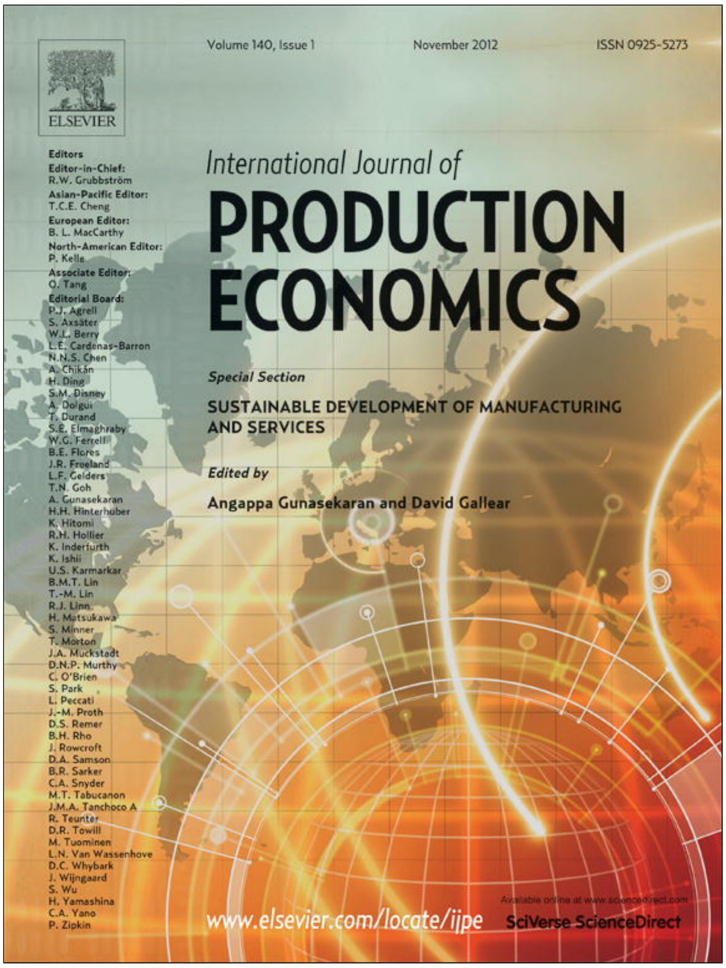
Volume 140, Issue 1
November 2012
ISSN 0925-5273
ELSEVIER
International Journal of
PRODUCTION ECONOMICS
Special Section
SUSTAINABLE DEVELOPMENT OF MANUFACTURING AND SERVICES
Edited by
Angappa Gunasekaran and David Gallear
www.elsevier.com/locate/ijpe
SciVerse ScienceDirect

Int. J. Production Economics 140 (2012) 466–472

Contents lists available at SciVerse ScienceDirect

Int. J. Production Economics

journal homepage: www.elsevier.com/locate/ijpe

Constrained network-based column generation for the multi-activity shift scheduling problem ☆

María I. Restrepo, Leonardo Lozano, Andrés L. Medaglia *

Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería Industrial, Universidad de los Andes, Cr 1E No. 19A-10, ML711 Bogotá, Colombia

ARTICLE INFO

Article history:
Received 30 April 2012
Accepted 25 June 2012
Available online 3 July 2012

Keywords:
Shift scheduling
Column generation
Shortest path with resource constraints

ABSTRACT

Real applications in shift scheduling often require handling rules such as multiple breaks, flexible shift lengths, overtime, multiple activities, among others. Because these rules demand a high level of flexibility, we model the problem as a Multi-Activity Shift Scheduling Problem (MASSP), where multiple activities can be scheduled in a shift. To solve the MASSP, we propose a column generation-based approach. The auxiliary problem is modeled as a Shortest Path Problem with Resource Constraints (SPPRC), where most difficult constraints are embedded in the underlying graph. To illustrate the solution approach, we present our experience solving a real-world problem from a large parking lot operator that schedules security staff and cashiers among several parking lots in Bogotá (Colombia). The results show a significant reduction on the staffing total costs and on man-hours used.

1. Introduction

The Shift Scheduling Problem (SSP) is the problem of selecting a set of shifts to satisfy a daily demand for staff requirements. This problem was first introduced by Edie (1954) to schedule toll booth operators. Later, Dantzig (1954) proposed the first integer programming model to solve the SSP. This pioneering work has led to a vast body of research on more realistic variations of the original model and their solution methods.

Current solution methods for the SSP include: enumeration techniques (Klabjan et al., 2001; Çezik et al., 2001), which explore a complete or partial set of solutions; constraint programming (CP) (Côté et al., 2009), often useful when shift rules are complex; local search (Curtis et al., 2000; Bennett and Potts, 1968) or constructive heuristics (Dijkstra et al., 1991), which provide good feasible solutions quickly; simulation (Zülch et al., 2004), which allows rich models full of detail and exploits the flexibility of human resources; and column generation (CG) combined with heuristic methods (Alefragis et al., 2000; Chu et al., 1997), with dynamic programming (Desrochers and Soumis, 1989), and with CP (Demassey et al., 2006; Yunes et al., 2000, 2005). For a survey of solution methods for the SSP, the reader is referred to Ernst et al. (2004a, 2004b).

