Hierarchical constraints and their applications in staff scheduling problems

Chao Li · Pieter Smet · Patrick De Causmaecker

Abstract In this work, the combinatorial structures which imply polynomial-time solvability in staff scheduling problems are investigated. We introduce *hierarchical constraints* to emphasize the hierarchical relation among constraints and contribute a characterization for a large class of tractable optimization problems with totally unimodular matrices in their integer linear programs. As a result, polynomial-time solvable personnel scheduling problems in literature can be further extended and generalized, as hierarchical management requirements are often considered in practice. Furthermore, an approach to derive the minimum cost network flow problems from the proposed *hierarchical constraints* is established. The newly obtained insight into the generalized boundary between tractable and intractable staff scheduling constraints enriches the theoretical studies of staff scheduling problems' complexity and may lead to efficient models and methods for complex variations.

Keywords Integer programming · Combinatorial optimisation · Staff scheduling · Polynomial-time models · Hierarchical constraints · Total unimodularity

AI in Flanders, KU Leuven Research Fund RKU-D2932-C24/17/012
Grants from the China Scholarship Council (No. 201703170230)

C. Li
KU Leuven, Department of Computer Science, CODeS, KULAK
E. Sabbelaan 53, 8500 Kortrijk, Belgium
E-mail: chao.li@kuleuven.be

P. Smet
KU Leuven, Department of Computer Science, CODeS
Gebroeders De Smetstraat 1, 9000 Gent, Belgium
E-mail: pieter.smet@kuleuven.be

P. De Causmaecker
KU Leuven, Department of Computer Science, CODeS, KULAK
E. Sabbelaan 53, 8500 Kortrijk, Belgium
E-mail: patrick.decausmaecker@kuleuven.be

1 Introduction

Staff scheduling or personnel scheduling is a common operational management challenge with significant impact on operating costs, employee satisfaction, etc. As practical personnel scheduling problems constantly arise from real world applications, various methods including exact algorithms, heuristics and meta-heuristics have been extensively investigated [4, 12, 2]. However, theoretical studies of their models are relatively limited, as their importance has been underestimated [11].

Many staff scheduling problems are NP-hard due to the presence of specific constraints [8]. Recently, polynomial time solvable models in staff scheduling have attracted growing attention, since they incorporate representative and fundamental constraints involved in many industrial variants and they are straightforward to analyze. The first systematic study on staff scheduling models was presented by Brucker, et al [3], where several polynomial-time solvable rostering problems were recognized using minimum cost network flow models. The class of tractable rostering problems was extended by applying new techniques of network flow reformulation [1, 10, 11, 7]. However, these studies lack generality, as they are based on network flow models to be customized for different problems.

To obtain general insights into characteristics of tractable staff scheduling problems, we consider the polyhedra of associated linear programs and find special hierarchical relations between constraints to develop a sufficient condition for identifying tractable scheduling problems. Models in literature [10, 11] are further extended to incorporate more realistic hierarchical requirements while preserving the polynomial-time solvability. Furthermore, a general method to derive minimum cost network flow problems from a collection of the proposed *hierarchical constraints* is presented to link this work with previous studies of staff scheduling based on network flow models.

This paper is organized as follows. Section 2 introduces basic terminology and properties of integer linear programs. Section 3 presents the hierarchical structures in constraints which make its integer program tractable and a consequent algorithm to identify the polynomial-time solvability of integer linear programs. Examples and applications of hierarchical constraints in staff scheduling problems are explored in Section 4. Finally, Section 5 includes conclusions and future work.

2 Preliminaries

The integer linear program (ILP) is one of the most frequently used models in staff scheduling. We consider an ILP P_0 in a generic form:

$$P_0 : min\{\mathbf{c}^{\mathrm{T}}\mathbf{x} : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0, \mathbf{x} \in \mathbb{Z}^n\} \tag{1}$$

of which $\mathbf{c}, \mathbf{b}$ are vectors and A is a matrix with integer entries.

$H = \{x|A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0\}$ defines the polyhedron of the linear programming relaxation (LPR) of P_0. In general, optimal solutions of the LPR of P_0 can be fractional and infeasible, but total unimodularity of A is an important property which guarantees the integrality, as shown in Theorem 1.

