Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 768374, 10 pages
http://dx.doi.org/10.1155/2014/768374

Research Article

Scheduling IT Staff at a Bank: A Mathematical Programming Approach

M. Labidi, M. Mrad, A. Gharbi, and M. A. Louly

Department of Industrial Engineering, College of Engineering, King Saud University, P.O. Box 800, Riyadh 11421, Saudi Arabia

Correspondence should be addressed to M. Labidi; mlabidi@ksu.edu.sa

Received 16 November 2013; Accepted 28 January 2014; Published 17 March 2014

Academic Editors: S. Casado and N. Matsatsinis

We address a real-world optimization problem: the scheduling of a Bank Information Technologies (IT) staff. This problem can be defined as the process of constructing optimized work schedules for staff. In a general sense, it requires the allocation of suitably qualified staff to specific shifts to meet the demands for services of an organization while observing workplace regulations and attempting to satisfy individual work preferences. A monthly shift schedule is prepared to determine the shift duties of each staff considering shift coverage requirements, seniority-based workload rules, and staff work preferences. Due to the large number of conflicting constraints, a multiobjective programming model has been proposed to automate the schedule generation process. The suggested mathematical model has been implemented using Lingo software. The results indicate that high quality solutions can be obtained within a few seconds compared to the manually prepared schedules.

1. Introduction

Scheduling of staff or personnel is applicable in a wide variety of areas and service delivery fields such as scheduling of medical personnel in hospitals, scheduling of airline and hotel personnel, and scheduling of patrol officers. A good schedule balances the needs of employees or customers and the tasks involved, and schedules can be created for various periods of time such as days, weeks, or months. Depending on the complexity of the schedule, some organizations can use scheduling software such as spread sheets software to complete the task. One thing to note regarding software is that it can be expensive and can also be catastrophic for large organizations if poorly managed.

Some of the most common challenges with scheduling arise due to scheduling conflicts or poorly arranged schedules. Scheduling conflicts arise, for example, when an employee is scheduled to work at two locations at the same time. This is usually caused by changes made to the schedule by different people or lack of verification of the impact of the change on the rest of the schedule. Poorly arranged schedules include features such as lack of breaks between shifts and double shifts. This leads to poor production due to low staff morale and overwork as well as increasing the risk of accidents or mistakes due to fatigue of the employees.

In their simplest form, these problems involve only the assignment of days off, for example, in some of the less complex settings for the scheduling of nurses. A typical problem of this form requires the scheduler to give appropriate days off to each of a number of employees who work standard shifts while assuring that the required numbers of employees are on duty throughout the day and week. Effective staff scheduling consists of allocating the required workload to staff subject to a number of constraints. The main objectives of staff scheduling are to efficiently utilize resources, produce a schedule with a balanced workload, and satisfy individual preferences as much as possible.

There are various scheduling methods available and are discussed briefly below. The chosen scheduling method should conform to the department workload pattern to enable creation of effective and efficient employee schedules. Some scheduling models include fixed schedule method, rotation schedule method, pattern schedule method, coverage schedule method, assignment schedule method, and team schedule method. The fixed schedule method is mainly used where the workload is predictable; that is, the work is consistent and does not require a lot of flexibility. The rotation schedule method is used to ensure a balance between shifts; for example, the employees share the burden to be covered in 24 hrs in shifts such as morning shifts and evening shifts. Pattern schedule method sets the shift rotation by day of the week and is mainly applied where the work days usually change cyclically from one schedule period to another. An example of the pattern schedule method is where every employee is off every 2nd weekend. The coverage schedule method is used where the pattern of the workload is not consistent and a lot of flexibility is required. In this method, schedules are set according to the prevailing pattern in the workload. Assignment schedule method combines the aspect of time when one will be working and the assignment to be carried out; for example, at different times an employee will be in different departments. In the team schedule method, the employees are scheduled as a group; that is, there is one schedule for a group of workers following the same pattern or completing the same tasks.

The remainder of this paper is organized as follows. In Section 2, we begin by giving some literature review. Next, we propose a multiobjective model with the description of notation and variables, constraints, and objective function. In Section 3, we report the results of a computational study of the proposed approach. Finally, in the last section we provide some concluding remarks and directions for future investigations.

