

The Human Resources Assignment with Multiple Sites Problem

Mohamed Afilal, Hicham Chehade, and Farouk Yalaoui

Abstract—Human resources assignment is the process of creating an employee's assignment in order to meet the demand of a set of tasks over time horizon. Many research works have been developed for similar problems in many field areas like health-caring, manufacturing, transportation... However in our study, we approach a general case where tasks do not have a specific daily or weekly pattern. Employees can be assigned to more than one task per day and taking in consideration multiple sites. The objective of our study is to find a feasible solution that respect different constraints relative to labor regulations and a constraint relative to multiple sites, balance the workload over employees and minimize overload hours. We propose a mixed integer programming model and a key performance indicator based heuristic to solve this problem. The results of the heuristic are very promising.

Index Terms—Human resources assignment, workload balancing, overload minimization, MIP model, heuristic.

I. INTRODUCTION

Human resources assignment (HRA) is a widespread problem in many industries like health-caring, airlines, manufacturing, transportation, etc. In most real world cases the problem was well stated as NP-hard [1]. Sometimes the challenge lies just in finding an assignment that respect a set of constraints usually relative to labor regulations. In many cases, such solution does not exist due to a maximum number of hours allowed per employee. Hence, overload work hours must be assigned to new employees in order to meet the demand.

The problem is more challenging if we take in consideration multiple sites. The main constraint of multiple sites states that: an employee cannot work in two different sites within a given period. This kind of situation can face large firms who have multiple sites and cannot assign their employees to two different sites in the same day, week or month (due to distance constraint for example).

The most frequent problem in the literature is the Nurse Rostering Problem (NRP). Many research works have been developed for this problem. In [1] and [2], authors have presented hybrid approaches for solving the nurse rostering problem. However, multiple sites were not considered in

their research. Moreover, their objective function is to minimize the sum of penalties caused by violation of soft constraints. In [3], authors have presented a Brunch and Price method for the same problem.

Transportation is also one of the important areas where HRA problem was treated. In [4] authors presented a technique called Improved Squeaky Wheel Optimization to solve driver scheduling problem, where drivers have to be assigned to trips taking in consideration different labor constraints. In [5] authors presented an adapted particle swarm optimization method to resolve the HRA problem in the logistics field with respect to hard constraints. The objective of the latter research is to minimize the sum of penalties generated by violating soft constraints.

Human resources assignment was also a major research subject in manufacturing area. In [6] and [7], authors considered a similar problem based on a real world case of a large production firm. Multiple locations were considered in their research. However the constraints stated is the required number of employees with particular skills for each site. Reassigning an employee to other site is not permitted in their case study. Authors presented an MIP model for the problem and a column generation approach. In [8], authors presented a multi-criteria genetic algorithm based approach to solve the HRA problem. The objectives are to maximize employee satisfaction and production rate. In [9], authors presented different MIP models to solve the workforce schedule problem for a single-shift taking in consideration constraints relative to labor regulations. In [10], authors presented different heuristics and a MIP model to solve the workforce management where employees have different skills. The problem is to determine the required staff over a time horizon and minimize workforce related costs (hire, fire, and cross-train).

Another approach for the problem is to consider the employees as parallel machines and the tasks have to be assigned to them. This perspective gives us another approach for the problem. In [11], [12] and [13], authors have presented methods, like MIP models and genetic algorithm, for solving workload balancing in parallel machines problem. However the only constraint considered in these articles is the task order.

In recent review (2013) of the personnel scheduling problem [14], authors have encouraged to consider multiple locations in the future as few previous work researches have approached this kind of problem before.

The present problem can be classified under the Shift Based Demand and Task Based Demand categories (refer to [15] for more comprehensive classification schemes on Personnel Scheduling and Rostering problems).

Manuscript received November 19, 2014; revised January 13, 2015.

Mohamed Afilal, Hicham Chehade, Farouk Yalaoui are with the Institut Charles Delaunay, LOSI, Université de Technologie de Troyes UMR 6281, CNRS, Troyes, France (e-mail: {mohamed.afilal, hicham.chehade, farouk.yalaoui}@utt.fr, {afilal, chehade}@opta-lp.com).

