CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

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

María I. Restrepo
Bernard Gendron
Louis-Martin Rousseau

November 2015

CIRRELT-2015-58

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

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

María I. Restrepo[1,2,*], Bernard Gendron[1,3] , Louis-Martin Rousseau[1,2]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Mathematics and Industrial Engineering, Polytechnique Montréal, P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C 3A7

[3] Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

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 and randomly generated instances show that the use of the stochastic model helps to prevent additional costs, when compared with the expected-value problem solutions.

Keywords. Stochastic multi-activity tour scheduling problem, L-shaped method, column generation, context-free grammars.

Acknowledgements. The authors would like to thank the Fonds de recherche du Québec - Nature et technologies (FRQNT) which supported this work with a grant.

---

* Corresponding author: Maria-Isabel.Restrepo@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2015

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 usually one day divided into *time periods* of equal length. Since employees can perform different *work activities* during the same shift, the MASSP is concerned with choosing the work activities and the rest periods to assign to shifts to respond to a demand for service, that is translated into a *demand* for the number of employees required for each work activity and time interval. 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 companies 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 and flexibility in the composition of daily shifts and weekly tours. Second, since demand is typically unknown when scheduling decisions needs to be taken, specialized solution techniques that allow to include this variability should be developed. Specifically, such 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 should be implemented to compensate deficiencies in the previously made schedules (e.g., undercovering and overcovering of demand).

In this paper, we address the discontinuous *stochastic multi-activity tour scheduling problem* (SMATSP) for employees with identical skills. In this problem, a long-term staffing decision needs to be made while hedging for the short-term demand uncertainty. The problem is formulated as a two-stage stochastic programming model, decomposable by days and by scenarios, in which first-stage decisions correspond to the assignment of employees to weekly tours, while second-stage decisions (recourse actions) are related with the allocation of work activities and breaks to daily shifts. The contribution of this paper lies in the proposal of an approach to efficiently solve practical instances of the problem. A heuristic multi-cut L-shaped method is implemented as a solution approach. Because the complete enumeration of weekly tours makes the problem intractable, the first-stage problem is solved via column generation. Additionally, the second-stage problems benefit from the use of *context-free grammars* to include work rules regarding the definition of shifts and to efficiently handle the multi-activity context.

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 with 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 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 [1, 15, 16, 32], only recently some attention has been given to the problem that deals with multiple work activities. Ritzman et al. [30] 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 with switching between work activities are not considered. Heuristic approaches that use column generation (CG) [27] and tabu search [10] 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. [12] and Lequy et al. [18] 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. [11] present a CP-based CG algorithm to model complex regulation constraints in a real-world MASSP. Quimper and Rousseau [25] use formal languages to model the work rules related with the composition of shifts in a multi-activity context. Côté et al. [7] 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. [8] 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. [9] and Boyer et al. [5] 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. [29] and Restrepo et al. [28] present approaches based on branch-and-price (B&P) and Benders decomposition (BD), respectively. In the former approach [29], two B&P algorithms are presented for the personalized MATSP.

In the latter approach [28], 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 [14] and Easton and Mansour [13] develop heuristic methods that aim to tackle problems where demand is uncertain. Easton and Rossin [14] propose a tabu search method to solve a stochastic goal programming model that integrates and optimizes labor demand and employee scheduling. Easton and Mansour [13] present a genetic algorithm to solve shift scheduling problems in which the recourse decisions are related with 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. [3] propose a heuristic two-stage model that addresses tour scheduling problems over a one-week planning horizon. First-stage variables are related with 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 [21] propose a heuristic two-stage model to solve a shift scheduling problem. The proposed model is based on a decomposition of Aykin's [2] implicit model, where first-stage variables are associated to 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. [24] 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 [17] 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 variables 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 [31] 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.

Very limited literature is available on stochastic workforce planning for employees who have various skills to work on different activities, tasks or unit departments. Zhu and Sherali [34] 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 [6]. The author presents a two-stage model decomposable by days and by scenarios, where first-stage decisions are related with 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 previously defined and only a small number of scenarios is considered (10 in total). Parisio and Jones [23] 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.

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 $G_d = \langle \Sigma_d, N_d, S_d, P_d \rangle$, 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 $A \to \alpha$, 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 can be used to generate new symbol sequences until only terminal symbols are part of the sequence.

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 $l > 1$. $A^{\Pi,k}_{dil}$ 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 $A^{\Pi,k}_{d1n} \in ch(O^{S}_{d1n})$. The children of or-node O^{π}_{dil} are represented by $ch(O^{\pi}_{dil})$ and its parents by $par(O^{\pi}_{dil})$. Similarly, the children of and-node $A^{\Pi,k}_{dil}$ are represented by $ch(A^{\Pi,k}_{dil})$ and its parents by $par(A^{\Pi,k}_{dil})$. 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 [26].

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 $n = 4$ periods and should contain exactly one break, b, of one period that can be placed anywhere during the shift except at the first or the last period. 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:

$G_1 = (\Sigma = (w_1, w_2, b), N = (S, X, W, B), P, S)$,
where productions P are: $S \rightarrow XW$, $X \rightarrow WB$, $W \rightarrow WW|w_1|w_2$, $B \rightarrow b$,
and symbol | specifies the choice of production.

In the previous example, productions $W \rightarrow w_1$, $W \rightarrow w_2$ 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 followed by a break to finish with more working time.

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 $w_1bw_1w_2$, and a dashed-line parse tree that generates shift $w_2w_2bw_1$.