2. Literature Review

Creating a staff duty roster is very challenging and also time consuming. Although there is no polynomial known mathematical algorithm that guarantees an optimal solution in a reasonable computation time, it is still possible to generate duty rosters which are better than those produced by a human planner. Scheduling problems occur mostly when events are competing for the same attention, that is, conflicting constraints. The occurrence of conflicts can be stressful and also detrimental to productivity so it is very vital for managers to focus on prevention of these problems and, when they occur, work on minimizing the impact of the disruption. Two common ways to prevent scheduling conflicts are to review the full schedule during and after making changes and also have one person in charge of the schedule. The first method helps to ensure no new conflicts are generated or conflicts are detected early while the second reduces the risk of mistakes due to many people working on the same schedule. In this section, we discuss the approaches taken by several authors to find a solution to generate an effective schedule.

In [1], the authors proposed a review of applications used for staff scheduling and rostering problems. They have presented specific applications areas, models, algorithms, and methods used for solving them. In [2], a simulated annealing heuristic for use in a continuously operating scheduling environment is presented. Their research is designed to minimize the number of employees required to satisfy forecast demand. From the results, in terms of solution quality, the simulated annealing based method dominates other heuristics and it also exhibits rapid convergence to a low-cost solution.

One example of an NP-hard combinatorial problem is nurse rostering, and addressing scheduling problems can be done by using constraint programming and linear programming techniques. However, we note that, depending on the scenario, heuristics may also be applied for a feasible solution. For example, in [3], heuristics were necessary to address the issue of nurse rostering in Belgian hospitals. This approach was necessary due to the overconstrained schedules, the need to reduce the time to generate a schedule, and the fact that the cyclic three-shift schedule could not meet the hospital requirements. The Plane algorithm presented is used to determine which nurse can perform particular duties (according to qualification) and also when there are insufficient staff. The Tabu search method which is a local search method used for mathematical optimization for solving combinatorial optimization problems is used in the Plane algorithm. The two algorithms used by Plane are Tabu search and diversification and Tabu search and greedy shuffling. These algorithms are aimed at providing reliable solutions in a short time and providing results that are better from the human point of view, respectively. Results show that better qualities of rosters were provided and the users were more satisfied with the results. While addressing the limitations of the cyclic scheduling for nurses in Belgium, authors in [4] investigate the benefit of integrating nurse-specific characteristics into the cyclic scheduling approach. They analyze the extent to which the characteristics should be incorporated and this is compared to the cyclical and flexible acyclical scheduling approaches.

Genetic algorithms are search heuristics used to generate useful solutions as well as optimize and search problems. When implementing genetic algorithms for solving scheduling problems, however, there is a challenge handling the conflict between constraints and objectives which are prevalent in scheduling tasks. One approach to handling this challenge is to use problem specific knowledge, and in [5] genetic algorithms were used to address a nurse scheduling problem in a major UK hospital. In [5], the research investigates the effect of three types of problem specific knowledge on the scheduling challenge. The results show that the solution provided is fast, robust, and flexible.

The authors in [6] also showed how the column generation technique can easily be extended to integrate both nurse and surgery schedules to provide cost savings and reduce schedule generation times. In [7], the nurse rostering problem was partitioned into subproblems and each was solved sequentially using Integer Mathematical Programming (IMP). IMP is a mathematical optimization program which restricts variables to integers. In addition to local optimization techniques, their approach uses two phases where the first phase determines the workload for each nurse for each day of the week and the second phase allocates the daily shifts. Also, Mixed Integer Programing (MIP) is used in [8] to assign technicians to tasks requiring multilevel skill requirements by repeatedly applying a flexible matching model. This model selects tasks to be completed then forms groups of technicians assigned to a combination of tasks. The

MIP model performs very well in the case of rare skills and is capable of revising technician-task allocations. Furthermore, a mathematical model for staff scheduling in a telecommunications center is proposed in [9] to replace the manually generated weekly schedule which was generating inefficient and unfair schedules. A zero-one linear goal programming model with the work patterns mathematically formulated as a set of soft constraints was used. The results show the optimal scheduling cycle and also significant increase in efficiency and staff satisfaction.

In [10], the authors focus on scheduling crew and aircrafts while considering possible disruption. The goals in this case are to reduce slack between flights and reserve resources more intelligently or efficiently. They research into methods of evaluating robust scheduling approaches for aircraft and crew and develop stability indicators based on delay propagation.