Mohamed Afilal and Hicham Chehade are with Opta-Lp, 2 rue Gustave Eiffel, 10430 Rosières-Près-Troyes, France.

The key contribution of this research paper is that it addresses for the first time the human resources assignment with multiple sites problem with flexible employee's assignment.

The paper is organized as follows: Section I: brief introduction and a literature review of the problem. Section II: detailed description of the problem. Section III: mathematical formulation of the problem. Section IV: introduction of a heuristic approach to solve the problem. Section V: computational results. Section VI: conclusion and further research.

II. PROBLEM DESCRIPTION

Given a set of tasks to be realized in multiple sites over a time horizon, employees have to be assigned to these tasks in order to meet the demand: every task has to be assigned a required number of employees. The assignment must respect different labor regulations. In addition, a constraint relative to multiple sites is considered and states that an employee cannot be assigned to two tasks of different sites in the same day.

Note that all employees have the required skills to perform all tasks, which means that any task can be assigned to any employee. Moreover, no Over or Under-Staffing is allowed, which means that a task has to be assigned exactly its required number of employees.

The constraints that must be respected in our case study are as follows:

- Maximum horizon, weekly and daily work hours must not be exceeded
- Average weekly work hours in 12 consecutive weeks must not be exceeded
- Maximum working week-ends must not be exceeded
- Employee must either work the entire week-end (Saturday and Sunday) or not work the week-end at all
- Maximum daily extent must be respected
- Minimum daily rest hours must be respected
- Employee cannot work in two different sites in the same day

Moreover, employees do not all have the same maximum horizon and weekly working hours to be respected. An example of this case is the difference between full-time and part-time workers. In order to simplify our approach, a maximum horizon and weekly base working hours is set and each employee has a coefficient that represents the percentage to be considered for him, example: maximum horizon base working hours is 1500, employee with coefficient 0.75 has 1125 hours as maximum horizon working hours.

This approach perspective enables a great flexibility to extend this problem to other similar problems like the NRP. In fact, the goal of the NRP is to find feasible roster by assigning to a shift its required number of nurses. In the NRP, a nurse can only be assigned to one shift per day, and cannot be assigned to two consecutive shifts in consecutive days. This can be fairly replaced by the maximum daily extent and minimum rest hours constraints in our case.

The present study is based on a real world industrial case of a firm that owns multiple sites. Every site has to be

assigned a specific number of employees during time slots on a day. The demand pattern varies according to period type: school, small holiday and summer holiday period. Each period has a different weekly pattern of demand. The firm aims to find an annual schedule for its employees.

The objective is to find a solution that respect all the constraints above, minimize the overload work hours and balance the workload over the employees. The second objective can be modeled as the difference between the maximum and minimum workload of employees [12].

III. MATHEMATICAL FORMULATION

In this section we are going to present a mixed integer programming (MIP) model for the problem described in the previous section. The main input data sets for the problem are the set of employees and the set of tasks. Due to the complexity of the problem a feasible solution is less likely to be found. To deal with this problem, a fictitious employee is introduced and modeled separately with an integer decision variable that can absorb the demand of the tasks. This operator is excluded from the constraints described above. The sum of hours assigned to this latter are the overload hours subject to minimization in our case study.

Let us denote:

T : Set of tasks

E : Set of employees

S : Set of sites

W : Set of weeks

D : Set of days

T_d : Tasks of day d

T_s : Tasks of the site s

D_w : Days of the week w

Sat_w : Saturday of the week w

Sun_w : Sunday of the week w

$Demand_i$: Required number of employees for task i

a_i : Arrival time of task i

b_i : Completion time of task i

p_i : Processing time of task i

$Coef_j$: coefficient of employee j

$Extent$: Maximum daily hours extent

$Rest$: Minimal rest hours

$maxWeekEnds$: Maximum number of working week-ends

Mh : Maximum horizon base working hours

Mw : Maximum weekly base working hours

Md : Maximum daily working hours

Ma : Weekly Average maximum working hours in 12 consecutive weeks

M : a big number

Note that a_i and b_i design the arrival and the completion time (in hours) of the task i on the day it belongs: their values are between 0 and 24.