Despite of the advances on the methods to solve the SSP, in real-world problems it is often necessary to include complex rules that increase the problem complexity (Quimper and Rousseau, 2010) and give rise to several extensions of the original problem. The Multi-Activity Shift Scheduling Problem (MASSP) is a relevant extension of the SSP where each employee may perform several work activities in the same shift and the assignment of activities to a shift may be restricted by work rules, including activity length, number of breaks, shift starting time, shift length, and maximum number of activities, among others. In the MASSP the shift schedule spans over a planning horizon of one or more days. When longer horizons are involved, this gives rise to the problem known as the *Tour Scheduling Problem*.

Given that the mathematical programming approach to the MASSP yields very large mixed-integer programming formulations (MIP), several researchers have proposed heuristics. Ritzman et al. (1971) proposed an strategy with heuristic assignment rules and a simulation component to schedule workers in a post office over a planning horizon of one week. In their work, they did not consider breaks nor rules related with switching between activities. Loucks and Jacobs (1991), followed by Brusco and Jacobs (1998), proposed a two-stage heuristic strategy to schedule, over one week, employees in fast food restaurants and an airline. However, similar to Ritzman et al. (1971), their work did not consider breaks, rules related with switching between activities, or different shift starting times. Finally, Quimper and Rousseau (2010) took advantage of the rich expressivity provided

☆ "Paper selected from presentations at the "5th International Conference on Production Research—Americas Region", Bogotá, Colombia, July 21–23, 2010. The selection process has been managed by Professor Luis Quezada, Department of Industrial Engineering, University of Santiago of Chile (USACH) and Professor José Ceroni, School of Industrial Engineering, Pontifical Catholic University of Valparaíso (PUCV), Chile."

* Corresponding author. Tel.: +57 1 3394949x2880.
E-mail address: amedagli@uniandes.edu.co (A.L. Medaglia).
URL: http://wwwprof.uniandes.edu.co/~amedagli (A.L. Medaglia).

0925-5273/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ijpe.2012.06.030

by regular or context-free languages to model complex scheduling rules. They derived specialized graph structures and used a Large Neighborhood Search (LNS) procedure to generate feasible shifts. They were able to provide near-optimal solutions for single-activity instances and their methodology scales well for instances with up to 10 work activities.

As an alternative to heuristic techniques, Brusco and Jacobs (2000) proposed a compact integer program that allows flexible shift starting times, shift lengths, and breaks placement. Rekik et al. (2004) used Benders decomposition to consider, similar to Brusco and Jacobs (2000), work rules that allow more modeling flexibility in the shifts. Although these authors provided the basis to solve more complex shift scheduling problems, they did not considered the multi-activity context (they solved instances with only one activity over a planning horizon of one week). Recently, Demassey et al. (2006) proposed a CP-based column generation approach to solve the shift scheduling problem in the multi-activity context. In their work, they considered regulation concerning the breaks and switching between activities. Their method is efficient for the linear relaxation of the problem, but when integrality is imposed, their approach only succeeds for small instances of up to three activities. Finally, Côté et al. (2011) developed a new implicit formulation to solve the MASSP and to overcome the scalability and performance problems of the approaches presented in Côté et al. (2007, 2009). They used context-free grammars to model work rules and solved to optimality instances of up to ten work activities defined over a one-day planning horizon.

To solve the MASSP, we propose a column generation approach coupled with a Shortest Path Problem with Resource Constraints (SPPRC) as auxiliary problem. Some of the most difficult constraints (work rules) such as multiple work activities and breaks, different break types, and irregular shift lengths, are tackled while building the underlying graph of the auxiliary problem. We extended an exact algorithm for the Constrained Shortest Path (CSP) to efficiently solve the SPPRC. Our approach solves real-world large-scale instances of up to 16 work activities over a one-week planning horizon, achieving provably near-optimal solutions compared against the bounds obtained from the linear relaxation.

Finally, the paper is organized as follows. In Section 2, we present the definition of the MASSP. Section 3 describes the column generation approach to solve the MASSP. Section 4 defines the underlying network used in the auxiliary problem and describes the core of the shift generation algorithm. In Section 5 we present the case study and the computational experiments. Finally, Section 6 concludes the paper and outlines the future work.

2. The multi-activity shift scheduling problem

Given a planning horizon discretized into *time intervals* of equal length, a set of *work activities*, and a set of feasible *shifts*, the MASSP assigns a number of employees to each shift, minimizes the staffing total cost, and meets the staff requirements for every work activity over the planning horizon. In the MASSP the shifts may include several work activities and *breaks*. Fig. 1 shows a snapshot of a schedule over a 12-time interval planning horizon with four shifts, three work activities (labeled A1, A2 and A3), multiple breaks, and overtime.

In the MASSP the feasibility of the shifts is usually specified by some *work rules*. In our particular case, these rules include: (1) the *regular time shift length* should fall between a minimum and maximum number of time intervals; (2) the *overtime shift length* should not be longer than a maximum number of time intervals; (3) it is possible to work overtime only after regular time has been exhausted; (4) there is a minimum and a maximum number of work activities per shift; (5) there is a minimum and maximum number of breaks per shift; (6) a shift may not start nor end with a break; (7) the length of the work activities in the shift should fall between a minimum and a maximum number of time intervals; (8) a shift may start at any time interval of the day allowing enough time to complete its minimum length; (9) a break is required to switch between work activities; (10) the break length should fall between a minimum and maximum number of time intervals; (11) it is possible to have a break within a work activity, yet the break must be scheduled after a minimum number of time intervals.

