J Intell Manuf (2012) 23:2623–2634
DOI 10.1007/s10845-011-0582-9

An agent-based algorithm for personnel shift-scheduling and rescheduling in flexible assembly lines

M. Sabar · B. Montreuil · J.-M. Frayret

Received: 19 March 2010 / Accepted: 22 July 2011 / Published online: 17 August 2011

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

Keywords Personnel shift scheduling and rescheduling · Multi-agent systems · Coalition · Kernel stability · Cross-training · Flexible assembly lines

M. Sabar (✉) · B. Montreuil · J.-M. Frayret
CIRRELT Research Center, Quebec, QC, Canada
e-mail: Mohamed.Sabar@cirrelt.ca

B. Montreuil
e-mail: Benoit.Montreuil@cirrelt.ca

J.-M. Frayret
e-mail: Jean-marc.Frayret@polymtl.ca

M. Sabar
Institut Supérieur de Commerce et d'Administration des Entreprises, Casablanca, Marocco

B. Montreuil
Canada Research Chair in Enterprise Engineering, Quebec, QC, Canada

B. Montreuil
NSERC/Bell/Cisco Business Design Research Chair, Quebec, QC, Canada

B. Montreuil
Université Laval, 2642, Pavillon Palasis-Prince, Quebec, QC G1K 7P4, Canada

J.-M. Frayret
École Polytechnique de Montréal, Quebec, QC, Canada

Introduction

The significance of the personnel scheduling problems as managerial and strategic issues has grown considerably with the expansion of quick-response, agile, lean and personalized manufacturing (Montreuil and Poulin 2005). The high requirements for both productivity and reactivity create a major challenge. It consists of adapting the personnel planning and scheduling processes in order to deal efficiently with the dynamic nature of manufacturing and ultimately to sustain competitive advantage.

In this paper, we focus on personnel scheduling and rescheduling in a dynamic environment of a paced multi-product assembly center. In such an environment, large assembly lines sequentially producing a variety of complex products often require tens up to hundreds of assemblers. The number of assemblers required at each station of the assembly line varies depending both on the product currently assembled and the assembly activities assigned to that station. Owing to product changeovers and the specific manpower competency requirements associated to each product at each station, there are often large waves of personnel moves among stations, which cause significant disruptions to operations, deterring the overall productivity of the line and causing dissatisfaction among the personnel. Furthermore, because an assembly center is unexpected and dynamic events occur, rescheduling is necessary to update a personnel schedule when a disturbance or change occur make it infeasible.

In this context, we present a multi-agent based approach (MAS) that aims to tackle the complexity of our targeted personnel scheduling and rescheduling problem through distributed problem-solving. This MAS approach is tested in a dynamic environment under different workload situations and personnel absenteeism events. Thereby, we evaluate this approach by comparing computational results with the optimal solution and those obtained through a simulated annealing approach (SA).

The remainder of the paper is organized as follows. Section "Problem definition and classification" introduces the personnel scheduling problem. Section "Multi-agent systems for scheduling problems" defines multi-agent settings and presents in detail the multi-agent architecture retained to sustain our scheduling and rescheduling approach. The fourth section presents the formal description of the multi-agent approach. The experimental setup and the results are discussed in Section "Computational experiments". Finally, the sixth section presents summary and conclusive remarks.

Problem definition and classification

Personnel scheduling problems are particular cases of allocation resource problems (Hao et al. 2004). They can take several configurations according to the characteristics of the organizational environment and the duration of the planning period. Generally, they aim to construct a working timetable for each employee by defining start time periods, duration of work, break intervals, as well as the workstation of 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 labour requirements, individual preferences or specific competencies (Thompson 1995; Ernst et al. 2004; Sabar et al. 2008). Personnel scheduling problems can be found in several types of industrial or service companies. They are recurring problems in domains such as transport (Brusco et al. 1995), health (Aickelin and Dowsland 2000), education (DeGans 1981), industrial production (Berman et al. 1997; Lee and Vairaktarakis 1997; Vairaktarakis and Kim Winch 1999), call centres (Atlason and Epelman 2004), as well as protection and emergency services (Ernst et al. 2004).

