A Two-Stage Stochastic Programming Approach for Multi-Activity Tour Scheduling

María I. Restrepo[a,b,*], Bernard Gendron[a,c], Louis-Martin Rousseau[a,b]

[a] *Centre interuniversitaire de recherche sur les réseaux d'Entreprise, la logistique et le transport, CIRRELT*
[b] *Département de mathématiques et de génie Industriel, Polytechnique Montréal, Montréal, Québec H3C 3A7, Canada*
[c] *Département d'Informatique et de Recherche Opérationelle, Université de Montréal, Montréal, Québec H3C 3J7, Canada*

Abstract

This paper addresses a discontinuous multi-activity tour scheduling problem under demand uncertainty and when employees have identical skills. The problem is formulated as a two-stage stochastic programming model, where first-stage decisions correspond to the assignment of employees to weekly tours, while second-stage decisions are related to the allocation of work activities and breaks to daily shifts. A multi-cut L-shaped method is presented as a solution approach. Computational results on real-based and randomly generated instances show that the use of the stochastic model helps to reduce understaffing and overstaffing costs, when compared with the expected-value problem solutions.

Keywords: Scheduling, Stochastic multi-activity tour scheduling problem, Two-stage stochastic programming model, L-shaped method, Context-free grammars.

*Corresponding author
Email addresses: maria-isabel.restrepo-ruiz@polymtl.ca (María I. Restrepo), bernard.gendron@cirrelt.ca (Bernard Gendron), louis-martin.rousseau@cirrelt.net (Louis-Martin Rousseau)

1. Introduction

The *multi-activity tour scheduling problem* (MATSP) is the integration of two problems, *the multi-activity shift scheduling problem* (MASSP) and the *days-off scheduling problem*. In the MASSP, the *planning horizon* is one day divided into *time periods* of equal length. The MASSP is concerned with the design of shifts, each characterized by a start time and a length. Since employees can perform several *work activities* during the same shift, this problem also deals with choosing the work activities and the rest periods to assign to shifts to respond to a *demand*. This demand corresponds to a demand for service, that is translated into the number of employees required for each work activity and time period. The days-off scheduling problem deals with the selection of employee *working days* and *days-off* over a planning horizon of at least one week. In the MATSP, the constraints characterizing the feasibility of daily shifts and weekly tours, as well as the *work rules* for the allocation of work activities and rest breaks to the shifts, are usually defined by *employee regulations* and *workplace agreements*. The MATSP can be categorized into different variants depending on the characteristics considered. For instance, the *personalized* version of the MATSP appears when employees have individual *preferences* and *skills*. The *anonymous* version of the MATSP corresponds to the case when employees have identical skills. When shifts are allowed to span from one day to another, the *continuous* version of the MATSP arises; otherwise, we have the *discontinuous* version of the problem. In this paper, we consider the discontinuous anonymous version of the MATSP.

Realistic applications of the MATSP for service companies such as retail stores, restaurants, call centers, and banks that operate outside the standard 8-hours shift, 5-days per week schedule and that face wide fluctuations in demand become challenging due to several factors. First, complex large-scale models result as a consequence of considering multiple work activities, multiple work rules, employee regulations, workplace agreements and flexibility in the composition of daily shifts and weekly tours. Second, since demand is typically unknown when scheduling decisions need to be taken, specialized models and solution techniques that allow to include this uncertainty should be developed. Specifically, such models and techniques allow to make a decision on the employee schedule before a realization of the demand is known. Then, after demand becomes known, a recourse action is implemented to compensate deficiencies in the previously made schedules (e.g., undercover-

ing and overcovering of demand). Recent research has shown that personnel scheduling methods that include demand uncertainty can lead to significant reductions in staffing costs and to important improvements in customer service levels.

In this paper, we study the discontinuous *stochastic multi-activity tour scheduling problem* (SMATSP) for employees with identical skills. In this problem, mid-term staffing decisions (allocation of weekly tours to employees) that are feasible to schedule without knowing the actual realization of demand, will be generated in advance (e.g., one week ahead), while short-term adjustments (allocation of work activities and breaks to daily shifts) are made once improved daily demand information becomes available. The aim of our study is to propose a model for the generation of a robust weekly tactical plan including: (i) the design of weekly schedules (tours) where days-off, working days, shift start times and shift lengths are defined; (ii) the allocation of employees to those weekly tours. This tactical plan will be able to accommodate changes in operational decisions (work activity and break allocation to daily shifts), due to the stochastic variation of demand. Our study is motivated by applications in service companies, where it is important to define and to post employee weekly schedules at least one week in advance to allow for choices and to reduce turnovers due to application of just-in-time scheduling. Moreover, these companies are also interested in robust solutions (robust weekly schedules) that can be easily adapted to possible perturbations in demand.

To the best of our knowledge, this is the first work in the literature to address the SMATSP problem. Therefore, the main contribution of this paper lies in the presentation of a two-stage stochastic programming model where employee weekly schedules (tactical first-stage decisions) have to be defined in advance and cannot be reviewed on a daily basis. On the contrary, work activities and break allocation to daily shifts (operational second-stage decisions) are allowed to be adjusted on a daily basis, depending on the realization of the stochastic demand. The paper also proposes, as a secondary contribution, the development and implementation of an extension of a decomposition method for the deterministic case, previously presented in Restrepo et al. (2015). In this extension, a heuristic multi-cut L-shaped algorithm is developed, where the first-stage problem is solved via column generation as the complete enumeration of weekly tours makes the problem intractable, and the second-stage problem decomposes by days and by demand scenarios. The importance of solving the proposed model is demonstrated with an ex-

tensive computational study, showing that including stochastic information about demand leads to robust solutions that help to reduce operating costs, by decreasing understaffing an overstaffing levels.

The paper is organized as follows. In Section 2, we review the relevant literature on shift scheduling and tour scheduling problems with multiple work activities and stochastic demand. Then, we present some background material related to the use of grammars for multi-activity shift scheduling problems. In Section 3, we describe the two-stage model for the SMATSP. In Section 4, we introduce the solution approach to solve the problem. Computational experiments are presented and discussed in Section 5. The concluding remarks and future work are presented in Section 6.

2. Background Material

In this section, we review some literature on the models and methods to solve the MASSP and the MATSP. We also present some references on workforce problems under stochastic demand. Then, we finish with an introduction on the use of grammars for the MASSP.

2.1. Literature Review on Multi-Activity Shift and Tour Scheduling

Although mono-activity shift scheduling and tour scheduling problems have been extensively studied in the literature during the last few decades Alfares (2004); Ernst et al. (2004a,b); Van den Bergh et al. (2012), only recently some attention has been given to the problem that deals with multiple work activities. Ritzman et al. (1976) propose one of the first approaches to solve the MATSP. The method is based on a heuristic solution approach that integrates a construction method with a simulation component. Although employees are assigned to specific operations, breaks and rules related to switching between work activities are not considered. Heuristic approaches that use column generation (CG) (Restrepo et al., 2012) and tabu search (Dahmen and Rekik, 2015) are also proposed to solve multi-activity shift scheduling problems over multiple days. Even though both approaches tackle long time horizons, the constraints characterizing the feasibility of weekly tours (e.g., total tour length) are not included in the formulation of the problem. In a similar way, Detienne et al. (2009) and Lequy et al. (2012) solve a multi-activity assignment problem by using decomposition techniques and heuristics based on CG and branch-and-bound (B&B) as solution methods.

Fixing the sequences of work, rest days, shift types and breaks, might reduce the complexity of personnel scheduling problems, but it can also lead to sub-optimal solutions. Constraint programming (CP) techniques aim to solve that difficulty by offering modeling languages to handle complex optimization problems. Demassey et al. (2006) present a CP-based CG algorithm to model complex regulation constraints in a real-world MASSP. Quimper and Rousseau (2010) use formal languages to model the work rules related to the composition of shifts in a multi-activity context. Côté et al. (2011a) propose two approaches for the MASSP: the first one uses an automaton to derive a network flow model, while the second one takes advantage of context-free grammars to obtain a MIP model in which an and/or graph structure is used. Côté et al. (2011b) present an implicit grammar-based model for the MASSP that addresses symmetry issues by using general integer variables. Computational results show that, in the mono-activity case, the solution times of the model are comparable and sometimes superior to the results presented in the literature and that, in the multi-activity case, the model is able to solve to optimality instances with up to ten work activities. Côté et al. (2013) and Boyer et al. (2014) present grammar-based CG methods to solve the personalized MASSP and the personalized multi-activity multi-task shift scheduling problem, respectively. Both approaches use formal languages and dynamic programming to efficiently formulate and solve the pricing subproblems, but some limitations regarding long time horizons (e.g., one week) are present. To overcome these issues, Restrepo et al. (2016) and Restrepo et al. (2015) present approaches based on branch-and-price (B&P) and Benders decomposition (BD), respectively. In the former approach (Restrepo et al., 2016), two B&P algorithms are presented for the personalized MATSP. In the latter approach (Restrepo et al., 2015), a combined BD and CG method is introduced for the anonymous MATSP. In both approaches, the work rules for the composition of multi-activity shifts are expressed with context-free grammars, while some constraints that guarantee the feasibility of weekly tours are embedded into a directed acyclic graph.

2.2. Literature Review on Stochastic Shift and Tour Scheduling

Different models and solution approaches have been proposed in the literature to deal with stochastic demand in personnel scheduling problems. As an illustration, Easton and Rossin (1996) and Easton and Mansour (1999) develop heuristic methods that aim at tackling problems where demand is uncertain. Easton and Rossin (1996) propose a tabu search method to solve

a stochastic goal programming model that integrates and optimizes labor demand and employee scheduling. Easton and Mansour (1999) present a genetic algorithm to solve shift scheduling problems in which the recourse decisions are related to the undercovering and overcovering of demand. Although both approaches aim to solve problems over a one-week planning horizon, employee patterns are previously defined and only a small set of stochastic scenarios is considered. Bard et al. (2007) propose a heuristic two-stage model that addresses tour scheduling problems over a one-week planning horizon. First-stage variables are related to the number of full-time and part-time employees hired, while second-stage decisions correspond to the allocation of employees to specific shifts during the week. Computational experiments on real instances that consider three stochastic scenarios (high, medium and low demand) show that significant savings are likely when the recourse problem is used.

Some studies that use decomposition approaches have been recently proposed as alternatives to solve workforce planning problems when demand is uncertain. Pacqueau and Soumis (2014) propose a heuristic two-stage model to solve a shift scheduling problem. The proposed model is based on a decomposition of Aykin's (Aykin, 1996) implicit model, where first-stage variables are associated with the allocation of full-time shifts to the employees and recourse decisions correspond to hiring part-time employees, using overtime for full-time shifts, the allocation of breaks and the allowance of understaffing. Punnakitikashem et al. (2013) introduce a stochastic nurse scheduling problem that aims to minimize staffing costs and excess workload. The authors present a BD approach, a Lagrangian relaxation with a BD approach and a nested BD approach as solution methods. Computational results suggest that simultaneously considering nurse staffing and assignment is more desirable than doing them sequentially. Similarly, Kim and Mehrotra (2015) present an integrated staffing and scheduling approach applied to nurse management when demand is uncertain. The problem is formulated as a two-stage stochastic integer program, where daily shifts and weekly patterns are previously enumerated. First-stage decisions correspond to the number of employees assigned to daily shifts and to weekly patterns, while second-stage decisions correspond to: 1) the possibility of adding or canceling daily shifts for every working pattern; 2) allowing undercovering or overcovering of demand. A set of valid mixed-integer rounding inequalities that describe the convex hull of feasible solutions in the second-stage problem are included. Consequently, the integrality of the second-stage decision vari-

ables can be relaxed. Computational experiments show that the use of the stochastic model prevents the hospital from being overstaffed. An L-shaped method is presented in Robbins and Harrison (2010) to solve a combined server-sizing and staff scheduling problem for call centers in which a service level agreement must be satisfied. First-stage decisions correspond to the employee staffing, while second-stage decisions correspond to the computation of a telephone service shortfall. Computational results show that ignoring variability is a costly decision, since the value of the stochastic solution for the model is substantially high.

Stochastic workforce planning for employees who have various skills to work on different activities, tasks or unit departments, has recently started receiving attention in the operations research literature. Song and Huang (2008) consider the planning problem of transferring, hiring, or firing employees between different departments of an organization, where demands are uncertain and turnover is random. The problem is formulated as a multistage workforce capacity planning problem and solved with a successive convex approximation method. Zhu and Sherali (2009) address a workforce planning problem for employees with multiple skills between service centers. A two-stage model under demand fluctuations is presented, where first-stage decisions correspond to personnel recruiting and allocation of employees to multiple locations, while second-stage decisions consists in reassigning the workforce among the locations. The scheduling of cross-trained workers in a multi-department service environment with random demand is addressed in Campbell (2011). The author presents a two-stage model decomposable by days and by scenarios, where first-stage decisions are related to the scheduling of days-off and second-stage decisions correspond to the allocation of available employees at the beginning of each day. In the approach, days-off are fixed beforehand and only a small number of scenarios is considered (10 in total). Parisio and Jones (2015) present a two-stage stochastic model for a multi-skill tour scheduling problem in retail outlets, where first-stage variables are associated with the assignment of employees to weekly schedules, while recourse decisions correspond to the allocation of overtime and to the undercovering and overcovering of demand. Although multiple work activities are included in the problem, the authors assume employees are allowed to work in only one activity per shift. The reader is referred to De Bruecker et al. (2015) for a recent review on workforce planning incorporating skills. In this paper, the current literature is classified based on, apart from other criteria, the type of problem: deterministic or stochastic.