3. Column generation scheme for the MASSP

Let Ω, $\mathcal{T}$, and $\mathcal{K}$ be the sets of feasible shifts, time intervals, and work activities, respectively. Let a_{tlj} be a binary parameter that takes the value of 1 if time interval t and work activity l are covered by shift j; it takes the value of 0, otherwise. Let d_{tl} be the staff requirements of work activity l at time interval t. Let c_j be the labor cost of shift j. The integer variable x_j represents the number of employees assigned to shift j. The master problem – $MP(\Omega)$ – for the MASSP follows:

$$\min \quad Z_{MP(\Omega)} = \sum_{j \in \Omega} c_j x_j \tag{1}$$

subject to

$$\sum_{j \in \Omega} a_{tlj} x_j \geq d_{tl}; \quad t \in \mathcal{T}, \; l \in \mathcal{K} \tag{2}$$

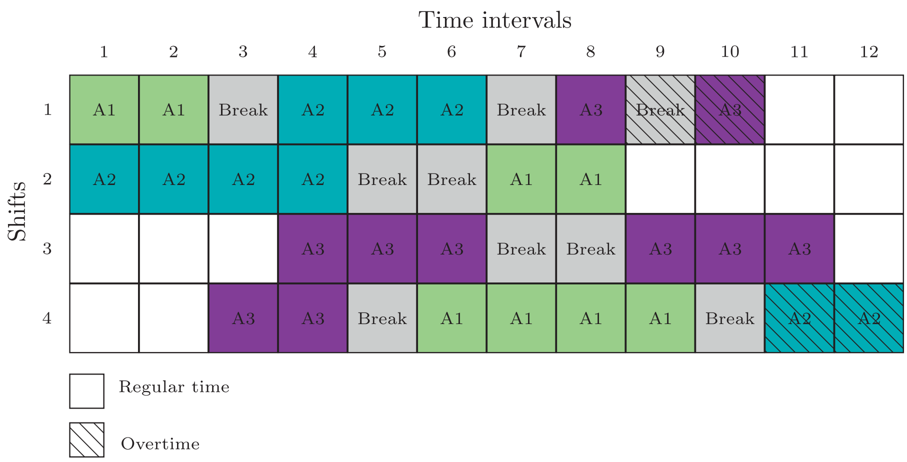

Fig. 1. Example of a multi-activity schedule with breaks and overtime.

$$x_j \in \mathbb{Z}^1_+, \quad j \in \Omega \tag{3}$$

The objective in the $MP(\Omega)$ is to minimize the total staffing cost (1). The set of constraints (2) enforces the staff requirements per time interval t and work activity l. The set of constraints (3) defines the integrality (and nonnegativity) of the decision variables x_j. Note that in the $MP(\Omega)$, defined by (1)–(3), the set of all feasible shifts Ω is known. However, Ω is often very large in practical settings, so we define a reduced set of shifts $\Omega' \subseteq \Omega$ that gives rise to the restricted master problem $RMP(\Omega')$:

$$\min \quad Z_{RMP(\Omega')} = \sum_{j \in \Omega'} c_j x_j \tag{4}$$

subject to

$$\sum_{j \in \Omega'} a_{tlj} x_j \geq d_{tl}; \quad t \in \mathcal{T}, \ l \in \mathcal{K} \tag{5}$$

$$x_j \geq 0, \quad j \in \Omega' \tag{6}$$

Note that in the $RMP(\Omega')$, aside from defining the variables over the reduced set $\Omega' \subseteq \Omega$, the integrality constraints (3) over the variables are now relaxed in (6).

Let $\mathbf{x}$ and $\boldsymbol{\pi}$ be the primal and dual variables associated with the $RMP(\Omega')$, respectively. The reduced cost $\bar{c}_j$ of any given column $\mathbf{a}_j$ is:

$$\bar{c}_j = c_j - \boldsymbol{\pi} \cdot \mathbf{a}_j \tag{7}$$

Under a column generation scheme, the auxiliary problem uses the dual variables $\boldsymbol{\pi}$ to find the minimum reduced cost column at every iteration. If a new column $\mathbf{a}_j$ with negative reduced cost $\bar{c}_j$ is found, the column is added to the $RMP(\Omega')$. Also, note that the coefficients of the newly generated column $\mathbf{a}_j$ define a new shift. On the other hand, if the minimal reduced cost of a newly generated column (shift) is at least zero, the optimal solution of the $RMP(\Omega')$ has been found. To obtain an integer solution to the MASSP we solve the $RMP(\Omega')$ forcing the integrality constraints on $\mathbf{x}$. If an integer solution is found, the solution may still not be optimal because only a reduced set of shifts $\Omega' \subseteq \Omega$ has been considered. However, the quality of the near-optimal (integer) solution can be measured against the lower bound achieved with the relaxed solution of $RMP(\Omega')$. As it will be seen in Section 5.2, our approach works remarkably well, even when compared against this strict (lower) bound.

4. The constrained-network auxiliary problem

To find good shifts – columns for $RMP(\Omega')$ – that meet the work rules presented in Section 2, we propose an auxiliary problem for the MASSP based on a Shortest Path Problem with Resource Constraints (SPPRC), where a path corresponds to a feasible shift. The time shift lengths (both regular and overtime), the number of work activities, and the number of breaks are modeled as *resources*. The regular time shift length leads to *local resource constraints* defined over each node, while the other resources define *global resource constraints*.

