2018)) and its special version (Valério de Carvalho and Rodrigues (1995)) in which each intermediate roll is cut into a single finished roll. The PSP2 for this problem is the integer knapsack problem, which allows elimination of many rolls I^2 in the first level of this method due to Proposition 4.1. Even though Valério de Carvalho and Rodrigues (1995) pre-enumerated the set of intermediate rolls, the columns that don't qualify for Q' are not taken into consideration in PSP2.

In some cases, the large cardinality of Q' results from the set packing constraints in PSP2 emerging in the form of (17c). For VRP extensions, the set packing constraints induce elementarity in the first level, which generates a permutation of elements of I^1. These constraints not only complicate the enumeration process in the first level but also proliferate the set Q' that forms the second level. Elementarity, i.e. each element of I^1 can appear at most once in each element of I^2, is common across many applications. On the other hand, for the OBPS and PBPM, it is the combination, rather than permutation in the case of VRP extensions, of elements in I^1 that coalesce in the first phase of the two-phase approach to form Q'. For the OBPS and PBPM, Muter and Öncan (2021) and Muter (2020), respectively, address the adverse effect of the packing constraints by simply relaxing them. The resulting problem is a 0-1 knapsack problem, which is easier to solve and lends itself to a more efficient enumeration in the first phase. However, the solution of the two-phase approach may contain some $i \in I^1$ more than once and thus only provides a lower bound. In order to obtain the optimal solution using this relaxation, Muter (2020) and Muter and Öncan (2021) employ a technique based on the enumeration of variables.

4.2.3 One-level Method.

The one-level method solves PSP2, given in (13), directly by generating the columns in S without a separate procedure for the generation of Q. In this approach, the MP2 results from a one-level decomposition defining $I^{2'}$—rather than performing the steps of a two-level decomposition sequentially. The mathematical model of (13) associated with this approach can be written as follows:

$$\begin{aligned} \text{minimize} \quad & (c-\pi A)x - \zeta, \\ \text{subject to} \quad & Bx \leq d, \\ & Ex \leq f, \\ & x \in \{0,1\}^n_+. \end{aligned}$$

Despite its modularity, this model is challenging to solve. For the MTVRP, this approach entails a model that contains all constraints of (7a)-(7g) except for (7b). In the solution of the VRP extensions where this subproblem is modeled as the ESPPRC, incorporating the constraints associated with the transition from I^2 to I^3 in the one-level method adds new resources associated with the members of this set.

A one-level method to solve the subproblem of MP2 of ILRSP is proposed in Akca (2010). This solution method employs a modified form of the labeling algorithm presented by Feillet et al. (2004), where *replenishment arcs* are utilized to allow the vehicles to continue their schedules after a route ends at a depot. This approach can be adapted for any VRP extension that has a two-level decomposable structure. Examples for the MDVRPI and MTVRPTW are presented by Muter et al. (2014) and Hernandez et al. (2016), respectively. The computational experiments by Akca (2010) and Muter et al. (2014) show that the two-phase approach typically outperforms the one-level method.

Zak (2002) applies the one-level method to solve the MSCS heuristically. The author modifies (18) by replacing the total width c_q and the reduced cost $\bar{c}_q$ of $q \in Q$ with $c_q = \sum_{i \in I^1} a_{iq} c_i$ and $\bar{c}_q = \sum_{i \in I^1} a_{iq} \pi_i$, respectively. The resulting problem is not only large-scale but also non-linear due to multiplication of λ_q and a_{iq}. In order to circumvent these difficulties, Zak (2002) limits the number of new finished roll cutting pattern to be added to the problem in each column generation iteration to one. This restriction enabled Zak (2002) to solve the MSCS heuristically as an extension of the knapsack problem. A similar heuristic approach is adopted by Wang and Tang (2010) to solve the PBPM in which only one new batch is allowed to reside in a machine schedule generated by the PSP2.

4.3 Solving MP1

Solving the first-level decomposition model MP1 directly is an alternative to solving MP2. Compared to the solution algorithms for MP2, which are hindered by PSP2 needing to generate $s \in S$, it has the advantage of dealing with column set Q in a single level. For example, generating routes in the solution of MTVRP is easier than generating schedules. However, restricting the decomposition to a single level misses the opportunity to exploit properties such as an improved lower bound and a more compact formulation that are only available after the second-level decomposition is applied. Even though solving MP1 by column generation is more practical in terms of solving the subproblem, a number of other issues affect the effectiveness of the overall branch-and-price scheme. For instance, considering MP1 for the MTVRP, given by (8), the LP relaxation bound is considerably loose. In fact, relaxing the knapsack constraints (8c) does not deteriorate this bound. Moreover, the assignment of routes to vehicles represented by variables λ_q^k causes difficulties in the branch-and-bound tree due to symmetry.

