Reactive scheduling using a multi-agent model: the SCEP framework

Bernard Archimède, Thierry Coudert

Reactive scheduling using a multi-agent model: the SCEP framework

B. Archimede*, T. Coudert

LGP (Laboratoire Génie de Production), ENIT (École Nationale d'Ingénieurs de Tarbes), 47, Avenue d'Azereix, BP 1629, 65016 Tarbes, France

Abstract

Multi-agent systems have been successfully applied to the scheduling problem for some time. However, their use often leads to poorly unsatisfactory disappointing results. A new multi-agent model, called supervisor, customers, environment, producers (SCEP), is suggested in this paper. This model, developed for all types of planning activities, introduces a dialogue between two communities of agents leading to a high level of co-operation. Its two main interests are the following: first it provides a more efficient control of the consequences generated by the local decisions than usual systems to each agent, then the adopted architecture and behaviour permit an easy co-operation between the different SCEP models, which can represent different production functions such as manufacturing, supply management, maintenance or different workshops. As a consequence, the SCEP model can be adapted to a great variety of scheduling/planning problems. This model is applied to the basic scheduling problem of flexible manufacturing systems, and it permits a natural co-habitation between infinite capacity scheduling processes, performed by the manufacturing orders, and finite capacity scheduling processes, performed by the machines. It also provides a framework in order to react to the disturbances occurring at different levels of the workshop.

Keywords: Decision support; Co-operative scheduling; Distributed approach; Multi-agent systems; Reactivity

1. Introduction

Present manufacturings systems are complex sets of interrelated activities and are often used in a highly unpredictable and competitive environment. The performance of these systems depends on the efficient control structures, which allow to manage their activities properly (Lin and Solberg, 1994). In order to remain competitive, these control structures require properties such as decentralisation, reaction capabilities, integration, robustness and flexibility.

Scheduling is one of the most significant activities of the control system since it aims at planning orders on machines in order to ensure customer satisfaction (through the respect of the due dates) and system profitability (through an efficient use of the resources). Manufacturing scheduling problems are usually solved by mono-criteria methods. However, the scheduling process should take into account objectives and constraints from related services, such as maintenance, manpower management or supply management. Therefore, the scheduling process should be closer to a compromise-based solving method than to a mono- or multi-criteria optimisation method.

Multi-agent systems (MAS) aim at solving complex problems by co-operation between several autonomous agents (Ferber, 1999; Wooldridge and Jennings, 1995). Conflict situations can be solved with negotiation techniques, in which the compromises can moderate the satisfaction and frustrations of the agents. The multi-agent approaches give a satisfactory robustness according to the context evolution and the occurrence of disturbances. Nevertheless, negotiation is usually based on local considerations, leading to a disappointing performance (such as respect of the due dates or minimisation of the work in progress).

A new multi-agent model, called supervisor, customers, environment, producers (SCEP), is suggested in this paper. It can be used in order to solve the scheduling problem. The multi-agent methods applied to the scheduling domain are analysed in Section 2, then the SCEP model is described and formalised in Section 3. The use of this model is detailed in Section 4 and illustrated in Section 5. The reaction capabilities of the model are developed in Section 6. The

*Corresponding author. Tel.: +33-5-6244-2734; fax: +33-5-6244-2708.

E-mail address: archi@enit.fr (B. Archimede).

Contract Net model (Smith, 1980), taken as a benchmark in Section 7, assesses the performance of the SCEP model. The perspectives of this approach are developed. They include the co-operation of different SCEP models in order to perform schedules based on co-operation between different production functions (manufacturing, maintenance, manpower management, supply management…) or different workshops.

