Multi-agent Approach for Personnel Scheduling and Rescheduling in Assembly Centers

M. Sabar

Multi-agent approach for Personnel Scheduling and Rescheduling in Assembly Centers

M. SABAR*

* Groupe ISCAE: Institut supérieur de commerce et d'administration des entreprises
Km 9,500 Route de Nouasseur BP. 8114 - Casablanca Oasis, Casablanca, Maroc
msabar@groupeiscae.com

Abstract. This article presents a multi-agent based algorithm for personnel scheduling and rescheduling in the dynamic environment of a paced multi-product assembly center. Our purpose is, on the one hand, to elaborate daily assignment of employees to workstations to minimize the operational costs as well as the personnel dissatisfactions and, on the other hand, to generate an alternative planning when the first solution has to be rescheduled due to disturbances related to operators' absenteeism. The proposed approach considers the individual competencies, mobility and preferences of each employee, as well as the personnel and competency requirements associated with each assembly activity given both the current master assembly schedule and the line balancing for each product. To benchmark the performance of the multi-agent approach, we use solutions obtained through a simulated annealing algorithm. Experimental results show that our multi-agent approach can produce high-quality and efficient solutions in a short computational time.

Keywords: Personnel shift scheduling, Multi-agent systems, Coalition, Kernel stability, Cross-training, Flexible assembly lines.

1 Introduction

Personnel scheduling problems are particular cases of resource allocation problems (Hao et al. 2004). They aim to construct a working timetable for each employee by defining start time periods, duration of work, break intervals, as well as the tasks to be fulfilled. The objective is for the timetable to optimize one or several criteria while respecting a set of constraints such as labor requirements, individual preferences, or specific competencies (Thompson, 1995; Ernst et al. 2004). Typical classifications of personnel scheduling problems tend to separate problems in three categories (Ernst et al. 2004): shift, days-off and tour scheduling problems.

In this article, the focus is on shift-scheduling problems in a large assembly line environment where the pace setting takt time between individual product units is preset, equal to at least a few minutes. We consider an assembly line with multiple workstations responsible to sequentially assemble different product-models. Each assembly task requires one or several employees with a specific competency profile. In addition, concerning the manpower pool, we take into account individual competencies, mobility and preferences. Specifically, we consider that:

- Each worker has a specific degree of cross-training, enabling them to carry one or several types of assembly activities during a work shift.
- Workers are allowed to move between workstations in order to fulfill specific assembly activities according to the product assembly schedule.

- Workers can be assigned to secondary activities, which can be either productive, administrative or learning, when the employees are not assigned to an assembly workstation.
- Each worker has a set of individual preferences related to (1) the shift duration, (2) the assignable activities and (3) the number of transfers between activities.

Due to product changeovers and the specific manpower competency requirements associated with each product at each station, there are often large waves of personnel moves among stations. This causes significant disruptions to operations, deterring the overall productivity of the line and causing dissatisfactions among the personnel (Sabar et al. 2008, Sabar et al. 2012). Also, this paper considers that if a disturbance related to operators' absenteeism occurs at a given time period, the variables representing the personnel' scheduling up to this period are fixed to the matching values from the original plan and, the disturbance' parameters are considered for the remaining time periods. Then, a new rescheduling has to be performed on the global level through complete regeneration of personnel schedule. Concerning the operators' absence, we distinguish full absenteeism (operator absents from work during the entire shift) from partial absenteeism (operator arriving late to work or leaving work during working hours due to sickness or personal affairs). In both cases, the rescheduling aims to generate a new allocation plan which replaces absent operators by transferring activities to the available and on-call workers.

In this context, we present a multi-agent based approach that aims to tackle the complexity of our targeted personnel scheduling / rescheduling problem through distributed problem-solving. The proposed approach is based on cooperation among several rational agents which encapsulate individual competencies and preferences of employees. In this approach, the agents negotiate to form coalitions which allow them to improve their individual schedules, and consequently to iteratively improve the global solution of personnel scheduling problem. To benchmark the performance of the multi-agent approach, we use solutions obtained through a simulated annealing approach.