Definition 1 *A matrix $\mathbf{A}_{m\times n}$ is totally unimodular (TU) if each square submatrix of A has determinant in $\{0, 1, -1\}$ [9].*

Theorem 1 *A matrix $\mathbf{A}$ is totally unimodular if and only if the polyhedron $\{x|A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0\}$ is integral for any integral vector $\mathbf{b}$ in P_0 [5].*

According to Theorem 1, if $\mathbf{A}$ is TU and constants $\mathbf{b}$ are all integers, P_0 is is solvable in polynomial time, since its optimal solutions are the same as its LPR's solutions.

3 Hierarchical constraints and their systems

In this section, we introduce hierarchical constraints which are key in establishing the tractability of the aforementioned ILP formulation P_0.

As shown in Section 2, constraints coefficients determine the complexity of an ILP. We consider the coefficients in each constraint as a vector and apply the Hadamard (entry-wise) product of constraint vectors to define the hierarchical relationship between constraints.

Definition 2 *The Hadamard product of two vectors $\mathbf{a} = (a_1, ..., a_n)$ and $\mathbf{b} = (b_1, ..., b_n)$ is denoted as $\mathbf{a} \odot \mathbf{b} = (a_1 b_1, ..., a_n b_n)$ [6].*

Definition 3 *Two constraints C_1, C_2 are hierarchical, if their coefficient vectors $\mathbf{c}_1, \mathbf{c}_1$ has a Hadamard product such that $\mathbf{c_1} \odot \mathbf{c_2} \in \{\mathbf{0}, \pm\mathbf{c_1}, \pm\mathbf{c_2}\}$. Additionally, every entry of $\mathbf{c_2}, \mathbf{c_2}$ must be in $\{0, 1, -1\}$ and constants in their constraints must be integers.*

The following is an explanation of the hierarchy of two constraints C_1, C_2, based on the vectorized computation of their constraint coefficient vectors $\mathbf{c_1}, \mathbf{c_1}$:

1) $\mathbf{c_1} \odot \mathbf{c_2} = \mathbf{0} \iff$ the two constraints are disjoint, i.e., they contain no common variable (with nonzero coefficients). For example, $C_1 : 1x_1 + 1x_2 + 0x_3 + 0x_4 \leq b_1$ and $C_2 : 0x_1 + 0x_2 + 1x_3 + 1x_4 \leq b_2$.

2) $\mathbf{c_1} \odot \mathbf{c_2} = \mathbf{c_1}$ or $-\mathbf{c_1} \iff$ all the variables in C_1 are included in C_2. For instance, $C_1 : 1x_1 + 1x_2 + 0x_3 + 0x_4 \leq b_1$ and $C_2 : -1x_1 - 1x_2 - 1x_3 - 1x_4 \leq b_2$;

3) $\mathbf{c_1} \odot \mathbf{c_2} = \mathbf{c_2}$ or $-\mathbf{c_2} \iff C_1$ has all the variables in C_2. In other words, C_1 contains C_2;

4) $\mathbf{c_1} \odot \mathbf{c_2} \notin \{0, \pm\mathbf{c_1}, \pm\mathbf{c_2}\} \iff C_1$ and C_2 are not hierarchical.

The relationship level of two hierarchical constraints is defined as follows.

Definition 4 *The hierarchical level of a constraint C_1 is lower than that of a constraint C_2 denoted as $C_1 \leq_H C_2$, if their coefficient vectors have a Hadamard product $\mathbf{c_1} \odot \mathbf{c_2} \in \{\pm\mathbf{c_1}\}$ and $\mathbf{c_1} \neq \pm\mathbf{c_2}$. Furthermore, $C_1 <_H C_2$ donates the case that there is no other constraint C_3 with a hierarchical level between that of C_1 and C_2, i.e., $C_1 \leq_H C_3 \leq_H C_2$.*

Since there are usually more than two constraints in an ILP of P_0, a special collection of constraints is defined in Definition 4.

Definition 5 *A hierarchical constraint system is a collection of constraints in which every pair of constraints is hierarchical.*

For example, constraints $x_1 + x_2 + x_3 \leq b_1$, $-x_2 - x_3 \leq b_2$ and $-x_1 \leq b_3$ are a system of hierarchical constraints.