Formally, let $G = (\mathcal{N}, \mathcal{A})$ be a directed acyclic graph with a set of nodes $\mathcal{N} = \{v_{(t,l)} | t \in \mathcal{T}, l \in \mathcal{K}\} \cup \{v_s, v_e\}$, where v_s and v_e are the source and sink nodes, respectively. The set of arcs $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4$ is divided into four subsets depending on the type of the arc, namely: shift start $\mathcal{A}_1 = \{(v_s, v_{(t,l)}) | v_{(t,l)} \in \mathcal{N}\}$; work activity $\mathcal{A}_2 = \{(v_{(t,l)}, v_{(t',l)}) | v_{(t,l)}, v_{(t',l)} \in \mathcal{N}, t' > t\}$; break or work activity switch $\mathcal{A}_3 = \{(v_{(t,l)}, v_{(t',l')}) | v_{(t,l)}, v_{(t',l')} \in \mathcal{N}, t' \geq t\}$; and shift end $\mathcal{A}_4 = \{(v_{(t,l)}, v_e) | v_{(t,l)} \in \mathcal{N}\}$. Fig. 2 shows the graph's general form including the sets of arcs $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ and $\mathcal{A}_4$, a planning horizon $|\mathcal{T}|$, and $|\mathcal{K}|$ different work activities.

The size of graph G depends on $|\mathcal{T}|$, $|\mathcal{K}|$, and the set of work rules defined. In the worst-case scenario, where the set of work rules is very permissive, the number of arcs can be calculated as: $|\mathcal{A}_1| = |\mathcal{T}| \cdot |\mathcal{K}|$ considering a start arc for every activity and time interval; $|\mathcal{A}_2| = (((|\mathcal{T}|-1) \cdot |\mathcal{T}|)/2) \cdot |\mathcal{K}| \cdot 2$ considering both regular and overtime arcs for all work activities; $|\mathcal{A}_3| = ((|\mathcal{T}| \cdot (|\mathcal{T}|+1))/2) \cdot |\mathcal{K}|^2 \cdot 2$ considering switches between all activities; and $|\mathcal{A}_4| = |\mathcal{T}| \cdot |\mathcal{K}| \cdot 2$ considering both regular and overtime shift end arcs for every activity and time interval. In the graph, every combination of time interval and work activity is represented by a node $v_{(t,l)} \in \mathcal{N}$, thus the number of nodes is $|\mathcal{N}| = |\mathcal{T}| \cdot |\mathcal{K}| + 2$. In summary, the number of arcs dominates the size of the graph, which ends having a storage requirement of $O(|\mathcal{T}|^2 \cdot |\mathcal{K}|^2)$.

Let $c_{v_{(t,l)}}$ be the regular labor cost of working at node $v_{(t,l)}$ and $b_{v_{(t,l)}}$ be the regular cost of assigning a break at node $v_{(t,l)}$; likewise, $c^o_{v_{(t,l)}}$ and $b^o_{v_{(t,l)}}$ are the corresponding costs for overtime work. Let $\pi_{v_{(t,l)}}$ be the dual variable associated to the staff requirements constraints (5) and node $v_{(t,l)}$. Each arc has a four-dimensional vector of attributes defined based on its type. The first attribute $\beta(v_{(t,l)}, v_{(t',l')})$ is a binary parameter that takes the value of 1 if the arc represents a break; and takes the value of 0 if the arc represents a work activity. The second attribute, denoted by $\gamma(v_{(t,l)}, v_{(t',l')})$, takes the value of 1 if the arc covers overtime intervals; and takes the value of 0 if the arc covers regular time intervals. The third attribute denoted by $\delta(v_{(t,l)}, v_{(t',l')})$ represents the number of nodes (time intervals) covered by the arc. The fourth and last attribute $\alpha(v_{(t,l)}, v_{(t',l')})$ represents the reduced cost contribution resulting from traversing the arc. Note that graph $G = (\mathcal{N}, \mathcal{A})$ allows *multiarcs*, that is, arcs with different attribute values, but sharing the same tail and head nodes.

For shift-start arcs $(v_s, v_{(t,l)}) \in \mathcal{A}_1$, the attributes are defined as follows. Attribute $\beta(v_s, v_{(t,l)}) = 1$ because these arcs do not represent a work activity, just the beginning of a shift. Attribute $\gamma(v_s, v_{(t,l)}) = 0$ because overtime arcs from node v_s to any node $(v_{(t,l)})$ do not exist. Attribute $\delta(v_s, v_{(t,l)}) = 1$ because these arcs cover just the first time interval of the shift. Finally, the reduced cost contribution $\alpha(v_s, v_{(t,l)})$ is calculated as

$$\alpha(v_s, v_{(t,l)}) \triangleq c_{v_{(t,l)}} - \pi_{v_{(t,l)}} \tag{8}$$

For work-activity arcs $(v_{(t,l)}, v_{(t',l)}) \in \mathcal{A}_2$, that is, arcs going from node $v_{(t,l)}$ to node $v_{(t',l)}$ in the same work activity, the attributes are defined as follows. The attribute $\beta(v_{(t,l)}, v_{(t',l)}) = 0$ because these arcs represent always a work activity, not a break. Because arc $(v_{(t,l)}, v_{(t',l)})$ can represent work on regular time or overtime, $\gamma(v_{(t,l)}, v_{(t',l)})$ can be either 0 or 1. These arcs cover the time intervals from $v_{(t+1,l)}$ to $v_{(t',l)}$, thus the value of the attribute $\delta(v_{(t,l)}, v_{(t',l)}) = t' - (t+1)$ is equivalent to the number of nodes covered by the arc. Finally, the reduced cost contribution $\alpha(v_{(t,l)}, v_{(t',l)})$ depends on $\gamma(v_{(t,l)}, v_{(t',l)})$ and is defined by