For the MTVRPTW with duration limit, Hernandez et al. (2014) and Macedo et al. (2011) propose MP1 formulations that rely on the pre-enumeration of the route set—as in Azi et al. (2010). Methodologies used in these studies deal with the selection of routes by incorporating the constraint set $Ex \leq f$ directly into MP1, rather than handling their assignment to vehicles. Hernandez et al. (2016) consider the MTVRPTW without a limitation on route duration, which proliferates the number of routes to be incorporated completely by pre-enumeration. In their MP1 formulation, the $Ex \leq f$ constraints emerge as the time-indexed constraints that limit the number of routes performed at any time period to the fleet size. Hernandez et al. (2016) also propose a one-level method for the solution of the MP2 formulation. The computational experiments show that the proposed methodology aimed at solving MP1 outperforms the one-level method.

Crevier et al. (2007) formulate an MP1 model for the MDVRPI, where the set Q corresponds to the inter-depot routes and the constraints $Ex \leq f$ impose the connectivity of inter-depot routes to form schedules. The authors solve this model heuristically by first generating a pool of inter-depot routes. Keeping the pool size relatively small, they managed to solve MP1 by an off-the-shelf solver in a reasonable amount of time.

4.4 Solving MP1-2

The structure of MP1-2 requires the use of specialized algorithms to effectively solve the corresponding problems. The main difficulty that arises from the application of Dantzig-Wolfe reformulation to form MP1-2 is that the constraint set is dependent on the column set. Since the size of the column set is exponential in the problem input, this formulation also bears a prohibitively large number of constraints. The proliferation of constraints has led to the development of algorithms combining a delayed constraint generation algorithm with a delayed column generation algorithm.

4.4.1 Simultaneous Column-and-Row Generation.

Applying column generation to MP1-2 entails the generation of both S and Q in a single model. However, when Q is replaced by $\bar{Q}$ to form the RMP, constraints (6c) associated with $Q \backslash \bar{Q}$ are absent from the model. Hence, the generation of a new $q \in Q \backslash \bar{Q}$ adds a constraint in (6c). Problems that meet a set of conditions and are amenable for this method are referred to as the problems with column-dependent rows (Muter et al. 2013), and MP1-2 falls into this problem class. Recall that algorithms developed for MP2 generate $q \in Q$ during the solution of PSP2, which produces $s \in S$, while those designed for MP1 only generate $q \in Q$. Since Q and S reside in the same level in MP1-2, the design issue becomes whether S and Q can be generated in separate subproblems at the same level. Regardless, the duals associated with (6c) must be taken into account in the subproblems that generate columns and trigger new constraints.

Note that due to constraints (6c), generating a $q \in Q \backslash \bar{Q}$ when there is no $s \in S$ in the RMP with $h_{qs} > 0$ results in a degenerate iteration, since λ_q is fixed to zero. Thus, it is algorithmically more accurate to generate variables in Q in tandem with S in a single subproblem. Generating columns from Q and S simultaneously in the same subproblem can be achieved using the approaches described above to solve MP2. Further, we can show that the reduced cost of column $s \in \bar{S}$ resulting from applying simultaneous column-and-row generation is equal to that resulting from applying column generation to MP2, which also attests to the equivalence of the subproblem of MP1-2 and PSP2.

Proposition 4.4 *The pricing subproblem of MP1-2 resulting from the simultaneous column-and-row generation is equivalent to PSP2.*

Proof 4.1 *The proof is given in the Appendix A.*

A motivation for solving MP1-2, which grows both in columns and rows and still hinges upon the solution of PSP2, lies in the capability of generating Q and S separately. For instance, in MSCS formulation given in

(10), the one-to-many relationship between J, the index set of (10c), and Q induces that many finished roll cutting patterns can be identified for a single intermediate roll width. This is in contrast with the other applications of MP1-2 where there is a one-to-one relationship. An example is the model for PBPM presented by Wang and Tang (2010) that leads to a single subproblem generating machine schedules. A separate $\lambda-$generating subproblem also brings about the benefit of warming up the dual variables from their initial erratic state. This subproblem can be invoked as long as it generates a negative reduced cost λ variable, which is then followed by PSP2 to generate y_s, $s \in S\backslash\bar{S}$. In Zak (2002) and Muter and Sezer (2018), the $\lambda-$generating subproblem is called for each $j \in \bar{J}$ to generate finished roll patterns from the existing intermediate rolls.