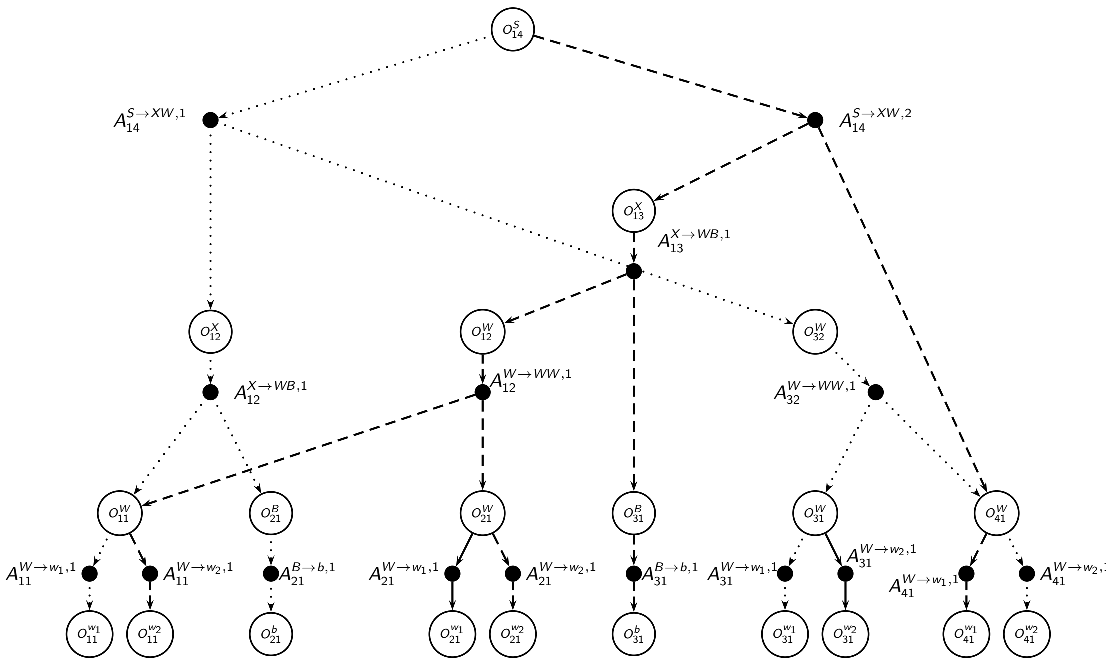

Figure 1 – DAG Γ_1 on words of length four and two work activities.

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. Hence, and-nodes $A_{d1n}^{\Pi,k}$ are characterized by their starting time $t_{d1n}^{\Pi,k}$, working length $w_{d1n}^{\Pi,k}$, and length including breaks $l_{d1n}^{\Pi,k}$. 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 intervals, a total length of four time intervals and both start at time interval 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. [28] 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. [8]. 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 number of employees assigned 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. [8]. 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 with the total working length, with the number of working days, with the rest time between consecutive shifts and with the allocation of days-off. Figure 2 presents an example of three tours composed with the shifts presented in 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 intervals, 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 $A^{S \to XW,1}_{d14} \to wbww$, S_2 corresponds to $A^{S \to XW,2}_{d14} \to wwbw$ 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, at the moment we can act on second-stage variables, the scenario for day $d \in D$ is fully known. Hence, staffing decisions (allocation of tours and daily shifts to employees) that are feasible to schedule without knowing in advance the demand, will be generated well ahead in time, while adjustments (allocation of work activities, position of breaks, overcovering and undercovering of demand) are made once improved (daily) demand information is available. We also assume that the random vector ξ representing the stochastic perturbations of demands is non-negative and 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 intervals at day $d \in D$;

E: set of employees;

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

First-stage problem
Parameter

$\delta_{dt}^{\Pi,k}$: parameter that takes value 1, if tour t includes the kth shift shell built with production Π for day d (variable $v_{d1n}^{\Pi,k}$), and assumes value 0 otherwise.

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 and-node built with production Π (children of the root node O_{d1n}^{S} from Γ_d).

Second-stage problem
Parameters

b_{dij}: deterministic demand for day d, time interval i and activity j;

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

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

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

c_{dij}^{+}, c_{dij}^{-}: demand overcovering and undercovering costs for day d, time interval i and activity j, respectively.

Decision variables

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

Additionally, let $\mathcal{V}$ denote the set of variables corresponding to the union, over the set of days $d \in D$, of variables $v_{d1n}^{\Pi,k}$. Given that notation, the formulation for the first-stage model, denoted $G_{\mathcal{T}}$, is as follows.

$$f(G_{\mathcal{T}}) = \min \quad \mathcal{Q}(\mathcal{V}) \tag{1}$$
$$v^{\Pi,k}_{d1n} = \sum_{t \in \mathcal{T}} \delta^{\Pi,k}_{dt} x_t, \quad \forall\, d \in D,\ A^{\Pi,k}_{d1n} \in ch(O^S_{d1n}), \tag{2}$$
$$\sum_{t \in \mathcal{T}} x_t = |E|, \tag{3}$$
$$x_t \geq 0 \text{ and integer}, \quad \forall\, t \in \mathcal{T}, \tag{4}$$
$$v^{\Pi,k}_{d1n} \geq 0, \quad \forall\, d \in D,\ A^{\Pi,k}_{d1n} \in ch(O^S_{d1n}). \tag{5}$$

The objective of $G_{\mathcal{T}}$, (1), is to minimize the expected recourse cost $\mathcal{Q}(\mathcal{V})$. Constraints (2) represent the link between daily shifts (children of root nodes O^S_{d1n} 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 $v^{\Pi,k}_{d1n}$.

