Solving a Multi-Activity Shift Scheduling Problem with a Tabu Search Heuristic

S. Pan · **M. Akplogan** · **L. Létocart** · **N. Touati** · **R. Wolfler Calvo**

Abstract Scheduling problems form a large class of optimization problems encountered in various industries and organizations. The problem that we consider belongs to the category of shift scheduling problems with a multi-activity work environment. In general these problems consist in the process of assigning employees to activities in order to meet workload requirements, while observing legal regulations, and other company's rules. Due to the exponential growth of the computational time as instances increase, exact methods are usually only applicable for small sized instances. Consequently, different optimization methods such as heuristics, have been developed and widely used to handle large scale practical problems. In this paper, we focus on constraints concerning activities' duration, which make the problem difficult to solve. We propose a tabu search approach to deal with the specific multi-activity shift scheduling problem taking into account workload requirements and activities' duration constraints. Computational results show the effectiveness of the proposed approach compared to CPLEX solver.

Keywords Tabu search · shift scheduling · integer linear programming

1 Introduction

In order to succeed in company's business, one of the vital factors contributing to productivity and service quality is human capital management. Various sectors have begun to see the management of human capital as central to their strategy for

Stefania Pan · Lucas Létocart · Roberto Wolfler Calvo
LIPN, UMR 7030 CNRS, Université Paris 13
99 avenue Jean-Baptiste Clément 93430, Villetaneuse, France
E-mail: {pan,lucas.letocart,roberto.wolfler}@lipn.univ-paris13.fr

Stefania Pan · Mahuna Akplogan · Nora Touati
Horizontal Software
9 Rue de l'Isly, 75008 Paris
E-mail: {span,makplogan,ntouati}@horizontalsoftware.com

improving organizational performance. Human capital management involves different components such as recruitment, talent development, and determination of optimized working schedules. Multi-activity shift scheduling problems arise in the latter aspect and they consist of assigning employees to activities over a given time horizon, taking into account organizational, legal and social constraints. In many cases, these problems consist not only in finding feasible assignments, but also assignments that optimize a given objective, for instance, the minimization of the overall costs and the maximization of employees' preferences satisfaction. Various constraints occur in real-world shift scheduling problems, their number and complexity quickly make these problems difficult to solve. Shift scheduling can be found in many different application areas: airlines and railways transportation systems, healthcare systems, emergency services such as police, call centre services, and service organizations as hotels, restaurants, and retail stores [7]. The specific requirements of different companies result in quite diverse models and, consequently, solution methods to assist managers and planners in defining employees' schedules. The literature exhibits a wide range of models and solution techniques such as integer programming, dynamic programming, column generation, Lagrangian relaxation, heuristics and many other approaches. Ernst et al. in [7], [8] present a comprehensive survey of personnel scheduling problems. More than 700 published papers are classified according to problem type, solution approach and application area. In a recent work [15], Van den Bergh et al. review more than 300 articles published between 2004 and 2012. Papers are categorized according to four main fields: 1) personnel characteristics (contract type, skills, individual/crew), decision delineation and shifts definition; 2) different constraints (hard/soft, coverage, time-related, fairness and balance), performance measures and flexibility; 3) solution method and uncertainty incorporation; 4) application area and applicability of research.

In this paper we focus on a particular multi-activity shift scheduling problem, taking into account workload requirements and constraints on minimum and maximum activities' duration. We propose a heuristic method based on the tabu search technique [9]. Quimper and Rousseau [12] show that formal languages, such as regular and context-free languages, can be used to model shift scheduling regulations. From these languages they derive a large neighborhood search procedure to solve the multi-activity shift scheduling problem. A column generation approach based on constraint programming is proposed by Demassey et al. [6]. Both these papers consider legal regulations, workloads requirements, and constraints on the minimum activities' duration, without any restriction on the maximum duration. Dahmen and Rekik [5] solve a similar problem with both minimum and maximum duration restriction of activities, combining the tabu search technique and the exact branch-and-bound procedure of CPLEX. This procedure is embedded in the tabu search at three stages: the neighborhood exploration, the intensification, and the diversification. In these three stages, they consider a restricted set covering model which is solved to optimality by the B&B procedure of CPLEX. The authors construct feasible schedules that minimize over and under assignments. Our heuristic has the opposite approach. It defines schedules which satisfy workload requirements and, in the second place, it minimizes the violation of activities'