Even though some authors have tried to tackle personnel scheduling problems under stochastic demand, none of the previous studies consider the integration of days-off scheduling with shift scheduling in a multi-activity context. The method proposed in this paper addresses the discontinuous MATSP when demand is uncertain and employee skills are identical. Unlike the previous approaches, employee patterns and daily shifts are not previously fixed and a high degree of flexibility is included in their composition. Additionally, the multi-activity context is efficiently handled with context-free grammars, which are reviewed next.

2.3. Grammars for Multi-activity Shift Scheduling

In shift scheduling, a *context-free grammar* (CFG) can be defined as a finite set of work rules that are used to generate valid sequences of work (shifts) for a given day $d \in D$, where $|D|$ denotes the number of days in the planning horizon. A CFG consists of a tuple $G_d = (\Sigma_d, N_d, S_d, P_d)$, where:

- Σ_d represents an alphabet of characters called the *terminal symbols* for day d, which consists of work activities, breaks, lunch breaks, and rest stretches.
- N_d is a finite set of *non-terminal symbols* for day d.
- $S_d \in N_d$ is the *starting symbol* for day d.
- P_d is a set of *productions* for day d, represented as $A \rightarrow \alpha$, where $A \in N_d$ is a non-terminal symbol and α is a sequence of terminal and non-terminal symbols. The work rules used to generate shifts are represented by the set of productions. The productions of a grammar are used to generate new sequences of symbols, until all non-terminals have been replaced by terminal symbols. A sequence of terminal symbols is called a *word*, or in the shift scheduling context, a shift.

A *parse tree* is a tree where each inner-node is labeled with a non-terminal symbol N_d and each leaf is labeled with a terminal symbol Σ_d. A grammar recognizes a sequence if and only if there exists a parse tree where the leaves, when listed from left to right, reproduce the sequence.

A *DAG* Γ_d is a *directed acyclic graph* that embeds all parse trees associated with words (shifts) for day d of a given length n recognized by a grammar. The DAG Γ_d has an and/or structure where the and-nodes represent

productions (work rules) from P_d and the or-nodes represent non-terminals from N_d and letters from Σ_d. An and-node is true if all of its children are true. An or-node is true if one of its children is true. The root node is true if the grammar accepts the sequence encoded by the leaves. In Γ_d, O^{π}_{dil} denotes the or-node associated with $\pi \in N_d \cup \Sigma_d$, i.e., with non-terminals from N_d or letters from Σ_d, that generates a subsequence at position i of length l for day d. Note that if $\pi \in \Sigma_d$, the node is a leaf and l is equal to one. On the contrary, if $\pi \in N_d$, the node represents a non-terminal symbol and $l \geq 1$. $A^{\Pi,k}_{dil}$ is the kth and-node representing production $\Pi \in P_d$ that generates a subsequence from position i of length l at day d. There are as many $A^{\Pi,k}_{dil}$ nodes as there are ways of using P_d to generate a sequence of length l from position i. In Γ_d, the root node is described by O^{S}_{d1n} and its children by $A^{\Pi,k}_{d1n} \in ch(O^{S}_{d1n})$. These children generate sequences that contain a combination of rest and work stretches of length n, from position 1 at day d. Observe that n represents the length of the whole planning horizon during day d and not only the time spanned by the working shift. The children of a node (or-node or and-node) are represented by $ch()$ and its parents by $par()$. The sets of or-nodes, and-nodes and leaves in Γ_d are denoted by O_d, A_d and L_d, respectively. The DAG Γ_d is built by a procedure proposed in Quimper and Walsh (2007). The reader is referred to Côté et al. (2011b), for more details on the use of grammars in multi-activity shift scheduling.

Grammar G_1 presents an example on the use of context-free grammars for multi-activity shift scheduling. Two activities, w_1 and w_2, must be scheduled, shifts have a length of $n = 4$ time periods and should contain exactly one break, b, of one time period that can be placed anywhere during the shift except at the first or the last time period. Six different productions are defined: $W \rightarrow w_1$, $W \rightarrow w_2$ and $B \rightarrow b$ generate the terminal symbols associated with working on activity 1, working on activity 2, or having a break, respectively. Production $W \rightarrow WW$ generates two non-terminal symbols, W, meaning that the shift will include a working subsequence. Production $X \rightarrow WB$ means that the shift will include working time followed by a break. Finally, production $S \rightarrow XW$ generates a sequence of length four (the daily shift), which includes working time W, followed by a break B, to finish with more working time W. For clarity, we do not include the subscript of the day in the notation of grammar G_1 and nodes from Γ_1. The grammar that defines the set of feasible shifts on this example follows:

$G_1 = (\Sigma = (w_1, w_2, b), N = (S, X, W, B), S, P),$

where productions P are: $S \to XW$, $X \to WB$, $W \to WW|w_1|w_2$, $B \to b$, and symbol | specifies the choice of production.

Figure 1 represents the DAG Γ_1 associated with G_1. Observe that there are 16 parse trees (different shifts) embedded in Γ_1. As an illustration, we present a dotted-parse tree that generates shift $w_1 b w_1 w_2$, and a dashed-line parse tree that generates shift $w_2 w_2 b w_1$.

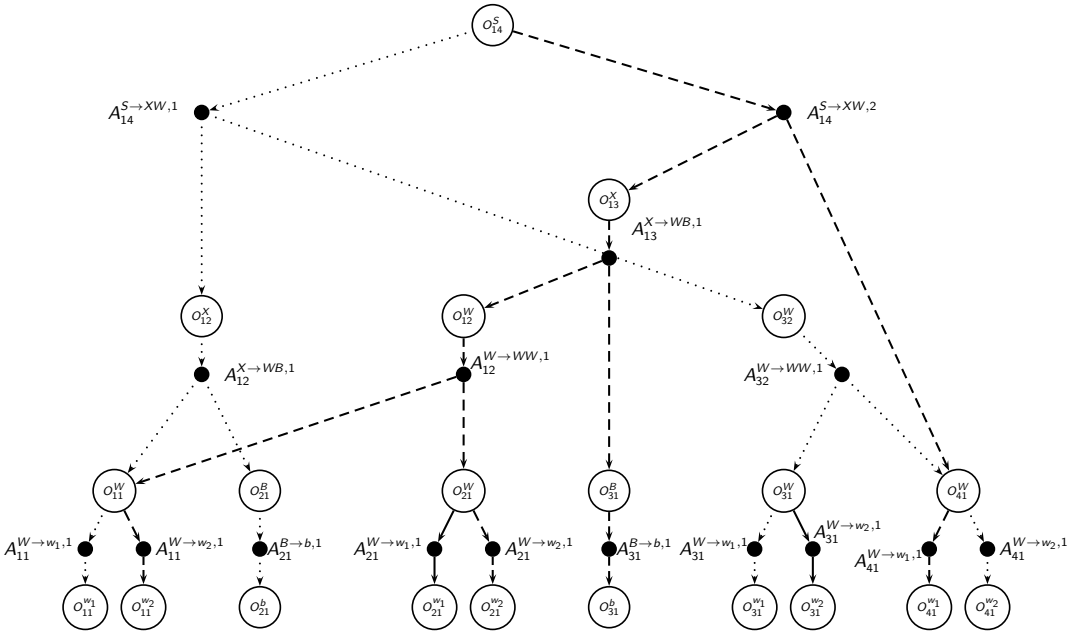

Figure 1: DAG Γ_1 on words of length four and two work activities.

Note that the children of the root node ($\{A_{14}^{S\to XW,1}, A_{14}^{S\to XW,2}\} \in ch(O_{14}^S)$) can be seen as shift "shells" because they do not consider the allocation of specific work activities to the shifts, only the shift starting time and its length including breaks. Hence, and-nodes $A_{d1n}^{\Pi,k}$ are characterized by their starting time $t_{d1n}^{\Pi,k}$, length including breaks $l_{d1n}^{\Pi,k}$ and working length $w_{d1n}^{\Pi,k}$. When the length of the time horizon (n) is longer than the maximum allowed shift length, $l_{d1n}^{\Pi,k} < n, \forall A_{d1n}^{\Pi,k} \in ch(O_{d1n}^S)$, the sequences generated by and-nodes $A_{d1n}^{\Pi,k}$ will include rest stretches. In Γ_1, and-node $A_{14}^{S\to XW,1}$ generates shift $wbww$, while and-node $A_{14}^{S\to XW,2}$ generates shift $wwbw$. Both shifts have a working length of three time periods, a total length of four time periods and

both start at time period one ($i = 1$).

Although the expressiveness of grammars allow to encode a large number of work rules for the composition of daily shifts, some limitations regarding shift total length are present when long planning horizons are included in the problem (e.g., one week). To circumvent this problem, Restrepo et al. (2015) present an approach that combines BD and CG to solve the deterministic discontinuous MATSP for employees with identical skills. The model combines an explicit definition of weekly tours with the implicit definition of daily shifts from Côté et al. (2011b). Since the model presents a nice block structure decomposable by days, it is used in the formulation of the two-stage stochastic problem, presented next.

3. Two-Stage Stochastic Problem

Stochastic shift and tour scheduling formulations extend and adapt deterministic models to allow schedule modifications at a time closer to the actual demand realization. Two-stage stochastic programming models give an example of such extensions. In these models, some decisions must be made in the first-stage before values of random variables are observed. Then, in the second-stage, a recourse action can be adopted after observing the actual values of the random variables to adjust any bad decision previously taken. In the model proposed, first-stage decisions correspond to the allocation of employees to each tour and to each daily shift shell, while second-stage decisions (recourse actions) correspond to the allocation of breaks and work activities to daily shifts and to the undercovering or overcovering of demand.

The second-stage problem is formulated with the implicit model proposed in Côté et al. (2011b). In this approach, the authors translate the logical clauses associated with $\Gamma_d, d \in D$, into linear constraints on integer variables, where the number of employees assigned to each and-node (A_d), each or-node (O_d) and each leaf (L_d) in Γ_d are represented by an integer variable.

In the first-stage problem, we define a feasible tour as the integration of daily shift shells (children of root nodes $O^S_{d1n}, d \in D$) and days-off, over the set of days in the planning horizon. Tours must meet the work rules related to the total working length, to the number of working days, to the rest time between consecutive shifts and to the allocation of days-off. Figure 2 presents an example of three tours built with the shifts shells from Γ_1. In this example, we assume that the DAG Γ_d for each day $d \in D$ is the same. Additionally, the planning horizon corresponds to seven days, the working length should

fall between 15 and 18 time periods, the number of working days must fall between 5 and 6, and there are no rules for the allocation of days-off and for the rest time between shifts. Finally, S_1 corresponds to $A_{d14}^{S\to XW,1} \to wbww$, S_2 corresponds to $A_{d14}^{S\to XW,2} \to wwbw$ and DO corresponds to a day-off.

Tours \ Days	1	2	3	4	5	6	7
1	S_1	DO	DO	S_1	S_2	S_1	S_2
2	S_1	S_1	S_2	S_2	S_2	DO	S_2
3	DO	S_1	S_2	S_2	DO	S_2	S_1

Figure 2: Weekly tours composed of shifts from Γ_1.

In defining a model for the discontinuous SMATSP, we assume that, the random vector ξ representing stochastic demands has a finite support. Henceforth, we define Ω as the set of its possible realizations and $p^{(w)} > 0$ as the probability of occurrence of scenario $w \in \Omega$ with $\sum_{w\in\Omega} p^{(w)} = 1$. The notation for the stochastic model follows.

Sets

J: set of work activities;

D: set of days in the planning horizon;

I_d: set of time periods at day $d \in D$;

E: set of employees;

$\mathcal{T}$: set of feasible tours.

First-stage problem

Decision variables

x_t: integer variable that represents the number of employees assigned to tour t;

$v_{d1n}^{\Pi,k}$: variable that represents the number of employees assigned to the kth child of the root node (shift shell) built with production Π from Γ_d.

Parameter

$\delta_{dt}^{\Pi,k}$: parameter that takes value 1, if tour t includes the kth shift shell from Γ_d at day d, and assumes value 0 otherwise.