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

$$Q(\mathcal{V}, \xi(w)) = \min \sum_{d \in D} \sum_{i \in I_d} \sum_{j \in J} c_{dij} y^{(w)}_{dij} + \sum_{d \in D} \sum_{i \in I_d} \sum_{j \in J} (c^+_{dij} s^{+(w)}_{dij} + c^-_{dij} s^{-(w)}_{dij}) \tag{6}$$
$$y^{(w)}_{dij} - s^{+(w)}_{dij} + s^{-(w)}_{dij} = b^{(w)}_{dij}, \quad \forall d \in D, i \in I_d,\ j \in J, \tag{7}$$
$$\sum_{A^{\Pi,k}_{dil} \in ch(O^\pi_{dil})} v^{\Pi,k,(w)}_{dil} = \sum_{A^{\Pi,k}_{dil} \in par(O^\pi_{dil})} v^{\Pi,k}_{d1n}, \quad \forall d \in D, O^\pi_{dil} \in ch(A^{\pi,k}_{d1n}) \setminus L_d, \tag{8}$$
$$\sum_{A^{\Pi,t}_{dil} \in ch(O^\pi_{dil})} v^{\Pi,k,(w)}_{dil} = \sum_{A^{\Pi,k}_{dil} \in par(O^\pi_{dil})} v^{\Pi,k,(w)}_{dil}, \quad \forall d \in D, O^\pi_{dil} \in O_d \setminus \{O^S_{d1n} \cup L_d \cup ch(A^{\pi,k}_{d1n})\}, \tag{9}$$
$$y^{(w)}_{dij} = \sum_{A^{\Pi,k}_{di1} \in par(O^j_{di1})} v^{\Pi,1,(w)}_{di1}, \quad \forall d \in D, i \in I_d,\ j \in J, \tag{10}$$
$$v^{\Pi,k,(w)}_{dil} \geq 0, \quad \forall\, d \in D, A^{\Pi,k}_{dil} \in A_d \setminus ch(O^S_{d1n}), \tag{11}$$
$$s^{+(w)}_{dij}, s^{-(w)}_{dij} \geq 0, \quad \forall d \in D, i \in I_d,\ j \in J, \tag{12}$$
$$y^{(w)}_{dij} \geq 0 \text{ and integer}, \quad \forall d \in D, i \in I_d,\ j \in J. \tag{13}$$

Problem (6)-(13) is based on the implicit model presented in Côté et al. [8]. 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 demand per day d, time interval i and work activity j is met for each demand realization w. Due to the structure of Γ_d, $d \in D$, constraints (8)-(9) guarantee for every or-node O^π_{dil}, excluding the root node O^S_{d1n} and the leaves L_d, that the summation of the value of its children, $ch(O^\pi_{dil})$, is the same as the summation of the value of its parents, $par(O^{\pi}_{dil})$. Constraints (10) set the value of variables $y^{(w)}_{dij}$ equal to the summation of the value of the parents of leaf nodes O^{j}_{di1}. Constraints (11)-(13) set the non-negativity of variables $v^{\Pi,k,(w)}_{dil}$, $s^{+(w)}_{dij}$, $s^{-(w)}_{dij}$ and the non-negativity and integrality of variables $y^{(w)}_{dij}$.

Observe that problem (1)-(5) (*first-stage problem*) has *complete recourse* because for any realization of the random vector ξ and value of variables $v^{\Pi,k}_{d1n}$, problem (6)-(13) (*second-stage problem*) is always feasible due to the allowance of undercovering and overcovering of demand. Additionally, note that problem $\mathcal{Q}(\mathcal{V}, \xi(w))$ is decomposable by days due to its particular block structure. Therefore, $\mathcal{V}$, Ω and $p^{(w)} > 0$ can also be decomposed by days: $\mathcal{V}_d$, Ω_d, $p^{(w)}_d > 0$, $d \in D$, and the expected recourse function $\mathcal{Q}(\mathcal{V})$ can be represented as:

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

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

4 Heuristic Multi-cut L-shaped Method

The basic idea behind the L-shaped method is to approximate the nonlinear term, $\mathcal{Q}(\mathcal{V})$, 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 $\mathcal{Q}(\mathcal{V})$, as in BD. Since ξ_d follows a non-negative distribution with a finite support, with Ω_d as the set of its possible realizations for each day $d \in D$, and $p^{(w)}_d > 0$ as the probability of occurrence of scenario $w \in \Omega_d$ ($\sum_{d \in D} \sum_{w \in \Omega_d} p^{(w)}_d = 1$), $\mathcal{Q}(\mathcal{V})$ can be expressed as $\mathcal{Q}(\mathcal{V}) = \sum_{d \in D} \sum_{w \in \Omega_d} p^{(w)}_d \mathcal{Q}(\mathcal{V}_d, \xi_d(w))$. By defining $\theta^{(w)}_d$ as an additional set of free variables, the two-stage stochastic problem $G_{\mathcal{T}}$ can be reformulated as the following model, denoted as $B_{\mathcal{T}}$.

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

$$v^{\Pi,k}_{d1n} = \sum_{t \in \mathcal{T}} \delta^{\Pi,k}_{dt} x_t, \quad \forall\, d \in D,\ A^{\Pi,k}_{d1n} \in ch(O^{S}_{d1n}), \tag{16}$$

$$\theta^{(w)}_d \geq p^{(w)}_d \mathcal{Q}(\mathcal{V}_d, \xi_d(w)), \quad \forall\, d \in D,\ w \in \Omega_d, \tag{17}$$

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

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

$$v^{\Pi,k}_{d1n} \geq 0, \quad \forall\, d \in D,\ A^{\Pi,k}_{d1n} \in ch(O^{S}_{d1n}). \tag{20}$$