duration constraints. In many applications, the exact fulfilment of workload requirements is often not possible. However, there are situations in which workloads are known and are essential to ensure the realization and the quality of some services. For instance, in the staff management of a museum, workload demand for employees in ticket office, cleaning services, and security of the collections in each room is often known. The importance of workload requirements in some companies, justifies our approach which first considers the fulfilment of workload requirements, and then the constraints on the duration of the activities, which make the problem difficult to solve when both lower and upper bounds are considered. It is also true that in this study we do not take into account legal regulations. Therefore, the legal regulations and integrations of these constraints should be investigated in future works. However, we point out that some legal constraints, such as minimum and maximum consecutive working hours without break, can be managed using duration constraints.
The paper is organized as follows. In Section 2 we provide the definition of the considered problem giving an integer linear programming formulation. The framework of the proposed tabu search method is discussed in Section 3. Computational results are shown in Section 4. Lastly, conclusions and perspectives are presented in Section 5.

2 Problem description: an integer linear programming formulation

This section describes the particular multi-activity shift scheduling problem we deal with, and gives an integer linear programming formulation. Given a time horizon T divided into time periods of equal length called slots, a set of activities K and a number of employees w_{jk} required at each slot $j \in J$ for each activity $k \in K$, the problem then becomes the assignment of the right number of employees to each activity and in each slot, in order to fulfil workload requirements. Under and over staffing are not allowed. Furthermore, the assignment of employees to activities have to respect duration restriction on activities stretch. The minimum and maximum durations of an activity k, denoted respectively l_k and u_k, implies that an employee cannot perform activity k less than l_k slots and more than u_k slots consecutively. The maximum duration is imposed on difficult activities, in order to protect the employees' well-being, they express specific health constraints. The problem, therefore, is based on three main entities: employee, slot and activity. We define the set I of employees, the set J of slots and the set K of activities. The considered problem consists in assigning employees to activities in slots over all the given time horizon, in order to satisfy the constraints defined below.

Workload: in each slot $j \in J$, a number w_{jk} of employees is required for each activity $k \in K$. We assume that workload requirements are given as input data of the problem.
Activities' duration: each activity $k \in K$ has a bounded duration. It has to be performed at least l_k consecutive slots and it cannot be performed more than u_k consecutive slots.

The problem lends itself to be modelled as binary linear program. Our main decision variable is x_{ijk}, which takes value 1 if employee i is assigned to activity k in slot j, and it takes value 0 otherwise. Furthermore, in order to define activities' duration constraints, we introduce the binary variable t_{ijk}, which is 1 if employee i starts activity k in slot j, and 0 otherwise. The problem consists in solving the following integer linear program.

find (x_{ijk}) *such that*

$$\sum_{i\in I} x_{ijk} = w_{jk}, \quad \forall j \in J, \forall k \in K, \tag{1}$$
$$t_{ijk} + x_{i\,j-1\,k} \leq 1, \quad \forall i \in I, \forall j \in J, \forall k \in K, \tag{2}$$
$$t_{ijk} - x_{ijk} \leq 0, \quad \forall i \in I, \forall j \in J, \forall k \in K, \tag{3}$$
$$x_{i\,j-1\,k} + t_{ijk} - x_{ijk} \geq 0, \quad \forall i \in I, \forall j \in J, \forall k \in K, \tag{4}$$
$$l_k t_{ijk} - \sum_{j'=j}^{j+l_k-1} x_{ij'k} \leq 0, \quad \forall i \in I, \forall j \in J, \forall k \in K, \tag{5}$$
$$\sum_{j'=j}^{j+u_k} x_{ij'k} + t_{ijk} - u_k - 1 \leq 0, \quad \forall i \in I, \forall j \in J, \forall k \in K, \tag{6}$$
$$\sum_{k\in K} x_{ijk} \leq 1, \quad \forall i \in I, \forall j \in J, \tag{7}$$
$$x_{ijk} \in \{0,1\}, \quad \forall i \in I, \forall j \in J, \forall k \in K, \tag{8}$$
$$t_{ijk} \in \{0,1\}, \quad \forall i \in I, \forall j \in J, \forall k \in K. \tag{9}$$