Personnel scheduling problems are typically classified in three categories (Bailey and Field 1985; Ernst et al. 2004). First, days-off scheduling problems deal with the assignment of days off to employees (e.g. scheduling 2-day or 3-day off patterns into one week). Second, shift scheduling problems typically deal with the elaboration of 8-h or 9-h shifts that must be allocated to employees across a daily planning horizon, in order to meet demand. Third, tour scheduling problems deal with the construction of a weekly set of work schedules for employees. Typically, these latter problems integrate the days-off and shift scheduling problems. To this basic classification can be added contextual parameters, which allow the description of specific personnel scheduling problems connected with a particular activity domain. Generally, these parameters reflect the organizational, temporal or spatial specificities. Examples are: (1) demand nature, such as cyclic versus acyclic and determinist versus stochastic (Baker 1976; Easton and Rossin 1996; Easton and Mansour 1999); (2) employee preferences (Topaloglu and Ozkarahan 2004; Sabar et al. 2008); (3) employee seniority (Volgenant 2004); (4) composition of working teams expressed in terms of part-time employee percentages in comparison to full-time employees (Brusco et al. 1995; Brusco and Jacobs 1998), and (5) scheduling flexibility (Bailey and Field 1985; Baker 1976; Sabar et al. 2008). Personnel scheduling problem have long been recognized as being complex and hard to solve, being identified as NP-complete (Garey and Johnson 1979; Bartholdi 1981).

In this article, we focus on shift scheduling and rescheduling 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 ($ST_1, ST_2, \ldots ST_M$) responsible to sequentially assemble different product-models. For each product, there is a predetermined line balancing which specifies the assembly tasks to be realized at each station when this product is assembled. An assembly activity is defined by the vector < workstation number, assembly task to be fulfilled >. Each assembly activity requires one or several employees who contribute to their execution. Each assembly activity a requires ω_a employees with competency profile $P_a = \langle c_{1a}, \ldots, c_{la}, \ldots, _{La} \rangle$. We consider that $c_{la} = 1$ if competence l is required for fulfilling a; and zero otherwise. The number of employees present in the system can vary according to demand. However, at the beginning of every shift, a minimum number of employees are required. If necessary, other operators can be introduced into the assembly line. We suppose that every employee e introduced into the assembly system has a required minimal presence duration equal to $d_{\min,e}$ periods.

In addition, concerning the manpower pool, we take into account individual competencies, mobility and preferences. Specifically, we consider that:

– Workers each have a specific degree of cross-training enabling them to carry one or several types of assembly activities during a work shift. So each worker e possesses a number of competencies referred to as competency vector $P_e = \langle c_{1e}, \ldots, c_{le}, \ldots, _{Le} \rangle$. We consider that $c_{le} = 1$ if worker e has acquired competence l; and zero otherwise. To allocate worker e to the execution of activity a, the inclusion relation must be satisfied between sets P_e and P_a. Indeed, they have to satisfy the relation: $P_a \subseteq P_e$.

- Workers are allowed to move between workstations in order to fulfill specific assembly activities as per the product assembly schedule.
- In cases where the number of present workers exceeds the total requirements of the assembly activities during period t, extra workers can be assigned to execute secondary activities, to be trained in other stations, or to perform elementary administrative operations, according to specified availability of such work.
- 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.
- Two levels of shift structure flexibility are taken into consideration: shift start-time flexibility and break-placement flexibility. For shift start-time flexibility, we define a set of planning periods T_q in which the shift can be started. Relative to break-placement flexibility, we define three types of pauses: first-half-shift break (pause type 1), lunch break (pause type 2), and second-half-shift break (pause type 3). For every kind of break "i" we associate two time intervals. The first interval reflects the periods along which the pause can spread out. The second interval defines the periods during which this break can be started. The duration of each break is fixed.

Figure 1 illustrates the organizational structure of the selected personnel scheduling problem. It shows an example of the master assembly schedule and the principles employees' states and assignment during a given shift.

In a previous article (Sabar et al. 2008) we have presented a formal description and mathematical modeling of this multi-objective decision problem. The experimental results have demonstrated, on the one hand, that for smaller cases, it is possible to reach the optimal solution using a commercial solver, and on the other hand, that optimal resolution times tend to be huge for larger size cases. Moreover, results have shown that even for smaller cases, commercial solvers cannot be reliably used in a fast interactive and uncertain environment requiring the generation of solutions in short times, pointing towards the need to develop heuristic optimization approaches. In the second article (Sabar et al. 2009), we have described a multi-agent based approach for personnel scheduling problems in the context of a paced multi-product assembly center, and we have benchmark the performance of our multi-agent approach against optimal solutions obtained through a linear programming model resolution using a commercial solver. Experimental results have shown that our multi-agent approach can produce high-quality and efficient solutions in a short computational time.