2. FMS scheduling multi-agent systems

Several works on MAS implementation have been achieved for flexible manufacturing systems. The eco-resolution principle presented by Ferber (Ferber, 1989) is used in Sohier et al. (1998) in order to take into account the real-time control of a flexible cell. A multi-agent system for the dynamic control of a production cell is presented in Ouelhadj et al. (1998). The communication protocol between the different agents is based on the Contract Net model (Smith, 1980). An approach in which the machine and the cell agents interact in order to absorb the internal machine disturbances is described in Tranvouez et al. (1998). Other approaches based on MAS are used to model different decision centres of a production system. A market-driven Contract Net has been proposed by Baker (Baker, 1998). It provides a direct negotiation between the customers and the shop floor agents associated with resources. In (Krothapalli and Deshmukh, 1999) a method for designing decentralised control protocols for discrete part manufacturing systems is suggested. This method includes both the concurrency and the pre-emption schemes within a common framework. An MAS in which the machines and the products are modelled by agents which interact by the Contract Net protocol (Smith, 1980) in order to schedule the operations is proposed in Saad et al. (1997). Two ways of using the system are considered for execution: (1) the machines select the product to manufacture or (2) the product selects the machine on which it will be carried out. The products are sequentially processed according to the order date. For each product, a reservation process is achieved on all the machines. This process consists in finding a schedule by scheduling all the operations of the routing one by one. In this example, the proposition of reservation of a new operation does not cancel the previous reservations. The best proposition is accepted without negotiation between the machine and the product agents, and the other propositions are cancelled. During the scheduling process, only local decisions are made. This method emphasises the speed to obtain a solution. This method is interesting for the real-time control of the production cells, and it can be considered as a manufacturing order positioning method performed in a distributed context.

The method suggested in this paper is similar to the method proposed in Saad et al. (1997) since it sets out a co-operation between the manufacturing order agents and the machine agents (Archimede, 1998; Archimede and Coudert, 1998). However, it presents a negotiation between these agents based on a distribution of the decisional activities. A kind of competition between the manufacturing order agents is developed. The suggested propositions made by the machine agents may be rejected by the manufacturing order agents if they consider that the proposition can be improved in the future. The propositions of the manufacturing order agents concerning all the operations of the routing are taken into account by the machine at the same time. Then, a global visibility (i.e. a time horizon in which the operations are taken into account) is identified, which allows to satisfy the customer objectives and the production system objectives. This method focuses first on the quality of the solution, then on the calculation time. This calculation time is acceptable in previsional scheduling.

3. SCEP multi-agent model

3.1. *Presentation*

The SCEP multi-agent model proposed (Fig. 1) is a distributed model, which introduces an indirect co-operation between manufacturing order agents representing the set of customers called C and machine agents representing the set of producers called P. This co-operation is performed synchronically via the blackboard environment E and is controlled by the supervisor agent S. The scheduling is achieved after a defined number of cycles, since the algorithm convergence is shown later on. Each cycle corresponds to an activation of customer agents followed by an activation of the producer agents.

In the following section the formalism used in order to show the convergence of this model is presented. The environment E is composed of a set of objects O that evolve according to the influence that they receive from the customer and producer agents. In E, the position of an object o is defined by two co-ordinates $([d,f], n)$, where $[d,f]$ represents the abscissa segment and n the ordinate of o. The abscissa segment $[d,f]$ is a continuous temporal interval

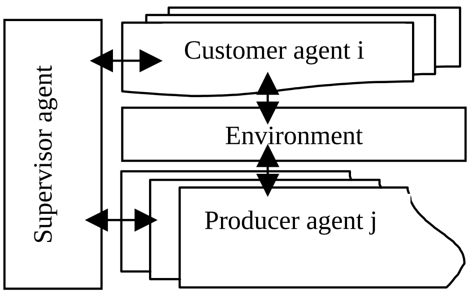

Fig. 1. The SCEP multi-agent model.

between a starting date d and a final date f. Whatever the co-ordinate, f is strictly superior to d, and n is a positive integer or zero. The co-ordinate is partially defined if the ordinate n is zero; otherwise it is completely defined.