The first set of constraints (1) ensures that workload requirements are satisfied in all slots and activities. Constraints (2)–(4) define the relation between the two decision variables x and t, imposing $t_{ijk} = 1$ when j is the starting slot of activity k. The set of constraints (5) and (6) ensure that, when activity k starts, it is performed at least l_k consecutive slots and maximum u_k consecutive slots. Constraints (7) make sure that each employee is assigned to only one activity in each slot. The final set of constraints state that all variables are binary.

3 A heuristic approach

This section describes a heuristic approach to handle the multi-activity shift scheduling problem previously presented. This method is a 2-phases heuristic. First, a greedy initial solution that satisfies only workload constraints is found, then a tabu search heuristic integrates activities' duration constraints.

3.1 First phase: a greedy heuristic

In the first phase our heuristic builds a greedy solution assigning employees to activities in order to satisfy only workload constraints. Initially, no employees are assigned and therefore the planning is empty. Activities are treated one by one and available employees are assigned in slots where the considered activity is required. The greedy heuristic's pseudo-code in given below.

Algorithm 1: Greedy heuristic (First phase)

```
X0 ← [ ];
foreach (activity k) do
    foreach (employee i) do
        foreach (slot j) do
            if (w_jk > 0 & i is available in j) then
                X0 ← Assign(i,j,k);
                w_jk − −;
return X0;
```

The greedy phase considers only workload requirements and it assigns employees to activities in each slot. This is done without checking if the assignments cause the violation of activities' duration constraints.

3.2 Second phase: a tabu search heuristic

Tabu search is a heuristic procedure which goes back to Glover [9] and has been used in many applications such as [1],[2], [3] and [4]. In the second phase of our heuristic, an initial solution satisfying workload constraints is available. As previously explained, this solution is built considering only the demand in each slot and no check is done on how long an activity is performed. We introduce an objective function to measure how far a solution is from satisfying activities' duration constraints. This function is given by the sum of all violations of either constraint (5) or constraint (6). We define $v_{ijk}(x)$ as

$$v_{ijk}(x) = \begin{cases} l_k t_{ijk} - \sum_{j'=j}^{j+l_k-1} x_{ij'k}, & \text{if duration} < l_k, \\ \sum_{j'=j}^{j+u_k} x_{ij'k} + t_{ijk} - u_k - 1, & \text{if duration} > u_k, \\ 0, & \text{otherwise.} \end{cases}$$

The violation $v_{ijk}(x)$ allows us to know which activity violates the duration constraint, moreover, when and which employee is performing that activity. Suppose that employee i starts activity k in slot j, and he/she performs it during n consecutive slots. If n does not satisfy duration constraints on activity k, $v_{ijk}(x)$ measures the violation. The tabu search heuristic aims to find the best quality solution (of the lowest total violation to activities' duration) satisfying workload requirements. Therefore, the goal is the minimization of the total violations to activities' duration:

$$\min \sum_{i \in I,\, j \in J,\, k \in K} \lambda_{ijk}\, v_{ijk}(x). \tag{10}$$

The violations $v_{ijk}(x)$ are weighted using multipliers λ_{ijk} which are iteratively updated. To minimize activities' duration violations, we use a basic tabu search method combined with intensification and diversification. In the following we describe the features of our tabu search heuristic.