Decision variables:

x_{ij} : binary = 1 if task i is assigned to employee j , =0 otherwise

xf_i : integer = demand of task i absorbed by the fictitious

employee

y_{wj} : binary = 1 if employee j works the week-end of the week w , = 0 otherwise

$start_{dj}$: starting work time of employee j the day d

end_{dj} : ending work time of employee j the day d

MaxLoad: maximum workload of employees

MinLoad: minimum workload of employees

The MIP model for the problem is as follow:

Minimize

$$Z : (MaxLoad - MinLoad) + \sum_{i \in T} x_{ij} \times p_i \quad (1)$$

Subject to:

$$\forall i \in T : \sum_{j \in E} x_{ij} + x_{ij}^f = Demand_i \quad (2)$$

$$\forall d \in D, j \in E : end_{dj} - start_{dj} \leq Extent \quad (3)$$

$$\forall d \in D - \{1\}, j \in E : start_{dj} + 24 - end_{d-1j} \geq Rest \quad (4)$$

$$\forall d \in D, i \in T_d, j \in E : start_{dj} \leq x_{ij} \times a_i + (1 - x_{ij}) \times M \quad (5)$$

$$\forall d \in D, i \in T_d, j \in E : end_{dj} \geq x_{ij} \times b_i \quad (6)$$

$$\forall j \in E : \sum_{i \in T} x_{ij} \times p_i \leq Mh \times Coef_j \quad (7)$$

$$\forall w \in W, j \in E : \sum_{d \in D_w} \sum_{i \in T_d} x_{ij} \times p_i \leq Mw \times Coef_j \quad (8)$$

$$\forall d \in D, j \in E : \sum_{i \in T_d} x_{ij} \times p_i \leq Md \quad (9)$$

$$\forall w \in W, j \in E : \sum_{k=w}^{\max(card(W), k+1)} \sum_{d \in D_k} \sum_{i \in T_d} x_{ij} \times p_i \leq Ma \times coef_j \times 12 \quad (10)$$

$$\forall w \in W, j \in E : \sum_{d \in Sat_w} \sum_{i \in T_d} x_{ij} \leq M \times \sum_{d \in Sun_w} \sum_{i \in T_d} x_{ij} \quad (11)$$

$$\forall w \in W, j \in E : \sum_{d \in Sun_w} \sum_{i \in T_d} x_{ij} \leq M \times \sum_{d \in Sat_w} \sum_{i \in T_d} x_{ij} \quad (12)$$

$$\forall w \in W, d \in Sat_w, i \in T_d, j \in E : x_{ij} \leq y_{wj} \quad (13)$$

$$\forall j \in E : \sum_{w \in W} y_{w,j} \leq \max WeeKends \quad (14)$$

$$\forall s_1 \in S, s_2 \in S - \{s_1\}, d \in D, i_2 \in T_{s_2} \cap T_d, j \in E :$$

$$\sum_{i_1 \in T_{s_1} \cap T_d} x_{i_1 j} \leq (1 - x_{i_2 j}) \times M \quad (15)$$

$$\forall j \in E : MaxLoad \times Coef_j \geq \sum_{i \in T} x_{ij} \times p_i \quad (16)$$

$$\forall j \in E : MinLoad \times Coef_j \leq \sum_{i \in T} x_{ij} \times p_i \quad (17)$$

Equation (1) The objective function is to minimize the difference between the maximum and minimum workload, hence balancing the workload over employees [12], and

minimize the sum of hours assigned to the fictitious employee designated by the decision variable x_{ij} .

Equation (2) Demand requirement constraint assure that every task have to be assigned it required number of employees.

Equation (3) Assure that extent working hours (the difference between starting and ending service time) of an employee must not be exceeded.

Equation (4) Assure that rest hours between two consecutive working days must be respected. The first day is not included because we consider that employees have not worked before the first day.

Equation (5): Assign proper value to the decision variable $start_{d,j}$. It assures that the variable must be less than all arrival times of assigned tasks to employee j in the day d .

Equation (6): The same way as (5), this constraint assigns proper value to the decision variable $end_{d,j}$.