Optimality cuts (17) ensure that the value of each variable $\theta^{(w)}_d$ 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 [4] 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. [28] 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 $\bar{\mathcal{Q}}(\mathcal{V}_d, \xi_d(w))$ denote the LP relaxation of problem $\mathcal{Q}(\mathcal{V}_d, \xi_d(w))$. Let $\rho_{dij}^{(w)}$, $\gamma_{dil}^{\pi,(w)}$ be the dual variables associated with constraints (7) and (8) from $\bar{\mathcal{Q}}(\mathcal{V}_d, \xi_d(w))$, 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 $\bar{\mathcal{Q}}(\mathcal{V}_d, \xi_d(w))$. Note that $\mathbf{\Delta}_d^{(w)}$ is itself a polyhedron [33]. 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 $B'_{\mathcal{T}}$:

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

Since these new optimality cuts are using linear approximations of $\mathcal{Q}(\mathcal{V}_d, \xi_d(w))$, 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 $l \geq 1$, 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 $E^l_{\mathbf{\Delta}_d^{(w)}} \subseteq E_{\mathbf{\Delta}_d^{(w)}}$. 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 $B^{LP}_{\widetilde{\mathcal{T}}}$, 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 to $B^{LP}_{\widetilde{\mathcal{T}}}$ as $B^{MILP}_{\widetilde{\mathcal{T}}}$, 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 [20], 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 [22] for the generation of strong optimality cuts. The author proposes an alternative to eliminate the necessity of solving the extra auxiliary subproblem introduced in Magnanti and Wong [19]. Additionally, since finding a *core point* for the problem is a difficult task, the author suggests to use an approximation of the core point that consists of a convex linear combination of the previously generated core point and the current solution for the first-stage problem.

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 [20] strategy. Let Int be a boolean variable that indicates whether the MILP ($B^{MILP}_{\widetilde{\mathcal{T}}}$) of the first-stage problem is solved (Int=true) or not (Int=false). Let ${\theta^{(w)}_{dl}}^*$ denote the optimal value of variables $\theta^{(w)}_d$ from $B^{LP}_{\widetilde{\mathcal{T}}}$, at iteration l. Note that $\underline{\theta}_l = \sum_{d\in D}\sum_{w\in\Omega_d} {\theta^{(w)}_{dl}}^*$ denotes a lower bound on $f(G_\mathcal{T})$ at iteration l. Since we are using a heuristic approach to find integer solutions for the first-stage model, we also denote $\overline{\theta_{dl}}^{(w)^*}$ as the optimal values of variables $\theta^{(w)}_d$ when $B^{MILP}_{\widetilde{\mathcal{T}}}$ is solved at iteration l. To simplify the algorithm description, we use the same notation when solving $B^{LP}_{\widetilde{\mathcal{T}}}$, even though in that case, we have ${\theta^{(w)}_{dl}}^* = \overline{\theta_{dl}}^{(w)^*}$ for each $d \in D, w \in \Omega_d$, since the CG algorithm is performed until all columns have non-negative reduced costs. Let ${s^{(w)}_{dl}}^*, \overline{s_{dl}}^{(w)^*}$ be the optimal value of second-stage problem (6)-(13) for day d, under scenario w at iteration l, when integrality constraints on variables $y^{(w)}_{dij}$ are imposed and relaxed, respectively. Note that $\bar{\theta}_l = \sum_{d\in D}\sum_{w\in\Omega_d} p^{(w)}_d {s^{(w)}_{dl}}^*$ is an upper bound on $f(G_\mathcal{T})$ at iteration l. Similarly, $\tilde{\theta}_l = \sum_{d\in D}\sum_{w\in\Omega_d} p^{(w)}_d \overline{s_{dl}}^{(w)^*}$ is an upper bound on $f(B'_\mathcal{T}) \leq f(G_\mathcal{T})$ at iteration l, which we call the approximated upper bound. Let ${v^{\pi,k}_{d1nl}}^*$, ${v^{\pi,k}_{d1nl}}^0$ denote an optimal solution and the core point approximation, respectively of first-stage variables $v^{\pi,k}_{d1n}$ from model $B'_\mathcal{T}$ at iteration l. The flow diagram of the algorithm is presented in Figure 3. The description of the multi-cut L-shaped method follows.