$$\alpha(v_{(t,l)}, v_{(t',l)}) \triangleq \begin{cases} \sum_{k=t+1}^{t'} (c_{v_{(k,l)}} - \pi_{v_{(k,l)}}), & \text{if } \gamma(v_{(t,l)}, v_{(t',l)}) = 0 \\ \sum_{k=t+1}^{t'} (c^o_{v_{(k,l)}} - \pi_{v_{(k,l)}}), & \text{if } \gamma(v_{(t,l)}, v_{(t',l)}) = 1 \end{cases} \tag{9}$$

Arcs $(v_{(t,l)}, v_{(t',l')}) \in \mathcal{A}_3$ represent either a switch between work activities or a break within a work activity. The attributes $\beta(v_{(t,l)}, v_{(t',l')}) = 1$ and, similar to the latter case (arcs in $\mathcal{A}_2$), $\gamma(v_{(t,l)}, v_{(t',l')})$ can be either 0 or 1. Attribute $\delta(v_{(t,l)}, v_{(t',l')}) = t' - (t+1)$ (number of nodes covered by the arc). The reduced cost contribution $\alpha(v_{(t,l)}, v_{(t',l')})$ depends on $\gamma(v_{(t,l)}, v_{(t',l')})$ and is defined by

$$\alpha(v_{(t,l)}, v_{(t',l')}) \triangleq \begin{cases} \sum_{k=t+1}^{t'} b_{v_{(k,l)}}, & \text{if } \gamma(v_{(t,l)}, v_{(t',l')}) = 0 \\ \sum_{k=t+1}^{t'} b^o_{v_{(k,l)}}, & \text{if } \gamma(v_{(t,l)}, v_{(t',l')}) = 1 \end{cases} \tag{10}$$

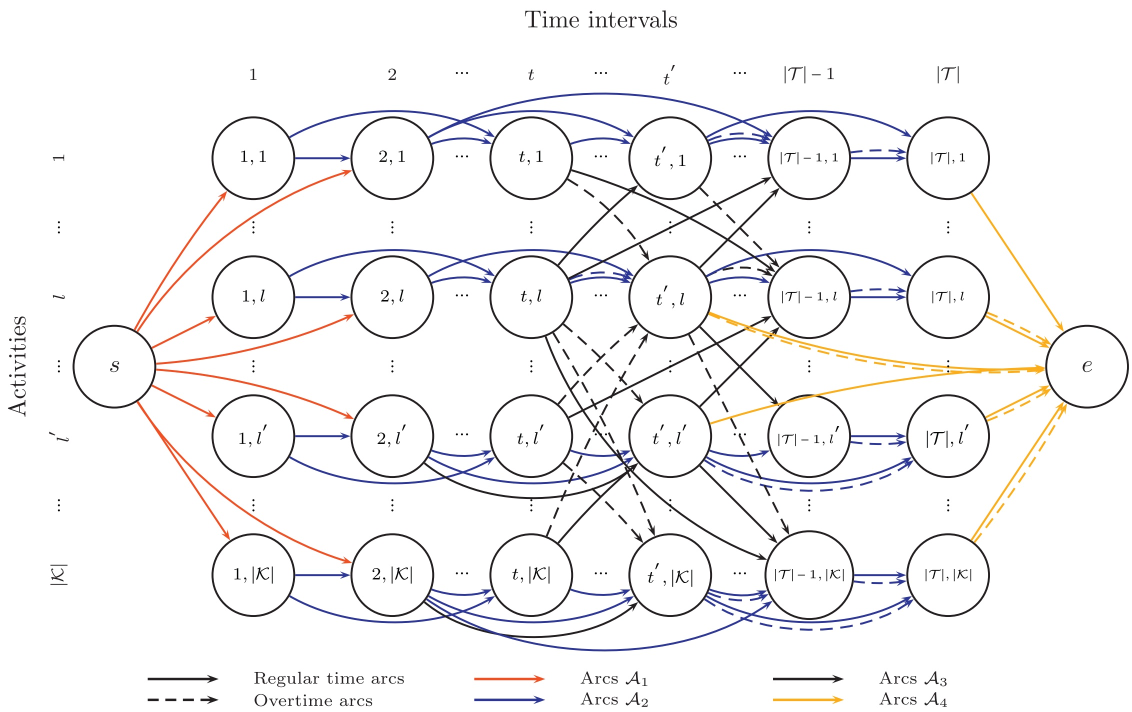

Fig. 2. Generic graph behind the MASSP auxiliary problem.

For shift-end arcs $(v_{(t,l)}, v_e) \in \mathcal{A}_4$ the attributes are defined as follows. The attribute $\beta(v_{(t,l)}, v_e) = 0$ because the arc represents the end of a shift. Attribute $\gamma(v_{(t,l)}, v_e)$ can be either 0 or 1 depending on whether the shift uses overtime or not. Finally, these arcs do not cover any time interval, therefore $\delta(v_{(t,l)}, v_e) = \alpha(v_{(t,l)}, v_e) = 0$.