Equation (1), (7), (8) and (9): Constraints that assure maximum horizon, weekly and daily work hours, note that employee coefficient is not applied to daily maximum hours Equation (9).

Equation (10): Assure that maximum average weekly hours must not be exceeded in 12 consecutive weeks.

Equation (11): Constraints assure that employee works on Saturday only if he works on Sunday.

Equation (12): Constraints assure that employee works on Sunday only if he works on Saturday.

Equation (13): Assign the value 1 to the decision variable y_{wj} if employee j works the week-end of the week w . This will be used in the nest equation to assure maximum working week-ends.

Equation (14): Respects the maximum number of working week-ends per employee.

Equation (15): Assure that employee must not work in two different sites in the same day: the decision variables x_{ij} that design if a task i that belong to a given site in a given day is assigned to employee j , can take the value 1 only if all the other variables that belong to the remaining sites in the same day are equal to 0. A previous version of the MIP contained an additional binary decision variable l_{sdj} that designs whether employee j works on site s in the day d or not, and a constraint stated that the sum of these variables must be less than or equal to 1. This latter was more complex, hence it takes more runtime to be solved comparing to this version.

Equation (16), Equation (17) assign proper value to decision variables *MaxLoad* and *MinLoad*. To take in consideration the difference between maximum working hours we scaled these variables to have a better balance of workload over employees.

This model will be, used, and solved using proper tools later to obtain optimal solutions in order to compare them with the ones obtained by the heuristic that we will present in the next section.

IV. CONSTRUCTIVE HEURISTIC APPROACH

The method we are introducing is a key performance indicator (KPI) based heuristic (Algorithm 1). The

procedure is as follows: given a task, employees who can be assigned to it without violating any constraint form a set called candidates list. We compute the KPI for each candidate and assign the task to the employee with the highest KPI value. If there is no candidate for the task, the latter will be assigned to an additional fictitious employee. The idea of the fictitious employee is the same as presented in the previous section. The sum of hours assigned to this latter represents the overload hours.

Since the tasks are already sorted by their arrival time, we do not have to worry about overlap problem when assigning tasks to employees. Hence, we are free to change the order of tasks subject to assignment during the process. In our case, the task subject to assignment is the one with the longest processing time. Moreover, since an employee has to work the entire week-end and maximum working week-ends constraint has to be respected, we can face, during the constructive process of assignment, a situation where the employee cannot be assigned to week-end's tasks due to this latter constraint. Hence, week-end's tasks are proceeded first to avoid this problem.

Let us denote:

S_j : Set of tasks already assigned to employee j .

$C(S_j)$: Total workload of employee j .

$C_w(S_j, i)$: weekly workload of employee j at the day of task i .

$C_d(S_j, i)$: daily workload of employee j at the day of task i .

The KPI is presented as follows:

$$KPI(S_j, i) = \frac{Mh - \frac{C(S_j)}{Coef_j}}{Mh} + \frac{Mw - \frac{C_w(S_j, i)}{Coef_j}}{Mw} + \frac{Md - \frac{C_d(S_j, i)}{Coef_j}}{Md}$$

This could be simplified to:

$$KPI(S_j, i) = 3 \times Coef_j - \frac{1}{Coef_j} \times \left(\frac{C(S_j)}{Mh} + \frac{C_w(S_j, i)}{Mw} + \frac{C_d(S_j, i)}{Md} \right)$$

```

1 Sort tasks by week-ends tasks and non-increasing order of processing
  time
2 for i in tasks
3   list = candidates for the task i
4   while ( i has not been assigned it's required number of employees)
5     if (list is empty)
6       assign remaining demand of i to the fictitious employee
7       set required number of employees of task i to zero
8     else
9       j = employee with the highest KPI within list
10      assign i to j
11      decrease required number of employees of task i by one
12      update Sj
13      remove j from list
14    end if
13 end while
14 end for
    
```

Algorithm 1. Pseudo code for the KPI heuristic.