4.4.2 Benders' Decomposition.

Problem MP1-2 exhibits a bordered block diagonal structure, with the border given by a set of linking variables. The application of Benders' decomposition to MP1-2 results in the following formulations of the master and subproblem:

$$
\begin{aligned}
\text{minimize} \quad & \varphi, \\
\text{subject to} \quad & \sum_{q \in Q} a_{iq} \lambda_q \geq b, \\
& \varphi \geq \sum_{q \in Q} \mu_q^{\omega} \lambda_q + \rho^{\omega}, \quad \forall \omega \in \mathcal{O}, \\
& \lambda_q \in \mathbb{Z}, \quad q \in Q, \\
& \varphi \in \mathbb{R}.
\end{aligned}
$$

$$
\begin{aligned}
z(\hat{\lambda}) = \text{minimize} \quad & \sum_{s \in S} c_s y_s, \\
\text{subject to} \quad & \sum_{s \in S} h_{qs} y_s \geq \hat{\lambda}_q, \quad q \in Q, \\
& \sum_{s \in S} y_s = 1, \\
& y_s \in \mathbb{Z}, \quad s \in S.
\end{aligned}
$$

The set $\mathcal{O}$ is an index set for extreme points of the dual feasible region of the subproblem, where μ_q^{ω} and ρ^{ω} are the dual values corresponding to the first and second constraints, respectively. The constraints induced by $\mathcal{O}$ are termed optimality cuts, and the auxiliary variable φ is an underestimator for the optimal subproblem objective value. In this decomposition, the linking and master variables are the λ_q variables, while the subproblem variables are the y_s variables. This is not the only decomposition achievable by applying Benders' decomposition. An equivalent decomposition can be found by interchanging the master and subproblem variables, and corresponding constraints.

The application of Dantzig-Wolfe reformulation and Benders' decomposition to solve the MP1-2 formulation of the crew pairing and assignment problem is presented by Zeighami and Soumis (2019). In the formulation presented by Zeighami and Soumis (2019), the assignment variables correspond to the y variables where the pairing variables correspond to the λ variables. The limitations of combining column generation and Benders' decomposition in this manner is addressed by Zeighami and Soumis (2019) through the formulation of *weak* Benders' optimality cuts. It is reported that finding the optimal solution using the MP1-2 formulation is very difficult, as such heuristic branching schemes are required. The experience of Zeighami and Soumis (2019) demonstrate the challenges associated with applying Benders' decomposition to problems with column-dependent rows. These challenges are addressed in Muter et al. (2018) where a combined Benders' decomposition and simultaneous column-and-row generation algorithm is proposed.

4.4.3 Alternative algorithm.

Motivated by the solution approaches for the multi-stage cutting stock problem, Gleixner et al. (2020) propose a column generation-based algorithm to solve the recursive ring packing problem. A Dantzig-Wolfe reformulation is developed to exploit one-level packings; thus, formulating the problem as MP1-2. The proposed formulation defines I^1 as the set of rings, the set of I^2 contains the packings of circles (of equivalent diameter as the rings in I^1) into rings and I^3 is the set of packings of circles into rectangles. The solution approach begins with a partial enumeration of circular packings, partitioning all packings into *feasible* and *unknown* sets. Branch-and-price is then performed to find feasible upper and lower bounds. If the bounding solutions use any packings from the *unknown* set, then feasibility of these packings is checked. If the used *unknown* packings are found to be infeasible, constraints forbidding their use are imposed and the algorithm continues.

4.5 Implicit Enumeration

An alternative to branch-and-price for solving multi-level optimization problems is predicated on a fundamental result of integer programming that capitalizes on tight optimality gaps. The implicit enumeration approach uses a reduced form of the master problem by enumerating a set columns that may potentially reside in the optimal solution. The resulting master problem is then solved by powerful off-the-shelf solvers. Baldacci and Mingozzi (2009) has employed this approach in the solution of the VRP and its extensions.