Second-stage problem

Decision variables

$y_{dij}^{(w)}$: variable that denotes the number of employees assigned to activity j, at time period i, for day d under scenario w;

$v_{dil}^{\Pi,k,(w)}$: variable that denotes the number of employees assigned to the kth and-node, representing production Π from Γ_d and that generates a sequence from i of length $l < n$, under scenario w (this set of variables excludes the children of the root node O_{d1n}^{S});

$s_{dij}^{+(w)}$, $s_{dij}^{-(w)}$: slack variables representing overcovering and undercovering of demand of activity j, at time period i, for day d under scenario w, respectively.

Parameters

$\xi_{dij}^{(w)}$: stochastic demand for day d, time period i, and activity j for scenario w;

$\bar{\xi}_{dij}$: mean demand for day d, time period i, and activity j;

c_{dij}: nonnegative cost associated with one employee working on activity j, at time period i, at day d;

c_{dij}^{+}, c_{dij}^{-}: demand overcovering and undercovering nonnegative costs for day d, time period i, and activity j, respectively ($c_{dij}^{+} < c_{dij}^{-}$).

Given the above notation, the formulation for the two-stage stochastic problem, denoted $G_{\mathcal{T}}$, is as follows.

$$f(G_{\mathcal{T}}) = \min \quad \mathcal{Q}(\mathbf{v}) \tag{1}$$

$$v_{d1n}^{\Pi,k} = \sum_{t \in \mathcal{T}} \delta_{dt}^{\Pi,k} x_t, \quad \forall\, d \in D,\, A_{d1n}^{\Pi,k} \in ch(O_{d1n}^{S}), \tag{2}$$

$$\sum_{t \in \mathcal{T}} x_t = |E|, \tag{3}$$

$$x_t \geq 0 \text{ and integer}, \quad \forall\, t \in \mathcal{T}, \tag{4}$$

$$v_{d1n}^{\Pi,k} \geq 0, \quad \forall\, d \in D,\, A_{d1n}^{\Pi,k} \in ch(O_{d1n}^{S}). \tag{5}$$

The objective of $G_{\mathcal{T}}$, (1), is to minimize the expected recourse cost $\mathcal{Q}(\mathbf{v})$. Constraints (2) represent the link between daily shifts (children of root nodes O_{d1n}^{S} in $\Gamma_d, d \in D$) and tours. Since a fixed number of employees is given and all the employees have the same skills, constraint (3) guarantees that exactly $|E|$ employees are assigned to the tours. Finally, constraints (4)-(5) set the non-negativity and integrality of variables x_t and the non-negativity of variables $v_{d1n}^{\Pi,k}$.

The *expected recourse function* is denoted by $\mathcal{Q}(\mathbf{v}) \equiv \mathbb{E}_{\xi}[\mathcal{Q}(\mathbf{v}, \xi)]$. The *recourse function* $\mathcal{Q}(\mathbf{v}, \xi(w))$, for a given realization w of ξ, is represented by:

$$\mathcal{Q}(\mathbf{v}, \xi(w)) = \min \sum_{d \in D} \sum_{i \in I_d} \sum_{j \in J} c_{dij} y_{dij}^{(w)} + \sum_{d \in D} \sum_{i \in I_d} \sum_{j \in J} (c_{dij}^{+} s_{dij}^{+(w)} + c_{dij}^{-} s_{dij}^{-(w)}) \tag{6}$$

$$y_{dij}^{(w)} - s_{dij}^{+(w)} + s_{dij}^{-(w)} = \xi_{dij}^{(w)}, \quad \forall d \in D, i \in I_d, j \in J, \tag{7}$$

$$\sum_{A_{dil}^{\Pi,k} \in ch(O_{dil}^{\pi})} v_{dil}^{\Pi,k,(w)} = \sum_{A_{dil}^{\Pi,k} \in par(O_{dil}^{\pi})} v_{d1n}^{\Pi,k},$$
$$\forall d \in D, O_{dil}^{\pi} \in ch(A_{d1n}^{\Pi,k}) \setminus L_d, \tag{8}$$

$$\sum_{A_{dil}^{\Pi,t} \in ch(O_{dil}^{\pi})} v_{dil}^{\Pi,k,(w)} = \sum_{A_{dil}^{\Pi,k} \in par(O_{dil}^{\pi})} v_{dil}^{\Pi,k,(w)},$$
$$\forall d \in D, O_{dil}^{\pi} \in O_d \setminus \{O_{d1n}^{S} \cup L_d \cup ch(A_{d1n}^{\Pi,k})\}, \tag{9}$$

$$y_{dij}^{(w)} = \sum_{A_{di1}^{\Pi,k} \in par(O_{di1}^{j})} v_{di1}^{\Pi,1,(w)}, \quad \forall d \in D, i \in I_d, j \in J, \tag{10}$$

$$v_{dil}^{\Pi,k,(w)} \geq 0, \quad \forall\, d \in D, A_{dil}^{\Pi,k} \in A_d \setminus ch(O_{d1n}^{S}), \tag{11}$$

$$s_{dij}^{+(w)}, s_{dij}^{-(w)} \geq 0, \quad \forall d \in D, i \in I_d, j \in J, \tag{12}$$

$$y_{dij}^{(w)} \geq 0 \text{ and integer}, \quad \forall d \in D, i \in I_d, j \in J. \tag{13}$$

Problem (6)-(13) is based on the implicit model presented in Côté et al. (2011b). The objective, (6), is to assign work activities to daily shifts in order to minimize the staffing cost plus the undercovering and overcovering of demand. Constraints (7) ensure that the total number of employees working on day $d \in D$, time period $i \in I_d$ and work activity $j \in J$ is equal to the particular demand realization w subject to some adjustments related to undercovering and undercovering. Due to the structure of Γ_d, $d \in D$, constraints (8)-(9) guarantee for every or-node O_{dil}^{π}, excluding the root node O_{d1n}^{S} and the leaves L_d, that the number of employees allocated to its children, $ch(O_{dil}^{\pi})$, is the same as the summation of the employees allocated to its its parents, $par(O_{dil}^{\pi})$. Constraints (10) represent the allocation of employees to work activity j, at time period i for a given day d and scenario w. Constraints (11)-(13) set he non-negativity of variables $v_{dil}^{\Pi,k,(w)}$, $s_{dij}^{+(w)}$, $s_{dij}^{-(w)}$ and the non-

negativity and integrality of variables $y_{dij}^{(w)}$.

Constraints (8)-(9) can be seen as flow conservation equations where or-nodes O_{dil}^{π} represent "transition nodes". The constraints for those transition nodes guarantee that if e employees are allocated to the productions generating the subsequence associated with node O_{dil}^{π}, those e employees have to be distributed along all the possible ways to use π to generate a sequence of length l from position i ($ch(O_{dil}^{\pi})$). Consider the following example using the DAG Γ_1 from Figure 1. Assume that three employees are assigned to variable $v_{12}^{X\to WB,1}$ representing and-node $A_{12}^{X\to WB,1}$ and that one employee is assigned to variable $v_{12}^{W\to WW,1}$, representing and-node $A_{12}^{W\to WW,1}$. Since these two and-nodes have one child in common, or-node O_{11}^{W}, the number of employees allocated to O_{11}^{W} is four. Now, since or-node O_{11}^{W} has two children $A_{11}^{W\to w_1,1}$, $A_{11}^{W\to w_2,1}$ the four employees have to be distributed among variables $v_{11}^{W\to w_1,1}, v_{11}^{W\to w_2,1} \in ch(O_{11}^{W})$.

Observe that the two-stage stochastic problem (1)-(5) has *complete recourse* because for any realization of the random vector ξ and value of variables $v_{d1n}^{\Pi,k}$, problem (6)-(13) (*second-stage problem*) is always feasible due to the allowance of undercovering and overcovering of demand. Additionally, note that since we assumed that at the moment we can act on second-stage variables the scenario for day $d \in D$ is fully known and since problem $\mathcal{Q}(\mathbf{v}, \xi(w))$ is decomposable by days due to its particular block structure, $\mathbf{v}$, Ω and $p^{(w)} > 0$ can also be decomposed by days: $\mathbf{v}_d$, Ω_d, $p_d^{(w)} > 0$, $d \in D$ and the expected recourse function $\mathcal{Q}(\mathbf{v})$ can be represented as:

$$\mathcal{Q}(\mathbf{v}) \equiv \mathbb{E}_{\xi}[\sum_{d\in D} \mathcal{Q}(\mathbf{v}_d, \xi_d)] \equiv \sum_{d\in D} \mathbb{E}_{\xi}[\mathcal{Q}(\mathbf{v}_d, \xi_d)] \tag{14}$$

In the following, we present the solution method proposed to solve the SMATSP.

4. Heuristic Multi-cut L-shaped Method

The L-shaped method was first proposed in Van Slyke and Wets (1969) for solving two-stage stochastic linear problems. The basic idea behind this method is to approximate the nonlinear term, $\mathcal{Q}(\mathbf{v})$, in the objective function of the two-stage stochastic problem (1)-(5). In particular, since the expected recourse function involves solving all second-stage recourse problems, the

main principle of the L-shaped method is to avoid numerous function evaluations by using an outer linearization of $\mathcal{Q}(\mathbf{v})$, as in BD. In the case of complete recourse, the L-shaped method and BD reduce to the same algorithm, as it is not longer required to take care of feasibility issues. Since ξ_d follows a discrete distribution with a finite support, with Ω_d as the set of its possible realizations for each day $d \in D$, and $p_d^{(w)} > 0$ as the probability of occurrence of scenario $w \in \Omega_d$ ($\sum_{d \in D} \sum_{w \in \Omega_d} p_d^{(w)} = 1$), $\mathcal{Q}(\mathbf{v})$ can be expressed as $\mathcal{Q}(\mathbf{v}) = \sum_{d \in D} \sum_{w \in \Omega_d} p_d^{(w)} \mathcal{Q}(\mathbf{v}_d, \xi_d(w))$. By defining $\theta_d^{(w)}$ as an additional set of free variables, the two-stage stochastic problem $G_{\mathcal{T}}$ can be reformulated as the following model, denoted as $B_{\mathcal{T}}$.

$$f(B_{\mathcal{T}}) = \min \sum_{d \in D} \sum_{w \in \Omega_d} \theta_d^{(w)} \tag{15}$$

$$v_{d1n}^{\Pi,k} = \sum_{t \in \mathcal{T}} \delta_{dt}^{\Pi,k} x_t, \quad \forall\, d \in D,\ A_{d1n}^{\Pi,k} \in ch(O_{d1n}^S), \tag{16}$$

$$\theta_d^{(w)} \geq p_d^{(w)} \mathcal{Q}(\mathbf{v}_d, \xi_d(w)), \quad \forall\, d \in D,\ w \in \Omega_d, \tag{17}$$

$$\sum_{t \in \mathcal{T}} x_t = |E|, \tag{18}$$

$$x_t \geq 0 \text{ and integer}, \quad \forall\, t \in \mathcal{T}, \tag{19}$$

$$v_{d1n}^{\Pi,k} \geq 0, \quad \forall\, d \in D,\ A_{d1n}^{\Pi,k} \in ch(O_{d1n}^S). \tag{20}$$

Optimality cuts (17) ensure that the value of each variable $\theta_d^{(w)}$ is larger than or equal to the optimal value of its corresponding second-stage problem for each day $d \in D$ and each scenario $w \in \Omega_d$. Observe that the structure of problem (15)-(20) allows the L-shaped method to be extended to include multiple cuts at each iteration, i.e., one per day and per scenario, instead of adding one aggregated cut. Birge and Louveaux (1988) showed that in an iterative algorithm, adding multiple cuts at the same iteration may speed up convergence and reduce the number of iterations.

Since the second-stage problems (6)-(13) are MILP models that do not possess the integrality property, we relax integrality constraints (13) on variables $y_{dij}^{(w)}$ because 1) the dual to the LP relaxation of each second-stage problem will produce a cut that forces $\theta_d^{(w)}$ to be at least as great as the objective value of the relaxation, which is a valid lower bound for the actual recourse function value; 2) Restrepo et al. (2015) showed that, in practice,

problems (6)-(13) do not exhibit a large integrality gap and that optimal or near-optimal solutions can be found by solving the LP relaxation of the second-stage problems.