Given two objects i (respectively j) whose co-ordinates in E are $([d_i,f_i], n_i)$ (respectively $([d_j,f_j], n_j)$), we can define the following relations between i and j as:

$i = j$(i has the same temporal position as that of j) $\Leftrightarrow d_i = d_j$ and $f_i = f_j$,

$i \neq j$(i does not have the same temporal position as that of j) $\Leftrightarrow d_i \neq d_j$ or $f_i \neq f_j$,

$i \approx j$(i has an equivalent temporal position as that of j) $\Leftrightarrow f_i = f_j$,

$i < j$(i has a best temporal position as that of j) $\Leftrightarrow f_i < f_j$ or $f_i = f_j$ and $d_i > d_j$,

$i \leqslant j$(i has a best or equivalent temporal position as that of j) $\Leftrightarrow f_i \leqslant f_j$ or $f_i = f_j$ and $d_i \geqslant d_j$,

$i \cong j$(i overlap j) $\Leftrightarrow [d_i,f_i[\cap[d_j,f_i[\neq \varnothing$.

An object can be influenced by only one customer agent and by several producer agents. Each customer agent i possesses an intervention domain $Dc^i(O)$ composed of all objects which may be influenced by him. This domain has the following property:

$$\forall (i,j) \in C^2, \quad Dc^i(O) \cap Dc^j(O) = \varnothing, \quad O = \cup Dc^i(O), \quad i = 1, 2, \ldots, |C|.$$

Each producer agent i possesses an intervention domain $Dp^i(O)$ composed of all objects which may be influenced by him. This domain has the following property:

$$\exists (i,j) \in P^2, \quad Dp^i(O) \cap Dp^j(O) \neq \varnothing \quad O = \cup Dp^i(O), \quad i = 1, 2, \ldots, |P|.$$

We note that $P(o) = \{i \in P / o \in Dp^i(O)\}$ the set of producer agents which influences the object o.

Remark. A necessary condition for the system is that each object of the environment belongs to the intervention domain containing at least one producer agent: $\forall o \in O, \ P(o) \neq \varnothing$.

In the environment, the state of an object depends on different influences received by the customer agent and the concerned producer agents. It is impossible for two objects to have the same final position. The final position of an object results in a compromise through time between the influences resulting from the customer agent and those resulting from the concerned producer agents.

Let PO be the set of all possible positions in the environment. The environment state E^k at a given moment k is a sub-set of $PO \times P(PO) \times P(PO)$ in which each element $e^k(o)$ represents the state of a particular object o.

Let $pe_m^k(o)$, (*resp.* $pp_m^k(o)$) be the effective position *(resp. potential)* of the object o resulting from the influence of cycle k of the producer agent m. The state $e^k(o)$ of the object o in cycle k is defined by the triplet $(ps^k(o), pe^k(o), pp^k(o))$ which represents the propositions resulting from the influences of the agents the object, where:

$ps^k(o)$ is the position requested by the customer in cycle k,

$pe^k(o) = \{pe_m^k(o), \ \forall m \in P(o)\}$ is the set of effective positions in cycle k for the object o,

$pp^k(o) = \{pp_m^k(o), \ \forall m \in P(o)\}$ is the set of potential positions in cycle k for the object o.

The effective position results from the scheduling of all the tasks associated with the propositions collected from the environment. The potential position results from the scheduling of one task associated with a proposition collected from the environment. We note:

$mpe^k(o) = i \in pe^k(o)/i<j, \quad \forall j \in pe^k(o)$ the best effective position for object o in cycle k and

$mpp^k(o) = i \in pp^k(o)/i<j, \quad \forall j \in pp^k(o)$ the best potential position for object o in cycle k.

The customer agent collects the tendencies (received propositions) from the environment, takes its decisions and product influences (sent propositions). While producing an influence in cycle k on an object o in its intervention domain $Dc^i(O)$, the customer agent i defines its state $e^k(o)$. The customer agent i tries to push o to its best position according to its own objectives, taking account of its state in cycle $k-1$. It takes into account the last wish that it expressed for this object and the tendencies of the environment in cycle $k-1$. This position can be defined partially or entirely. The influence of the customer agent i can formally be defined by the function:

$$c^i : Dc^i(O) \times E^{k-1} \to E^k$$

given $o \in Dc^i(O)$, $\quad ps^{k-1}(o) = ([d,f],0)$, $\quad mpe^{k-1}(o) = ([x,y],n)$ and $mpp^{k-1}(o) = ([z,t],u)$

$$(1) e^k(o) = c^i(o, e^{k-1}(o)) = (([d,f],n), \varnothing, \varnothing) \text{ if } ps^{k-1}(o) = mpe^{k-1}(o) \text{ and } [d,f] \neq [0,0],$$

$$(2) e^k(o) = c^i(o, e^{k-1}(o)) = (([x,y],n), \varnothing, \varnothing) \text{ if } ps^{k-1}(o) \neq pe^{k-1}(o), mpe^{k-1}(o) \approx mpp^{k-1}(o)) \text{ and } [d,f] \neq [0,0],$$

$$(3) e^k(o) = c^i(o, e^{k-1}(o)) = (([r,s],0), \varnothing, \varnothing) \text{ if } [d,f] = [0,0],$$

$$(4) e^k(o) = c^i(o, e^{k-1}(o)) = (([a,b],0), \varnothing, \varnothing) \text{ if } ps^{k-1}(o) \neq mpe^{k-1}(o), mpp^{k-1}(o) < mpe^{k-1}(o)) \text{ and } [d,f] \neq [0,0],$$
$$\text{where } a \geqslant z \text{ and } b = a + f - d.$$

In Eq. (3), $([r,s],0)$ represents the initial influence of the customer agent i for object o. The evaluation of abscissa $[r,s]$ only depends on internal constraints of the customer agent i.

In Eq. (4), the evaluation of abscissa $[a,b]$ depends on the internal constraints, on the customer agent i and on the possible availability of the producer agents concerned with object o. The producer agent gets the tendencies of the environment, makes its decisions and produces its influences. While producing an influence in cycle k on object o in the intervention domain $Dp^i(O)$, the producer agent i modifies its state $e^k(o)$. The producer agent i tries to push o to two completely defined positions where he would like to see him: on a potential and effective position. The producer agent i only influences an object o if the latter is on a partially defined position. The influence producer agent can be formally defined by the following function:

$$p^i : Dp^i(O) \times E^k \to E^k$$

given $o \in Dp^i(O)$, $\quad ps^k(o) = ([d,f],n)$, $\quad pe^k(o) = \varnothing$ and $pp^k(o) = \varnothing$

$$e^k(o) = p^i(o, e^k(o)) = \begin{cases} (([d,f],n), pe^k(o) \cup \{pe_i^k(o)\}, pp^k(o) \cup \{pp_i^k(o)\}) \text{ if } n = 0 \text{ with} \\ pe_i^k(o) = ([x,y],i) \text{ and } pp_i^k(o) = ([z,t],i), \\ \text{where } x \geqslant d,\ y \geqslant f,\ y - x \geqslant f - d,\ z \geqslant d,\ t \geqslant f,\ t - z \geqslant f - d,\ t - z = y - x, \\ ([z,t],i) \leqslant ([x,y],i),\ ([d,f],n) \leqslant ([x,y],i),\ ([d,f],n) \leqslant ([z,t],i). \end{cases}$$

The evaluation of $pe_i^k(o)$ or $pp_i^k(o)$ depends on the state and the internal behaviour of the producer agent i. A state of a producer agent i is defined in cycle k by $Cf_i^k(O)$: the set of objects in its intervention domain for which he has found a completely defined position.

$$Cf_i^k(O) = \{o \in Dp^i(O)/e^k(o) = (([x,y],i), \varnothing, \varnothing)\}.$$

With an internal behaviour imposing a strict sequence between the objects, the definition of the function has to be enriched by the following constraint where the effective propositions of two distinct objects cannot overlap.

$$\forall (o_1, o_2) \in Dp^i(O)^2 \text{ with } pe_i^k(o_1) = ([x_1, y_1], i),\ pe_i^k(o_2) = ([x_2, y_2], i) \text{ then } [x_1, y_1[\cap [x_2, y_2[= \varnothing.$$

3.2. Dynamic of the model

The following algorithm describes the behaviour of the supervisor agent. This agent begins by creating an agent society, generating the objects and initialising the environment. After this it launches the co-operation process. It first activates the customer agents and tells the producer agents to wait. Each customer agent performs its actions at the same moment and independent from the behaviour of the other customer agents. The customer agent informs the supervisor once its actions are finished. Once the end of the action of the last customer agent has been recorded, the supervisor activates the producer agents and tells the customer agents to wait.