Consider the first- or second-level master problem, namely MP1 or MP2 respectively. Applying column generation to the LP relaxation provides a lower bound on the optimal objective value of the respective master problem, which is denoted by $\underline{z}^k$ where $k = 1, 2$ indicates the level of the master problem being tackled. When column generation terminates for the k^{th}-level model, it also outputs dual feasible solutions for this model. Since upper bounding methods are out of the scope of this paper, we assume that an upper bound $\bar{z}$ is at our disposal. The following proposition that is defined for a given level k strips the immaterial variables out from the k^{th} level model when constructing the reduced form of the master problem.

PROPOSITION 4.5 *The columns whose reduced cost is larger than $\bar{z} - \underline{z}^k$ can be discarded in the solution of MPk, $k = 1, 2$.*

Let Q^* and S^* refer to the sets of columns in the reduced form of MP1 and MP2, respectively. The enumeration of one of these sets is carried out after the bounds of the respective model are obtained. The drawbacks of employing MP1 in an implicit enumeration scheme have already been addressed in the previous section. One significant drawback is the poor quality of the LP relaxation bound. On the other hand, the enumeration of Q^* is intrinsically simpler than that of S^*, since the former is handled in a single step from I^1 to I^2 while the latter involves a further step from I^2 to I^3. Therefore, if MP2 and one of the methods designed for its solution is selected, a tight lower bound ensues at the expense of a more burdensome enumeration. In several studies on multi-level problems (Mingozzi et al. 2013, Muter 2020, Muter and Öncan 2021), MP2 has been adopted to reap the benefits of tighter lower bounds; however, the set Q^* is enumerated to form the reduced MP1. The gap between this lower bound $\underline{z}^2$ and a given upper bound $\bar{z}$ sets a limit on the reduced costs in the enumeration of Q^* due to the proposition given below.

PROPOSITION 4.6 *Let $\hat{\pi}$ and $\hat{\zeta}$ denote the dual values outputted at the termination of column generation applied to MP2. In the optimal solution of MP1, a column q' satisfying $\bar{c}_{q'} = c_{q'} - \hat{\pi} a_{q'} > \bar{z} - \underline{z}^2$ is never selected.*

PROOF 4.2 *The proof is given in the Appendix B.*

As alluded to above, Q^* is easier to enumerate than S^* and its cardinality is also considerably smaller. However, we remark that the application of this methodology in the literature has shown that the solution of MP1 is substantially more difficult than MP2, which boasts a compactness that is achieved through the second-level decomposition.

Since PSP2 is itself an integer programming problem, implicit enumeration has implications on this problem as well. These implications are two-fold. First, in regards to the two-phase approach for solving PSP2, an implicit enumeration procedure is required to identify the set of negative reduced cost columns Q'. For PSP2, a readily available upper bound is zero as this is the reduced cost of a basic variable of RMP2. If a fast lower bounding method can be developed, which provides a lower bound z^{LB} and duals $\hat{\theta}$ of PSP2, all columns $q \in Q'$ that violate $c_q - \hat{\pi} a_q - \hat{\theta} e_q \leq 0 - z^{LB}$ can be eliminated. This methodology to reduce the size of Q', which forms the reduced PSP2 in the second phase, has been applied by Muter et al. (2014) and Muter and Sezer (2018) in solving PSP2 of the MDVRPI and MSCS, respectively. In the latter work, a knapsack bound and duals were used, while in the former work, a more complex LP relaxation bound is obtained. Second, implicit enumeration is an alternative to solving PSP2 by branch-and-price. Rather than resorting to a branch-and-bound tree search after PSP2 is solved by column generation, one can make use of the lower bound z^{CG} alongside the duals $\hat{\pi}$ and $\hat{\theta}$ and the trivial upper bound of zero in the enumeration of $q \in Q'$ satisfying $c_q - \hat{\pi} a_q - \hat{\theta} e_q \leq 0 - z^{CG}$.

5 Final Remarks and Conclusion

This paper provides an overview of nested multi-level optimization problems. This overview covers the various formulations resulting from the application of decomposition techniques, a discussion of applications that have been examined in the literature and a presentation of state-of-the-art solution methods. As a summary, the main points and contributions of this paper are:

- Three alternative models based on Dantzig-Wolfe reformulation are presented and their computational effectiveness with respect to applications from the literature has been discussed. It is observed that most of the literature on nested decomposition tackle the MP2 formulations. Branch-and-price is the prevalent method for the solution of PSP2.