In this article, we are primarily concerned with rescheduling problem. We consider that if a disturbance related to operators' absenteeism occurs at a given time period, the variables representing the personnel' scheduling up to

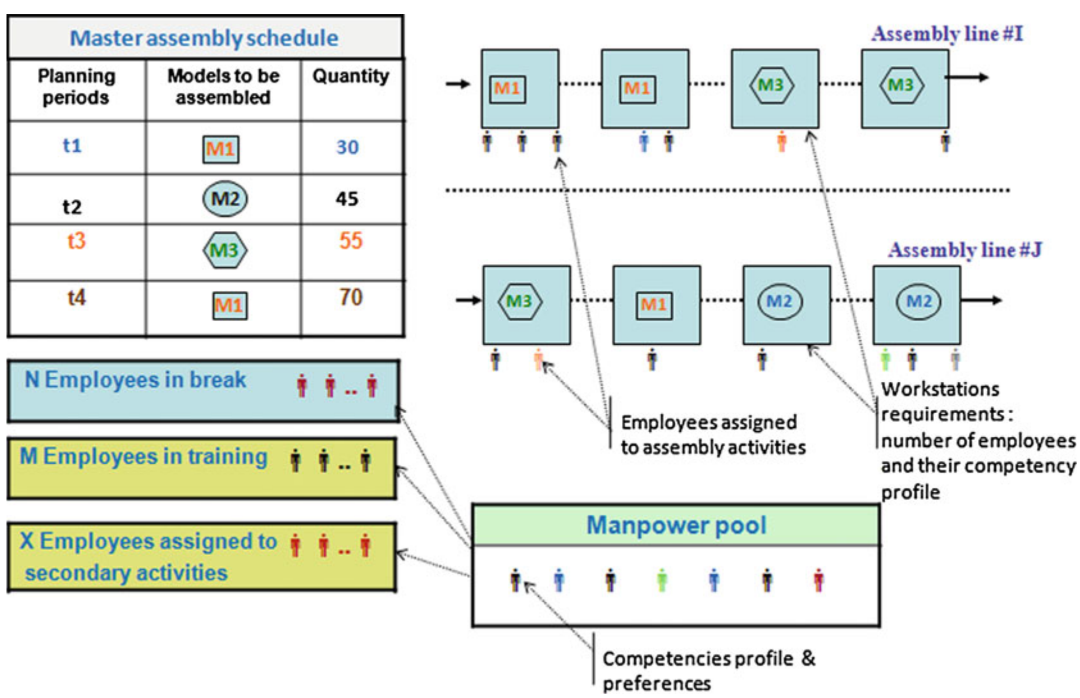

Fig. 1 Organizational structure of the selected personnel scheduling/rescheduling problem

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. Regarding 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 who 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.

In order to evaluate the performance of our MAS scheduling & rescheduling approach, we perform two sets of benchmark analyses:

- For pure scheduling problems, we report optimal solutions and those obtained through MAS and SA approaches for small and medium scale problems. The aim of this test is to evaluate the deviation of MAS and SA approaches from the optimal solution. Experiments have been performed for an assembly line consisting of 5-workstations. We consider employees in attendance. A sample data set is provided in Sabar et al. (2009).
- For scheduling & rescheduling problems, we compare MAS solutions against simulated annealing approach for large-scale problems. Taken into consideration that linear programming based approach for personnel scheduling optimization becomes difficult to solve once the problem dimensions, given by time horizon, set of workstations and employees' characteristics become large. Therefore, we choose to validate the MAS approach by comparing the deviation of 5 large-scale and generic problem instances with the solutions obtained with the SA approach. Experiments have been performed for an assembly line consisting of 40-workstations. 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.

Multi-agent systems for scheduling problems

Definitions

Multi-agent systems paradigm takes inspiration from the distributed artificial intelligence field. This paradigm provides a new approach to deal with the complexity of manufacturing systems scheduling through distributed problem-solving. The main element in multi-agent systems is the "agent". A number of researchers have proposed definitions of an "agent". Russel and Norvig (1995, p. 32) define an agent as "anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators". According to Wooldridge (2002, p. 15), "An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives". There is, as yet, no consensus definition of an intelligent agent. However, the following properties are often associated with the notion of an intelligent agent: autonomy, social ability, reactivity and pro-activeness (Wooldridge and Jennings 1995).

Multi-agent system is defined as a collection of autonomous agents that communicate with one another and coordinate their activities to reach an overall goal, while simultaneously each agent pursues individual objectives (Oliveira et al. 1999). In order to solve complex problems in a distributed manner, agents interact through mechanisms such as negotiations, cooperation, coordination or simple messages passing (Wooldridge 2002). In agent systems, collaboration may lead to emergent properties (Lesser 1999) and results which cannot be predicted by the analysis of each agent' actions separately. This is a desired characteristic of MAS systems.