The remainder of the article is organized as follows. Section 2 defines multi-agent settings and presents in detail the multi-agent architecture and principles retained to sustain our scheduling/rescheduling approach. Section 3 presents the formal description of the proposed multi-agent approach. The experimental setup and results are discussed in section 4. Finally, section 5 presents conclusive remarks.

2 Multi-agent architecture proposed for personnel scheduling and rescheduling problem

In this research, the real environment to model is an assembly line system. It is made up of several workstations which constantly require a mix of skilled employees. In order to model such a system, we developed a multi-agent system composed of heterogeneous and autonomous agents, which cooperate with one another to produce a personnel schedule. Each agent represents a physical entity of the assembly system, or encapsulates a planning and decision making function. Our multi-agent system includes four categories of autonomous agents: a production-agent; a station-agent for each workstation; a coordinating agent; and an employee-agent representing each employee. These agents are autonomous, rational and able to communicate with each other.

(1) Production-agent: elaborates and manages the production planning. It determines the dynamic sequence and quantity of the product models to be assembled. The production plan is then commu-

nicated to stations-agents. The Production agent uses a set of priority rules to decide which job orders are to be planned. Its objective is to optimize criteria such as the maximization of the workstations' utilization or the minimization of delays. In this article, we consider that the production planning decisions are independent of the influence of human resources management.

(2) Station-agent: manages and controls the assembly activities of a workstation. Based on the production planning, this agent defines the needs of the workstation concerning number of employees and their required competencies, which it sends to the coordinator-agent.

(3) Coordinator-agent: is responsible for coordinating the employee-agents. First, it elaborates an initial solution of personnel scheduling. Then, it takes the active role of mediator in the negotiation process among the employee-agents who will try to improve their initial work plans through activities swapping.

(4) Employee-agent: represents the individual interests of an employee. It encapsulates the state as well as the main characteristics of the matching employee, in particular his competencies, his preferences and his allocation history. These agents can negotiate and cooperate among them in order to maximize their profit and their satisfaction. In the proposed architecture, they are coordinated by the coordinator-agent which plays a mediator's role.

In our approach, the personnel scheduling process is supported by the Coordinator-agent and the Employee-agents. In the first step, the coordinator-agent uses priority rules to produce an initial solution taking into account employees 'needs and competencies. Next, the initial work plan of each employee is transmitted to the corresponding Employee-agent. Considering that the initial solution is often mediocre quality in terms of total cost and employees' satisfaction, we use coalitions to improve the performance of the schedule. The issue of coalition formation has been studied in the game theory literature in the context of cooperative N-person games (Rapoport, 1970).

In our approach, coalitions are formed among Employee-agents who negotiate a potential mutual agreement on what activities to swap in order to increase their individual profits and ultimately improve the global personnel scheduling solution. Each Employee-agent is rational and self-interested. It has interest in forming coalitions which releases him from less satisfying assembly activities and/or allows him to get a more satisfying set of activities. The proposed coalition approach is round based. In each round at most one coalition is formed. Each round involves two phases:

- Phase 1 aims to generate a stable coalitional configuration, which consists of a partition of the set of employee-agents into disjoint stable coalitions. To maintain polynomial complexity of the formation process, we restrict our research on coalitions of size two. At the beginning of each round, each employee-agent contacts other employee-agents with whom it has common competencies. It seeks to identify whether there are assembly activities that can be swapped and sends coalition offers. In the scheduling case, all activities' exchange possibilities are tested. However, in the rescheduling case, only activities which are not carried out yet are taken into consideration. A coalition offer contains the list of tasks that may be switched, and the corresponding payoff. For each received coalition-offer by an employee-agent, it uses the Kernel stability concept (Davis and Maschler, 1965) to test the coalition equilibrium. The Kernel is based on the idea that the members of a coalition should be in an equilibrium concerning their power to object to each other's payoff. If an employee-agent finds that it can outweigh the other according to the initial payoff, it uses the Transfer Scheme proposed in Streans (1968) to demand a side payments transfer. Using a stable payoff distribution, the employee-agents can compare different coalition structures. In fact, each employee-agent compares the payoff of all received proposals. It chooses the coalition which is most beneficial for the employee it represents (i.e. the one that maximizes his utility function), and informs the Coordinator-agent about the accepted coalitions.