To solve the auxiliary problem we extended an specialized algorithm for the Constrained Shortest Path (CSP) proposed by Lozano and Medaglia (in press) to the SPPRC. The algorithm consists of a depth-first search exploration of the graph with additional *pruning strategies*, which are rules that avoid exploring a vast number of paths based on the constraints. Similar to branch and bound, optimality is guaranteed because a complete enumeration of all possible paths is done implicitly. Because the strength of the algorithm relies on the pruning strategies, we devised several strategies on the following constraints: the minimum and maximum shift length for regular time over each node, the maximum shift length for overtime, the minimum and maximum number of work activities and breaks allowed in the shift, overtime can be assigned only after regular time, and shifts may not start nor end with a break. These constraints are modeled as resources that are consumed by the paths (shifts), in such a way that as the graph is being explored, the algorithm checks for the feasibility of the partial path. If the path (or all possible extensions) is proved to be infeasible, it is pruned. The algorithm is extremely fast and it is able to solve the auxiliary problem for the case study described in Section 5 in a fraction of a second.

5. Case study

We applied our approach to solve the MASSP in one of the largest parking operators in Bogotá, Colombia. The company has more than 400 employees spread over 120 parking lots, grouped by areas according to their proximity. Aside from using the proposed approach to schedule security staff and cashiers a key objective of the case study was to evaluate the impact of a new policy that allows staff movements between parking lots on the staffing cost and in the man-hours required. The current staff scheduling policy (baseline) does not allow staff movements and is planned over a week and updated every four months.

5.1. Problem description

In the case study, we model the parking lots as work activities and the staff movements between parking lots as changes between work activities. The planning horizon is defined over seven days (one week), where each day is divided into 48 periods of 30 min. Each shift may cover more than one parking lot, meaning that, staff movements between parking lots are allowed during the same shift. Every shift follows the work rules presented in Section 2, more specifically: all shifts have the same regular time length of 8 h; overtime is allowed and its maximum length is 4 h; the shifts must cover at least 3 h at one parking lot before switching to another parking lot; if a shift only covers one parking lot, the shift does not have a break, but if a shift covers more than one parking lot, there must be a break representing the switching between parking lots (the break length depends on the distance between parking lots); any given shift involves working in at most two parking lots; the shifts may start at any time interval of the day; and, finally, the shifts span over a maximum of two consecutive days. The latter has the important implication that the auxiliary problem is separable into seven different subproblems considering each pair of consecutive days (i.e., Monday–Tuesday, Tuesday–Wednesday, and so forth).

5.2. Computational experiments

We tested our approach on several real instances with data provided by the company. The instances are labeled following the format `AX-TILX-DX-VX` where `A`, `TIL`, `D` and `V` are the prefixes for the number of activities (parking lots), the time interval length in minutes, the planning horizon in days, the instance's version depending on whether overtime work is allowed (value of 1) or not (value of 2), respectively. We also measured the effect of the size of the time intervals on the performance of our approach.

Table 1
Computational performance on instances with a one-day planning horizon.

Instance	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{A}\rvert$	LR-RMP (s)	IP-RMP (s)	Gap (%)	Iter.	Columns
A5-TIL30-D1-V1	242	3087	2.39	2.41	0	123	123
A5-TIL30-D1-V2	242	2247	0.42	0.44	0	17	17
A7-TIL30-D1-V1	338	5047	16.61	16.67	0	388	388
A7-TIL30-D1-V2	338	3815	2.81	2.88	0	96	96
A13-TIL30-D1-V1	626	14,704	404.47	404.70	0	1909	1909
A13-TIL30-D1-V2	626	12,359	38.52	38.61	0	396	396
A16-TIL30-D1-V1	770	12,516	131.56	131.70	0	997	997
A16-TIL30-D1-V2	770	10,205	13.53	13.59	0	200	200
A21-TIL30-D1-V1	1010	28,447	3650.10	3654.83	0	3776	3776
A21-TIL30-D1-V2	1010	24,639	200.42	202.31	0	727	727

Table 2
Computational performance on instances with a one-week planning horizon.

Instance	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{A}\rvert$	LR-RMP (s)	IP-RMP (s)	Gap (%)	Iter.	Columns
A5-TIL30-D7-V1	482	6589	2629.15	2634.43	0.01	1517	10,619
A5-TIL30-D7-V2	482	4270	16.54	17.05	0.00	81	567
A7-TIL30-D7-V1	674	5180	4471.20	4476.43	0.02	1812	12,684
A7-TIL30-D7-V2	674	3661	81.90	83.08	0.00	154	1078
A13-TIL30-D7-V1[a]	1250	21,542	28,807.95	28,878.46	–	2120	14,840
A13-TIL30-D7-V2	1250	17,293	1749.50	1755.79	0.00	605	4235
A16-TIL30-D7-V1[a]	1538	19,784	28,805.60	28,861.84	–	3199	22,393
A16-TIL30-D7-V2	1538	15,507	624.12	628.29	0.21	363	2541

[a] 8-h limit reached solving the LR-RMP.

The experiments were performed on a DELL Precision 7400 with 8 GB of RAM, two processors Intel® Xeon® X5450 running at 3 GHz, on a 64-bit Windows Vista Ultimate operating system. Both the algorithm by Lozano and Medaglia (in press) and the column generation logic were coded in Java using the compiler on Eclipse SDK version 3.4.0, while the set covering and its relaxation were solved using Xpress-MP Optimizer version 19.00.00.