Let $\overline{\mathcal{Q}}(\mathbf{v}_d, \xi_d(w))$ denote the LP relaxation of problem $\mathcal{Q}(\mathbf{v}_d, \xi_d(w))$. Let $\rho_{dij}^{(w)}$, $\gamma_{dil}^{\pi,(w)}$ be the dual variables associated with constraints (7) and (8) from $\overline{\mathcal{Q}}(\mathbf{v}_d, \xi_d(w))$, respectively. Let $\mathbf{\Delta}_d^{(w)}$ be the projection over the space of variables $\rho_{dij}^{(w)}$, $\gamma_{dil}^{\pi,(w)}$ of the polyhedron defined by the constraints associated with the dual of model $\overline{\mathcal{Q}}(\mathbf{v}_d, \xi_d(w))$. Note that $\mathbf{\Delta}_d^{(w)}$ is itself a polyhedron (Wolsey and Nemhauser, 1999). Let $E_{\mathbf{\Delta}_d^{(w)}}$ be the set of extreme points of $\mathbf{\Delta}_d^{(w)}$. Inequalities (17) in model $B_{\mathcal{T}}$ are replaced by the following ones, defining formulation $B_{\mathcal{T}}^{'}$:

$$\theta_d^{(w)} \geq p_d^{(w)} \Bigg(\sum_{i \in I_d} \sum_{j \in J} \xi_{dij}^{(w)} \rho_{dij}^{(w)} + \sum_{O_{dil}^{\pi} \in ch(A_{d1n}^{\Pi,k}) \setminus L_d} \gamma_{dil}^{\pi,(w)} \sum_{A_{d1n}^{\Pi,k} \in par(O_{dil}^{\pi})} v_{d1n}^{\Pi,k} \Bigg),$$
$$\forall\, d \in D,\, w \in \Omega_d,\, (\rho_d, \gamma_d) \in E_{\mathbf{\Delta}_d^{(w)}}. \qquad (21)$$

Since these new optimality cuts are using linear approximations of $\mathcal{Q}(\mathbf{v}_d, \xi_d(w))$, model $B_{\mathcal{T}}^{'}$ is a MILP relaxation of $B_{\mathcal{T}}$ i.e., $f(B_{\mathcal{T}}) \geq f(B_{\mathcal{T}}^{'})$. Optimality cuts (21) do not need to be exhaustively generated, since only a subset of them are active in the optimal solution of the problem. Hence, an iterative algorithm can be used to generate only the subset of cuts that will represent the optimal solution.

The algorithm consists in a multi-cut version of the L-shaped method where, at each iteration $m \geq 1$, a relaxation of the first-stage problem is solved. Such relaxation is obtained by replacing the set of extreme points at each day $d \in D$ and each scenario $w \in \Omega_d$, by subsets $E^m_{\mathbf{\Delta}_d^{(w)}} \subseteq E_{\mathbf{\Delta}_d^{(w)}}$. Note that in the first-stage model, $B_{\mathcal{T}}^{'}$, it is assumed that the complete set of tours $\mathcal{T}$ is known. However, with the incorporation of shift and tour flexibility, the complete enumeration of the set of feasible tours might be intractable. To address this issue, we propose a heuristic CG approach in which a master problem $B_{\widetilde{\mathcal{T}}}^{LP}$, is defined as the LP relaxation of $B_{\mathcal{T}}^{'}$ over a restricted set of tours $\widetilde{\mathcal{T}} \subseteq \mathcal{T}$. We also define the MILP associated with $B_{\widetilde{\mathcal{T}}}^{LP}$ as $B_{\widetilde{\mathcal{T}}}^{MILP}$, where for a given subset of columns $\widetilde{\mathcal{T}} \subseteq \mathcal{T}$, the integrality constraints on x_t variables are imposed to obtain a heuristic integer solution

(i.e., $B^{MILP}_{\widetilde{\mathcal{T}}}$ is solved by a state-of-the-art B&B method, using only the columns corresponding to $\widetilde{\mathcal{T}}$). The algorithm for the L-shaped method is then divided into two parts: multi-cut generation and CG.

Two algorithm enhancements were implemented to speed-up the convergence of the multi-cut L-shaped method. First, we adopted the strategy proposed in McDaniel and Devine (1977), which consists in initially solving the LP relaxation of the first-stage problem to generate, in a fast way, a number of valid cuts. Then, when some criterion is met (i.e., the relative gap between the upper bound and the lower bound of the problem is smaller than a certain value), the method then solves the MILP of the first-stage problem. Second, we implemented the method presented in Papadakos (2008) for the generation of strong optimality cuts (the so-called Algorithm 3). The author proposes an alternative to eliminate the necessity of solving the extra auxiliary subproblem introduced in Magnanti and Wong (1981). Additionally, since finding a *core point* for the problem is a difficult task, the author suggests to use an *approximate core point* consisting of a convex linear combination of the previously generated core point and the current solution for the first-stage problem. The use of this approximate core point is justified by Corollary 9 in Papadakos (2008). Since the conditions of this Corollary hold for our problem, we will use this approximation in our algorithm.

Let ϵ_1 be the tolerance that defines if an optimality cut is added or not to the first-stage problem and let ϵ_2 be the tolerance we used to stop solving $B^{LP}_{\widetilde{\mathcal{T}}}$, i.e., stop the McDaniel and Devine (1977) strategy. Let Int be a Boolean control variable for the McDaniel and Devine strategy, that indicates whether the MILP ($B^{MILP}_{\widetilde{\mathcal{T}}}$) of the first-stage problem is solved (Int=true) or not (Int=false). Let $\theta^{(w)LB}_{dm}$ denote the optimal value of variables $\theta^{(w)}_{d}$ from $B^{LP}_{\widetilde{\mathcal{T}}}$, at iteration m. Note that $\theta^{LB}_{m} = \sum_{d\in D}\sum_{w\in\Omega_d}\theta^{(w)LB}_{dm}$ denotes a lower bound on $f(G_{\widetilde{\mathcal{T}}})$ at iteration m. Since we are using a heuristic approach to find integer solutions for the first-stage model, we also denote $\theta^{(w)UB}_{dm}$ as the optimal values of variables $\theta^{(w)}_{d}$ when $B^{MILP}_{\widetilde{\mathcal{T}}}$ is solved at iteration m. To simplify the algorithm description, we use the same notation when solving $B^{LP}_{\widetilde{\mathcal{T}}}$, even though in that case, we have $\theta^{(w)LB}_{dl} = \theta^{(w)UB}_{dm}$ for each $d \in D, w \in \Omega_d$, since the CG algorithm is performed until all columns have non-negative reduced costs. Let $q^{(w)UB}_{dm}$, ${q_{dm}}^{(w)LB}$ be the optimal value of second-stage problem (6)-(13) for day d, under scenario w at iteration m, when integrality constraints on variables $y^{(w)}_{dij}$ are imposed and relaxed, respectively. Note that

$Q_m^{UB} = \sum_{d\in D}\sum_{w\in\Omega_d} p_d^{(w)} q_{dm}^{(w)UB}$ is an upper bound on $f(G_{\mathcal{T}})$ at iteration m. Similarly, $\widetilde{Q}_m^{UB} = \sum_{d\in D}\sum_{w\in\Omega_d} p_d^{(w)} q_{dm}^{(w)LB}$ is an upper bound on $f(B'_{\mathcal{T}}) \leq f(G_{\mathcal{T}})$ at iteration m, which we call the approximate upper bound. Let $v_{d1nm}^{\Pi,k\ *}$, $v_{d1nm}^{\Pi,k\ 0}$ denote an optimal solution and the core point approximation, respectively of first-stage variables $v_{d1n}^{\Pi,k}$ from model $B'_{\mathcal{T}}$ at iteration m. The flow diagram of the algorithm is presented in Figure 3. The description of the multi-cut L-shaped method follows.

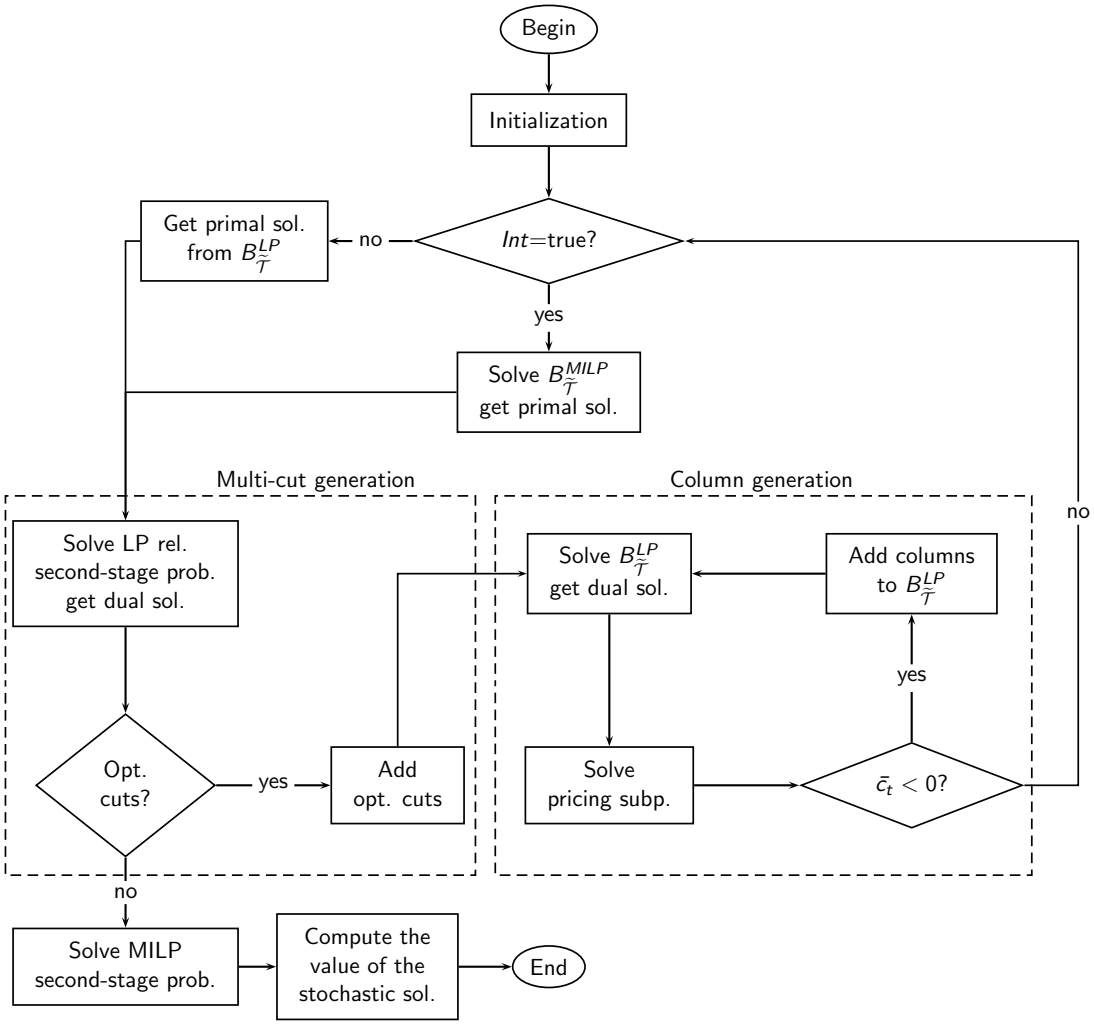

Figure 3: Flow chart for the multi-cut L-shaped method.

- *Initialization*: The multi-cut L-shaped algorithm starts with an empty set of optimality cuts, $E^m_{\boldsymbol{\Delta}_d^{(w)}} = \varnothing$, $d \in D, w \in \Omega_d$. In this step, the number of iterations m is set to zero, the lower and upper bounds of the problem are initialized as $Q^{UB}_m = \infty, \widetilde{Q}^{UB}_m = \infty, \theta^{LB}_m = -\infty$, the Boolean variable Int is set to false, and an initial set of columns, generated with the procedure shown in the *Column generation* step, is added to $B^{LP}_{\widetilde{\mathcal{T}}}$.

- *Int=true?*: In this step of the algorithm, we verify if the Boolean variable Int is true or false. If $Int = false$, we continue with the step *Get primal sol. from* $B^{LP}_{\widetilde{\mathcal{T}}}$. If $Int = true$ we continue with the step *Solve* $B^{MILP}_{\widetilde{\mathcal{T}}}$ *get primal sol.* The value of Int is changed from false to true when $(\widetilde{Q}^{UB}_m - \theta^{LB}_m)/\widetilde{Q}^{UB}_m < \epsilon_2$ and $Int = false$.

- *Get primal sol. from* $B^{LP}_{\widetilde{\mathcal{T}}}$: In this step of the algorithm, we get the primal solution ${v^{\Pi,k}_{d1nm}}^*$, $\theta^{(w)LB}_{dm}$ from $B^{LP}_{\widetilde{\mathcal{T}}}$. Then, we calculate the approximation of the core point as: ${v^{\Pi,k}_{d1nm}}^0 = \frac{1}{2}{v^{\Pi,k}_{d1n\,m-1}}^0 + \frac{1}{2}{v^{\Pi,k}_{d1nm}}^*$, $\forall\, d \in D$, $A^{\Pi,k}_{d1n} \in ch(O^S_{d1n})$ and we update the lower bound of the problem as: $\theta^{LB}_m = \sum_{d\in D}\sum_{w\in\Omega_d}\theta^{(w)LB}_{dm}$. The values of ${v^{\Pi,k}_{d1nm}}^*$, ${v^{\Pi,k}_{d1nm}}^0$ are sent to the second-stage problems. The approximate core point ${v^{\Pi,k}_{d1nm}}^0$ is initially set equal to the solution of the LP relaxation of the first-stage problem.