Several research projects have already investigated the application of multi-agents systems in the manufacturing field, in particular for the area of enterprise integration (Maturana and Norrie 1996; Shen and Norrie 1998), and the domain of manufacturing scheduling and control (Duffie and Piper 1986; Kouiss et al. 1997; Miyashita 1998; Parunak et al. 2001; Shen et al. 2006; Monostori et al. 2006; Frayret et al. 2007).

Multi-agent architecture proposed for personnel scheduling problem

A multi-agent architecture can be defined as a set of system design guidelines in which decisions and reasoning are distributed among the agents (Gokturk and Polat 2003). Such architecture generally describes two levels. The first level contains the entities related to the real environment which we seek to model (e.g. physical entities; external entities and decisional entities). The second level represents the agents which are responsible for handling and controlling the actions of these entities. In addition, multi-agent architecture requires addressing many issues, including: what agents to design and what solution tasks to assign to each agent; how to manage information exchange and communication; and how to coordinate agent interaction and control execution of agent tasks.

In this article, the real environment to model is an assembly lines system. It is made up of several workstations, which constantly require a mix of cross-trained employees. In order to model such 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, as depicted in Fig. 2: a production-agent; station-agents (one for each workstation); a coordinator agent; and employee-agents (one for each employee). These agents are autonomous, rational and able to communicate with each other.

(1) **Production-agent**: It 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 communicated 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. This decision process is thus an input required for the definition of a personnel schedule. In further research, we intend to build a bidirectional interaction between production and personnel planning, but it is out of scope for this article.
(2) **Station-agent**: It manages and controls the assembly activities of a workstation. Based on the production planning, this agent defines the needs of the workstation in terms of a number of employees and required competencies, which are sent to the coordinator-agent. A Station-agent behaves like a material requirement planning system (MRP) to define the aggregate demand for employees at the corresponding station.
(3) **Coordinator-agent**: It is responsible for coordinating the employee-agents. First, it elaborates an initial solution of personnel scheduling. Then, it takes the active role of a mediator in the negotiation process among the employee-agents who will try to improve their initial work plans through activities swapping.
(4) **Employee-agent**: It 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 between 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.

These four types of agents are deliberative. They have an internal symbolic reasoning model of the world and themselves (Wooldridge 2002). Each agent uses its model to reason about situations that are desirable. We consider that Coordinator-agent and Employee-agents are utility-based agents, whereas Production-agent and Station-agents are goal-based agents (Russel and Norvig 1995).

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 the needs for employees and their competencies. Next, the initial work plan of each employee is transmitted to the corresponding Employee-agent.

Considering that the initial solution is often of mediocre quality in terms of total cost and employees' satisfaction, we use the concept of coalition 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 an interest in forming coalitions, which release him from less satisfying assembly activities and/or allow 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 to 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