- Phase 2 proceeds to select the coalition to enact. Once all accepted coalitions have been received by the Coordinator-agent, it randomly selects a coalition among the group of coalitions that have a bilateral acceptance of the two members. In a rescheduling case, the priority is given to a coalition among those which disengage completely or partially an agent-employee of his activities during his absence periods. Next, the Coordinator-agent informs the two coalition's members about the agreement. These two agents complete the process by exchanging tasks. Based on the new task distribution, the employee-agents start a new round of coalition formation.

These two phases are structured within the framework of an anytime algorithm. Such a type of algorithm improves gradually the quality of its solution as computation time increases and can be interrupted at any time during computation to provide a solution (Russell & Zilberstein, 1991).

3 Multi-agent approach for personnel scheduling and rescheduling

In this section, we formally present our multi-agent approach for personnel scheduling. We describe on one hand the algorithm for initial solution generation, and on the other hand, sequentially the algorithm to generate coalitions with K-stable payoff distributions and the algorithm for coalition selection. It should be noted that whenever the process of solution improvement by coalition formation is stopped, it produces the best currently available solution. This process is generally interrupted when the desired state is reached or when having to answer an immediate need of assembly lines about some employee's allocation, for example when an employee becomes out-of-kilter.

The initial solution of employee allocation is performed at the coordinator-agent level based on a priority dispatching rule. Using the production planning, each station-agent has a view of local requirements concerning the number of employees and their competency profile. The used dispatching rule involves the selection, period by period, of the workstation with the least extra number of employees that have the required competency profile. At a given period, the extra number of employees is equal to the difference between the number of available employees and the required number of employees. For each selected workstation, the coordinator-agent assigns the least cross-trained employee available among those who have the required competency profile.

At the end of the first stage, each employee-agent $i \in N = \{1,...,n\}$ possesses a vector of activities to perform $\rho^i = \left[a^i_{nm,0},...,a^i_{kl,t}\right]$, where $a^i_{kl,t}$ is the activity k to execute on station l in the period t by the employee-agent i. To evaluate the utility of each employee-agent $i \in N = \{1,...,n\}$, we use a linear function: $V_i(\rho^i) = S - f_i(\rho^i)$, where S is a constant which corresponds to an initial amount allocated to each employee-agent, it represents an artificial gain that each employee earns if it succeeds to totally release himself from duty. f_i is an increasing linear function of work duration and dissatisfaction of employee i. In fact, $f_i = F1 + F2 + F3 + F4 + F5 + F6 + F7 + F8$, where[1,] :

- $F1$: Salary cost of employee i ;
- $F2$: Activity assignment cost of employee *i*;
- $F3$: Idleness penalty cost of employee;
- $F4$: Cost savings generated by the assignment of employee i to secondary activities;
- $F5$: Transfer cost of employee i ;
- $F6$: Penalty cost associated to the deviation from the number of transfers preferred by employee i ;
- $F7$: Penalty cost associated to the deviation from the total work duration preferred by employee i ;
- $F8$: Penalty cost (positive or negative) associated to the dissatisfaction or satisfaction of employee i for its assignment to a set of activities.

The algorithm to generate coalitions with K-stable payoff distributions (**Phase 1**) is as follows:

Each agent-employee i ***:***

1. Maintains a register concerning the references of the employee-agents with whom it can exchange certain activities.

2. For each employee-agent j indexed on its register, i tests all possible permutations of activities and calculates the value of each potential coalition $v(C_{ij}) = V_i(\rho^i_{new}) + V_j(\rho^j_{new})$. In case of several permutation possibilities with an agent j, i retains the one which generates the highest coalition value. In rescheduling case, the coalitions are based on the exchange of the activities which are not carried out yet (i.e. activities planned between the periods of absence' notification and the shift-end). Activities carried out are considered fixed and cannot be changed. However, they are taken into account in assessing the total cost of personnel staff scheduling.