- *Solve* $B^{MILP}_{\widetilde{\mathcal{T}}}$ *get primal sol.*: In this step of the algorithm, we get the primal solution $\theta^{(w)LB}_{dm}$ from $B^{LP}_{\widetilde{\mathcal{T}}}$ to update the lower bound of the problem as: $\theta^{LB}_m = \sum_{d\in D}\sum_{w\in\Omega_d}\theta^{(w)LB}_{dm}$. Then we solve $B^{MILP}_{\widetilde{\mathcal{T}}}$ to get the primal solution $\theta^{(w)UB}_{dm}, {v^{\Pi,k}_{d1nm}}^*$ and to calculate the approximation of the core point as: ${v^{\Pi,k}_{d1nm}}^0 = \frac{1}{2}{v^{\Pi,k}_{d1n\,m-1}}^0 + \frac{1}{2}{v^{\Pi,k}_{d1nm}}^*$, $\forall\, d \in D$, $A^{\Pi,k}_{d1n} \in ch(O^S_{d1n})$. The values of ${v^{\Pi,k}_{d1nm}}^*$, ${v^{\Pi,k}_{d1nm}}^0$ are sent to the second-stage problems.

- *Multi-cut generation*: The objective of the multi-cut step is to generate optimality cuts (21) in order to approximate the recourse function $\mathcal{Q}(\mathbf{v}, \xi)$. The procedure to generate and to add new optimality cuts to the first-stage problem follows.

– *Solve LP rel. second-stage prob.*: In this step of the algorithm, we solve the LP relaxation of the second-stage problems twice. First, we fix variables $v_{d1n}^{\Pi,k}$ with the value of core point $v_{d1nm}^{\Pi,k\ 0}$ to get a dual solution $(\rho_d, \gamma_d)^0$. Second, we fix variables $v_{d1n}^{\Pi,k}$ with the value of point $v_{d1nm}^{\Pi,k\ *}$ to update the approximate upper bound of the problem as: $\widetilde{Q}_m^{UB} = \min\{\widetilde{Q}_m^{UB}, \sum_{d\in D}\sum_{w\in\Omega_d} p_d^{(w)} q_{dm}^{(w)LB}\}$.

– *Opt. cuts?* and *add opt. cuts*: In order to add optimality cuts to the first-stage problem, we verify if $(q_{dm}^{(w)LB} - \theta_{dm}^{(w)UB})/\ q_{dm}^{(w)LB} > \epsilon_1, d \in D, w \in \Omega_d$, in which case a new optimality cut (the one generated with the approximate core point) is added to the problem. After adding the optimality cuts, we increase by one the number of iterations m.

- *Column generation*: The CG method consists of a master problem $B_{\widetilde{\mathcal{T}}}^{LP}$ and a pricing subproblem. The former problem, as mentioned before, is the LP relaxation of model $B_{\mathcal{T}}^{'}$ over a reduced set of tours $\widetilde{\mathcal{T}} \subseteq \mathcal{T}$. The latter problem is responsible for finding tours with negative reduced cost that will be added to $B_{\widetilde{\mathcal{T}}}^{LP}$ in an iterative way.

  Let $\lambda_{d1n}^{\Pi,k}$ and δ be the dual variables associated with the constraints (16) and (18) from $B_{\widetilde{\mathcal{T}}}^{LP}$, respectively. Let $\mathcal{S} = \bigcup_{d\in D} ch(O_{d1n}^S)$ be a set of shift shells, defined as the union, over the set of days in the planning horizon, of all the children of root nodes $O_{d1n}^S, d \in D$. Let $G(\mathcal{N}, \mathcal{A})$ be a directed acyclic graph, composed of a set of nodes $\mathcal{N} = \{n_s \mid s \in \mathcal{S}\} \cup \{n_b, n_e\}$, where n_s corresponds to shift s and n_b, n_e are the source and sink nodes, respectively. Each shift $s \in \mathcal{S}$ holds, besides a set of attributes inherited from its corresponding and-node (start time period, working time, and length including breaks), a "reduced cost contribution" corresponding to the value of dual variable $\lambda_{d1n}^{\Pi,k}$. The set of arcs $\mathcal{A}$ represents the connection between nodes depending on the work rules for the allocation of days-off and rest time between consecutive shifts.

  New columns for $B_{\widetilde{\mathcal{T}}}^{LP}$ correspond to resource-constrained shortest paths over $G(\mathcal{N}, \mathcal{A})$. More specifically, each feasible tour $t \in \widetilde{\mathcal{T}}$ must meet the work rules related to the minimum and maximum number of working days in a tour, to the minimum and maximum tour working length in time periods, to the maximum number of days-off, and to the

minimum rest time between two consecutive daily shifts. Additionally, the reduced cost $\bar{c}_t$ of tour t is given by

$$\bar{c}_t = \left(\sum_{d \in D} \sum_{A^{\Pi,k}_{d1n} \in ch(O^S_{d1n})} \lambda^{\Pi,k}_{d1n} \delta^{\Pi,k}_{dt} \right) - \sigma. \tag{22}$$

The procedure to generate and add new columns for $B^{LP}_{\widetilde{\mathcal{T}}}$ is as follows:

- *Solve $B^{LP}_{\widetilde{\mathcal{T}}}$, get dual sol.*: In this step of the algorithm, we solve problem $B^{LP}_{\widetilde{\mathcal{T}}}$ to get the dual solution (λ, σ) that will be sent to the pricing subproblem.
- *Solve pricing subp.*: New variables (tours) for the $B^{LP}_{\widetilde{\mathcal{T}}}$ are generated by using a label setting algorithm for the resource-constrained shortest-path problem over graph $G(\mathcal{N}, \mathcal{A})$. In the algorithm, the total length of the tour and the number of working days represent global resources that are consumed by the labels while they are extended.
- $\bar{c}_t < 0$? and *add columns to $B^{LP}_{\widetilde{\mathcal{T}}}$*: In this step of the algorithm we evaluate if negative reduced cost columns were found by the pricing subproblem. If yes, such columns are sent to $B^{LP}_{\widetilde{\mathcal{T}}}$ which is re-optimized to start a new iteration.

- *Solve MILP second-stage prob.*: In this step of the algorithm, we solve the MILP of the second-stage problems when $v^{\Pi,k}_{d1n}$ variables are fixed with the value of $v^{\Pi,k}_{d1nm}{}^{*}$. In this step, we also compute the value of the upper bound as $Q^{UB}_m = \sum_{w \in \Omega} \sum_{d \in D} p^{(w)}_d q^{(w)UB}_{dm}$ and the gap with respect to the approximate upper bound: $100 \times (Q^{UB}_m - \widetilde{Q}^{UB}_m)/Q^{UB}_m$. This gap helps to measure the quality of the solution obtained when the integrality constraints on second-stage variables $y^{(w)}_{dij}$ are relaxed.

- *Compute the value of the stochastic sol.*: The multi-cut L-shaped algorithm ends with the computation of the *value of the stochastic solution* (VSS). The VSS is a standard measure that indicates the expected gain from solving a stochastic model rather than its deterministic counterpart, the *mean value problem* (EV). The value of the stochastic solution

is defined as $VSS = EEV - HN$, where HN corresponds to the optimal value of problem (1)-(5) and EEV corresponds to the expected value of using the EV solution. EV is simply problem (1)-(5) evaluated using the mean value for demands, $\bar{\xi}_d$, for each day $d \in D$. Given an EV solution $\bar{v}_d^*$, EEV corresponds to: $EEV = \sum_{d \in D} \sum_{w \in \Omega_d} p_d^{(w)} \mathcal{Q}\left(\bar{v}_d^*, \xi_d(w)\right)$. A large VSS means that uncertainty is important for the quality of the resulting optimal solution. On the contrary, a small VSS means that a deterministic approach based on the expected values of the random variables might be sufficiently good to take a decision. Recall that, since second-stage problems are MILP models that do not possess the integrality property, the final (heuristic) solution of the two-stage stochastic problem might not be optimal and $HN = Q_m^{UB} \geq \widetilde{Q}_m^{UB}$.

5. Computational Experiments

In this section, we test the proposed multi-cut L-shaped method on real and randomly generated instances of the SMATSP. First, we describe the generation and the characteristics of the set of instances used. Second, we present the problem definition and the grammar built for the composition of daily shifts. Third, we report and analyze the computational results.

The computational experiments were performed on a 64-bit GNU/Linux operating system, 96 GB of RAM and 1 processor Intel Xeon X5675 running at 3.07GHz. The multi-cut L-shaped algorithm was implemented in C++. The LP relaxation of both the first-stage problem and the second-stage problems was solved by using the barrier method of CPLEX version 12.5.0.1. The time limit to solve each instance was set as two hours for instances with one and two work activities and to three hours for the rest of instances. A relative gap tolerance of 0.01 was set as a stopping criterion for solving the MILPs with CPLEX. The maximum number columns sent to the master problem at each CG iteration was 30. The value of tolerances ϵ_1, ϵ_2 were set to 0.0001 and 0.02, respectively.

5.1. Instances Generation

The set of instances used to test our method is divided into two groups: randomly generated instances and instances based on the demand pattern from a small retail shop. The deterministic demand patterns for the real-world instances are presented in Côté et al. (2011b). These demand profiles

present a constant (uniform) demand across hours (level shape) and are labeled as level/level. The deterministic demand patterns for the set of random instances were generated such that they include three dimensions: *mean demand* (μ), *amplitude of demand* (σ) and *shape of demand*. Mean demand is the average number of employees required across all periods of the day; amplitude of demand is defined as the maximum difference in employee requirements. The mean demand ranges between $\mu = [2, 11]$, the amplitude of demand ranges between $\sigma = [2, 21]$, and different daily/weekly shapes were generated. For each day in the planning horizon we generated unimodal (unim.) and bimodal (bimod.) shapes. For the week, we generated unimodal and level shapes. Hence the daily/weekly demand patterns created are: unim./level, unim./unim., bim./level and bim./unim. Bechtold and Showalter (1987) noted that, depending on the nature of the service sector, a wide variety of demand shapes can be encountered, and that in a tour scheduling environment, demand can assume shapes across both the day and the week. For instance, daily-unimodal/weekly-unimodal, is representative of service sectors that have a single peak of demand during each day and a single peak during the week. Restaurants that focus on the evening meal, with Friday and Saturday as the principal days for demand, would expect to see a pattern of this nature. For the random instances operations start between 6am and 8am and finish at 10pm. For the real-world instances, operations start between 8am and 11am and finish between 4pm and 9pm. We generated instances with up to five work activities and ten different demands for each work activity and daily/weekly shape combination; therefore, the resulting number of instances for the deterministic demand patterns is $5 * 10 * 5 = 250$. We tested instances with one work activity because the stochastic model is still relevant in this case, as the allocation of breaks to shifts is included in the set of recourse actions.

The generation of the deterministic demand patterns described above allows us to determine the value of the mean demand, $\bar{\xi}_{dij}$, for each work activity, time period, and day of the planning horizon. We assume that the demand for each work activity at each day, follows a non-homogeneous Poisson process (NHPP) with mean $\bar{\xi}_{dij}$ that changes at each time period i, at each day d. We also assume that such demand can be translated into a minimum employee requirements by dividing it by a constant service rate. Service systems such as hospital emergency rooms, call centers and banks normally have strong time-varying arrival rates. Thus, a NHPP is a natural way to model these arrival processes. Since the probability distribution of

demand leads to an exponential growth of the numbers of scenarios, we used Monte Carlo-sampling to generate $|\Omega_d^{'}|$ equiprobable observations, at each day, of the random demand and then solve the stochastic program obtained by replacing the original distribution with the sampled one. We set $|\Omega_d^{'}| = 100$. The NHPP was simulated with the thinning algorithm presented in Lewis and Shedler (1979).

5.2. Problem Definition and Grammar

The work rules for shift and tour generation, as well as the grammar used in the problem are as follows. The work rules for shift generation are based on a real-world timetabling problem, presented in Demassey et al. (2006).

Tour generation

1. The planning horizon is seven days, where each day is divided into 96 time periods of 15 minutes.
2. Shifts are not allowed to span from one day to another (discontinuous problem).
3. The tour working length should fall between 35 and 40 hours per week.
4. The number of working days in the tour should fall between five and six.
5. There must be a minimum rest time of twelve hours between consecutive shifts.

Daily shift generation

1. Shifts can start at any time period during any day d, allowing enough time to complete their duration in day d.
2. Three types of shifts are considered: 8-hour shifts with 1-hour lunch break in the middle and two 15-minute breaks. 6-hour shifts with one 15-minute break and no lunch, and 4-hour shifts with one 15-minute break and no lunch.
3. If performed, the duration of a work activity is at least two hours and at most five hours.
4. A break (or lunch) is necessary between two different work activities.
5. Work activities must be inserted between breaks, lunch and rest stretches.
6. A fixed number of employees $|E|$ is given, therefore undercovering and overcovering of demand is allowed.