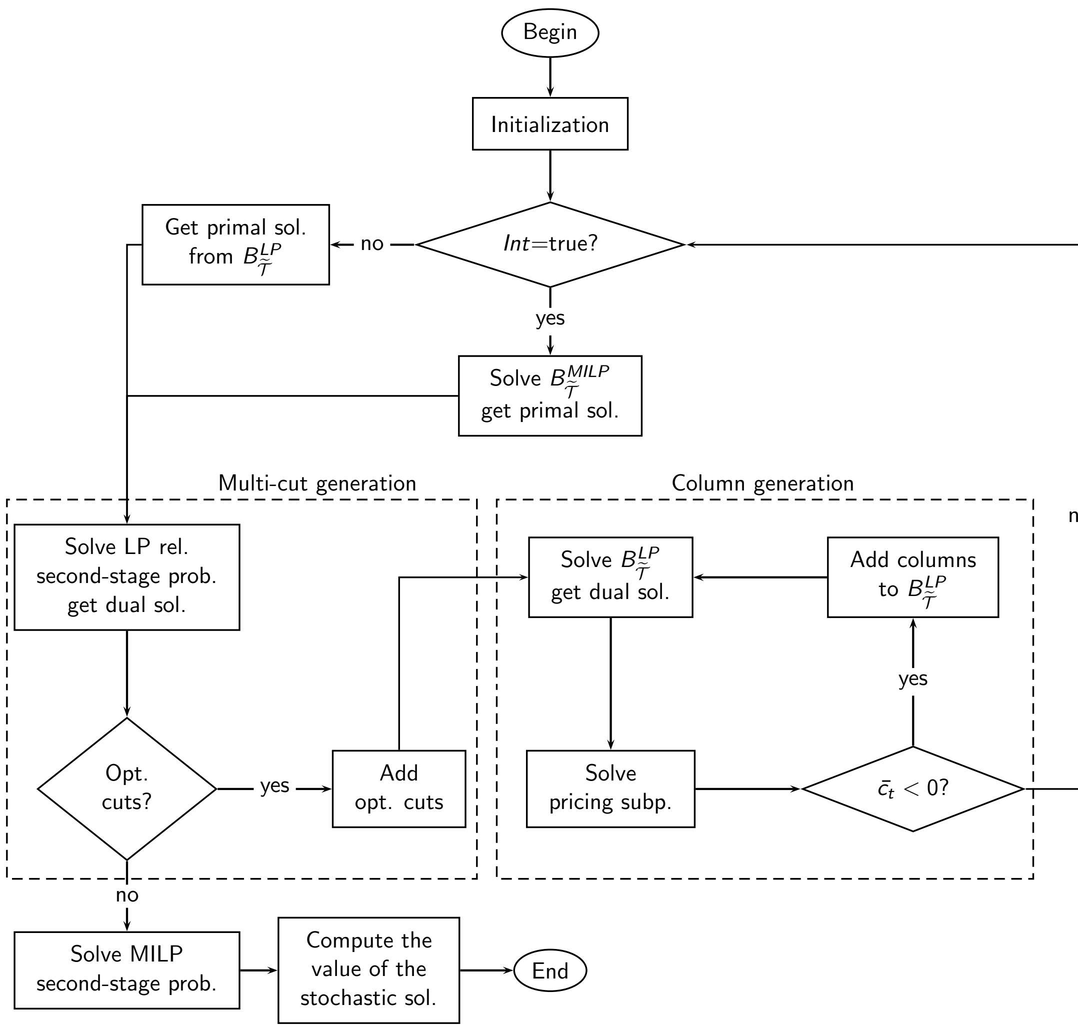

Figure 3 – Flow chart for the multi-cut L-shaped method.

- *Initialization*: The multi-cut L-shaped algorithm starts with an empty set of optimality cuts, $E^l_{\mathbf{\Delta}^{(w)}_d} = \varnothing$, $d \in D, w \in \Omega_d$. In this step, the number of iterations l is set to zero, the lower and upper bounds of the problem are initialized as $\bar{\theta}_l = \infty, \tilde{\theta}_l = \infty, \underline{\theta}_l = -\infty$, 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 $B^{LP}_{\widetilde{T}}$.

- *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* $B^{LP}_{\widetilde{T}}$. If $Int = true$ we continue with the step *Solve* $B^{MILP}_{\widetilde{T}}$ *get primal sol.* The value of Int is changed from false to true when $(\tilde{\theta}_l - \underline{\theta}_l)/\tilde{\theta}_l < \epsilon_2$ and $Int = false$.

- *Get primal sol. from* $B^{LP}_{\widetilde{T}}$: In this step of the algorithm, we get the primal solution $v^{\pi,k\ *}_{d1nl}$, $\theta^{(w)*}_{dl}$ from $B^{LP}_{\widetilde{T}}$. Then, we calculate the approximation of the core point as:

${v^{\pi,k}_{d1nl}}^0 = \frac{1}{2}{v^{\pi,k}_{d1n\,l-1}}^0 + \frac{1}{2}{v^{\pi,k}_{d1nl}}^*$, $\forall\, d \in D$, $A^{\Pi,k}_{d1n} \in ch(O^S_{d1n})$ and we update the lower bound of the problem as: $\underline{\theta}_l = \sum_{d\in D}\sum_{w\in\Omega_d} {\theta^{(w)}_{dl}}^*$. The values of ${v^{\pi,k}_{d1nl}}^*$, ${v^{\pi,k}_{d1nl}}^0$ are sent to the second-stage problems.

- *Solve $B^{MILP}_{\widetilde{\mathcal{T}}}$ get primal sol.*: In this step of the algorithm, we get the primal solution ${\theta^{(w)}_{dl}}^*$ from $B^{LP}_{\widetilde{\mathcal{T}}}$ to update the lower bound of the problem as: $\underline{\theta}_l = \sum_{d\in D}\sum_{w\in\Omega_d} {\theta^{(w)}_{dl}}^*$. Then we solve $B^{MILP}_{\widetilde{\mathcal{T}}}$ to get the primal solution ${\overline{\theta_{dl}}^{(w)}}^*, {v^{\pi,k}_{d1nl}}^*$ and to calculate the approximation of the core point as: ${v^{\pi,k}_{d1nl}}^0 = \frac{1}{2}{v^{\pi,k}_{d1n\,l-1}}^0 + \frac{1}{2}{v^{\pi,k}_{d1nl}}^*$, $\forall\, d \in D$, $A^{\Pi,k}_{d1n} \in ch(O^S_{d1n})$. The values of ${v^{\pi,k}_{d1nl}}^*$, ${v^{\pi,k}_{d1nl}}^0$ 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 $\mathcal{Q}(\mathcal{V}, \xi)$. 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 $v^{\pi,k}_{d1n}$ with the value of core point ${v^{\pi,k}_{d1nl}}^0$ to get a dual solution (ρ_d, γ_d). Second, we fix variables $v^{\pi,k}_{d1n}$ with the value of point ${v^{\pi,k}_{d1nl}}^*$ to recover the real objective value of the second-stage problems and to update the approximated upper bound of the problem as: $\tilde{\theta}_l = \min\{\tilde{\theta}_l, \sum_{d\in D}\sum_{w\in\Omega_d} p^{(w)}_d {\overline{s_{dl}}^{(w)}}^*\}$.
  - *Opt. cuts?* and *add opt. cuts*: In order to add optimality cuts to the first-stage problem, we verify if $({\overline{s_{dl}}^{(w)}}^* - {\overline{\theta_{dl}}^{(w)}}^*)/{\overline{s_{dl}}^{(w)}}^* > \epsilon_1$, in which case a new optimality cut is added for scenario w and day d. After adding the optimality cuts, we increase by one the number of iterations l.
- *Column generation*: The CG method consists of a master problem $B^{LP}_{\widetilde{\mathcal{T}}}$ 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 $B^{LP}_{\widetilde{\mathcal{T}}}$ in an iterative way.