3. If $v(C_{ij}) \geq V_i(\rho^i) + V_j(\rho^j)$, then i sends a coalition proposal PR_{ij} to j. The proposal encapsulates the set of activities to be permuted; the coalition value $v(C_{ij})$ and the initial proposed payoff $u_j = V_j(\rho^j) + \frac{v(C_{ij}) - (V_i(\rho^i) + V_j(\rho^j))}{2}$ (i.e. Dividing the profit generated by the coalition into two equal parts).

4. Receive coalition proposals from the other employees-agents.

5. Evaluate the received coalition proposals:
   (a) Use the Kernel concept to test the proposals coalitions equilibrium;
   (b) If employee-agent i dominates any other agent, it uses the Streans' transfer scheme to evaluate the side-payment demand and informs the concerned agent.

6. For each instable coalition, send or receive a part of payoff equal to the side-payment demand.

[1] For more details concerning the function f_i, see Sabar et al. 2008.

The utility function V_i is designed so as to generate more profit for an employee who succeeds to release himself from duty or acquire a set of activities which creates a higher satisfaction.

Given a pair of employee-agents (i , j) with the activity vectors ρ^i and ρ^j , we define the potential value of the coalition $C_{i,j}$ as: $v(C_{i,j}) = V_i(\rho^i_{new}) + V_j(\rho^j_{new})$, where ρ^i_{new} and ρ^j_{new} are the new activity vectors of i and j if they agree to form the coalition by permuting a part of their initial activities. To accept a coalition, the payoff of each agent after the redistribution of the coalition value must be at least equal to its initial self-value, i.e. $v(C_{ij}) = u_i + u_j$; $u_i \geq V_i(\rho^i)$ and $u_j \geq V_j(\rho^j)$. Each employee-agent uses the Kernel concept to evaluate the offered payoff and to assess its power to object to its partner's payoff. A general strategy used by employee-agents for coalition formation and payoff distribution is defined as follows.

3.1. Initialization of the regression coefficient : $\eta = 1$

3.2. Each employee agent $i \in \{1,\ldots,n\}$:

- Elaborates the list $\Lambda_{i,\eta}$ of K-stable coalitions that give him a payoff at least equal to $\eta \times u_{i,\max}$;
- Sends the list to the coordinator-agent.

3.3. Based on all the received lists, the coordinator-agent selects the set BC of coalitions which have a bilaterally acceptance from the two members i.e. the coalitions $C_{ij} \mid (C_{ij} \in \Lambda_{i\eta}) \wedge (C_{ij} \in \Lambda_{j\eta})$; $\forall i, j \in \{1,\ldots,n\}$. At this level, there are two possible scenarios :

- $BC \neq \phi$: several coalitions have a bilaterally acceptance of their two members:
  - In scheduling cases, the Coordinator-agent randomly selects a coalition from BC . In re-scheduling cases, it selects in priority a coalition among those which release completely or partially an agent-employee of duties during his absence periods. Then, informs the two coalition's members about the agreement.
  - These two agents finalize the process by exchanging tasks. Based on the new tasks distribution, the employee-agents start a new round of coalition formation (return to Stage 2).
- $BC = \phi$: no consensus is reached, then the regression coefficient will be decreased $\eta \leftarrow \eta - \varepsilon$:
  - If $\eta \geq 0$ return to 3.2.
  - If $\eta < 0$ the global solution has reached a local optimum (i.e. given the current activities distribution, employee-agents have no benefit by forming coalitions), then we introduce an artifice for fictitious payoffs distribution. This artifice randomly generates and attributes factitious profits to a certain number of employee-agents in such manners as to incite them to form coalitions. Return to stage 2 in order to generate new K-stable coalitions.