Neighborhood. The neighborhood of a solution is defined by the operator called `Swap`. Given a subset of consecutive slots $J' = \{j, j+1, \dots\}$ (this subset can contain also one single slot) and two employees i_1 and i_2, the solution obtained after applying `Swap` $(i_1, J', k_1) \rightarrow (i_2, J', k_2)$ is equal to the current solution except that employees i_1 and i_2 exchange their activities in all the slots j in J'. The neighborhood of a solution s is built according to the following: we select the most violated constraint, $N(s)$ consists of all the solutions we can achieve by applying a `Swap` move on a subset J' of slots of the selected constraint. We remark that, for each slot j in J', the two employees embedded in the move, exchange the activities in j, ensuring that all the solutions in the neighborhood keep on satisfying workload constraints. Regarding the neighborhood exploration, we tested both best and first improvement. The first one consists in looking for the `Swap` that gives the best objective function improvement, while the second one consists in accepting the first `Swap` that improves the objective function. Due to the high number of evaluations needed, the first improvement gives better results considering both solving time and quality of solutions.

Tabu list. We employ a dynamic tabu list. We fix minimum l_{min} and maximum l_{max} lengths. The tabu list length changes between l_{min} and l_{max} according to the evolution of the objective function: it increases when the best solution known does not improve after Δ_tabu iterations, while it decreases when the best solution known improves. Regarding the tabu list update, we use a FIFO policy.

Intensification. The basic tabu search is combined with intensification. When an improving `Swap` $(i_1, J', k_1) \rightarrow (i_2, J', k_2)$ is found, the neighborhood is deeply explored, and moves in adjacent slots are evaluated.

Diversification. We also combine the basic tabu search with diversification. In our heuristic, when the best solution cannot be improved any more using the basic tabu search with intensification, we employ a perturbation operator to destruct the obtained local optimum solution. Starting from the best solution known, we apply `Swap` move in all slots in which we have a violated duration constraint. To be more precise, for each slot we first evaluate the swap moves with all other employees, then we select the move that leads the objective function to the lowest deterioration. As a result, the new solution preserves part of the best solution's feasibility and differs where duration constraints are violated. Then, the basic tabu search with intensification is restarted. Diversification is performed $div\_max = 5$ times.

Stopping criteria. Many stop conditions are possible for the tabu search, such as the fixed numbers of iterations, the maximum number of iterations without improvement in cost function and the total amount of solving time. Since the basic tabu search is integrated with a perturbation operator, it stops when the best solution cannot be improved within a given number of iteration it_max, currently set to 100.

Multipliers λ_{ijk} update. To conclude, multipliers λ_{ijk} are iteratively updated according the evolution of the corresponding violation $v_{ijk}(x)$ after applying the `Swap` move. Fixing $\delta \in (0,1)$ as small constant value, $\lambda_{ijk} = (1+\delta)\lambda_{ijk}$ if the violation increases and $\lambda_{ijk} = (1-\delta)\lambda_{ijk}$ if the violation decreases. The pseudo-code of the tabu search heuristic in given below.

Algorithm 2: Tabu search heuristic (Second phase)

```
X ← X0;
XBest ← X;
for (t = 1,.., div_max) do
    while (cost_best > 0 & it < it_max) do
        X ← ExploreNeighborhood(X);
        if (f(X) < f(XBest)) then
            X ← Intensification(X);
            XBest ← X;
        else
            UpdateTabuList();
    X ← Diversification(X);
    it_max ← 2 * it_max;
return XBest;
```

4 Computational results

We solved the multi-activity shift scheduling problem for 33 instances. Eleven instances sets are generated by varying the number of employees ($|I| = 10, 20, \ldots, 110$). The time horizon is fixed to one week and slots have a time unit of 15 minutes, which results in $|J| = 672$. All instances sets differ in workload requirements, which are defined as approximation of realistic instances coming from quick service restaurants. However, even if the number of employees required changes from set to set of instances, workloads have basically the same structure: in each day of the week, there are two rush periods (lunchtime and dinner time) wherein all employees are required, while outside mealtime half of the employees is in average needed, as shown in Figure 1.