Branch and price methods are used for solving integer linear programming and mixed integer linear programming problems which include many variables. The branch and price methods have been applied to scheduling of train drivers in [11]. Their approach looks at minimizing the number of duties and maximizing the schedule for outside disruptions. A duty in this case consists of a sequence of trips carried out on a given day by a train driver. An implicit column generation solution approach is used where a dynamic programming algorithm for the pricing-out of columns forms the basis of the heuristic brand and price algorithm. Also, [12] uses the brand and price models to solve workforce scheduling problems and propose an integer multicommodity network flow formulation to solve the problem.

3. Problem Statement

This paper addresses scheduling constraints encountered by bank IT staff. The problem can be described as finding a schedule that both respects the constraints of the staff and fulfils the objectives of the organization. Currently, there is a manual process for producing the monthly schedules using excel sheets; however, there are scenarios where the generated schedule violates some constraints. In order to deal with this issue, an automated scheduling system which will take into account the numerous constraints and goals is required for supervisors and operators in the IT Department. The next paragraphs present the major constraints and data related to the research.

There are four types of shifts (1 shift = 8 hours):

(i) morning shift (M): 06:00–14:00,

(ii) evening shift (E): 14:00–22:00,

(iii) night shift (N): 22:00–06:00,

(iv) external shift (S): 08:00–16:00.

The department rules or constraints to be taken into account while generating the schedule for IT staff can be summarized as follows.

(1) There are 9 operators and 5 supervisors who work morning, evening, and night in 8-hour shifts to provide a service that operates 24 hours a day and 7 days a week.

(2) The morning and evening shifts should include three employees (i.e., one supervisor and two operators).

(3) Each employee is required to work an average of 40 hours per week.

(4) The employee should not work on day shift after night shift.

(5) Each employee must have at most one shift per day.

(6) Overtime is estimated by the number of hours that exceed 40 hours per week.

(7) Only operator staff can work at an external/outside branch.

(8) Operator staff that leaves to work at an external/outside branch should have 2 days off on Thursday and Friday.

(9) The preferences of day off of each employee are taken into consideration.

4. Model Formulation

The staff scheduling problem has been formulated as a multiobjective programming model. Considering the fact that some of the constraints might be conflicting, however, the model will try to reduce these violations by minimizing the number of overtimes and maximizing the preferences of the staff.

4.1. Notation. The following notations are used to specify the model.

N_{jkl}: the requirement of staff of category j in shift k in day l,

n_j: the number of employees of category j,

S_w: the set of days of week w,

$S_{\text{off}_{(i,j)}}$: the preferable set of days off of employee i of category j,

AL_{ij}: the set of annual leave days of employee i of category j,

s_{1w}: the first 5 days of each week w,

s_{2w}: the 2 last days of each week (Thursday and Friday).

4.2. Decision Variables. The used decision variables can be described as follows:

$$x_{ijkl} = \begin{cases} 1, & \text{if employee } i \text{ of category } j \\ & \text{is working in shift } k \text{ on day } l \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

D_{ijw}: the number of overtime shifts of employee i of category j in week w,

R_{ij}: the number of days where the employee i of category j worked and normally he does not want to work,

$$y_{iw}=\begin{cases}1, & \text{if employee } i \text{ is working external} \\ & \text{shift during the week } w \\ 0, & \text{otherwise.}\end{cases} \qquad (2)$$

4.3. Model Constraints. The department rules for scheduling have been modelled into constraints and presented in this section.

(1) Each employee should work at most one shift per day:

$$\sum_{k=1}^{4} x_{ijkl} \leq 1 \qquad \begin{array}{r}\forall i = 1 \cdots n_j \\ \forall j = 1 \cdots 2 \\ \forall l = 1 \cdots 28.\end{array} \qquad (3)$$

(2) Set the staff requirement in each shift:

$$\sum_{i=1}^{n_j} x_{ijkl} = N_{jkl} \qquad \begin{array}{r}\forall j = 1 \cdots 2 \\ \forall k = 1 \cdots 4 \\ \forall l = 1 \cdots 28.\end{array} \qquad (4)$$

(3) Each employee should not work in annual leave days:

$$\sum_{k=1}^{4} \sum_{l \in \mathrm{AL}(i,j)} x_{ijkl} = 0 \qquad \begin{array}{r}\forall i = 1 \cdots n_j \\ \forall j = 1 \cdots 2.\end{array} \qquad (5)$$