Theorem 2 *If the constraints of an ILP consist of no more than two hierarchical constraint systems, the problem can can be solved in polynomial time.*

Proof According to Theorem 1, Definition 3 and Definition 5, we only need to prove total unimodularity in the constraint coefficient matrix A of an ILP with at most two hierarchical constraint systems $\mathcal{H}_1$ and $\mathcal{H}_2$.

Mathematical induction is applied to show that an arbitrary square submatrix B of A, has determinant $det(B) \in \{0, \pm 1\}$. If only one element is in B, $det(B) \in \{0, \pm 1\}$ by definition. Assume $\det(B_{k\times k}) \in \{0, \pm 1\}$ for any square submatrix with k dimensions ($k > 1$), the case ($det(B_{(k+1)\times(k+1)}) \in \{0, \pm 1\}$) of any k+1 dimensions square submatrix remains to be verified.

It is noted that $det(B_{(k+1)\times(k+1)})$ will at most change its sign and preserve magnitude by applying the following elementary row operations. For a row in $B_{(k+1)\times(k+1)}$ from a constraint with level n in $\mathcal{H}_1$ or $\mathcal{H}_2$, we replace it by the entrywise sum or difference with other rows from constraints in level $n-1$ of the same hierarchical constraint system. Repeating this procedure until we get a new matrix $B'_{(k+1)\times(k+1)}$ of which at most two nonzero entries are in each column, because there are at most two hierarchical constraint systems. Finally, the proof is concluded as follows.

1) If $B'_{(k+1)\times(k+1)}$ has a column with only zero entries, $det(B'_{(k+1)\times(k+1)}) = det(B_{(k+1)\times(k+1)}) = 0$;

2) If $B'_{(k+1)\times(k+1)}$ contains a column of a single nonzero entry (1 or -1), $det(B'_{(k+1)\times(k+1)}) = \pm det(B_{(k+1)\times(k+1)}) \in \{0, \pm 1\}$, using determinant expansion by minors;

3) If all columns in $B'_{(k+1)\times(k+1)}$ have two nonzero entries, the sum of the rows from $\mathcal{H}_1$ must equal to that from $\mathcal{H}_2$ by Definition 3 and Definition 5. Consequently, $det(B'_{(k+1)\times(k+1)}) = det(B_{(k+1)\times(k+1)}) = 0$.

Hierarchical constraint systems can be used for fast identification of a wide range of polynomial-time solvable problems with no more than two hierarchical constraint systems according to Theorem 2. The number of hierarchical constraint systems is counted by using the Hadamard products of pairwise constraints' coefficient vectors in Definition 3. If two constraints are not hierarchical, they must be in different hierarchical constraint systems.

4 Applications of hierarchical constraints in staff scheduling

4.1 Tractable personnel scheduling problems

A general personnel rostering problem is to assign employees $i \in E = \{1, 2, ..., |E|\}$ to shifts $k \in S = \{1, 2, ..., |S|\}$ on days $j \in T = \{1, 2, ..., |T|\}$, considering operational constraints and objectives (e.g., costs or employee satisfaction) [11]. The assignment of employee i to shift k on day j is denoted as $x_{ijk} = 1$, otherwise $x_{ijk} = 0$.

Problem P_1 [10] is an example of staff rostering with hierarchical constraints and is presented here to demonstrate an application of Theorem 2 to other real problems. A cost c_{ijk} is incurred by the assignment $x_{ijk} = 1$. The objective is to minimize the total cost of the final schedule. The total number of shifts (or days) that employee i should work is a_i. The minimum and maximum number of employees required for shift k on day j are d^l_{jk} and d^u_{jk} respectively. The ILP of P_1 is formulated as follows.