Two different scenarios for the allocation, undercovering and overcovering costs were tested. In the first scenario, activity allocation costs c_{dij} are equal to 15, overcovering costs c_{dij}^{+} are equal to 20, and undercovering costs

are computed as: $c^{-}_{dij} = c_{dij} * 4 = 60$. In the second scenario, activity allocation costs c_{dij} are equal to 15, overcovering costs c^{+}_{dij} are equal to 0, and undercovering costs are computed as: $c^{-}_{dij} = c_{dij} * 6 = 90$.

Let a_j be a terminal symbol that defines a time period of work activity $j \in J$. Let b, l and r be the terminal symbols that represent break, lunch and rest periods, respectively. In productions $\Pi \in P$, $\Pi \rightarrow_{[\min,\ \max]}$ restricts the subsequences generated by a given production to a length between a minimum and maximum number of time periods. The grammar and the productions that define the multi-activity shifts are as follows:

$$
\begin{aligned}
G =& (\Sigma = (a_j\ \forall j \in J, b, l, r),\\
& N = (S, F, Q, N, W, A_j\ \forall j \in J, B, L, R), P, S),\\
& S \rightarrow RFR|FR|RF|RQR|QR|RQ|RNR|NR|RN,\ B \rightarrow b,\ L \rightarrow llll,\\
& F \rightarrow_{[38,38]} NLN,\ Q \rightarrow_{[25,25]} WBW,\\
& N \rightarrow_{[17,17]} WBW,\ R \rightarrow Rr|r,\\
& W \rightarrow_{[8,20]} A_j\ \forall j \in J,\ A_j \rightarrow A_j a_j | a_j\ \forall j \in J.
\end{aligned}
$$

5.3. Instances Size

Ten different demands were tested for each activity (*Nb.Act*) and for each version on the demand shape (*Shape.*) Table 1 shows the size of these instances including the average number of employees (*Nb.Emp*) and several grammar-related statistics: the average number of children of the root node (*Nb.ChRoot*), the average number of and-nodes (*Nb.AndN*) without including the children of the root node, the average number of or-nodes (*Nb.OrN*) without including the leaves, and the average number of leaves (*Nb.Leaves*) of DAG Γ. We also present the average number of nodes (*Nd.Nodes*) and the average number of arcs (*Nb.Arcs*), in the directed acyclic graph from the pricing subproblem to generate columns. Observe that the number of variables and constraints in $B'_{\mathcal{T}}$ is different at each iteration m. The number of variables is equal to *Nd.Nodes* + $|\widetilde{\mathcal{T}}^m|$, where $|\widetilde{\mathcal{T}}^m|$ represents the total number of columns generated until iteration m. The number of constraints is equal to 1 + *Nb.nodes*+$\sum_{d\in D}\sum_{w\in\Omega_d}|E^m_{\mathbf{\Delta}^{(w)}_d}|$, where $|E^m_{\mathbf{\Delta}^{(w)}_d}|$ represents the total number of cuts generated until iteration m for day d and scenario w. The number of variables at each second-stage problem is equal to $|I_d| \times |J| \times 2$ + *Nb.AndN* + *Nb.Leaves*. The number of constraints at each second-stage

problem is equal to $|I_d| \times |J| + Nb.OrN + Nb.Leaves$. The average range of the mean demand μ, and the average amplitude of demand σ are also presented for each demand shape.

Shape.	*Nb.Act*	*Nb.Emp*	*Nb.ChRoot*	*Nb.AndN*	*Nb.OrN*	*Nb.Leaves*	*Nb.Nodes*	*Nb.Arcs*
Bim./level	1	20	175	6,089	5,721	286	1,224	425,679
	2	36	175	7,956	7,015	370	1,225	426,405
$\mu = [4.5, 5]$	3	55	175	9,816	8,302	454	1,225	426,405
$\sigma = [8, 9]$	4	72	175	11,676	9,589	538	1,225	426,405
	5	90	175	13,536	10,876	622	1,225	426,405
Bim./unim.	1	19	171	5,937	5,578	280	1,198	413,575
	2	36	172	7,816	6,889	365	1,205	417,207
$\mu = [2, 8.5]$	3	52	172	9,662	8,169	448	1,206	417,855
$\sigma = [2, 15]$	4	69	172	11,508	9,448	532	1,207	418,301
	5	86	173	13,389	10,754	616	1,211	420,352
Unim./level	1	13	138	4,654	4,370	237	971	274,551
	2	23	144	6,480	5,692	319	1,012	296,130
$\mu = [2.5, 3]$	3	34	145	8,157	6,867	395	1,021	300,572
$\sigma = [5, 6]$	4	45	146	9,818	8,025	472	1,026	304,314
	5	55	148	11,548	9,239	549	1,036	310,304
Unim./unim.	1	17	139	4,696	4,411	238	992	294,909
	2	31	141	6,325	5,554	313	1,017	306,104
$\mu = [1.5, 10.5]$	3	45	141	7,901	6,649	386	1,017	305,928
$\sigma = [2, 21]$	4	59	142	9,578	7,826	462	1,020	307,312
	5	73	143	11,190	8,947	536	1,020	307,312
Level/level	1	30	97	3,188	3,001	182	681	139,147
	2	50	97	4,400	3,847	240	681	140,245
$\mu = [11, 18]$	3	76	106	6,076	5,086	317	746	166,676
$\sigma = [11, 20]$	4	103	108	7,468	6,070	382	758	171,344
	5	128	109	8,842	7,036	446	765	174,948

Table 1: Size of SMASSP instances

5.4. Computational Results

Table 2 presents the computational results on stochastic weekly instances dealing with up to five work activities and for cost scenario 1 (*CostScen1*), where $c_{dij} = 15$, $c^{+}_{dij} = 20$, $c^{-}_{dij} = 60$. These results include the average CPU time to solve the problem (*T.time*), the average CPU time spent in the CG approach (*T.CG*), which includes the time to solve the pricing sub-problems and the time to solve the LP relaxation when new columns are added, the average CPU time to solve the first-stage problem (*T.F-S*), and the average CPU time to solve the second-stage problems (*T.S-S*). CPU times are presented in seconds (s). The average gap between the best upper bound and best lower bound is presented in *Gap1*. This gap is computed as: $Gap1 = 100 \times (Q^{UB}_m - \theta^{LB})/Q^{UB}_m$. Since the second-stage problems

are MILPs that do not possess the integrality property and we are relaxing integrality constraints on variables $y_{dij}^{(w)}$, we also calculate the average gap between the upper bound Q_m^{UB} and the approximate upper bound $\widetilde{Q}_m^{UB}$ as: *Gap2*= $100 \times (Q_m^{UB} - \widetilde{Q}_m^{UB})/Q_m^{UB}$. *Conv.* presents the number of instances that converged to a near-optimal solution, i.e., the algorithm stopped when no more optimality cuts were added to the first-stage model. Results for cost scenario 2 (*CostScen2*) with $c_{dij} = 15$, $c_{dij}^{+} = 0$, $c_{dij}^{-} = 90$ are presented in AppendixA.

Shape.	*Nb. Act*	*T.time (s.)*	*T.CG (s.)*	*T.F-S (s.)*	*T.S-S (s.)*	*Gap1*	*Gap2*	*Conv.*
Bim./level	1	706.96	130.79	203.30	372.87	0.00%	0.00%	10
	2	2,185.52	385.21	552.39	1,247.92	0.00%	0.00%	10
	3	3,525.60	530.20	740.38	2,255.02	0.01%	0.01%	10
	4	4,121.34	477.77	748.03	2,895.54	0.01%	0.01%	10
	5	4,770.41	488.40	601.63	3,680.39	0.03%	0.02%	10
Bim./unim.	1	1,007.05	153.44	506.19	347.42	0.00%	0.00%	10
	2	3,454.93	762.29	1,575.52	1,117.12	0.01%	0.01%	10
	3	6,438.78	1,116.84	2,972.61	2,349.33	0.02%	0.02%	10
	4	6,141.87	1,272.00	1,756.42	3,113.45	0.03%	0.03%	10
	5	7,832.38	1,340.90	2,925.55	3,565.93	0.06%	0.04%	10
Unim./level	1	2,607.69	256.51	2,055.44	295.74	0.00%	0.00%	10
	2	6,023.31	764.94	3,931.98	1,326.40	0.24%	0.00%	10
	3	6,045.06	500.55	3,691.78	1,852.74	0.17%	0.02%	10
	4	6,799.45	413.99	3,970.50	2,414.97	0.28%	0.04%	9
	5	10,473.59	861.24	5,705.51	3,906.85	0.90%	0.05%	5
Unim./unim.	1	2,814.60	802.80	1,441.24	570.57	0.00%	0.00%	10
	2	6,287.41	1,705.67	3,001.79	1,579.94	0.29%	0.00%	9
	3	8,456.22	1,839.14	3,605.73	3,011.36	0.19%	0.02%	9
	4	10,054.78	1,974.96	4,155.02	3,924.80	0.84%	0.04%	7
	5	10,565.48	1,891.22	3,727.33	4,946.93	1.31%	0.05%	3
Level/level	1	222.56	35.58	53.58	133.39	0.00%	0.00%	10
	2	1,095.15	271.99	289.37	533.80	0.00%	0.00%	10
	3	5,897.56	1,998.89	2,270.83	1,627.85	0.82%	0.00%	8
	4	5,875.22	1,816.50	2,006.79	2,051.93	0.48%	0.01%	8
	5	6,483.81	1,957.59	1,619.63	2,906.60	0.84%	0.03%	8

Table 2: Computational effort on stochastic instances under CostScen1.

From Table 2, we can conclude that our method was able to find near-optimal solutions for almost all the instances with up to three work activities, when evaluated on 100 demand scenarios. Our method was not able to find solutions that converged for all of the instances with four and five activities and 100 demand scenarios. However, for most of these instances, the final gap is, on average, less than 1.31%. Although integrality constraints on second-stage variables were relaxed, we can conclude that in most cases, the approximate upper bound $\widetilde{Q}_m^{UB}$ was the same as or very close to the real

upper bound Q_m^{UB}. The above can be observed from the average values of $Gap2$, which are at most 0.05%.

Regarding the distribution of the CPU time, we observe that in most of the instances, the time to solve the LP relaxation of the second-stage problems is very similar to the time to solve the first-stage problem. These CPU times increase, in most of the cases, with the numbers of activities, and they appear to be influenced by the shape of the demand (e.g., instances with a weekly level profile converged faster than instances with an unimodal weekly profile). Finally, we observe that the time spent in the CG method is not affected by the number of activities (at least for instances from three to five activities), as weekly patterns are composed with shifts that do not include activity allocation (shift shells).

The multi-cut L-shaped method was compared with a heuristic "one-tree" approach. This approach consists of two phases. In the first phase, we add columns and optimality cuts to the LP relaxation of problem $B_{\widetilde{\mathcal{T}}}^{LP}$. Then, when the criteria of the McDaniel and Devine strategy is met ($(\widetilde{Q}_m^{UB} - \theta_m^{LB})/\widetilde{Q}_m^{UB} < \epsilon_2$), we switch to the second phase. In this phase we start a branch-and-bound procedure, where new optimality cuts are added to the problem, as CPLEX callbacks, at the nodes with integer solutions. The one-tree approach is faster in terms of total CPU time, as it can be observed in Table B.8 from AppendixB. However, the solution quality measured as the percentage change between the final upper bound value for the one-tree method and the final upper bound value for the multi-cut L-shaped method, shows that the multi-cut L-shaped method always provides better quality solutions, as new columns are generated in all the process and not only when solving the LP relaxation of the first-stage problem.

5.5. Value of the Stochastic Solution

Table 3-4 present a comparison between the results and computational effort of the two-stage stochastic problem and the mean value problem. Table 3 presents the average CPU time to solve the stochastic problem ($T.time$), the average CPU time to solve the mean value problem ($T.timeEV$), the average value of the stochastic solution, associated with tree factors: allocation costs (VSS_{AL}), overcovering costs (VSS_{OC}), and undercovering costs (VSS_{UC}). Column VSS_T presents the average of the total value of the stochastic solution. These values are computed as: $VSS_c = 100 \times (EEV_c - HN_c)/HN_T, \forall c = \{T, AL, OC, UC\}$, where $EEV_T, EEV_{AL}, EEV_{OC}, EEV_{UC}$

correspond to the expected value of the total cost, allocation cost, overcovering cost, and undercovering cost when using the EV solution, respectively. In the same way, $HN_T, HN_{AL}, HN_{OC}, HN_{UC}$ correspond to the final value of the total cost, allocation cost, overcovering cost, and undercovering cost, when using the stochastic model, respectively. The last column of Table 3 ($TA.Dif$) presents an index that measures the difference between the shift allocation from the two-stage stochastic problem and the mean value problem. This index is computed as: $TA.Dif = 100 \times \left(\frac{\sum_{d \in D} \sum_{i \in I_d} 1 \times AV_{di}}{(\sum_{d \in D} \sum_{i \in I_d} 1)}\right)$, where $u_{di}, \bar{u}_{di}$ denote the number of employees working at day d and time period d, for the two-stage stochastic problem and for the mean value problem, respectively, and AV_{di} takes value one if $|u_{di} - \bar{u}_{di}| > 0$, and value zero otherwise.