Fig.1. Algorithm for coalition selection (Phase 2)

At the end of this stage, we obtain a set of potential coalitions with stable payoff distributions. Each employee-agent may have several offers of coalitions with various profits. Since each employee-agent i is rational, it tries to form the coalition, among all possibilities, in which it earns the greatest payoff $u_{i,\max}$. However, if an agent i chooses to form a coalition with the agent j , nothing guarantees that agent j will accept because j may earn more by forming another coalition with a third agent k . In case of conflicts of interest between employees-agents, we introduce a regression function f_{reg} which allows agents to reduce the value of their aimed payoff in order to reach a consensus.

For an employee-agent i, this function is defined as $f_{reg}(i,\eta): u_{i,\max} \to \eta \times u_{i,\max}$, where $\eta \in [0,1]$ represents the rate of payoff's decreasing. Considering the reduced payoff, each employee-agent $i \in \{1,\ldots,n\}$ chooses among its K-stable coalitions those which give him a payoff at least equal to $\eta \times u_{i,\max}$. After that, it communicates the results to the coordinator-agent which randomly selects a coalition among the group of coalitions that have a bilaterally acceptance of the two members. The detailed procedure for coalition's selection is defined as detailed on Figure 1.

4 Computational Experiments

In this section, we present experimental results concerning several shift scheduling problems in the context of a paced multi-product assembly center. A set of five problems is conducted to test the performance of the proposed multi-agent approach. For each problem, the production planning is spread out over 60* 9-hour shift. Experiments have been performed in an assembly line consisting of 40 workstations. The takt time between two product units is preset equal to 15 minutes. For each workstation, the employee requirements in a given period are determined according to the assembly activities to be fulfilled on the scheduled product according to the preset line balancing. Concerning the staff, we consider that the offer and the demand per shift for employees vary between 150 and 200 employees. The daily absenteeism rate varies according to the shift number, from a minimum of 1% to a maximum of 5% of total employees.

To evaluate the quality and the efficiency of the proposed multi-agent approach, we report solutions obtained on the same problems through the simulated annealing approach (SA). A detailed description of the proposed simulated annealing algorithm and its use for resolution of personnel scheduling problems can be found in (Sabar 2008, Sabar et al. 2012). For each shift s, we report the cumulated deviation CD_s between the best solutions founded by these two approaches for a computation time equal to 10 minutes for scheduling and 3 minutes if rescheduling is required.

$$\mathbf{CD_s} = \frac{\text{MAScumulated cost at (s) - SA cumulated cost at (s)}}{\text{SA cumulated cost at (s)}}.$$

Figure 2 exhibits the evolution of this deviation between SMA and SA results. It shows clearly that for the five test problems the proposed SMA approach leads to high quality solutions in comparison with the SA approach. It is interesting to observe that SMA systematically outperforms SA for all shift results. Indeed, we notice that the deviations of the SMA approach solutions from SA range between -4.2 % and -0.7 %. These results demonstrate that the proposed multi-agent approach for personnel scheduling is effective and can generate high-quality solutions fast and reliably.

5 Conclusion

In this article, we developed a multi-agent approach for the personnel scheduling/rescheduling problem in the context of a paced multi-product assembly center. The proposed approach is based on cooperation among several rational agents which encapsulate individual competencies and preferences of workers. The experiments we have performed demonstrate that the multi-agent approach can produce high-quality and efficient solutions in comparison with simulated annealing approach.

Our future research will focus on the impact of dynamic random events such as product quality issues on the line, and probabilistic operation times potentially depending on the operator's skill level. In addition, we will investigate the impact of modeling employee preferences on the quality of the scheduling solutions obtained.