$$P_1: \quad Min \sum_{i \in E} \sum_{j \in J} \sum_{k \in S} c_{ijk} x_{ijk} \tag{2}$$

$$s.t. \quad \sum_{k \in S} x_{ijk} \leq 1, \ \forall i \in E, j \in T \tag{3}$$

$$\sum_{j \in T} \sum_{k \in S} x_{ijk} = a_i, \ \forall i \in E \tag{4}$$

$$\sum_{i \in E} x_{ijk} \leq d^u_{jk}, \ \forall j \in T, k \in S \tag{5}$$

$$\sum_{i \in E} -x_{ijk} \leq -d^l_{jk}, \ \forall j \in T, k \in S \tag{6}$$

$$x_{ijk} \in \{0, 1\}, \ \forall i \in E, j \in T, k \in S \tag{7}$$

Inequalities (3) are *single assignment* constraints, which restrict that an employee work at most one shift per day. The *total assignment constraints* (4) ensure that the total assignments of employee i equals a_i. The *coverage constraints* (5) and (6) define the range of requirements of employees for shift k on day j. Integrality of decision variables is constrained by (7). As the decision variables and their coefficients in constraints of P_1 are integers, all the constants $(a_i, d^l_{jk}, d^u_{jk})$ in the form in constraints must be integers, otherwise we just round them into integers.

It is trivial to convert P_1 into the generic problem P_0 by replacing the *total assignment constraints* (4) with two inequality constraints. Fig. 1 shows the constraint coefficient matrix of P_1 with $|E| = |T| = |S| = 2$. There are two hierarchical constraint systems partitioned by the line in Fig. 1, of which the correctness can be easily verified by Definition 3. It is noted that the integer constraints (7) are hierarchical with all constraints in P_1, since their linear relaxations have an identity matrix of coefficients. Therefore, P_1 not only has a polynomial-time solvable network flow model as shown in [10], the ILP of P_1 is also tractable by Theorem 2.

	$x_{1,1,1}$	$x_{1,1,2}$	$x_{1,2,1}$	$x_{1,2,2}$	$x_{2,1,1}$	$x_{2,1,2}$	$x_{2,2,1}$	$x_{2,2,2}$
Con(3)	1	1						
			1	1				
					1	1		
							1	1
Con(4)	1	1	1	1				
	−1	−1	−1	−1				
					1	1	1	1
					−1	−1	−1	−1
Con(5)	1				1			
		1				1		
			1				1	
				1				1
Con(6)	−1				−1			
		−1				−1		
			−1				−1	
				−1				−1
Con(7)	I_8							

Fig. 1: The partition of constraint matrix of A_1

An important tractable extension to P_1 is adding constraints (8), where $\bar{D}_i$ is a set of pairwise disjoint subsets of the day set T for each employee i, i.e., $F_1 \cap F_2 = \emptyset, F_1 \subseteq T, F_2 \subseteq T, \forall F_1, F_2 \in \bar{D}_i$. This new type of constraint can limit the number of working shifts (or days) of an employee i in the range $[m^l_{iF}, m^u_{iF}]$ within disjoint periods such as weekends [11].

$$m^l_{iF} \le \sum_{j \in F} \sum_{k \in S} x_{ijk} \le m^u_{iF}, \ \forall i \in E, F \in \bar{D}_i \tag{8}$$

Constraints (8), (3) and (4) are in the same hierarchical constraint system according to Definition 5. Here we present a more general set of constraints (9) replacing constraints (8) to preserve polynomial-time solvability. In constraints (9), G_i is a set of subsets of the day set T for employee i, and $F_1 \cap F_2 \in \{F_1, F_2, \emptyset\}, F_1 \subseteq T, F_2 \subseteq T, \forall F_1, F_2 \in G_i$. This extension enables the inclusion of assignment restrictions of employees within hierarchical periods in P_1. For example, the constraints that restrict the range of assignments (workload) of an employee in hierarchical periods such as weekends, weeks and months.

$$m^l_{iF} \le \sum_{j \in F} \sum_{k \in S} x_{ijk} \le m^u_{iF}, \ \forall i \in E, F \in G_i \tag{9}$$

Furthermore, constraints with a hierarchy among employees can also be included in P_1. A common situation is that there are coverage constraints in terms of particular groups of employees (such as skilled workers, interns and general workers) required for a shift k on a day j, as formulated by constraints (10). Q_k are a collection of subsets of the employee set E such that $L_1 \cap L_2 \in \{L_1, L_2, \emptyset\}, L_1 \subseteq E, L_2 \subseteq E, \forall L_1, L_2 \in Q_k$. In other words, all sets in Q_k are hierarchical instead of pairwise disjoint.

$$m^l_{kL} \leq \sum_{i \in L} \sum_{j \in T} x_{ijk} \leq m^u_{kL}, \ \forall k \in S, L \in Q_k \tag{10}$$