Table 4 shows a comparison of the computational effort to solve the stochastic problem and the mean value problem. This information includes the average number of iterations for the multi-cut L-shaped method ($I.L\text{-}S$), the average number of iterations for the CG method ($I.CG$), the average number of optimality cuts added to the master problem ($Nb.Cuts$), and the average number of columns added to the master problem ($Nb.Col.$).

Shape.	*Nb. Act*	*T.time (s.)*	*T.timeEV (s.)*	VSS_{AL}	VSS_{OC}	VSS_{UC}	VSS_T	$TA.Dif$
	1	706.96	213.30	0.02%	1.31%	3.88%	5.21%	57.87%
	2	2,185.52	3,771.84	0.00%	0.48%	1.43%	1.90%	62.68%
Bim./level	3	3,525.60	8,594.19	0.00%	1.06%	3.19%	4.25%	68.81%
	4	4,121.34	8,754.93	0.00%	1.41%	4.23%	5.64%	70.00%
	5	4,770.41	9,451.65	0.00%	1.24%	3.72%	4.95%	71.96%
	1	1,007.05	175.83	0.00%	0.85%	2.56%	3.42%	58.74%
	2	3,454.93	264.16	0.00%	0.98%	2.93%	3.91%	67.65%
Bim./unim.	3	6,438.78	392.94	0.00%	1.05%	3.14%	4.19%	73.08%
	4	6,141.87	425.20	0.00%	1.07%	3.22%	4.28%	71.31%
	5	7,832.38	425.20	0.00%	1.12%	3.37%	4.50%	75.27%
	1	2,607.69	42.38	0.00%	1.86%	5.59%	7.45%	47.51%
	2	6,023.31	5,821.94	0.00%	2.85%	8.54%	11.39%	59.21%
Unim./level	3	6,045.06	8,597.50	0.00%	3.51%	10.53%	14.03%	61.74%
	4	6,799.45	9,529.08	0.00%	3.73%	11.20%	14.94%	63.21%
	5	10,473.59	10,607.98	0.00%	3.98%	11.95%	15.93%	64.52%
	1	2,814.60	7,492.79	0.00%	1.65%	4.96%	6.62%	48.14%
	2	6,287.41	7,398.80	0.00%	1.22%	3.66%	4.88%	54.78%
Unim./unim.	3	8,456.22	10,592.67	-0.06%	0.86%	2.81%	3.61%	56.44%
	4	10,054.78	10,054.61	-0.05%	0.23%	0.88%	1.06%	58.87%
	5	10,565.48	10,831.52	-0.07%	0.25%	1.01%	1.19%	61.28%
	1	222.56	21.67	0.00%	1.14%	3.42%	4.56%	35.97%
	2	1,095.15	522.70	0.00%	0.53%	1.59%	2.12%	46.68%
Level/level	3	5,897.56	9,624.00	0.00%	1.35%	4.04%	5.39%	53.04%
	4	5,875.22	8,303.40	0.00%	1.28%	3.84%	5.12%	56.25%
	5	6,483.81	9,222.51	-0.01%	1.55%	4.68%	6.22%	56.59%

Table 3: Value of the stochastic solution under CostScen1.

Results from Table 3 suggest that the two-stage stochastic model can lead to significant reductions in the total cost when compared to the mean value program, since the VSSs are always positive values ranging from 1.06% to 15.93%. Observe that activity allocation costs are very similar for the stochastic problem and for the mean value problem, as all values for VSS_{AL} are close to zero. For the instances tested, a zero value of VSS_{AL} means that the solution for the total number of hours worked by all employees is the same for both problems (stochastic problem and mean value problem). However, a VSS_{AL} equals to zero does not indicate that first-stage solutions are the same for the two problems, as the allocation of shift start times, days-off, and working days to the employees might be different. Column $TA.Dif$ presents an index to measure such difference. $TA.Dif$ indicates the percentage of time periods in which the absolute value of the difference between the number of employees allocated with the two-stage stochastic problem and the number of employees allocated with the mean value problem is greater than zero. Positive values for $TA.Dif$ indicate that the schedule allocation decisions (first-stage solutions) found with the stochastic problem are different from

the schedule allocation decisions found with the mean value problem. Because these first-stage solutions are different, the positive values for VSS_{OC}, VSS_{UC} confirm that the tactical plan built by using stochastic information is robust enough to allow changes in the allocation of activities and breaks to shift schedules, preventing the occurrence of additional understaffing and overstaffing costs.

A small percentage of improvement in costs by including variability in demand in personnel scheduling problems for service companies is still meaningful, since most of the operational costs depend on the staffing. Additionally, a reduction in undercovering and overcovering levels by using the stochastic model will certainly have a positive impact on customer service levels, reputation, and total revenues. The size and importance of this improvement depends on the values of the allocation, undercovering and overcovering costs, as well as on the type of service sector. For instance, if undercovering costs are relatively high when compared to overcovering and allocation costs (e.g., healthcare applications), a small reduction in the understaffing by using a stochastic model will lead to large percentage improvements. Note that the VSS presented in Table A.6 from AppendixA demonstrates the previous remark (larger VSSs when compared to the values in Table 3, for most of the instances tested). The instances used for the experiments in Table A.6 are the same 250 instances tested above, but overcovering and undercovering penalties are changed from 20 to 0 and from 60 to 90, respectively.

We note that in most of the instances there is no significant difference between the total CPU time to solve the stochastic problem and the total CPU time to solve the deterministic problem. This might be explained by the fact that the multi-cut L-shaped method provides more cuts to support the recourse function from below, and most likely reduce the number of iterations (Table 4, Columns 5 and 7). However, the introduction of more constraints in the master problem, may potentially slow the computational times, as observed in the CPU total time from Table 3 for the Bim./unim. demand shape.

		Stochastic problem				*Mean value problem*			
Shape.	*Nb. Act*	*I.L-S*	*I.CG*	*Nb.Cuts*	*Nb.Col.*	*I.L-S*	*I.CG*	*Nb.Cuts*	*Nb.Col.*
	1	21	17	11,498	477	44	17	316	419
	2	25	20	14,451	582	174	69	1,237	1,961
Bim./level	3	27	20	15,461	567	239	138	1,590	3,831
	4	27	21	16,064	585	225	112	1,614	3,139
	5	27	20	10,076	573	259	162	1,685	4,344
	1	23	51	11,326	1,488	45	125	258	3,486
	2	28	64	15,160	1,899	50	138	298	3,831
Bim./unim.	3	30	76	16,985	2,262	50	134	308	3,594
	4	31	68	16,804	1,998	54	136	334	3,729
	5	33	88	17,994	2,607	56	140	352	3,967
	1	20	111	11,306	3,312	50	19	383	546
	2	30	87	16,072	2,589	294	90	1,986	2,676
Unim./level	3	27	82	15,607	2,436	289	112	2,059	3,336
	4	26	85	15,406	2,532	290	131	2,102	3,912
	5	30	94	19,302	2,784	312	149	2,277	4,428
	1	35	45	18,515	1,314	134	184	728	5,043
	2	38	62	23,604	1,836	150	237	820	6,359
Unim./unim.	3	41	69	25,837	2,037	147	305	831	8,661
	4	41	78	27,129	2,295	149	337	865	8,443
	5	40	74	27,367	2,190	145	348	817	9,356
	1	18	16	9,466	456	26	16	172	419
	2	26	32	14,016	924	84	44	575	1,177
Level/level	3	41	58	25,349	1,716	283	159	1,942	4,411
	4	40	49	24,845	1,428	279	185	1,831	5,087
	5	41	48	26,201	1,407	275	183	1,854	5,187

Table 4: Computational effort comparison between the stochastic problem against the mean value problem under CostScen1.

6. Concluding Remarks

In this paper, we presented a two-stage stochastic programming approach to solve the discontinuous multi-activity tour scheduling problem when demand is uncertain and employees have identical skills. In the model, first-stage decisions correspond to the allocation of employees to days-off and to daily shifts (weekly tours), while second-stage decisions correspond to the allocation of work activities and breaks to shifts and to the undercovering and overcovering of demand. Since the number of tours becomes large with an increase in shift and tour flexibility, the first-stage problem was solved via CG. Second-stage problems were modeled with context-free grammars in order to efficiently handle the work rules for the composition of the shifts and the allocation of work activities to the shifts.

A heuristic multi-cut L-shaped method was implemented as a solution approach. Two algorithmic refinements were used to enhance the performance of the method. First, we adopted the strategy of McDaniel and Devine

(1977) in order to generate an initial set of valid cuts in a fast way. Second, we implemented the idea of Papadakos (2008) to generate strong optimality cuts. Additionally, we showed that, although second-stage problems are MILP models that do not possess the integrality property, high-quality solutions can be achieved by relaxing the integrality constraints to generate optimality cuts.

An extensive computational study suggests that our algorithm was able to provide solutions with a maximum average optimality gap of 1.31% for instances dealing with up to five work activities and 100 scenarios. The value of the stochastic solution, ranging from 1.06% to 15.93%, demonstrates that the two-stage model leads to robust solutions that show a significant reduction in labour costs when compared to the mean value program, since it prevents the occurrence of additional understaffing and overstaffing costs.

The two-stage stochastic programming formulation proposed in this paper presents some limitations that can be subject to further investigation in future works. These limitations are related to the definition of second-stage decisions, which currently only includes the allocation of breaks and work activities to shifts. The use of additional recourse actions such as hiring part-time employees or adding overtime to the employee schedules, will probably generate more robust solutions that are expected to respond better to demand uncertainty. Moreover, first-stage decisions can also be modified at the expense of increasing the complexity of the second-stage problems. Specifically, first-stage decisions can include the composition of days-off schedules (work days and off-work days) for each employee, while second-stage decisions will include the definition of shift start times, shift lengths, as well as the allocation of breaks and work activities to shifts, while guaranteeing the rules for the composition of employee tours (e.g., minimum rest time between consecutive shifts and weekly tour length). Future research can also include the use of an algorithm for scenario reduction that will compute the (nearly) best approximation of the discrete probability distribution for employee requirements, by a measure with fewer scenarios.

Acknowledgments

The authors would like to thank the Fonds Québécois de Recherche sur la Nature et les Technologies which supported this work with a grant. The authors would also like to thank the anonymous reviewers for their valuable comments and suggestions that helped to improve the quality of this paper.

AppendixA. Computational Effort and Solution Quality for Stochastic Instances under CostScen2.

Shape.	*Nb. Act*	*T.time (s.)*	*T.CG (s.)*	*T.F-S (s.)*	*T.S-S (s.)*	*Gap1*	*Gap2*	*Conv.*
Bim./level	1	1,224.97	317.54	444.38	463.04	0.00%	0.00%	10
	2	2,800.96	555.43	802.39	1,443.15	0.00%	0.00%	10
	3	3,772.30	645.91	873.41	2,252.99	0.01%	0.01%	10
	4	5,804.63	931.09	1,259.34	3,614.20	0.01%	0.01%	10
	5	6,840.95	854.26	1,178.08	4,808.61	0.02%	0.02%	10
Bim./unim.	1	635.43	92.89	218.17	324.37	0.00%	0.00%	10
	2	2,620.24	493.50	700.56	1,426.18	0.01%	0.01%	10
	3	3,384.86	470.99	780.31	2,133.57	0.03%	0.02%	10
	4	4,873.55	532.73	1,133.92	3,206.91	0.04%	0.03%	10
	5	6,998.45	765.05	1,697.54	4,535.87	0.05%	0.04%	10
Unim./level	1	4,184.30	418.25	3,401.88	364.18	0.26%	0.00%	9
	2	7,415.12	458.01	5,859.52	1,097.59	0.84%	0.00%	6
	3	9,602.98	476.12	7,187.82	1,939.04	1.01%	0.02%	5
	4	10,097.47	449.47	7,120.33	2,527.67	0.91%	0.05%	5
	5	11,091.03	477.28	7,376.86	3,236.90	0.56%	0.08%	8
Unim./unim.	1	5,666.34	1,089.01	3,966.30	611.03	0.19%	0.00%	10
	2	7,251.99	1,590.45	3,839.30	1,822.24	0.59%	0.00%	8
	3	9,524.86	1,993.68	4,068.15	3,463.04	0.60%	0.03%	6
	4	10,607.88	2,171.00	3,386.11	5,050.77	3.89%	0.03%	3
	5	10,599.08	2,046.25	2,160.65	6,392.19	4.81%	0.04%	0
Level/level	1	291.28	58.67	85.74	146.87	0.00%	0.00%	10
	2	1,724.12	537.00	506.50	680.62	0.00%	0.00%	10
	3	3,180.14	857.16	989.47	1,333.52	0.00%	0.00%	10
	4	4,598.09	1,133.21	1,405.46	2,059.43	0.01%	0.01%	10
	5	5,760.51	1,138.13	1,783.36	2,839.01	0.07%	0.02%	10

Table A.5: Computational effort on stochastic instances under CostScen2.