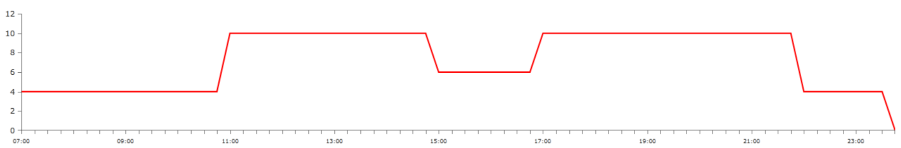

Fig. 1 Workload requirements

For each instances set, we consider two different activities ($|K| = 2$) and we generate 3 instances by varying the interval (l_k, u_k) which defines the minimum and the maximum duration. More precisely, the three intervals are respectively set equal to (1 hour, 3 hours), (2 hours, 3 hours) and (2 hours, 4 hours). We assume that employees are homogeneous and they have the skills to perform different activities. We implemented the formulation provided in Section 2 using CPLEX 12.6. The tests were performed on Intel Core i3 2.40GHz CPU and 8GB of RAM, with a maximum time limit of 2 hours. The heuristic was implemented using C# and run on the same machine.

Table 1 Computational results (time limit 2 hours)

Instances		Tabu search heuristic			CPLEX
# empl	durat (l_k,u_k)	Initial cost	Final cost	Elapsed time (s)	Elapsed time (s)
10	(1h,3h)	2324	0	12	6
10	(2h,3h)	2324	0	9	554
10	(2h,4h)	1932	0	9	167
20	(1h,3h)	5250	0	36	252
20	(2h,3h)	5250	0	71	5511
20	(2h,4h)	4522	0	30	1511
30	(1h,3h)	6930	0	59	684
30	(2h,3h)	6930	0	83	-
30	(2h,4h)	5810	0	54	3444
40	(1h,3h)	10780	0	93	1098
40	(2h,3h)	10780	0	791	-
40	(2h,4h)	9380	0	92	-
50	(1h,3h)	13020	0	133	1218
50	(2h,3h)	13020	0	1069	-
50	(2h,4h)	11340	0	157	-
60	(1h,3h)	15260	0	183	-
60	(2h,3h)	15260	0	1947	-
60	(2h,4h)	13300	0	195	-
70	(1h,3h)	17710	0	269	-
70	(2h,3h)	17710	0	2497	-
70	(2h,4h)	15190	0	325	-
80	(1h,3h)	19950	0	331	-
80	(2h,3h)	19950	- (2)	-	-
80	(2h,4h)	17150	0	560	-
90	(1h,3h)	22400	0	412	-
90	(2h,3h)	22400	- (14)	-	-
90	(2h,4h)	19040	0	890	-
100	(1h,3h)	24640	0	497	-
100	(2h,3h)	24640	- (28)	-	-
100	(2h,4h)	21000	0	1184	-
110	(1h,3h)	28490	0	599	-
110	(2h,3h)	28490	- (313)	-	-
110	(2h,4h)	24570	0	1968	-

Table 1 presents our computational results. The first two columns provide the instances according to the number of employees considered and the lower and upper bounds on activities' duration. The third column shows the cost of the solution found by the greedy heuristic, that is the violation of activities' duration constraints of the solution meeting only workload requirements. The final cost in the fourth column indicates that the tabu search heuristic is able to solve to optimality, in a reasonable elapsed time, all the instances with high amplitude of the interval (l_k, u_k), that is (1 hour, 2 hours) and (2 hours, 4 hours). CPLEX does not manage to solve the problem in less than two hours when the number of employees exceeds respectively 50 and 30. Furthermore, instances with interval (2 hours, 3 hours) are not solved by the heuristic when the number of employees exceeds 70. This is due to the low amplitude of the interval (l_k, u_k), which makes the problem difficult to solve. As we can see, in this case CPLEX solves only the two instances with 10 and 20 employees. In brackets we show the value of the objective function of the tabu search at the end of the time limit. This value points out a small optimality gap and, therefore, a near optimal solution is found.