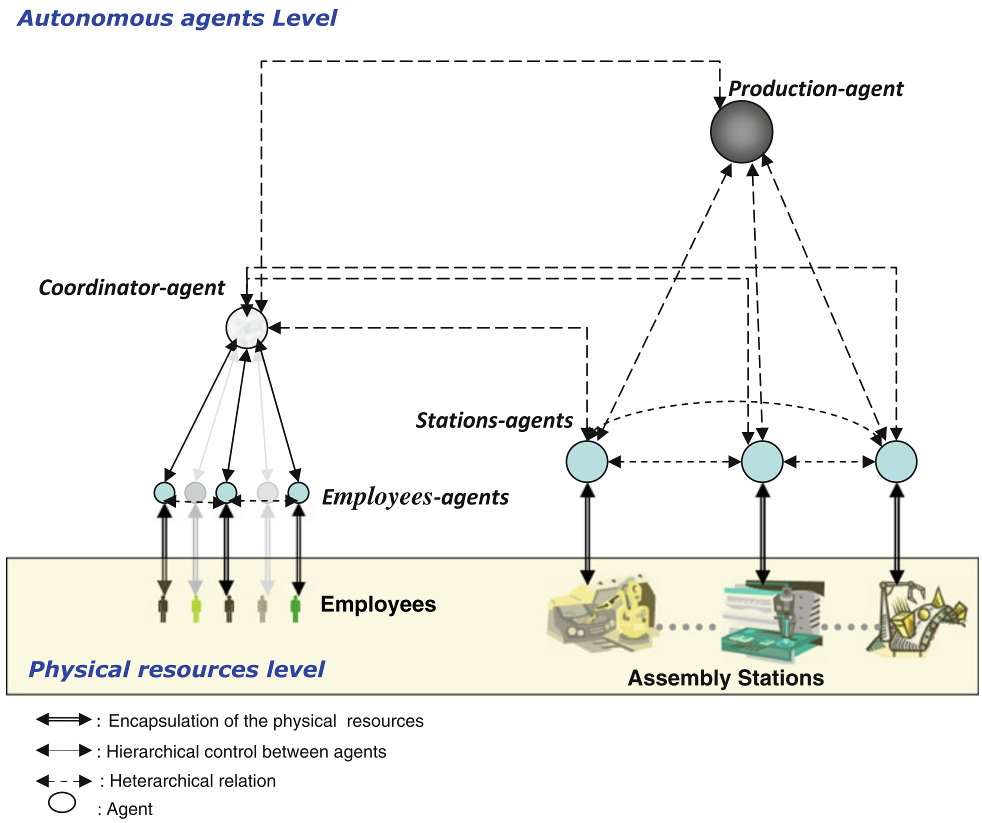

Fig. 2 Multi-agent architecture for personnel scheduling problem

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 Stearns (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 from 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 and Zilberstein 1991).

Multi-agent approach for personnel scheduling and reshduling

In this section, we formally present our multi-agent approach for personnel scheduling and rescheduling. 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.[1] It should be noted down 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.

[1] Details and principles of the proposed personnel scheduling approach, including Coalition formation, payoff distribution and Kernel concepts are detailed on Sabar et al. (2009).

Table 1 Algorithm to generate coalitions with K-stable payoff distributions (Phase 1)

Each agent-employee i:

2.1.	Maintains a register concerning the references of the employee-agents with whom it can exchange certain activities.
2.2.	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 permutations 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 period 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.
2.3.	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).
2.4.	Receive coalition proposals from the other employees-agents.
2.5.	Evaluate the received coalition proposals :
2.5.1.	Use the Kernel concept to test the proposals coalitions equilibrium;
2.5.2.	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.
2.6.	For each instable coalition, send or receive a part of payoff equal to the side-payment demand.

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 competencies 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 $i \in N = \{1, \ldots, n\}$ possesses a vector of activities to perform $\rho^i = \left[a^i_{nm,0}, \ldots, a^i_{kl,t}\right]$, 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 $i \in N = \{1, \ldots, n\}$, 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:[2]

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

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 higher a 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 (Table 1).

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

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

Table 2 Algorithm for coalition selection (Phase 2)

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 coordinator-agent.

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

•$BC \neq \phi$: several coalitions have a bilaterally acceptance from their two members:

– In scheduling cases, Coordinator-agent randomly selects a coalition from BC. In rescheduling 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.

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 $f_{reg}(i, \eta) : u_{i,\max} \rightarrow \eta \times u_{i,\max}$, 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 from the two members. The detailed procedure for coalition's selection is defined as follows (Table 2).

Computational experiments

In this section, we present experimental results concerning several shift scheduling and rescheduling problems in the context of a paced multi-product assembly center. Two data sets, made up of different combinations of shifts requirements and employees characteristics (i.e. competencies and preferences) to represent realistic situations, were generated to compare the results of the solutions MAS and simulated annealing to the optimal. Our empirical evaluation is composed of two sections:

- First, for pure scheduling problems, a set of six problems is conducted to test the performance of the proposed multi-agent approach. We report optimal solutions and those obtained through simulated annealing approach for small and medium scale problems. Experiments have been performed for an assembly line consisting of 5-workstations. We consider employees in attendance. A sample data set is provided in Sabar et al. (2009).
- Then, for scheduling and rescheduling problems, we compare MAS solutions against simulated annealing approach for five large-scale problems with long time horizon (60-shifts). Experiments have been performed for an assembly line consisting of 40-workstations. 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.

For all instances, the daily planning horizon equals to a 9-hour shift. Therefore, 60 of these daily problems are solved for each of the five scenarios, for a total of 300 daily scheduling problems. The takt time between two product units is preset equal to 15 min. Consequently, the employees' daily schedule is spread out over 36 15-min periods. 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. This line balancing states the assembly activities to be performed for each product at each station.

MAS and SA approaches were implemented in AnyLogic^{TM} tool, which offers an environment for agent based approaches development.[3] The optimal schedule was found by resolving the linear programming model (Sabar et al. 2008) in CPLEX 10.

Concerning the results obtained by CPLEX, we do not specify an upper bound on computation time. Our interest is to find optimal solutions, and to have an idea about the computation time needed to reach optimality. These elements will

[3] Technologies Company (http://www.xjtek.com).

enable us to evaluate the quality and the efficiency of MAS and SA approaches. For these two approaches, we report the value of the best solutions founded within various durations for solution improvement. The selected durations are: d = 15 s, d = 30 s, d = 60 s, d = 120 s and d = 300 s.