Shape.	*Nb. Act*	*T.time (s.)*	*T.timeEV (s.)*	VSS_{AL}	VSS_{OC}	VSS_{UC}	VSS_{T}
	1	1,224.97	31.87	0.00%	0.00%	8.51%	8.51%
	2	2,800.96	1,774.08	-0.01%	0.00%	5.26%	5.25%
Bim./level	3	3,772.30	7,428.56	0.00%	0.00%	5.44%	5.44%
	4	5,804.63	10,595.80	-0.01%	0.00%	7.15%	7.14%
	5	6,840.95	9,539.50	-0.01%	0.00%	6.69%	6.68%
	1	635.43	234.01	0.00%	0.00%	5.53%	5.53%
	2	2,620.24	311.84	0.00%	0.00%	4.71%	4.71%
Bim./unim.	3	3,384.86	272.60	0.00%	0.00%	4.40%	4.40%
	4	4,873.55	208.32	0.00%	0.00%	4.06%	4.05%
	5	6,998.45	406.52	0.00%	0.00%	4.71%	4.71%
	1	4,184.30	9.12	0.00%	0.00%	11.22%	11.22%
	2	7,415.12	2,946.02	0.02%	0.00%	13.72%	13.74%
Unim./level	3	9,602.98	9,514.23	-0.08%	0.00%	17.44%	17.36%
	4	10,097.47	9,546.24	-0.17%	0.00%	20.11%	19.94%
	5	11,091.03	10,558.17	-0.12%	0.00%	21.68%	21.57%
	1	5,666.34	7,621.11	-0.02%	0.00%	9.45%	9.42%
	2	7,251.99	7,485.70	-0.12%	0.00%	5.82%	5.70%
Unim./unim.	3	9,524.86	10,590.23	-0.12%	0.00%	5.92%	5.81%
	4	10,607.88	10,589.35	-0.27%	0.00%	3.39%	3.12%
	5	10,599.08	10,654.40	-0.24%	0.00%	1.97%	1.73%
	1	291.28	13.25	0.00%	0.00%	8.18%	8.18%
	2	1,724.12	28.54	-0.04%	0.00%	5.73%	5.69%
Level/level	3	3,180.14	3,838.55	-1.31%	0.00%	10.13%	8.82%
	4	4,598.09	3,782.92	-0.42%	0.00%	9.24%	8.82%
	5	5,760.51	6,987.04	-0.65%	0.00%	11.78%	11.13%

Table A.6: Value of the stochastic solution under CostScen2.

		Stochastic problem				*Mean value problem*			
Shape.	*Nb. Act*	*I.L-S*	*I.CG*	*Nb.Cuts*	*Nb.Col.*	*I.L-S*	*I.CG*	*Nb.Cuts*	*Nb.Col.*
	1	25	20	13,714	573	44	17	253	338
	2	27	22	15,926	642	174	69	796	1,010
Bim./level	3	29	23	17,188	666	239	138	1,709	2,894
	4	32	27	19,335	792	225	112	2,183	3,618
	5	32	27	19,489	792	259	162	1,928	3,792
	1	21	39	10,737	1,134	45	125	274	3,861
	2	29	31	14,622	906	50	138	333	4,428
Bim./unim.	3	29	29	15,617	852	50	134	329	4,052
	4	32	35	16,534	1,026	54	136	291	3,737
	5	34	38	17,647	1,107	56	140	364	4,234
	1	24	115	11,775	3,417	50	19	269	246
	2	26	88	15,189	2,604	294	90	1,299	1,341
Unim./level	3	28	90	16,312	2,664	289	112	2,425	3,408
	4	27	101	16,468	3,009	290	131	2,187	3,786
	5	26	110	15,932	3,276	312	149	2,349	4,125
	1	41	80	20,720	2,355	134	184	680	3,736
	2	36	60	22,105	1,773	150	237	741	9,274
Unim./unim.	3	41	47	25,201	1,392	147	305	844	11,452
	4	38	51	24,822	1,503	149	337	842	13,124
	5	39	35	25,095	1,014	145	348	781	14,497
	1	19	21	9,881	609	26	16	152	402
	2	32	40	16,319	1,164	84	44	214	672
Level/level	3	32	49	18,750	1,440	283	159	1,227	2,622
	4	37	53	21,664	1,572	279	185	1,221	3,093
	5	37	59	22,899	1,743	275	183	1,626	4,434

Table A.7: Computational effort comparison between the stochastic problem against the mean value problem under CostScen2.

AppendixB. Computational Effort and Solution Comparison for the One-Tree Solution Method

Table B.8 presents a comparison on the computational efficiency and solution quality for the multi-cut L-shaped method presented in Section 4 and for the one-tree approach when ϵ_2 is set to 0.1. Columns two and three present, for each demand shape (*Shape*) and number of activities (*Nb.Act*), the average total CPU time to solve the problem with the multi-cut L-shaped method and with the one-tree approach, respectively. The percentage changes in allocation, overcovering, undercovering and total costs are given by $Cost_{AL}$, $Cost_{OC}$, $Cost_{UC}$, and $Cost_T$, respectively. These values are computed as: $Cost_c = 100 \times (OT_c - HN_c)/OT_c, \forall c = \{AL, OC, UC, T\}$, where $OT_{AL}, OT_{OC}, OT_{UC}, OT_T$ correspond to the value for the allocation costs, overcovering costs, undercovering costs, and total costs when using the one-tree approach, respectively. In the same way, $HN_{AL}, HN_{OC}, HN_{UC}, HN_T$ correspond to the final value for the allocation costs, overcovering costs,

undercovering costs, and total costs, when using the stochastic model, respectively.

Shape	*Nb.Act*	*T.time (s.)*	*T.timeH (s.)*	$Cost_{AL}$	$Cost_{OC}$	$Cost_{UC}$	$Cost_T$
Bim./level	1	1,043.61	683.49	0.00%	0.08%	40.26%	5.91%
	2	1,839.01	1,725.98	0.00%	-0.57%	37.45%	6.51%
	3	2,401.77	2,391.33	0.00%	-0.21%	38.68%	5.58%
	4	3,589.84	3,238.35	0.00%	-0.31%	38.37%	5.61%
	5	4,170.37	3,647.60	0.00%	-0.20%	38.72%	5.30%
Bim./unim.	1	1,494.54	1,258.71	0.00%	0.04%	40.08%	10.38%
	2	3,582.10	2,055.92	0.00%	0.25%	40.60%	11.95%
	3	4,478.56	2,130.84	0.00%	0.54%	41.19%	12.97%
	4	4,659.44	3,098.57	0.00%	0.20%	40.44%	12.56%
	5	5,498.33	4,706.74	0.00%	0.22%	40.48%	12.67%

Table B.8: Computational effort and solution quality comparison between the multi-cut L-shaped method and the one-tree method under CostScen1.

References

Alfares, H. K. (2004). Survey, categorization, and comparison of recent tour scheduling literature. *Annals of Operations Research*, 127(1-4):145–175.

Aykin, T. (1996). Optimal shift scheduling with multiple break windows. *Management Science*, 42(4):591–602.

Bard, J. F., Morton, D. P., and Wang, Y. M. (2007). Workforce planning at USPS mail processing and distribution centers using stochastic optimization. *Annals of Operations Research*, 155(1):51–78.

Bechtold, S. E. and Showalter, M. J. (1987). A methodology for labor scheduling in a service operating system. *Decision Sciences*, 18(1):89–107.

Birge, J. R. and Louveaux, F. V. (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384–392.

Boyer, V., Gendron, B., and Rousseau, L.-M. (2014). A branch-and-price algorithm for the multi-activity multi-task shift scheduling problem. *Journal of Scheduling*, 17(2):185–197.

Campbell, G. M. (2011). A two-stage stochastic program for scheduling and allocating cross-trained workers. *Journal of the Operational Research Society*, 62(6):1038–1047.

Côté, M.-C., Gendron, B., Quimper, C.-G., and Rousseau, L.-M. (2011a). Formal languages for integer programming modeling of shift scheduling problems. *Constraints*, 16(1):54–76.

Côté, M.-C., Gendron, B., and Rousseau, L.-M. (2011b). Grammar-based integer programming models for multiactivity shift scheduling. *Management Science*, 57(1):151–163.

Côté, M.-C., Gendron, B., and Rousseau, L.-M. (2013). Grammar-based column generation for personalized multi-activity shift scheduling. *INFORMS Journal on computing*, 25(3):461–474.

Dahmen, S. and Rekik, M. (2015). Solving multi-activity multi-day shift scheduling problems with a hybrid heuristic. *Journal of Scheduling*, 18(2):207–223.

De Bruecker, P., Van den Bergh, J., Beliën, J., and Demeulemeester, E. (2015). Workforce planning incorporating skills: State of the art. *European Journal of Operational Research*, 243(1):1–16.

Demassey, S., Pesant, G., and Rousseau, L.-M. (2006). A cost-regular based hybrid column generation approach. *Constraints*, 11(4):315–333.

Detienne, B., Péridy, L., Pinson, É., and Rivreau, D. (2009). Cut generation for an employee timetabling problem. *European Journal of Operational Research*, 197(3):1178–1184.

Easton, F. F. and Mansour, N. (1999). A distributed genetic algorithm for deterministic and stochastic labor scheduling problems. *European Journal of Operational Research*, 118(3):505–523.

Easton, F. F. and Rossin, D. F. (1996). A stochastic goal program for employee scheduling. *Decision Sciences*, 27(3):541–568.

Ernst, A., Jiang, H., Krishnamoorthy, M., Owens, B., and Sier, D. (2004a). An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127:21–144.

Ernst, A., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004b). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27.

Kim, K. and Mehrotra, S. (2015). A two-stage stochastic integer programming approach to integrated staffing and scheduling with application to nurse management. *Operations Research*, 63(6):1431–1451.

Lequy, Q., Bouchard, M., Desaulniers, G., Soumis, F., and Tachefine, B. (2012). Assigning multiple activities to work shifts. *Journal of Scheduling*, 15(2):239–251.

Lewis, P. A. and Shedler, G. S. (1979). Simulation of nonhomogeneous poisson processes by thinning. *Naval Research Logistics Quarterly*, 26(3):403–413.

Magnanti, T. L. and Wong, R. T. (1981). Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484.

McDaniel, D. and Devine, M. (1977). A modified Benders' partitioning algorithm for mixed integer programming. *Management Science*, 24(3):312–319.

Pacqueau, R. and Soumis, F. (2014). Shift scheduling under stochastic demand. Technical Report G-2014-46, GERAD.

Papadakos, N. (2008). Practical enhancements to the Magnanti–Wong method. *Operations Research Letters*, 36(4):444–449.

Parisio, A. and Jones, C. N. (2015). A two-stage stochastic programming approach to employee scheduling in retail outlets with uncertain demand. *Omega*, 53:97–103.

Punnakitikashem, P., Rosenberber, J. M., and Buckley-Behan, D. F. (2013). A stochastic programming approach for integrated nurse staffing and assignment. *IIE Transactions*, 45(10):1059–1076.

Quimper, C.-G. and Rousseau, L.-M. (2010). A large neighbourhood search approach to the multi-activity shift scheduling problem. *Journal of Heuristics*, 16(3):373–392.

Quimper, C.-G. and Walsh, T. (2007). Decomposing global grammar constraints. In *Principles and Practice of Constraint Programming–CP 2007*, pages 590–604. Springer.

Restrepo, M. I., Gendron, B., and Rousseau, L.-M. (2015). Combining Benders decomposition and column generation for multiactivity tour scheduling. Technical Report 57, CIRRELT.

Restrepo, M. I., Gendron, B., and Rousseau, L.-M. (2016). Branch-and-price for multi-activity tour scheduling. *INFORMS Journal on Computing*, 28(2):1–17.

Restrepo, M. I., Lozano, L., and Medaglia, A. L. (2012). Constrained network-based column generation for the multi-activity shift scheduling problem. *International Journal of Production Economics*, 140(1):466–472.

Ritzman, L. P., Krajewsky, L. J., and Showalter, M. J. (1976). The disaggregation of aggregate manpower plans. *Management Science*, 22(11):1204–1214.

Robbins, T. R. and Harrison, T. P. (2010). A stochastic programming model for scheduling call centers with global service level agreements. *European Journal of Operational Research*, 207(3):1608–1619.

Song, H. and Huang, H.-C. (2008). A successive convex approximation method for multi-stage workforce capacity planning problem with turnover. *European Journal of Operational Research*, 188(1):29–48.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L. (2012). Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385.

Van Slyke, R. M. and Wets, R. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663.

Wolsey, L. A. and Nemhauser, G. L. (1999). *Integer and combinatorial optimization*. John Wiley & Sons.

Zhu, X. and Sherali, H. D. (2009). Two-stage workforce planning under demand fluctuations and uncertainty. *Journal of the Operational Research Society*, 60(1):94–103.