We also investigate the impact of *diversification* and *intensification* components on the whole algorithm. We consider all the solved instances with 10 employees up to 60 employees. Table 2 shows that tabu search is able to solve only half of the instances, while adding intensification or diversification a higher percentage of instances is solved. We also remark, from our tests, that diversification allows to approach the optimal solution while intensification speeds up the algorithm.

Table 2 Impact of algorithm's components on the solution

Components	% of solved instances
Tabu search	50%
Tabu search + intensification	67%
Tabu search + diversification	83%

5 Conclusions and perspectives

In this paper, we presented a multi-activity shift scheduling problem taking into account workload requirements and activities with minimum and maximum duration. This problem is interesting due to the second type of constraints. Indeed, a problem wherein only workloads are considered can be quickly solved by a commercial ILP solver. However, the computational difficulty increases when activities have both minimum and maximum durations restrictions.

We developed an integer linear programming formulation and a heuristic approach to solve the staff scheduling problem considered. In the first place, our method finds a greedy initial solution satisfying only workload constraints. Afterwards, activities' duration constraints are integrated using a basic tabu search combined with intensification and diversification. The effective performance of our heuristic is shown by comparison with CPLEX. In almost all cases, feasible solutions are provided in reasonable elapsed time, while CPLEX does not manage to solve the problem in two hours.

Future works are intended to integrate other kind of constraints, such as legal regulations, using activities' duration constraints to deal with some of them. As already mentioned, minimum and maximum consecutive working hours without break, can be managed using duration constraints. Furthermore, we aim to develop other methods getting inspiration from hybrid metaheuristic or parallel hybrid metaheuristic (hybridation of metaheuristic with other optimization techniques), to provide more efficient behavior and a higher flexibility.

Acknowledgements This work has been funded by BPIfrance in the frame of the French cooperative project ADAMme, Projet Investissement Avenir (FSN: AAP CISN2).

References

1. U. Al-Turki, C. Fedjki and A. Andijani, Tabu search for a class of single-machine scheduling problems, *Computers & Operations Research*, 28(12), 1223–1230, (2001).
2. E. K. Burke, P. D. Causmaecker, S. Petrovic and G. V. Berghe, Metaheuristics for handling time interval coverage constraints in nurse scheduling, *Applied Artificial Intelligence*, 20(9), 743–766, (2006).
3. M. Chiarandini, A. Schaerf and F. Tiozzo, Solving employee timetabling problems with flexible workload using tabu search, In *Proceedings of the 3 rd Int. Conf. on the Practice and Theory of Automated Timetabling (E. Burke & W. Erben, Eds.)*, 298–302,(2000).
4. P. De Bruecker, J. Van den Bergh, J. Belien and E. Demeulemeester, A Tabu search heuristic for building aircraft maintenance personnel rosters, *Available at SSRN 2464033*, (2014).
5. 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).
6. S. Demassey, G. Pesant and L. M. Rousseau, A cost-regular based hybrid column generation approach, *Constraints*, 11(4), 315–333, (2006).
7. 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).
8. 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(1-4), 21–144, (2004).
9. F. Glover, Tabu search-part I, *ORSA Journal on computing*, 1(3), 190–206, (1989).
10. F. Glover and M. Laguna, Tabu Search, *Kluwer Academic Publishers*, (1997).
11. Z. Lü and J. K. Hao, Adaptive tabu search for course timetabling, *European Journal of Operational Research*, 200(1), 235–244, (2010).
12. 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).
13. A. Schaerf and A. Meisels, Solving employee timetabling problems by generalized local search, In AI* IA 99 : *Advances in Artificial Intelligence*, 380–389, Springer, (2000).
14. G. M. Thompson, Labor Scheduling: A Commentary, *Cornell Hospitality Quarterly*, 44(5-6), 149–155, (2003).
15. 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, (2013).