(4) The employee should not work on day shift after night shift:

$$x_{ij3l} + x_{ij1(l+1)} \leq 1 \qquad \begin{array}{r}\forall i = 1 \cdots n_j \\ \forall j = 1 \cdots 2 \\ \forall l = 1 \cdots 28.\end{array} \qquad (6)$$

(5) Overtimes will be considered starting from the sixth shift during each week w:

$$\sum_{k=1}^{3} \sum_{l \in s_w} x_{ijkl} - d_{ijw} \leq 5 \qquad \begin{array}{r}\forall w = 1 \cdots 4 \\ \forall i = 1 \cdots n_j \\ \forall j = 1 \cdots 2.\end{array} \qquad (7)$$

(6) Overtimes should not exceed two shifts per week:

$$d_{ijw} \leq 2 \qquad \begin{array}{r}\forall w = 1 \cdots 4 \\ \forall i = 1 \cdots n_j \\ \forall j = 1 \cdots 2.\end{array} \qquad (8)$$

(7) Counting the number of working shifts during the preferable off days:

$$\sum_{k=1}^{4} \sum_{l \in S_{\mathrm{off}(i,j)}} x_{ijkl} = R_{(i,j)} \qquad \begin{array}{r}\forall i = 1 \cdots n_j \\ \forall j = 1 \cdots 2.\end{array} \qquad (9)$$

Table 1: Results notation.

Number	Identifier	Description
1	M	Morning shift
2	E	Evening shift
3	N	Night shift
4	S	External shift
5	X	Off day

(8) The operator who leaves to work at an external branch should take two days off per week (Thursday and Friday):

$$\begin{gathered}\sum_{L \in s_{1w}} x_{i24l} = 5 y_{iw} \qquad \begin{array}{r}\forall i = 1 \cdots n_2 \\ w = 1 \cdots 4;\end{array} \\ \sum_{k=1}^{4} \sum_{l \in s_{2w}} x_{i2kl} \leq 2\left(1 - y_{iw}\right) \qquad \begin{array}{r}\forall i = 1 \cdots n_2 \\ w = 1 \cdots 4.\end{array}\end{gathered} \qquad (10)$$

4.4. Objective Function. The objectives are to minimize the overtimes and maximise the satisfaction of employees in terms of desired days off. These objectives are weighted according to their priority such that ($\beta 1 > \beta 2$)

$$\operatorname{Min} Z = \beta 1 \sum_{i=1}^{n_j} \sum_{j=1}^{2} \sum_{w=1}^{4} d_{ijw} + \beta_2 \sum_{i=1}^{n_j} \sum_{j=1}^{2} R_{ij}. \qquad (11)$$

The highest importance level ($\beta 1$) is given to minimize the overtime, whereas the second highest priority level ($\beta 2$) is given to satisfaction of desired employee preferences. Since our model involves determining the weights of the two objectives, therefore, for our problem we use the two levels; level 2 represents the major objective of the problem: to minimise the overtime. Level 1 represents the degree of satisfaction of employee preferences. The weights βi $(i = 1, 2)$ will be equal to the grade of the level ($\beta 1 = 2$ and $\beta 2 = 1$).

5. Results

The following section presents the results of the research. Shift notations are summarized in Table 1. The results for the supervisor staff are shown in Table 2 (old solution) and Table 3 (new solution). These results compare the generated schedules of the working shift of each supervisor per day over a 4-week period. The results for the operator staff are shown in Table 4 (old solution) and Table 5 (new solution). These results compare the generated schedules of the working shift of each operator per day over a 4-week period.

5.1. Supervisors Shift. From comparing the generated schedules using the old and new methods, there are several things to be noted. First of all, the old solution gave total overtime of 8 shifts while the new solution has total overtime of only 4 shifts. This represents a 50% reduction in overtime costs. In addition, the old solution showed two supervisors (Supervisors 3 and 4) with higher amounts of overtime

Table 2: Supervisors shift—old solution.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	Overtime
Sup1	Vacation													M	M	M	M	M	M	X	N	N	N	N	N	N	X	X	1
Sup2	X	X	E	E	E	E	X	X	M	M	M	M	M	X	X	Vacation													0
Sup3	E	E	M	M	M	M	M	M	N	N	N	N	N	N	N	X	X	X	E	E	E	E	E	X	X	X	M	M	4
Sup4	M	M	X	X	N	N	N	N	N	X	X	E	E	E	E	E	E	E	X	M	M	M	M	M	M	M	N	N	3
Sup5	Vacation																												0