- The constraints $\mathcal{C}^2$ and $\mathcal{C}^3$ are important determinants in the selection of algorithms to solve MP2. If $q \in Q$ and $s \in S$ are allowed to contain multiple occurrences of an element of I^1, the resulting subproblems in both levels are exempt from the complicating elementarity requirement. Furthermore, if I^2 and I^3 are formed of combinations, rather than permutations, of I^1 and I^2, respectively, the respective subproblems can generally be formulated on topologically sorted networks that limit the solution space considerably. These structures favor enumerative algorithms, such as the two-phase approach, which avoid the need to apply branch-and-price in the solution of PSP2.

- Unlike branch-and-price and the two-phase approach, the one-level method solves MP2 by generating S directly through $I^{2'}$, which involves both $\mathcal{C}^2(x)$ and $\mathcal{C}^3(x)$. In the examined applications, this approach incorporates constraints associated with the third level $\mathcal{C}^3(x)$ into PSP1, which is typically solved by dynamic programming. This brings about an explosion in the state space, leading to the one-level method being outperformed by the other methods that have been discussed.

- The relaxation of constraints in MP1 leads to well-studied problems in the literature. Depending on the relaxed constraint, the resulting lower bound of MP1 can be very strong for instances with large $|I^1|$. Moreover, one can capitalize on the available algorithms to solve MP1, which involve aggregation of elements only from I^1 to Q, if its formulation encapsulates the formation of S from Q effectively.

- Tackling MP1-2 gives rise to simultaneous column-and-row generation, which triggers generation of constraints along with columns. We show that the subproblems of MP2 and MP1-2 are similar so that the algorithms presented for MP2 are still required for the solution of MP1-2. Furthermore, MP1-2 may be preferable over MP2 only if a one-to-many relation exists between the linking constraints and the variable set Q.

- Implicit enumeration of columns is an effective solution technique that has been employed in several applications to find integer optimal solutions. Finding tight upper and lower bounds is of paramount importance for the success of this method. We show that the levels in which enumeration and column generation take place can be different and should be chosen according to the simplicity of the algorithm in the respective level.

This paper has shown that common multi-level structures are routinely encountered in various application domains. Using the theoretical framework and generalized model formulations, we have shown that the most effective solution techniques are broadly applicable to solve numerous applied optimization problems. With the increasing prevalence of generic branch-cut-and-price solvers, such as Coluna (Coluna 2021), DIP (Galati et al. 2012), and GCG (Bergner et al. 2015) the generalization of solution approaches will drive the development of software for the deployment of state-of-the-art column generation algorithms. Future work will involve the integration of the key results from this paper into generic branch-cut-and-price solvers to support the formulation and solution of nested multi-level optimization problems.

A Proof of Proposition 4.4

Let π, w and ζ denote the duals associated with (6b), (6c) and (6d), respectively. The reduced cost of $s \in S \backslash \bar{S}$ is

$$\bar{c}_s = c_s - \sum_{q \in \bar{Q}} w_q h_{qs} - \sum_{q \in Q \backslash \bar{Q}} w_q h_{qs} - \zeta, \tag{20}$$

where w_q, $q \in Q \backslash \bar{Q}$ is unknown as the corresponding constraint is missing from the RMP. If $h_{qs} > 0$ for some $q \in Q$, adding s to the problem will introduce variable λ_q and constraint (6c) associated with q. The reduced cost of λ_q is

$$\bar{c}_q = 0 - \pi a_q + w_q. \tag{21}$$

The value of w_q influences both $\bar{c}_s$ and $\bar{c}_q$ but in different directions, i.e., it has negative sign in the former and positive in the latter. Introducing any $q \in Q$ with $\bar{c}_q < 0$ to MP1-2 causes a degenerate iteration as $y_s = 0$ for $h_{qs} > 0$ and $s \in S \backslash \bar{S}$. In this iteration, if λ_q is selected as basic at constraint (6c) associated with q, which leads to $\bar{c}_q = 0$, all dual variable values stay intact, and the new dual variable attains the value $w_q = \pi a_q$ (See Muter et al. (2013) for the proof). Thus, the reduced cost of s decreases by $w_q = \pi a_q$ with each new $q \in Q \backslash \bar{Q}$ and $h_{qs} > 0$ since $\pi \geq 0$. Plugging the value w_q gives

$$\bar{c}_s = c_s - \sum_{q \in \bar{Q}} w_q h_{qs} - \sum_{q \in Q \backslash \bar{Q}} (\pi a_q) h_{qs} - \zeta. \tag{22}$$

For the currently existing $q \in \bar{Q}$, the dual value w_q has been retrieved from the solution of the RMP. This subproblem and PSP2 given in (14) are similar in that h_{qs} corresponds to λ_q defined in the PSP2. Hence, the methods presented for the solution of PSP2 also solve this PSP.