Similarly, there are constraints with a hierarchy in terms of shifts, represented by constraints (11). This kind of constraints models the restriction on the number of shifts in several categories ($Y \in W_i$) worked by an employee i, where $Y_1 \cap Y_2 \in \{Y_1, Y_2, \emptyset\}$, $Y_1 \subseteq T, Y_2 \subseteq S$, $\forall Y_1, Y_2 \in W_i$. For example, contractual constraints restrict the maximum workload in terms of morning shifts, daytime shifts and evening shifts defined in W_i. In this case, the daytime shifts include morning shifts (their intersection equals to the morning shifts). The set of evening shifts are disjoint with the morning shift set and the daytime shift set.

$$m^u_{iY} \leq \sum_{i \in E} \sum_{k \in Y} x_{ijk} \leq m^u_{iY}, \ \forall i \in E, Y \in W_i \tag{11}$$

However, at most two kinds of constraints from (9), (10) and (11) can be included in a polynomial-time solvable model, as there are at most two hierarchical constraints systems according to Theorem 2.

It is also noted that the inclusion of soft constraints [7] to P_1 is tractable, if no additional hierarchical constraints systems are introduced.

4.2 Derivation of network flow problems

Network flow models can be more efficient than the ILP formulations for a polynomial-time solvable rostering problem [10]. The challenge is that the network layout varies from problem to problem. Hence, this section presents a general method to derive a minimum cost flow problem from an ILP with at most two hierarchical constraint systems ($\mathcal{H}_1$, $\mathcal{H}_2$), as an application of the proposed hierarchical constraints. The main procedures are described below. Without loss of generality, we assume that $\mathcal{H}_1$ and $\mathcal{H}_2$ are not empty.

1. Node generation: add a *variable node* for each decision variable; generate *constraint nodes* corresponding to constraints in $\mathcal{H}_1$ and $\mathcal{H}_2$

2. Arc generation I: add an arc from a *variable node* to *constraint nodes* associated with the lowest-level constraints in $\mathcal{H}_1$, if the corresponding variable has nonzero coefficient (± 1) in that constraint; connect *variable nodes* with the *constraint nodes* from $\mathcal{H}_2$ with the same manner but in the opposite direction.

2. Arc generation II: add an arc (u, v) between *constraint nodes* u and v which correspond to constraints C_1 and C_2 with adjacent hierarchical levels in $\mathcal{H}_1$ respectively such that $C_1 <_H C_2$; develop arcs in the reverse direction among *constraint nodes* from $\mathcal{H}_2$ using the same rule.

3. Parameter configuration: set the supply and demand of a source node s and a sink node t and connect them with the *constraint nodes* associated with top-level constraints in $\mathcal{H}_1$ and $\mathcal{H}_2$ respectively; set capacity of arcs according to the constants of their relevant constraints in $\mathcal{H}_1$ and $\mathcal{H}_2$.