  Let $\lambda^{\Pi,k}_{d1n}$ and δ be the dual variables associated with the constraints (16) and (18) from $B^{LP}_{\widetilde{\mathcal{T}}}$, respectively. Let $\mathcal{S} = \bigcup_{d\in D} ch(O^S_{d1n})$ 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 $O^S_{d1n}, d \in D$. Let $G(\mathcal{N}, \mathcal{A})$ be a directed acyclic graph, composed of a set of nodes $\mathcal{N} = \{v_s \mid s \in \mathcal{S} \cup \{v_b, v_e\}\}$, where v_s corresponds to shift s and v_b, v_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 period, working time, and length including breaks), a "reduced cost contribution" corresponding to value of the dual variable $\lambda^{\Pi,k}_{d1n}$. 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 $B^{LP}_{\widetilde{\mathcal{T}}}$ 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 with the minimum and maximum number of working days in a tour, with the minimum and maximum tour length in time intervals, with the maximum number of days-off, and with 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 $B^{LP}_{\tilde{\mathcal{T}}}$ is as follows:

- *Solve $B^{LP}_{\tilde{\mathcal{T}}}$, get dual sol.*: In this step of the algorithm, we solve problem $B^{LP}_{\tilde{\mathcal{T}}}$ to get the dual solution (λ, σ) that will be sent to the pricing subproblem.
- *Solve pricing subp.*: New variables (tours) for the $B^{LP}_{\tilde{\mathcal{T}}}$ 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 $B^{LP}_{\tilde{\mathcal{T}}}$*: 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 $B^{LP}_{\tilde{\mathcal{T}}}$ 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 ${v^{\pi,k}_{d1nl}}^*$. In this step, we also compute the value of the upper bound as $\bar{\theta}_l = \sum_{w \in \Omega} \sum_{d \in D} p^{(w)}_d {s^{(w)}_{dl}}^*$ and the gap with respect to the approximated upper bound: $100 \times (\bar{\theta}_l - \tilde{\theta}_l)/\bar{\theta}_l$. 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) which is defined as $VSS = EEV - HN$. HN corresponds to the value of the two-stage stochastic programming problem and EEV corresponds to the expected value of the *expected-value problem* (EV). 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 $HN = \bar{\theta}_l \geq \tilde{\theta}_l$.

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

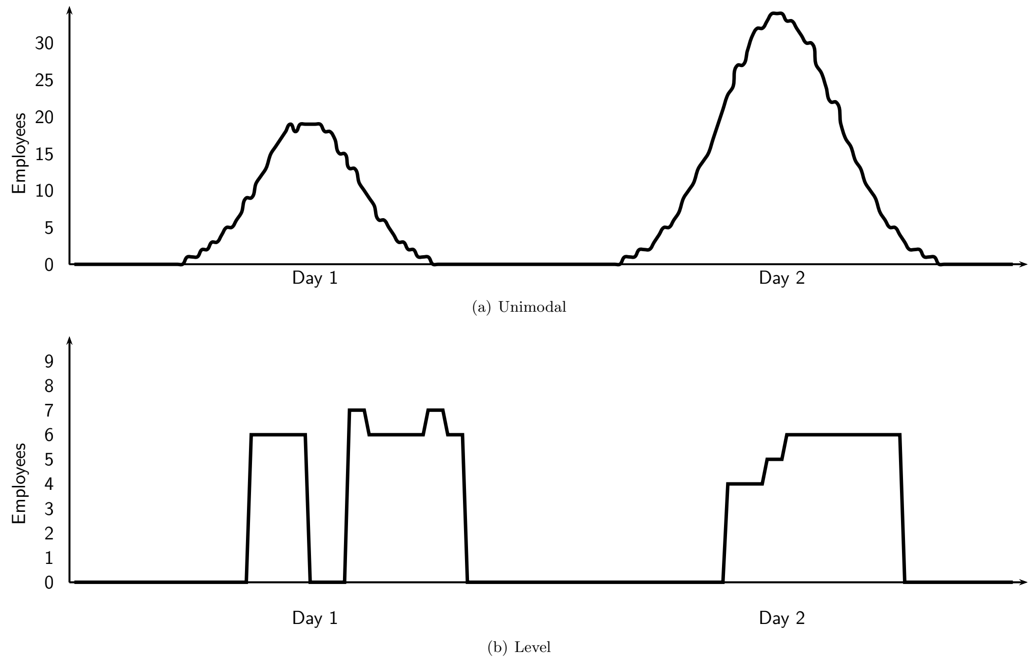

Figure 4 – Deterministic demand profiles.

the second-stage problems was solved by using the barrier method of CPLEX version 12.5.0.1. We set a time limit of 3 hours to solve each instance. Additionally, a relative gap tolerance of 0.01 was set as a stopping criterion for solving the MILPs with CPLEX. The value of tolerances ϵ_1, ϵ_2 were set to 0.0001 and 0.01, respectively.

5.1 Instances Generation

The set of instances used to test our method is divided into two groups: randomly generated instances and real instances from a small retail shop. The deterministic demand profiles for the set of random instances were generated such that they follow a unimodal behavior. The deterministic demand profiles for the real instances, are presented in Côté et al. [8]. These demand profiles present a constant (uniform) demand across hours (level behavior). Figure 4 shows an illustration, over two days, of the demand profiles used for the computational experiments.