The producer agents perform their actions independently and they inform the supervisor as soon as they finish their activity. The end of the action of the last producer agent sends the control to the customer agents.The alternation between the activation of the customer agents and the producer agents is repeated until the final positioning of all the environmental objects is effective. The algorithm is briefly described:

Algorithm

```
1. Initialisation
$k = 0$
$\forall i \in O, e^k(i)) = (([0,0],0), \varnothing, \varnothing)$
2. while $\exists i \in O / ps^k(i) = ([z,t],0)$ do
        begin
        for all $l \in C / \exists j \in Dc^l(O)$ verifying $ps^k(j) = ([x,y],0)$ do
                begin
                for all $j \in Dc^l(O) / ps^k(j) = ([x,y],0)$ do evaluate $e^{k+1}(j) = c^l(j, e^k(j))$
                end
        for all $m \in P / \exists j \in Dp^m(O)$ verifying $ps^{k+1}(j) = ([x,y],n)$ do
                  begin
                  $Cf_m^{k+1}(O) = Cf_m^k(O)$
                  for all $j \in Dp^m(O)$ verifying $ps^{k+1}(j) = ([x,y],n)$ do
                          begin
                          if $n = 0$ then evaluate $e^{k+1}(j) = p^m(j, e^{k+1}(j))$
                          else if $(n = m$ and $j \notin Cf_m^{k+1}(O))$ then $Cf_m^{k+1}(O) = Cf_m^{k+1}(O) \cup \{j\}$
                          end
                  end
        $k = k + 1$
        end
```

3.3. Convergence of the model

We suppose that the model is convenient and the producer agents are always available. Each object of the environment belongs to the intervention domain of at least one producer agent and belongs to the intervention domain of a unique customer agent. The latter algorithm finishes when all objects have found a definite position in the environment. The analysis of the definition of the customer and producer agents shows that the producer agents will always try to push the objects to completely defined positions. This is not the case for the customer agents. So, the algorithm will diverge if the customer agents push the objects constantly to partially defined positions.

The purpose of this paragraph is to show that in each cycle, the final position of at least one object is found. Therefore, we suppose that the k first resolution cycles have been achieved, and we note O_1 (respectively O_2) the set of objects where the coordinates are partially defined (respectively entirely defined), whatever the considered cycle.

$O_1 \cap O_2 = \varnothing$ and $O_1 \cup O_2 = O$.

We are interested in the beginning of the $k+1$ cycle of the algorithm, and in the evaluation of the c^i function, on a partially defined object o in its intervention domain: $o \in Dc^i(O)$.

We distinguish between three possible cases:

1. $ps^k(o) = mpe^k(o) = mpp^k(o)$. $ps^{k+1}(o)$ is completely defined. The object is pushed to a final position.

$O_1 = O_1 - \{o\}$; $O_2 = O_2 \cup \{o\}$.

2. $ps^k(o) \neq mpe^k(o)$ and $mpe^k(o) \approx mpp^k(o)$. Waiting is useless. The best effective position proposed by the producer agents cannot be improved. $ps^{k+1}(o) = mpe^k(o)$. The object is pushed to a final position corresponding to the best effective position in cycle k.

   $$O_1 = O_1 - \{o\};\ O_2 = O_2 \cup \{o\}.$$

3. $ps^k(o) \neq mpe^k(o)$ and $mpp^k(o) < mpe^k(o)$. The best effective position can be improved. We pass to the cycle $k+1$ where the best potential position of cycle k becomes the required position. (we note $ps^{k+1}(o) = mpp^k(o)$). The final position of the object cannot be specified at the beginning of cycle $k+1$. Let $CH^k(o) = \{o_i \in Dp^m(O)/pe_m^k(o_i) \cong ps^k(o),\ \forall m \in P(o)\}$ be the set of objects having received in cycle k an influence from one of the producer agents concerned by o to result in an overlapping with the wished position in cycle k for o.