Pure scheduling problems

Table 3 provides information regarding the optimal solutions generated by CPLEX. It gives the number of iterations up to optimality, the running time, optimal objective function value and optimal number of employees for each problem.

Tables 4 and 5 present the characteristics of the best solutions found through the MAS and SA approaches. They provide, on the one hand, details concerning the generated initial solutions, on the other hand, the evolution of solutions quality according to the duration of computation time.

Table 6 exhibits the results from the experiments expressed as a percentage deviation from optimality of the best solutions founded after 300 s running time.

The effectiveness of the MAS and SA approach in dealing with personnel scheduling problem is apparent from the results shown in Table 6. When running with 300 s computational time, the MAS and SA approaches yielded average

Table 3 Computational results for CPLEX

Problems	Number of iterations	Running time (s)	Optimal objective function ($)	Optimal employees' number
1	795,865	2,520.47	2,468.4	25
2	1,228,410	3,791.20	2,901.4	28
3	1,288,361	5,934.64	2,870.6	30
4	2,109,420	11,817.23	3,121.2	34
5	2,589,034	16,317.99	4,143.9	42
6	4,392,967	28,419.56	5,098.8	51

Table 4 Computational results for multi-agent approach

Problems	Initial solution			MAS solution improvement (coalitional process)									
				15 s		30 s		60 s		120 s		300 s	
	Run time (s)	Obj. func ($)	Emp nbr.	Obj. func ($)	Emp nbr.	Obj. func ($)	Emp nbr.	Obj. func ($)	Emp nbr.	Obj. func ($)	Emp nbr.	Obj. func ($)	Emp nbr.
1	4.4	2,787	28	2,562	26	2,559	26	2,545	26	2,526	26	2,515	25
2	4.8	3,325	30	3,091	29	3,043	29	2,991	29	2,990	29	2,958	29
3	5.1	4,389	40	2,964	31	2,966	30	2,919	31	2,903	30	2,894	30
4	5.9	4,153	50	3,371	37	3,313	35	3,234	35	3,203	35	3,203	35
5	7.1	5,492	47	4,293	44	4,250	44	4,160	43	4,160	43	4,160	43
6	7.6	6,296	54	5,360	54	5,296	53	5,227	52	5,126	52	5,126	52

Table 5 Computational results for simulated annealing approach

Problems	Initial solution			SA solution improvement (iterations)									
				15 s		30 s		60 s		120 s		300 s	
	Run time (s)	Obj. func ($)	Emp nbr.	Obj. func ($)	Emp nbr.	Obj. func ($)	Emp nbr.	Obj. func ($)	Emp nbr.	Obj. func ($)	Emp nbr.	Obj. func ($)	Emp nbr.
1	4.4	2,787	28	2,639	27	2,585	27	2,570	26	2,551	26	2,616	25
2	4.8	3,325	30	3,153	30	3,073	30	3,111	29	3,110	30	3,017	29
3	5.1	4,389	40	3,053	32	2,990	31	2,986	31	2,948	31	2,923	31
4	5.9	4,153	50	3,446	38	3,405	36	3,331	36	3,299	35	3,229	35
5	7.1	5,492	47	4,422	46	4,293	45	4,285	43	4,220	44	4,180	43
6	7.6	6,296	54	5,521	56	5,384	55	4,349	54	5,231	53	5,177	53

Table 6 Percentage deviation from optimality

Algorithm	Problem					
	1 (%)	2 (%)	3 (%)	4 (%)	5 (%)	6 (%)
MAS						
Objective function	1.89	1.95	0.82	2.62	0.39	0.53
Number of employees	0.00	3.57	0.00	2.94	2.38	1.96
SA						
Objective function	5.98	3.98	1.83	3.45	0.87	1.53
Number of employees	0.00	3.57	3.33	2.94	2.38	3.92

cost performance within 0.39 to 5.98% of optimality for all six problems configurations.

These results also indicate that the proposed multi-agent approach can potentially lead to high quality solutions compared to SA. The differences between MAS and SA cost performance were statistically significant. On average, it is equal 1.58%.

Overall, the multi-agent approach behaves in a similar way for the six test problem in terms of solution improvement. Indeed, we notice that initials solutions are significantly improved by the coalition's formation process within the first 30 s as shown in Table 4. The deviations of optimal solutions decrease drastically when the running time is extended to 300 s. Specifically, we observe, in this case, that the deviations from optimality of the multi-agent approach solutions range between 0.39 and 2.62%. On average, the deviation equals 1.37%. Furthermore, concerning the number of employees, an examination of results shows that for two instances the multi-agent approach values match the optimal values obtained by the CPLEX solver. The others instances have a deviation from optimal number equals one employee.