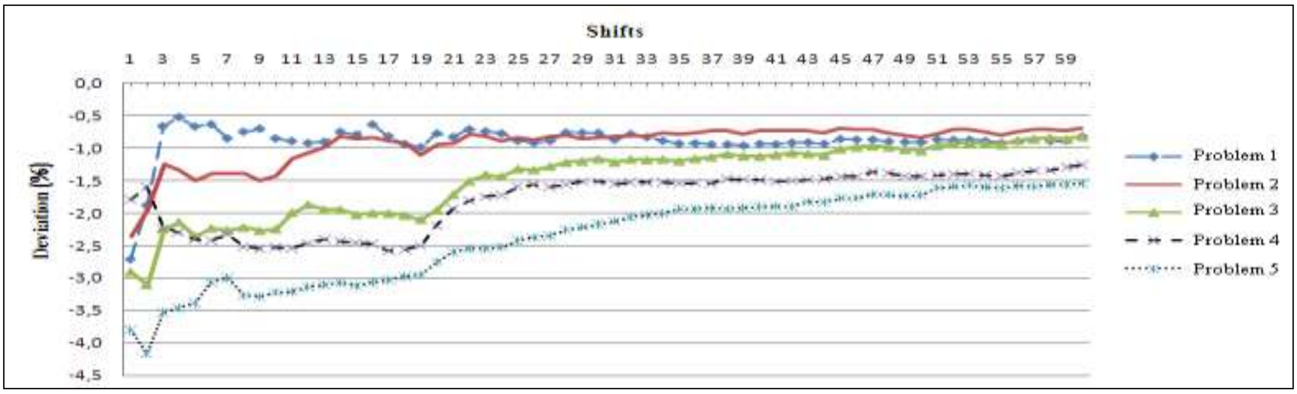

Fig. 2. Deviation between MAS and SA results

6 References

1. A. Schuldt,(2011), Multiagent coordination enabling autonomous logistics. Berlin:Springer Verlag .

2. Davis, M., Maschler, M., (1965). The Kernel of a cooperative game. *Naval research Logistics Quarterly*. 12: 223-259.

3. Ernst, AT., Jiang, H., Krishnamoorthy, M., Sier, D. (2004). Staff Scheduling and Rostering, A review of applications, methods and models, *European Journal of Operational Research*. 153 : 3-27.

4. Hao, G., Lai, K.K., and Tan, M. (2004). A Neural Network Application in Personnel Scheduling. *Annals of Operations Research*. 128: 65-90.

5. Leitão, P., Barbosa, J., Trentesaux, D.: Bio-inspired Multi-Agent Systems for Re-configurable Manufacturing Systems. Engineering Applications of Artificial Intelligence 25, 934–944 (2012)

6. Oliveira E., Fischer K., Stepankova O. (1999). Multi-agent systems: which research for which applications. *Robotics and Autonomous Systems*. 27: 91-106.

7. Rapoport, A. (1970). *N-person game theory: concepts and applications*. University of Michigan Press.

8. Russell, S.J. , Zilberstein,S. (1991). Composing Real-Time Systems. *Proceedings of the Twelfth International Conference on Artificial Intelligence*. 212-217.

9. Sabar, M., Montreuil, B., Frayret, J.-M. (2008). *Competency and preference based personnel scheduling in large assembly lines. International Journal of Computer Integrated Manufacturing*. 21: 468 - 479.

10. Sabar, M. (2008). *A multi-agent based approach for personnel scheduling in assembly centers*, Doctoral Thesis, Université Laval, Canada.

11. Sabar M, Montreuil B, Frayret J-M (2012), An agent-based algorithm for personnel shift-scheduling and rescheduling in flexible assembly lines. Journal of Intelligent Manufacturing23(6):2623–2634

12. Stearns, R. E. (1968). Convergent Transfer Schemes for N-Person Games. *Transactions of the American Mathematical Society*. 134: 449-459.

13. Thompson, G.M. (1995). Improved implicit optimal modeling of the labor shift scheduling problem. *Management Science*. 41: 595-607.

14. Wooldridge, M. (2002). An Introduction to Multi-agent Systems. *Published by John Wiley & Sons*, Chichester, England