Table 1 reports the computational effort of the proposed column-generation scheme on ten one-day planning horizon instances with up to 21 different work activities. We report the name of the instance, the number of nodes and arcs in the graph of the auxiliary problem, the time (in seconds) required to solve the linear relaxation of the restricted master problem (LR-RMP), the total execution time (including column generation) to find the best integer solution (IP-RMP) from the generated columns for LR-RMP, the gap between the relaxed solution and the integer solution, the total number of calls to the auxiliary problem (column labeled Iter.), and the total number of columns generated in the problem. We set a time limit of 8 h to solve the LR-RMP.

With the one-day planning horizon, our approach solved to optimality (zero gap) all instances (ten out of ten). More precisely, six out of ten instances were solved in less than 1 min; three out of ten instances were solved in less than 7 min; and just one barely exceeded 1 h. It is noteworthy that the time needed to solve the set covering problem is almost negligible (under 4 s).

We also wanted to explore the performance of our approach over instances with a longer one-week planning horizon. Table 2 reports the computational effort on eight long-horizon instances with up to 16 different work activities. Note that for these instances solving the auxiliary problem generates multiple columns at each iteration.

As noted in Table 2, our approach was able to solve six out of eight instances near to optimality. Even for the two instances that reached the 8-h limit, we were able to find integer feasible solutions, but we were not able to assess its quality (gap). For the case where overtime is not allowed, we found the solution for all instances in less than 11 min; solving even large-scale instances with up to 16 different work activities over a planning horizon of one week. In the case where overtime is allowed, we were able to consistently solve instances with up to seven different work activities over a planning horizon of one week. These (overtime) instances are more difficult because the number of overtime arcs increases the size of the network in the auxiliary problem. Regardless of the instance, the quality of the solutions found is consistently high, as the small gap (always under 0.3%) guarantees nearly optimal solutions.

In our last experiment, we tested the effect of the time interval lengths on the performance of our approach. We derived new instances with different time interval lengths ranging from 15 to 60 min from a medium-sized instance with seven work activities. Table 3 reports the effect of the granularity of the time intervals on the computational effort.

Our approach was able to solve to optimality the instance with time interval length of 60 min in less than 3 min. For the instances with time interval length of 45 and 30 min, we found near optimal solutions with gaps under 0.3%. Although the instance with the 15-min time interval reached the time limit, we still found an integer feasible solution in this case. As expected, by the theoretical analysis outlined in Section 4, the size of the underlying network of the auxiliary problem is bounded by a polynomial which depends quadratically on the number of time intervals $|\mathcal{T}|$. The longer the time interval lengths, the smaller the network, thus the auxiliary problem becomes easier to solve.

5.3. Comparing the new and current staffing policies

We compared the results allowing staff movements between parking lots in the same shift against the current policy (where movements are not allowed) on the long-horizon instances. We measured the improvement on total staffing cost (*TSC*) given by (4) and the improvement on the man-hours required (*MHR*)

Table 3
Measuring the effect of the time interval length on the computational effort.

Instance	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{A}\rvert$	LR-RMP (s)	IP-RMP (s)	Gap (%)	Iter.	Columns
A7-TIL60-D7-V1	338	1744	135.00	136.00	0.00	411	2877
A7-TIL45-D7-V1	450	2763	748.97	751.28	0.03	888	6216
A7-TIL30-D7-V1	674	5180	4468.93	4474.25	0.02	1812	12,684
A7-TIL15-D7-V1[a]	1346	17,131	28,808.50	28,831.50	–	2540	17,780

[a]8-h limit reached solving the LR-RMP.

Table 4
Reduction (in %) on *TSC* and *MHR* by allowing staff movements between parking lots.

Instance	*TSC* (%)	*MHR* (%)
A5-TIL30-D7-V1	12.4	20.8
A5-TIL30-D7-V2	4.8	9.9
A7-TIL30-D7-V1	8.2	12.7
A7-TIL30-D7-V2	2.3	2.7
A13-TIL30-D7-V1	0.3	1.5
A13-TIL30-D7-V2	0.0	0.0
A16-TIL30-D7-V1	16.0	18.7
A16-TIL30-D7-V2	7.5	9.6
Average	6.4	9.5

calculated as follows:

$$MHR \triangleq \frac{\sum_{t \in \mathcal{T}} \sum_{l \in \mathcal{K}} \sum_{j \in \Omega'} a_{tlj} x_j^*}{\sum_{t \in \mathcal{T}} \sum_{l \in \mathcal{K}} d_{tl}} \tag{11}$$

Note that in (11), the ideal case has a limit value of $MHR=1$. Table 4 summarizes the reduction on total staffing cost and on man-hours required resulting from allowing staff movements between parking lots.

In most cases we achieved a significant reduction on both measures, improving up to 16% on *TSC* and 20.8% on *MHR*. Note that one of the instances without a provable optimality gap (`A16-TIL30-D7-V1`) according to Table 1, reduces significantly both *TSC* and *MHR* (16% and 18.7%, respectively).

6. Concluding remarks

In this paper we propose an approach based on column generation to solve multi-activity shift scheduling problems. Within the column generation scheme we define an auxiliary problem based on a constrained network. The network structure of the auxiliary problem allows us to incorporate some of the most difficult constraints while building the graph and to generate shifts (columns) using an extension of an specialized algorithm for the constrained shortest path problem. The flexibility of the approach allows us to incorporate different work rules depending on problem specific information. We were able to efficiently achieve provably near-optimal solutions in instances of up to 16 work activities over a one-week planning horizon; and provably optimal solutions in instances of up to 21 work activities over a one-day planning horizon. Based on a theoretical and experimental analysis, we show that our approach is sensitive to the time interval length; as we increase the length of the time intervals, the size of the auxiliary problem reduces and the overall performance of the algorithm increases.