B Proof of Proposition 4.6

A column that can be non-zero in the optimal solution of MP2 satisfies

$$\bar{c}_s = \sum_{q \in Q} (c_q - \hat{\pi} a_q) \lambda_q - \hat{\zeta} \leq \bar{z} - \underline{z}^2, \tag{23}$$

where the first equality follows from the substitution in the second-level decomposition and the inequality follows from Proposition 4.5. When column generation terminates at the LP optimal solution of MP2, $\bar{c}_s \geq 0$, $s \in S$ is ensured, which leads to

$$\hat{\zeta} \leq \sum_{q \in Q} (c_q - \hat{\pi} a_q) \lambda_q \leq \bar{z} - \underline{z}^2 + \hat{\zeta}. \tag{24}$$

Due to (6c), if $\lambda_{q'} = 1$ for some $q' \in Q$, it should reside in one $s \in S$. Suppose that q' satisfies $(c_{q'} - \hat{\pi} a_q) > \bar{z} - \underline{z}^2$. Any $s \in S$ containing q' satisfies (23), i.e., $\bar{c}_s \leq \bar{z} - \underline{z}^2$, if the following condition holds:

$$\sum_{q \in Q \setminus q'} (c_q - \hat{\pi} a_q) \lambda_q < \hat{\zeta},$$

This is not possible at the LP optimal solution of MP2, otherwise, those $q \in Q \setminus q'$ with $\lambda_q = 1$ would form a schedule s' with a negative reduced cost $\bar{c}_{s'} < 0$ due to the first inequality in (24). Therefore, a column q is in Q^* only if $c_q - \hat{\pi} a_q \leq \bar{z} - \underline{z}^2$.

References

Akca, Z. (2010). *Integrated Location, Routing and Scheduling Problems: Models and Algorithms*. PhD thesis, Lehigh University.

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. and Mingozzi, A. (2009). A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120:347–380.

Bard, J. F., Huang, L., Dror, M., and Jaillet, P. (1998). A branch and cut algorithm for the VRP with satellite facilities. *IIE Transactions*, 30:821–834.

Bergner, M., Caprara, A., Ceselli, A., Furini, F., Lübbecke, M., Malaguti, E., and Traversi, E. (2015). Automatic Dantzig-Wolfe reformulation of mixed integer programs. *Mathematical Programming*, 149(1–2):391–424.

Bettinelli, A., Cacchiani, V., and Malaguti, E. (2017). A branch-and-bound algorithm for the knapsack problem with conflict graph. *INFORMS Journal on Computing*, 29(3):457–473.

Coluna (2021). Coluna.jl. See `https://github.com/atoptima/Coluna.jl`. Last accessed: 11-10-2021.

Crevier, B., Cordeau, J.-F., and Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176(2):756–773.

Desaulniers, G., Desrosiers, J., Dumas, Y., Marc, S., Rioux, B., Solomon, M., and Soumis, F. (1997). Crew pairing at Air France. *European Journal of Operational Research*, 97(2):245–259.

Desaulniers, G., Desrosiers, J., and Solomon, M., editors (2005). *Column Generation*, New York. Springer.

Desaulniers, G., Errico, F., Irnich, S., and Schneider, M. (2016). Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6):1388–1405.

Dohn, A. and Mason, A. (2013). Branch-and-price for staff rostering: An efficient implementation using generic programming and nested column generation. *European Journal of Operational Research*, 230(1):157–169.

Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44:216–229.

Galati, M., Ralphs, T., and Wang, J. (2012). Computational experience with generic decomposition using the DIP framework. In *Proceedings of RAMP 2012*.

Gilmore, P. C. and Gomory, R. E. (1963). A linear programming approach to the cutting-stock problem—Part II. *Operations Research*, 11(6):863–888.

Gleixner, A., Maher, S., Müller, B., and Pedroso, J. P. (2020). Price-and-verify: A new algorithm for recursive circle packing using Dantzig-Wolfe decomposition. *Annals of Operations Research*, 284:527–555.