The algorithm starts by sorting tasks by week-ends (week-end's tasks are proceeded first) and by non-increasing order of processing time (line 1). For every next task, construct a candidates list containing employees who can be assigned to it without violating any of the constraints presented in Section II (line 3). While the task has not been assigned it's required number of employees: if the candidates list is empty: assign remaining demand to the

fictitious employee (line 6) and set task's required number of employees to 0 (line 7), else: select the best employee within list (the one with the highest KPI) (line 9), assign the task to him (line 10) decrease task's requires number by 1 (line 11) update assignment of the selected employee (line 12) and remove him from the list (line 13) (employee should not be assigned twice to the same task). To assure minimum number of employees working in a given site in a given day, in the process of selection of the employee with highest KPI (line 9), employees who were assigned to the same site at the same day as the task subject to assignment have higher priority over the ones who were not assigned to any other task (employees who were assigned to other sites are excluded from the candidate list by the constraint of multiple sites).

V. COMPUTATIONAL RESULTS

In this section we are going to present the computational experiments for our case study.

First of all, and since there are no benchmark instances for the case of multiple sites human resources assignment problem, we have decided to test the developed heuristic on a particular case of the problem which is the single site problem. After that, and once the efficiency of the heuristic for the single site problem will be proved, we had to generate our own instances for the multiple sites case.

The model presented in Section III was written in the OPL language using a student version of the "IBM ILOG CPLEX Optimization Studio" software. The model was solved by the integrated CPLEX solver.

TABLE I: COMPUTATIONAL RESULTS FOR NRP (MONO-SITE CASE)

Instance	Cplex	Time Cplex	KPI Heuristic	Time Heuristic
Gpost	0	1s	0	1s
BCDT-Sep	8	1s	8	1s
BCV-4.13.1	8	1s	8	1s
Azaiez	8	1s	8	1s
BCV-3.46.2	1	3m46s	2	1s
CHILD	0,75	>2h	1,75	1s
HED01	2	>2h	7	3s

The KPI heuristic was coded using JAVA programming language.

A data manipulation tool was developed using JAVA programming language to transform data from Data Sheets, XML and raw text to structured usable data by the CPLEX's solver, and produce data sheet reports to display detailed results.

These instances presented in the next sub-sections were solved separately using a PC running Windows 8 with an Intel i5 2.5 GHz CPU and a 6 Go RAM (see Table I).

A. Mono Site Case

For this particular case, we used benchmark instances of the NRP found in [3] <http://www.cs.nott.ac.uk/~tec/NRP/>. These instances were used to get a real world in order to test the sturdiness of our heuristic, they are provided by different hospitals over the world. Note that the objective function we aim to optimize is different from the one in [3], as authors in

this latter aimed to minimize the sum of penalties caused by violation of soft constraints, we aimed to balance the workload over employees and minimize the sum of overload

hours. The NRP problem can be seen as the particular case of mono-site of our case study.

TABLE II: COMPUTATIONAL RESULTS FOR MULTI-SITE INSTANCES

Number of Sites	Number of Weeks	Number of Tasks	Number of Employees	Cplex's best solution	Time Cplex	KPI Heristic	Time Heuristic
2	4	399	36	893,5	>1h	518	1s
	12	1186	33	--	>1h	2730	3s
	24	2297	33	--	>1h	7677	2s
	48	4686	38	--	>1h	17052	2s
4	4	735	67	--	>1h	1252	2s
	12	2371	66	--	>1h	4407	3s
	24	4585	62	--	>1h	16316	3s
	48	9289	62	--	>1h	41604	5s
8	4	1561	129	--	>1h	2203,5	3s
	12	4631	139	--	>1h	8412	4s
	24	9293	128	--	>1h	31257,5	5s
	48	18624	129	--	>1h	77192	5s

The below table represents the results:

In general, the results of the heuristic were very good in term of solution's quality and run-time. In tactical decision making, where time horizon can be relatively long (6 to 12 months) or when number of employees is relatively big (> 20), having an optimal solution can take hours and days (instances CHILD, HED01: Cplex's best solution within 2 hours).

B. Multiple Site Case

In this section we are going to present parameters we used to generate our instances for the multiple site case and computational results for these instances.