Table 3: Supervisors shift—new solution.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	Overtime
Sup1	Vacation													N	N	X	M	E	E	X	M	M	M	M	X	M	M	M	1
Sup2	E	X	M	M	M	M	M	X	M	M	M	M	M	M	E	Vacation													2
Sup3	M	M	X	X	N	N	N	N	N	N	N	N	N	X	M	E	X	M	X	E	N	N	N	X	M	N	X	N	1
Sup4	X	E	E	E	E	E	X	M	N	X	X	E	E	E	X	M	E	X	M	M	E	E	E	N	N	X	N	X	0
Sup5	Vacation																												0

Table 4: Operators shift—old solution.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	Overtime
Op1	M	M	M	M	M	E	E	X	N	N	N	N	N	N	N	X	X	X	E	E	E	E	E	X	X	M	M	M	3
Op2	S	S	S	S	S	X	X	E	E	E	X	X	X	N	N	N	N	N	N	X	X	E	E	E	E	E	E	X	1
Op3	N	N	N	N	X	E	E	E	E	E	X	X	X	M	M	N	N	N	N	N	N	N	X	E	E	E	E	E	4
Op4	M	E	E	E	E	X	X	S	S	S	S	S	X	X	M	M	M	M	X	N	N	X	X	E	E	E	E	E	1
Op5	Vacation													E	E	E	E	E	X	X	M	M	M	N	N	N	N	E	2
Op6	X	X	M	M	M	M	M	M	M	M	M	M	M	X	S	S	S	S	S	X	X	N	N	N	N	X	X	X	1
Op7	E	E	E	E	E	X	X	M	M	M	M	M	M	M	X	X	E	E	E	E	E	X	X	M	M	M	M	M	2
Op8	E	M	X	X	N	N	N	N	X	X	E	E	E	E	E	E	X	X	M	M	M	M	M	M	M	X	X	N	0
Op9	N	N	N	N	X	M	M	E	E	E	E	E	E	X	X	M	M	M	M	M	X	S	S	S	S	S	X	X	2

Table 5: Operators shift—new solution.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	Overtime
Op1	X	X	M	E	E	E	E	E	E	E	E	E	E	E	N	N	X	E	N	N	X	E	X	N	N	X	E	E	2
Op2	N	N	N	N	N	X	M	N	X	M	M	M	M	X	M	E	E	N	N	X	E	S	S	S	S	S	X	X	2
Op3	E	E	E	N	X	E	X	M	M	M	N	X	X	N	S	S	S	S	S	X	X	M	E	E	E	E	X	M	1
Op4	M	M	M	M	E	X	M	M	M	N	X	X	N	N	X	M	M	M	M	M	M	E	E	X	M	M	M	N	3
Op5	Vacation													M	X	E	N	X	E	E	E	N	X	M	E	E	X	E	0
Op6	S	S	S	S	S	X	X	X	N	X	M	M	M	M	M	M	M	M	M	X	N	X	X	M	M	M	M	M	1
Op7	M	M	X	M	M	N	N	S	S	S	S	S	X	X	E	N	X	N	X	N	N	X	M	E	E	N	N	X	1
Op8	N	N	N	X	M	M	E	E	E	E	E	N	X	X	E	N	N	X	X	M	M	N	N	N	N	X	E	X	1
Op9	E	E	E	E	X	M	E	E	E	E	E	E	E	E	N	X	E	E	E	E	X	M	M	E	X	E	E	E	4

compared to the rest which was an indication of unfair allocation of work. In the new solution, however, the overtime is more distributed indicating that the work has been more fairly allocated. It should also be noted that in the new solution, on two occasions, the overtime was taken during the vacation period. This is preferable to taking extra hours on a normal working day especially if this overtime occurs at the beginning or end of the supervisor vacation. In the old solution, supervisors could be seen working for more than five days without a day off while this is not visible in the new solution. Finally, the new solution has implemented the supervisors' shift preferences and this can be seen through Supervisor 2 who is mostly on the morning shift and Supervisor 4 who is mostly on the evening shift. From analysis of the results, we can conclude that the initial objectives of reducing overtime and catering for supervisor preferences have been achieved.