We tested our approach in a real application from a parking lot operator that schedules security staff and cashiers over one hundred parking lots in the city of Bogotá. We found that by allowing staff movements between parking lots in the same shift the company is able to reduce the staffing total costs and the man-hours required. Also, our approach significantly reduces the time to build the staff schedule and allows the company to evaluate several what-if scenarios aimed to improve their staffing policies.

Acknowledgments

The authors would like to thank Carlos E. González, Alvaro Estupiñan, and Natalia Carrillo at Parking International for sharing with us their data and knowledge throughout the development of this project. The authors would also like to thank Dawn Mazzanti at FICO who provided us with Xpress-MP licenses under the Academic Partner Program.

References

Alefragis, P., Sanders, P., Takkula, T., D., W., 2000. Solving large scale crew scheduling problems. Annals of Operations Research 99 (1), 141–166.
Bennett, B., Potts, R., 1968. Rotating rosters for a transit system. Transportation Science 2 (1), 14–34.
Brusco, M.J., Jacobs, L.W., 1998. Personnel tour scheduling when starting time restrictions are present. Management Science 44 (4), 534–547.
Brusco, M.J., Jacobs, L.W., 2000. Optimal models for meal-break and start-time flexibility in continuous tour scheduling. Management Science 46 (12), 1630–1641.
Çezik, T., Günük, O., Luss, H., 2001. An integer programming model for the weekly tour scheduling problem. Naval Research Logistics 48 (7), 607–624.
Chu, H.D., Gelman, E., Johnson, E.L., 1997. Solving large scale crew scheduling problems. European Journal of Operational Research 97 (2), 260–268.
Côté, M.-C., Gendron, B., Rousseau, L.-M., 2007. Modeling the regular constraint with integer programming. In: Pascal Van Hentenryck, Laurence A. Wolsey (Eds.), Lecture Notes in Computer Science, Proceedings of CPAIOR, vol. 4510, pp. 29–43.
Côté, M.-C., Gendron, B., Rousseau, L.-M., 2011. Grammar-based integer programming models for multi-activity shift scheduling. Management Science 57 (1), 1–13.
Côté, M.-C., Gendron, B., Quimper, C.-G., Rousseau, L.-M., 2009. Formal languages for integer programming modeling of shift scheduling problems. Constraints 16 (1), 54–76.
Curtis, S.D., Smith, B.M., Wren, A., 2000. Constructing driver schedules using iterative repair. In: Proceedings of PACLP, pp. 59–78.
Dantzig, G.B., 1954. A comment on Edie's traffic delays at toll booths. Journal of the Operations Research Society of America 2, 339–341.
Demassey, S., Pesant, G., Rousseau, L.-M., 2006. A cost-regular based hybrid column generation approach. Constraints 11, 315–333.
Desrochers, M., Soumis, F., 1989. A column generation approach to the urban transit crew scheduling problem. Transportation Science 23 (1), 1–13.
Dijkstra, M.C., Kroon, L.G., van Nunen, J.A.E.E., Salomon, M., 1991. A DSS for capacity planning of aircraft maintenance personnel. International Journal of Production Economics 23, 69–78.
Edie, L.C., 1954. Traffic delays at toll booths. Journal of the Operations Research Society of America 2, 107–138.
Ernst, A., Jiang, H., Krishnamoorthy, M., Owens, B., Sier, D., 2004a. An annotated bibliography of personnel scheduling and rostering. Annals of Operations Research 127, 21–144.
Ernst, A., Jiang, H., Krishnamoorthy, M., Sier, D., 2004b. Staff scheduling and rostering: a review of applications, methods and models. European Journal of Operational Research 153 (1), 3–27.
Klabjan, D., Johnson, E., Nemhauser, G., 2001. Solving large airline crew scheduling problems: random pairing generation and strong branching. Computational Optimization and Applications 20 (1), 73–91.

Loucks, J.S., Jacobs, F.R., 1991. Tour scheduling and task assignment of a heterogeneous work force: a heuristic approach. Decision Sciences 22 (4), 719–738.

Lozano, L., Medaglia, A.L. On an exact method for the constrained shortest path problem. Computers & Operations Research. http://dx.doi.org/10.1016/j.cor.2012.07.008, In press.

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

Rekik, M., Cordeau, J.-F., Soumis, F., 2004. Using Benders decomposition to implicitly model tour scheduling. Annals of Operations Research 128, 111–133.

Ritzman, L.P., Krajewski, L.J., Showalter, M.J., 1971. The disaggregation of aggregate manpower plans. Management Science 17 (8), 533–549.

Yunes, T.H., Moura, A.V., de Souza, C.C., 2000. A hybrid approach for solving large scale crew scheduling problems. In: Proceedings of the 2nd International Workshop on Practical Aspects of Declarative Languages, Springer-Verlag, London, UK, pp. 293–307.

Yunes, T.H., Moura, A.V., de Souza, C.C., 2005. Hybrid column generation approaches for urban transit crew management problems. Transportation Science 39 (2), 273–288.

Zülch, G., Rottinger, S., Vollstedt, T., 2004. A simulation approach for planning and re-assigning of personnel in manufacturing. International Journal of Production Economics 90, 265–277.