We generated instances for 2, 4 and 8 site cases, for each one of them we generated instances for 4, 12, 24 and 48 weeks as time horizon. This gives us 12 different test cases.

For each case, parameters were generated as follows:

For each site and for each day in the time horizon generate task's parameters using the following expressions:

$$a_i = b_{i-1} + \lfloor N \sim (0,1) \times 3 \rfloor \times 0.5$$

$$p_i = \lfloor N \sim (0,1) \times 10 \rfloor \times 0.5 + 0.5$$

$$b_i = a_i + p_i$$

$$Demand_i = \lfloor N \sim (0,1) \times 9 \rfloor + 2$$

$N \sim (0,1)$ is a random number generated using the Normal distribution. Note that b_{i-1} is set to 0 when generating parameters for the first task in a day.

As for the Mw , Md , $Extent$ and $Rest$ parameters, we used the values approved and used by the majority of European countries:

$$Mh = Mw \times card(W) / card(W) = \text{number of weeks}$$

$$Mw = 48$$

$$Ma = 44$$

$$Md = 10$$

$$Extent = 12$$

$$Rest = 11$$

To generate the set of employees we used the following algorithm:

```

1 E = φ
2 TotalDemand = ∑ Demand_i × p_i
3 TotalCapacite = ∑ coef_j × card(W) × 48
4 while (TotalCapacite < TotalDemand)
5     generate employee j with coef_j = ⌊ N ~ (0,1) × 3 ⌋ × 0.5 + 0.5
6     E = E ∪ j
7     recalculate TotalCapacite
8 endwhile
    
```

Algorithm 2. Generation of employee's set.

The following table represents the results for the multiple site case:

Solving a multiple sites problem can be really hard and takes very long time as shown in the results of Table II. In big instances Cplex was unable to find a feasible solution within 1 hour, on the other hand, the heuristic produced feasible solutions very fast (5 seconds maximum) which makes it a very good decision tool.

Besides the runtime, when solving big instances using operational research, a considerable amount of free storage space is needed to store node files of the resolution process (minimum of 2Go in most cases). The heuristic on the other hand, need a negligible amount of storage space (few Mo), in our case only the space required to store data about employees assignment.

VI. CONCLUSION

The case study presented here concerns a task base demand human resources assignment solution tool that was developed for industrial purposes. The tool uses operational research to solve an MIP model for the problem for small instances, and a heuristic to solve big instances.

The heuristic presented in this study can be easily adapted to satisfy new constraint, the process of constructing candidates list can easily and swiftly check for any constraint. Moreover, the KPI of the heuristic can be modified to optimize other objectives other than the ones introduced in our study.

Further research involves introducing a local search for the heuristic to improve solution quality and adaptation of the heuristic to other similar problems.

REFERENCES

- [1] M. Stølevik, T. E. Nordlander, A. Riise, and H. Frøyseth, "A Hybrid approach for solving real-world nurse rostering problems," *Principles and Practice of Constraint Programming – CP 2011*, Springer Berlin Heidelberg, 2011, pp. 85–99.
- [2] E. K. Burke, T. Curtois, G. Post, R. Qu, and B. Veltman, "A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem," *European Journal of Operational Research*, vol. 188, no. 2, pp. 330–341, juillet 2008.
- [3] E. K. Burke and T. Curtois, "New approaches to nurse rostering benchmark instances," *European Journal of Operational Research*, vol. 237, no. 1, pp. 71–81, Aug. 2014.
- [4] U. Aickelin, E. K. Burke, and J. Li, "Improved squeaky wheel optimisation for driver scheduling," *Parallel Problem Solving from Nature -PPSN IX*, T. P. Runarsson, 2006, pp. 182–191.
- [5] M. Günther and V. Nissen, "Particle swarm optimization and an agent-based algorithm for a problem of staff scheduling," *Applications of Evolutionary Computation*, 2010, pp. 451–461.
- [6] H. D. S. S M Al-Yakoob, "A column generation approach for an employee scheduling problem with multiple shifts and work locations," *Journal of the Operational Research Society*, vol. 59, no. 1, pp. 34–43, 2007.
- [7] S. M. Al-Yakoob and H. D. Sherali, "Mixed-integer programming models for an employee scheduling problem with multiple shifts and work locations," *Ann Oper Res*, vol. 155, no. 1, pp. 119–142, Nov. 2007.
- [8] S. A. Cuesta, J. A. D. Mas, L. C. Darós, and C. A. Romano, "A genetic algorithm for the design of job rotation schedules considering ergonomic and competence criteria," *Int J Adv Manuf Technol*, vol. 60, no. 9–12, pp. 1161–1174, Jun. 2012.
- [9] C. S. Azmat, T. Hürlimann, and M. Widmer, "Mixed integer programming to schedule a single-shift workforce under annualized hours," *Annals of Operations Research*, vol. 128, no. 1–4, pp. 199–215, Apr. 2004.