Scheduling and rescheduling problems

To evaluate and confirm the efficiency of the proposed multi-agent approach, we report solutions obtained for five large-scale problems through the simulated annealing approach (SA). The planning horizon equals to a 60 × 9-h shift. A detailed description of the concerned problems can be found in (Sabar 2008). 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 min for scheduling and 3 min if rescheduling is required.

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

Figure 3 exhibits the evolution of this deviation between MAS (multi-agent system) and SA results. It shows clearly that for the five test problems the proposed MAS approach leads to high quality solutions in comparison with the SA approach. It is interesting to observe that MAS systematically outperforms SA for all shift results. Indeed, we notice that the deviations of the MAS 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 generates high-quality solutions fast and reliably.

The computational experiments demonstrate that the proposed multi-agent approach for personnel scheduling is effective and could generate in a short time high-quality quality solutions. This approach becomes even more interesting when considering the tradeoff between solution quality and computational effort especially in a fast interactive environment requiring the generation of best solutions in short times.

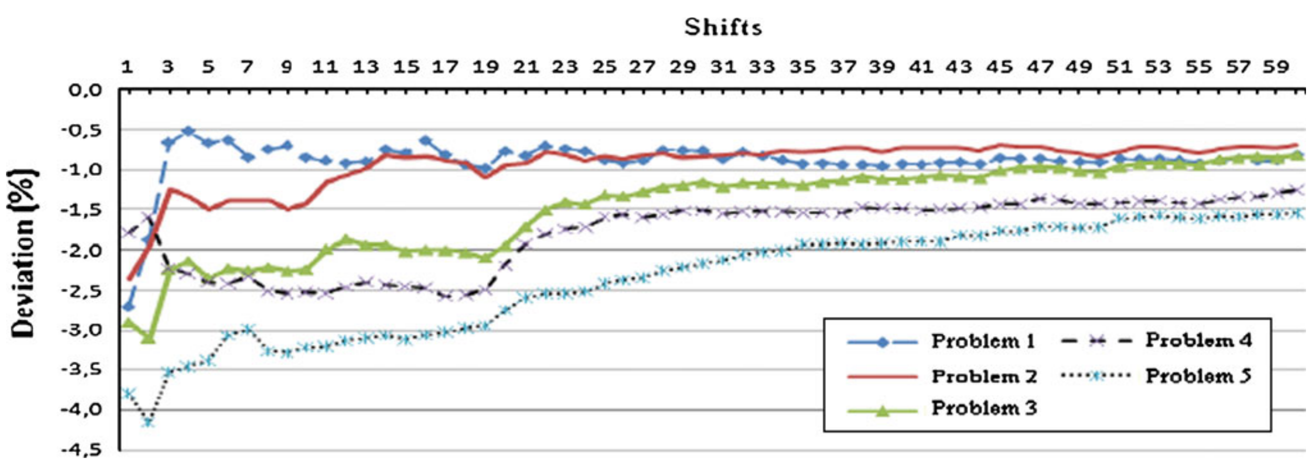

Fig. 3 Deviation between MAS and SA results

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.

There are at least two major directions for future research. First, our following 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. Then, we will investigate the impact of modeling employee preferences on the quality of the scheduling solutions obtained.

References

Aickelin, U., & Dowsland, K. (2000). Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem. *Journal of Scheduling, 3*, 139–153.

Atlason, J., & Epelman, M. A. (2004). Call center staffing with simulation and cutting plan. *Annals of Operations Research, 127*, 333–358.

Bailey, J., & Field, J. (1985). Personnel scheduling with flexshift models. *Journal of Operations Management, 5*, 327–338.

Baker, K. (1976). Workforce allocation in cyclical scheduling problems, A survey. *Operational Research Quarterly, 27*, 155–167.

Bartholdi, J. (1981). A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering. *Operations Research, 29*, 501–510.

Berman, O., Larson, R. C., & Pinker, E. (1997). Scheduling workforce and workflow in a high volume factory. *Management Science, 43*, 158–172.

Brusco, M. J., Jacobs, L. W., Bongiorno, R. J., Lyons, D., & Tang, B. (1995). Improving personnel scheduling at airline stations. *Operations Research, 43*, 741–751.

Brusco, M. J., & Jacobs, L. W. (1998). Personnel tour scheduling when starting-time restrictions are present. *Management Science, 44*, 534–547.

Davis, M., & Maschler, M. (1965). The kernel of a cooperative game. *Naval Research Logistics Quarterly, 12*, 223–259.