Hennig, F., Nygreen, B., and Lübbecke, M. E. (2012). Nested column generation applied to the crude oil tanker routing and scheduling problem with split pickup and split delivery. *Naval Research Logistics*, 59(3-4):298–310.

Hernandez, F., Feillet, D., Giroudeau, R., and Naud, O. (2014). A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. *4OR*, 12(3):235–259.

Hernandez, F., Feillet, D., Giroudeau, R., and Naud, O. (2016). Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows. *European Journal of Operational Research*, 249(2):551–559.

Ioachim, I., Desrosiers, J., Dumas, Y., Solomon, M., and Villeneuve, D. (1995). A request clustering algorithm for door-to-door handicapped transportation. *Transportation Science*, 29(1):63–78.

Karabuk, S. (2009). A nested decomposition approach for solving the paratransit vehicle scheduling problem. *Transportation Research Part B*, 43(4):448–465.

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

Macedo, R., Alves, C., Clautiaux, F., and Hanafi, S. (2011). Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. *European Journal of Operational Research*, 214(3):536–545.

Maher, S. J. (2016). Solving the integrated airline recovery problem using column-and-row generation. *Transportation Science*, 50(1):216–239.

Mason, A. J. and Smith, M. C. (1998). A nested column generator for solving rostering problems with integer programming. In *International conference on optimisation: techniques and applications*, pages 827–834. Curtin University of Technology Perth, Australia.

Mingozzi, A., Roberti, R., and Toth, P. (2013). An exact algorithm for the multi-trip vehicle routing problem. *INFORMS Journal on Computing*, 25(2):193–207.

Muter, İ. (2020). Exact algorithms to minimize makespan on single and parallel batch processing machines. *European Journal of Operational Research*, 285(2):470–483.

Muter, İ., Birbil, Ş. İ., and Bülbül, K. (2013). Simultaneous column-and-row generation for large-scale linear programs with column-dependent-rows. *Mathematical Programming*, 142(1-2):47–82.

Muter, İ., Birbil, Ş. İ., and Bülbül, K. (2018). Benders decomposition and column-and-row generation for solving large-scale linear programs with column-dependent-rows. *European Journal of Operational Research*, 264(1):29–45.

Muter, İ., Cordeau, J.-F., and Laporte, G. (2014). A branch-and-price algorithm for the multidepot vehicle routing problem with interdepot routes. *Transportation Science*, 48(3):425–441.

Muter, İ. and Öncan, T. (2021). Order batching and picker scheduling in warehouse order picking. *IISE Transactions*, Published Online, http://dx.doi.org/10.1080/24725854.2021.1925178.

Muter, İ. and Sezer, Z. (2018). Algorithms for the one-dimensional two-stage cutting stock problem. *European Journal of Operational Research*, 271(1):20–32.

Song, S. H. (2009). A nested column generation algorithm to the meta slab allocation problem in the steel making industry. *International Journal of Production Research*, 47(13):3625–3638.

Tilk, C., Drexl, M., and Irnich, S. (2019). Nested branch-and-price-and-cut for vehicle routing problems with multiple resource interdependencies. *European Journal of Operational Research*, 276(2):549–565.

Valério de Carvalho, J. M. and Rodrigues, A. G. (1995). An LP-based approach to a two-stage cutting stock problem. *European Journal of Operational Research*, 84(3):580–589.

Vance, P., Barnhart, C., Johnson, E., and Nemhauser, G. (1997). Airline crew scheduling: A new formulation and decomposition algorithm. *Operations Research*, 45(2):188–200.

Vanderbeck, F. (2000). On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research*, 48(1):111–128.

Vanderbeck, F. (2001). A nested decomposition approach to a three-stage, two-dimensional cutting-stock problem. *Management Science*, 47(6):864–879.

Wang, G. and Tang, L. (2010). A row-and-column generation method to a batch machine scheduling problem. In *Proceedings of the Ninth International Symposium on Operations Research and Its Applications (ISORA10)*, pages 301–308, Chengdu-Jiuzhaigou, China.

Zak, E. (2002). Row and column generation technique for a multistage cutting stock problem. *Computers & Operations Research*, 29(9):1143–1156.

Zeighami, V. and Soumis, F. (2019). Combining Benders' decomposition and column generation for integrated crew pairing and personalized crew assignment problems. *Transportation Science*, 53(5):1479–1499.