- [10] J. W. Fowler, P. Wirojanagud, and E. S. Gel, "Heuristics for workforce planning with worker differences," *European Journal of Operational Research*, vol. 190, no. 3, pp. 724–740, Nov. 2008.
- [11] S. Rajakumar, V. P. Arunachalam, and V. Selladurai, "Workflow balancing in parallel machines through genetic algorithm," *Int J Adv Manuf Technol*, vol. 33, no. 11–12, pp. 1212–1221, Aug. 2007.
- [12] Y. Ouazene, F. Hnaïen, F. Yalaoui, and L. Amodeo, "The joint load balancing and parallel machine scheduling problem," in *Operations Research Proceedings 2010*, B. Hu, K. Morasch, S. Pickl, and M. Siegle, Eds. Springer Berlin Heidelberg, 2011, pp. 497–502.
- [13] S. Rajakumar, V. P. Arunachalam, and V. Selladurai, "Workflow balancing strategies in parallel machine scheduling," *Int J Adv Manuf Technol*, vol. 23, no. 5–6, pp. 366–374, Mar. 2004.
- [14] 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*, vol. 226, no. 3, pp. 367–385, May 2013.
- [15] A. T. Ernst, H. Jiang, M. Krishnamoorthy, B. Owens, and D. Sier, "An annotated bibliography of personnel scheduling and rostering," *Annals of Operations Research*, vol. 127, no. 1–4, pp. 21–144, Mar. 2004.



Mohamed Afilal received his engineering degree in logistics and transports in 2013, followed by a master's degree in systems optimization and safety in 2014. Currently he is preparing a Ph.D. degree in systems optimization and safety at the University of Technology of Troyes (UTT, France). His research include concerns mainly human resources optimization, demand forecasting and system simulation. He has also other research topic such as lot-sizing and vehicle routing problems.



Hicham Chehade received his engineering degree in industrial systems engineering and his master's degree in systems optimization and safety in 2005, followed by his Ph.D. degree in systems optimization and safety from the University of Technology of Troyes (UTT, France) in 2009. He is currently an assistant professor in the Industrial Systems Optimization Laboratory (ICD - LOSI, UMR CNRS 6281) in the University of Technology of Troyes in

France. His research fields include mainly the optimization of production, assembly and robotic lines design. He has also other research topic such as scheduling problems, operations research, modeling and simulation. He is author and co-author of 2 books, 2 book chapters, 14 journal papers and more than 60 communications in international conferences.



Farouk Yalaoui obtained his degree in industrial engineering from the Polytechnics School of Algiers in 1995, his master's degree in industrial system engineering from Polytechnics Institute of Lorraine in 1997, his Ph.D. degree in production management from the Troyes University of Technology (UTT) in 2000 and followed by a *Habilitation à diriger les recherches* from Compiegne University of

Technology in 2006. He is currently a full professor at UTT, where he is the head of the LOSI laboratory (ICD-LOSI, UMR CNRS 6281). His research topic focuses on scheduling problems, system design, operations research, modeling, analysis and optimization of logistic and production systems, reliability and maintenance optimization. He is the author and co-author of more than 300 contributions, communications or publications with 3 books, 9 book chapters and 58 articles in journals. He also published more than 220 papers in conference proceedings. He is a member of several international conference committees and International journal Boards. He had presented 19 invited speeches (seminaries or conferences plenary sessions).