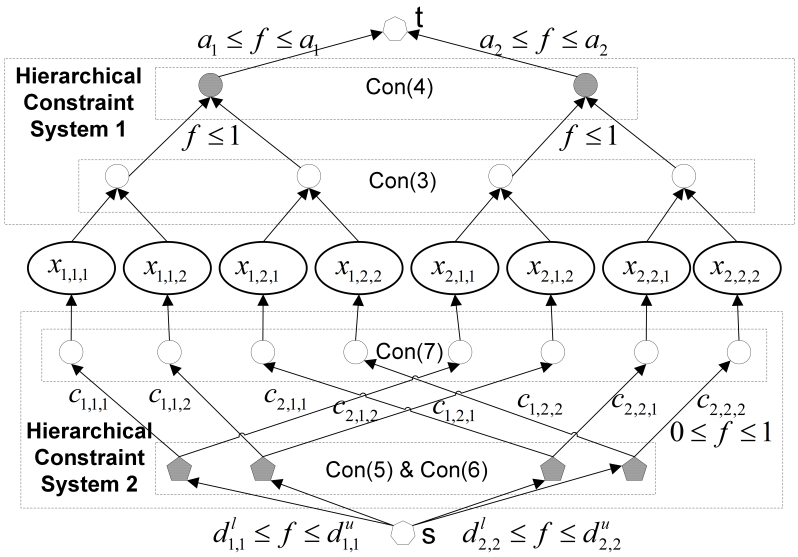

Fig. 2: The derived network flow problem from P_1

Generating the network is generic for any ILP with one or two hierarchical constraint systems, but the parameter configuration is problem dependent. Fig. 2 illustrates the network created by applying the presented approach on an instance of P_1. The supply of the source s and the demand of the sink t are equal to $\sum_{i \in E} a_i$. The lower and upper bound of flows from s to *constraint nodes* associating the *coverage constraints* (5) and (6) in $\mathcal{H}_2$ are d^l_{jk} and d^u_{jk} respectively. Then a flow from these *constraint nodes* to their upper *constraint nodes* corresponding to constraints (7) is limited in the range of [0,1]. There is a unit cost c_{ijk} of the flow to the *variable node* associating the decision variable x_{ijk} from the *constraint nodes* associated with constraints (7). As a result, costs of assignments can be calculated as the costs of flows. Furthermore, the flow from the bottom hierarchical level nodes associated with constraints (3) in $\mathcal{H}_1$ to their upper nodes has a capacity of one unit, according to the *single constraint* (3). Finally, the flow from the top-level *constraint nodes* representing *total assignment constraints* (4) in $\mathcal{H}_1$ to t is bounded by a_i, i.e., the required number of total assignments (working days) of employee i. Therefore, the derived minimum cost network flow problem is an equivalent of P_1.

5 Conclusions

This paper presents theoretical results concerning the constraints' effects on the complexity of optimization problems in a general setting of integer linear programs. Hierarchical constraints are identified and formulated to characterize the ILPs with polynomial-time solvability for a large class of personnel scheduling problems, without problem-dependent reformulations into tractable network flow problems [3, 10, 11, 7]. Consequent applications are introduced, including vectorized testing of polynomial-time solvability, several extensions of well-known tractable problems and the derivation of network flow problems from hierarchical constraints systems. These examples and applications validate that the hierarchy in constraints is one of the structural reasons why some staff scheduling problems/models are easy and can be remodeled as tractable minimum cost network flow problems in previous work [3, 10, 11, 7].

In the future, our focus will shift to the integration of tractable models into solution methods for complex personnel scheduling problems.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms and Applications. Prentice Hall, Upper Saddle River, New Jersey, USA (1993)
2. Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., De Boeck, L.: Personnel scheduling: A literature review. European Journal of Operational Research **226**(3), 367–385 (2013)
3. Brucker, P., Qu, R., Burke, E.: Personnel scheduling: Models and complexity. European Journal of Operational Research **210**(3), 467–473 (2011)
4. Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: A review of applications, methods and models. European journal of operational research **153**(1), 3–27 (2004)
5. Hoffman, A.J., Kruskal, J.B.: Integral boundary points of convex polyhedra. In: 50 Years of Integer Programming 1958-2008, pp. 49–76. Springer (2010)
6. Horn, R.A., Johnson, C.R.: Matrix analysis. Cambridge university press (2012)
7. Li, C., Smet, P., De Causmaecker, P.: Polynomial-time personnel scheduling with soft constraints. In: Proceedings of the 12th international conference on the practice and theory of automated timetabling, pp. 501–505. PATAT (2018)
8. Osogami, T., Imai, H.: Classification of various neighborhood operations for the nurse scheduling problem. In: International Symposium on Algorithms and Computation, pp. 72–83. Springer (2000)
9. Schrijver, A.: Combinatorial optimization: polyhedra and efficiency, vol. 24. Springer Science & Business Media (2003)

10. Smet, P., Brucker, P., De Causmaecker, P., Vanden Berghe, G.: Polynomially solvable formulations for a class of nurse rostering problems. In: Proceedings of the 10th international conference on the practice and theory of automated timetabling, pp. 408–419 (2014)
11. Smet, P., Brucker, P., De Causmaecker, P., Vanden Berghe, G.: Polynomially solvable personnel rostering problems. European Journal of Operational Research **249**(1), 67–75 (2016)
12. Vermuyten, H., Rosa, J.N., Marques, I., Belien, J., Barbosa-Póvoa, A.: Integrated staff scheduling at a medical emergency service: An optimisation approach. Expert Systems with Applications **112**, 62–76 (2018)