DeGans, O. B. (1981). A computer timetabling system for secondary schools in the Netherlands. *European Journal of Operational Research, 7*, 175–182.

Duffie, N. A., & Piper, R. S. (1986). Non-hierarchical control of manufacturing systems. *Journal of Manufacturing Systems, 5*, 39–137.

Easton, F. F., & Mansour, N. (1999). A distributed genetic algorithm for deterministic and stochastic labor scheduling problems. *European Journal Of Operational Research, 118*, 505–523.

Easton, F. F., & Rossin, D. F. (1996). A stochastic goal program for employee scheduling. *Decision Sciences, 27*, 541–568.

Ernst, A. T., 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.

Frayret, J.-M., D'Amours, S., Rousseau, A., Harvey, S., & Gaudreault, J. (2007). Agent-based supply chain planning in the forest products industry. *International Journal of Flexible Manufacturing Systems, 19*(4), 358–391.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability, a guide to the theory of NP-completeness*, WH Freeman & Co, New York.

Gokturk, E., & Polat, F. (2003). Implementing agent communication for a multiagent simulation infrastructure. In *On HLA. Proceedings of the international symposium on computer and information science, LNCS*, Springer, New York.

Hao, G., Lai, K. K., & Tan, M. (2004). A neural network application in personnel scheduling. *Annals of Operations Research, 128*, 65–90.

Kouiss, K., Pierreval, H., & Mebarki, N. (1997). Using multi-agent architecture in FMS for dynamic scheduling. *Journal of Intelligent Manufacturing, 8*(1), 41–47.

Lee, C., & Vairaktarakis, G. (1997). Workforce planning in mixed model assembly systems. *Operations Research, 45*(4), 553–567.

Lesser, V. R. (1999). Cooperative multiagent systems: A personal view of the state of the art. *IEEE Transactions on Knowledge and Data Engineering, 11*(1), 133–142.

Maturana, F., & Norrie, D. (1996). Multi-agent mediator architecture for distributed manufacturing. *Journal of Intelligent Manufacturing, 7*, 257–270.

Miyashita, K. (1998). CAMPS: A constraint-based architecture for multi-agent planning and scheduling. *Journal of Intelligent Manufacturing, 9*(2), 147–154.

Montreuil, B., & Poulin, M. (2005). Demand and supply network design scope for personalised manufacturing. *Production Planning & Control, 15*, 454–469.

Monostori, L., Vancza, J., & Kumara, S. R. T. (2006). Agent-based systems for manufacturing. *CIRP Annals-Manufacturing Technology, 55*, 697–720.

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

Parunak, H. V. D., Baker, A. D., & Clark, S. J. (2001). The AARIA agent architecture: From manufacturing requirements to agent-based system design. *Integrated Computer-Aided Engineering, 8*, 45–58.

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

Russell, S. J., & Zilberstein, S. (1991). Composing real-time systems. In *Proceedings of the twelfth international conference on artificial intelligence*, pp. 212–217.

Russel, S., & Norvig, P. (1995). *Artificial intelligence: A modern approach*. Prentice Hall, New Jersey.

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.

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

Sabar, M., Montreuil, B., & Frayret, J. -M. (2009). A multi-agent-based approach for personnel scheduling in assembly centers. *Engineering Applications of Artificial Intelligence, 22*(7), 1080–1088.

Shen, W., Hao, Q., Yoon, H. J., & Norrie, D. H. (2006). Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced Engineering Informatics, 20*, 415–431.

Shen, W., & Norrie, D. H. (1998). An agent-based approach for distributed manufacturing and supply chain management. In G. Jacucci (Ed.), *Globalization of manufacturing in the digital* *communications era of the 21st century: Innovation, agility, and the virtual enterprise*. Boston: Kluwer Academic Publishers.

Stearns, R. E. (1968). Convergent transfer schemes for N-person games. *Transactions of the American Mathematical Society, 134*, 449–459.

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

Topaloglu, S., & Ozkarahan, I. (2004). An implicit goal programming model for the tour-scheduling problem considering the employee work preferences. *The Annals of OR Special Issue on Staff Scheduling and Rostering, 128*, 135–150.

Vairaktarakis, G., & Kim Winch, J. (1999), Worker cross-training in paced assembly lines. *Manufacturing & Service Operations Management, 1*(2), 112–131.

Volgenant, A. (2004). A note on the assignment problem with seniority and job priority constraints. *European Journal of Operational Research, 154*, 330–335.

Wooldridge, M. (2002). *An introduction to multi-agent systems*. Chichester: Wiley.

Wooldridge, M., & Jennings, N. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review, 10*(2), 115–152.