Stochastic instances were created by adding to the deterministic demand profile, a random perturbation that follows a discrete uniform distribution. We created instances with 11, 49, 81 and 125 scenarios. Their description follows.

- Instances with 11 scenarios: In this group of instances, one large perturbation is generated for the complete week. Such perturbation follows a discrete uniform distribution between -5 and 5.

- Instances with 49 scenarios: In this group of instances, two perturbations that follow a discrete uniform distribution between -3 and 3 are generated. The first perturbation will affect the first 12 hours of each day during the week, while the second perturbation will affect the last 12 hours of each day during the week.

- Instances with 81 scenarios: In this group of instances, we generate perturbations for each day of the week every two hours between 10am and 6pm. Such perturbations follow a discrete uniform distribution between -1 and 1.

- Instances with 125 scenarios: In this group of instances, we generate perturbations for each day of the week every two hours between 11am and 5pm. The perturbations follow a discrete uniform distribution between -2 and 2.

It is important to highlight that the demand is not perturbed when $b_{dij}^{(w)} = b_{dij} + \xi_{dij}^{(w)} \leq 0$. However, this case only applies when $b_{dij} = 0$ because when the demand takes a positive value, this value is always higher than the value of the lower realization of the stochastic perturbation $\xi_{dij}^{(w)}$.

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.

Tour generation

1. The planning horizon is seven days, where each day is divided into 96 time intervals 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 interval 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 one hour 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.

Let a_j be a terminal symbol that defines a time interval 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 $\Pi \in P$, $\Pi \rightarrow_{[\text{min, max}]}$ restricts the subsequences generated by a given production to a length between a minimum and maximum number of periods. The grammar and the productions that define the multi-activity shifts are as follows:

$$
\begin{aligned}
G =(&\Sigma = (a_j \ \ \forall j \in J, b, l, r),\\
&N = (S, F, Q, N, W, A_j \ \ \forall j \in J, B, L, R), P, S),\\
&S \rightarrow RFR|FR|RF|RQR|QR|RQ|RNR|NR|RN, \ B \rightarrow b, \ L \rightarrow llll,\\
&F \rightarrow_{[38,38]} NLN, \ Q \rightarrow_{[25,25]} WBW,\\
&N \rightarrow_{[17,17]} WBW, \ R \rightarrow Rr|r,\\
&W \rightarrow_{[4,20]} A_j \ \ \forall j \in J, \ A_j \rightarrow A_j a_j | a_j \ \ \forall j \in J.
\end{aligned}
$$

5.3 Computational Results

Tables 1 - 2 present the computational results on stochastic weekly instances dealing with up to five work activities. Ten different demands were tested for each activity (*Nb.Act*) and for each version on the number of scenarios (*Scen.*). We present the average CPU time in seconds to solve the problem (*T. time*), the average CPU time spent in the CG approach (*Time CG*), which includes the time to solve the pricing subproblems and the time to solve the LP relaxation when new columns are added, the average CPU time to solve the first-stage problem (*Time F-S*), and the average CPU time to solve the second-stage problems (*Time S-S*). The average gap between the best upper bound and best lower bound is presented in *Gap1*. This gap is computed as: *Gap1*$= 100 \times (\bar{\theta} - \underline{\theta})/\bar{\theta}$. 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 $\bar{\theta}$ and the approximated upper bound $\tilde{\theta}$: *Gap2*$= 100 \times (\bar{\theta} - \tilde{\theta})/\bar{\theta}$. *Conv.* presents the number of instances that converged to a near-optimal solution, i.e., the algorithm stopped when no more optimality cuts are added to the first-stage model. The average value of the stochastic solution (*VSS*), in percentage, is presented in the last column. This value is computed as: $VSS = 100 \times (EEV - HN)/EEV$ and it is only calculated for instances that converged (*Conv.*=1).

Scen.	*Nb. Act*	*T. time*	*Time CG*	*Time F-S*	*Time S-S*	*Gap1*	*Gap2*	*Conv.*	*VSS*
11	1	242.75	23.04	11.14	185.99	1.01%	0.00%	10	5.40%
	2	1,587.45	97.11	188.8	1,123.27	0.91%	0.00%	10	9.05%
	3	2,114.18	94.6	200.1	1,655.17	0.85%	0.00%	10	8.63%
	4	2,602.97	101.37	133.23	2,227.37	0.87%	0.00%	10	8.97%
	5	3,502.28	117.21	134.1	3,092.78	1.09%	0.02%	10	7.69%
49	1	990.11	47.96	41.04	872.56	0.66%	0.00%	10	2.70%
	2	4,344.01	126.26	144.8	4,001.32	0.88%	0.00%	10	5.48%
	3	8,069.22	568.29	544.45	6,390.64	0.95%	0.00%	7	8.01%
	4	8,463.06	385.56	386.02	7,226.97	1.14%	0.01%	6	8.08%
	5	10,664.8	647.37	488.32	8,632.84	11.85%	0.02%	0	-
81	1	2,171.04	221.63	163.12	1,671.75	0.76%	0.00%	10	0.29%
	2	8,577.45	669.39	661.43	7,023.38	0.86%	0.00%	8	0.92%
	3	9,603.11	292.43	285.7	8,909.98	0.82%	0.01%	7	1.54%
	4	10,537.88	267.65	250.83	9,927.05	1.34%	0.01%	2	1.61%
125	1	2,189.07	212.81	237.51	1,630.95	0.89%	0.00%	10	1.51%
	2	9,415.76	635.78	726.83	7,826.62	0.84%	0.00%	10	2.07%
	3	10,321.03	371.64	419.09	9,409.37	0.74%	0.01%	6	2.78%
	4	10,619.85	192.11	197.47	10,144.59	3.23%	0.01%	2	3.17%

Table 1 – Results on stochastic weekly instances with unimodal demand shape.