5.2. Operators Shift. From comparing the generated schedules using the old and new methods, there are several things to be noted for the operators' shifts. First of all, the old solution gave total overtime of 16 shifts while the new solution has total overtime of 15 shifts. This represents a 6.25% reduction in overtime costs. The smaller reduction in overtime when compared to the supervisor case can be explained as due to the fact that there is an additional shift for operators and also more operators are required per shift compared to supervisors. When we compare the old and new solutions, there is a slightly better distribution of overtime; for example, in the old solution there are 3 staff with 2 shifts of overtime and 3 staff with 1 shift of overtime whereas in the new solution there are 2 staff with 2 shifts of overtime and 4 staff with 1 shift of overtime each. One limitation to note however is that with both the old and new solutions there are still cases of high amounts of overtime, for example, 3 and 4 shifts of overtime. This indicates that there is still some unfair allocation of work in some cases. Finally, we also note that the staffs' preferences have been catered for in the new solution (see, e.g., Operator 9 who is majorly on the evening shift). We also note that all operators have more regular days off when compared to the old solution where most operators were on all four shifts and also off days were not allocated regularly. From analysis of the results, we can conclude that the initial objectives of reducing overtime and catering for operator preferences have been achieved.

6. Conclusion and Future Work

Creating a staff schedule is a challenging and time consuming task. This requires generating a schedule which respects the constraints of the staff and fulfils the organization objectives. Various techniques are used to achieve this such as use of Tabu search algorithms, genetic algorithms linear programming, or mixed integer programming. This paper has proposed a solution for scheduling of IT staff in a bank by using a multiobjective programming method. The solution was implemented using Lingo software and the results were compared with the existing manual method in use. Improvements noted were in reduction in overtime for both the operator and supervisor shift and also the new method allows for the staff to choose the preferred day off. Future work includes implementing the solution in different departments in the live environment in the bank and also adapting the solution to provide flexibility in case of disruptions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for its funding of this research through the Research Group Project no. RGP-VPP-296.

References

[1] 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*, vol. 153, no. 1, pp. 3–27, 2004.

[2] M. J. Brusco and L. W. Jacobs, "A simulated annealing approach to the cyclic staff-scheduling problem," *Naval Research Logistics*, vol. 40, no. 1, pp. 69–84, 1993.

[3] E. K. Burke, P. de Causmaecker, and G. Vanden Berghe, "A hybrid tabu search algorithm for the nurse rostering problem," in *Proceedings of the 2nd Asia-Pacific Conference on Simulated Evolution and Learning (SEAL '98)*, pp. 187–194, Canberra, Australia, November 1998.

[4] U. Aickelin and K. A. Dowsland, "Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem," *Journal of Scheduling*, vol. 3, no. 3, pp. 139–153, 2000.

[5] B. Maenhout and M. Vanhoucke, "The impact of incorporating nurse-specific characteristics in a cyclical scheduling approach," *Journal of the Operational Research Society*, vol. 60, no. 12, pp. 1683–1698, 2009.

[6] J. Beliën and E. Demeulemeester, "A branch-and-price approach for integrating nurse and surgery scheduling," *European Journal of Operational Research*, vol. 189, no. 3, pp. 652–668, 2008.

[7] C. Valouxis, C. Gogos, G. Goulas, P. Alefragis, and E. Housos, "A systematic two phase approach for the nurse rostering problem," *European Journal of Operational Research*, vol. 219, no. 2, pp. 425–433, 2012.

[8] M. Firat and C. A. J. Hurkens, "An improved MIP-based approach for a multi-skill workforce scheduling problem," *Journal of Scheduling*, vol. 15, no. 3, pp. 363–380, 2011.

[9] M. A. Louly, "A goal programming model for staff scheduling at a telecommunications center," *Journal of Mathematical Modelling and Algorithms in Operations Research*, vol. 12, no. 2, pp. 167–178, 2013.

[10] V. Dück, L. Ionescu, N. Kliewer, and L. Suhl, "Increasing stability of crew and aircraft schedules," *Transportation Research C*, vol. 20, no. 1, pp. 47–61, 2012.

[11] A. Alfieri, L. Kroon, and S. van de Velde, "Personnel scheduling in a complex logistic system: a railway application case," *Journal of Intelligent Manufacturing*, vol. 18, no. 2, pp. 223–232, 2007.

[12] C. Stark and J. Zimmermann, "An exact branch-and-price algorithm for workforce scheduling," in *Operations Research Proceedings 2004*, vol. 2004 of *Operations Research Proceedings*, pp. 207–212, Springer, Tilburg, The Netherlands, 2005.