Scen.	*Nb. Act*	*T. time*	*Time CG*	*Time F-S*	*Time S-S*	*Gap1*	*Gap2*	*Conv.*	*VSS*
	1	152	18.02	5.34	81.44	0.56%	0.00%	10	1.96%
	2	279.16	14.31	7.78	203.67	0.46%	0.02%	10	0.87%
11	3	616.06	27.13	30.33	452.17	0.72%	0.02%	10	0.81%
	4	607.65	20.78	11.56	504.91	0.50%	0.02%	10	1.55%
	5	962.64	31.6	22.07	763.5	0.54%	0.03%	10	1.27%
	1	575.32	59.7	25.18	387.53	0.60%	0.00%	10	0.00%
	2	1,451.61	75.4	42.33	10,59.27	0.56%	0.01%	10	0.43%
49	3	2,838.99	87.35	153.01	2,123.57	0.83%	0.02%	10	0.42%
	4	3,166.09	93.83	78.68	2,625.82	0.54%	0.01%	10	1.11%
	5	4,557.47	94.44	127.76	3,813.78	0.59%	0.02%	10	0.95%
	1	1301.52	224.75	81.07	694.34	0.66%	0.00%	10	-0.75%
	2	2,812.83	269.58	117.27	1,804.07	0.65%	0.01%	10	-0.77%
81	3	6,177.93	298.59	611.18	4,174.32	0.80%	0.03%	9	0.01%
	4	6,261.87	486.49	169.65	4,719.92	0.53%	0.01%	10	-0.03%
	5	9,766.28	263.26	535.85	7,452.06	0.72%	0.01%	9	0.14%
	1	1,842.3	368.16	155.31	877.35	0.69%	0.00%	10	-0.47%
	2	4,322.38	597.87	283.89	2,376.46	0.58%	0.02%	10	-0.06%
125	3	7,510.87	513.18	436.21	4,891.14	0.64%	0.03%	10	0.37%
	4	8,129.87	607.49	324.75	5,912.99	0.51%	0.01%	10	1.53%
	5	10,818.7	371.01	388.89	7,572.3	22.19%	0.01%	0	-

Table 2 – Results on stochastic weekly instances with level demand shape.

From Table 1, we can conclude that the proposed approach was able to find high quality solutions for most of the instances with up to 125 stochastic scenarios and three work activities. Our method was not able to find solutions that converged for any of the instances with five activities when evaluated on 49 stochastic scenarios, providing solutions with 11.85% average optimality gap. However, for most of the other instances, the final gap is, on average, less than 1.15%. From Table 2, we can conclude that our method was able to find near-optimal solutions for almost all the instances with up to five work activities when evaluated on 11, 49 and 81 stochastic scenarios and with up to four work activities when evaluated on 125 stochastic scenarios. Regarding the CPU time, we observe that the most time-consuming component is related with the LP solution of the second-stage problems, which increases significantly with the number of activities and scenarios.

Although integrality constraints on second-stage variables were relaxed, we can conclude that in most cases, the approximated upper bound $\tilde{\theta}$ was the same as or very close to, the real upper bound $\bar{\theta}$. The above can be observed from the average values of $Gap2$, which are at most 0.03%.

We observe some negative values for the VSS since the algorithm is heuristic. These cases are rare, however: they are observed only on the instances with level demand shape and when the stochastic perturbations of demand do not exhibit a lot of variability (81 and 125 scenarios). The values of the stochastic solution (VSS) depend on the demand profile used. When a unimodal demand profile is tested, the stochastic model prevents the occurrence of additional staffing costs when compared with the expected value problem. On the contrary, when a level demand profile is used, a deterministic approach based on the expected demand appears to be sufficient, especially when 81 and 125 stochastic scenarios are used and not a lot of variability is included in the stochastic perturbation of demand.

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 weekly tours and to daily shifts, 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 [20] in order to generate an initial set of valid cuts in a fast way. Second, we implemented the idea of Papadakos [22] 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.

Computational results suggest that the performance of the method depends on the distribution of the demand profile, as well as on the number of scenarios and work activities included. Specifically, the multi-cut L-shaped method exhibited a better performance, in terms of CPU time, when evaluated on instances with a level demand behavior than when evaluated on instances with a unimodal behavior. However, we observed that the use of the stochastic model has a larger impact on instances with unimodal demand behavior, since it prevents the occurrence of additional staffing costs when compared with the expected value problem. On the contrary, since the value of the stochastic solution is close to zero, a deterministic approach based on the expected demand often appears to be sufficient for instances with a level demand behavior.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[17] K. Kim and S. Mehrotra. A two-stage stochastic integer programming approach to integrated staffing and scheduling with application to nurse management. Technical report, Department of Industrial Engineering and Management Sciences, Northwestern University, 2014.

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

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

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

[21] R. Pacqueau and F. Soumis. Shift scheduling under stochastic demand. Technical Report G-2014-46, GERAD, 2014.

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

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

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

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

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

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

[28] M. I. Restrepo, B. Gendron, and L.-M. Rousseau. Combining Benders decomposition and column generation for multiactivity tour scheduling. Technical report, CIRRELT, 2015.

[29] M. I. Restrepo, B. Gendron, and L.-M. Rousseau. Branch-and-price for multi-activity tour scheduling. *INFORMS Journal on Computing*, forthcoming.

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

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

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

[33] Laurence A Wolsey and George L Nemhauser. *Integer and combinatorial optimization*. John Wiley & Sons, 1999.

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