Let us note $o_x \in CH^k(o) \Rightarrow \exists i \in P(o) \cap P(o_x)/pe_i^k(o_x) \cong ps^k(o)$. According to our hypotheses $ps^k(o) \neq mpe^k(o) \Rightarrow ps^k(o) < pe_j^k(o) \forall j \in P(o)$. We deduce that

$$\forall o_i \in CH^k(o),\ \ \exists m \in P(o) \cap P(o_i),\quad pe_m^k(o_i) < pe_j^k(o)\ \ \forall j \in P(o).$$

Two cases are possible:

(a) $\exists$ at least $o_i \in CH^k(o)$ verifying case 1 or 2, then

$$O_1 = O_1 - \{o_i\};\ O_2 = O_2 \cup \{o_i\}.$$

(b) $\forall o \in O_{1,}$ there is no $o_i \in CH^k(o)$ verifying the cases 1 or 2. According to our hypotheses, the cardinal of O is finished, so we deduct:

$\exists$ at least one sequence $o_1, o_2, \ldots, o_x$ verifying $\quad o_2 \in CH^k(o_1),\quad o_3 \in CH^k(o_2), \ldots, o_x \in CH^k(o_{x-1}),\quad o_1 \in CH^k(o_x)$

$$\begin{aligned}
o_2 \in CH^k(o_1) \Rightarrow &\quad \exists p_2 \in P(o_1) \cap P(o_2)/pe_{p2}^k(o_2) < pe_p^k(o_1) \forall p \in P(o_1),\\
o_3 \in CH^k(o_2) \Rightarrow &\quad \exists p_3 \in P(o_2) \cap P(o_3)/pe_{p3}^k(o_3) < pe_p^k(o_1) \forall p \in P(o_2),\\
\vdots\ \ \vdots\qquad &\qquad\qquad\qquad \vdots\ \ \vdots\ \ \vdots\ \ \vdots\\
o_1 \in CH^k(o_x) \Rightarrow &\quad \exists p_1 \in P(o_x) \cap P(o_1)/pe_{p1}^k(o_1) < pe_p^k(o_x) \forall p \in P(o_x).
\end{aligned}$$

We deduce that

$$pe_{p1}^k(o_1) < pe_{px}^k(o_x) < \ldots < pe_{p3}^k(o_3) < pe_{p2}^k(o_2) < pe_{p1}^k(o_1) \Rightarrow pe_{p1}^k(o_1) < pe_{p2}^k(o_2) < pe_{p1}^k(o_1).$$

The effective proposition of the producer agent p_1 for the object o_1 is better than the effective proposition of the producer agent p_2 for the object o_2, which is better than the effective proposition of the producer agent p_1 for the object o_1.

We deduce that $pe_{p1}^k(o_1) \approx pe_{p2}^k(o_2)$. Since $pe_{p2}^k(o_2) \cong ps^k(o)$, $ps^k(o_1) \cong pe_{p1}^k(o_1)$ which contradicts the hypotheses.

So $(o_i, o_j) \in O_1^2$, $o_i \in CH^k(o_j)/o_i$ exists, verifying 1 or 2.

$$O_1 = O_1 - \{o_i\};\ O_2 = O_2 \cup \{o_i\}.$$

We have shown that, at the end of each cycle, the definite position from at least one object has been found. The number of objects of the set O_1 decreases from at least one unit per cycle. The algorithm converges after $|O|$ cycles in the worst situation.

4. Application of the scheduling model on the shop

4.1. Context of the study

In this study we assume that the workshop can achieve several activities like turning, milling, drilling and so on. An activity that has to be performed on each manufacturing order (MO) depending on its routing. The routing is a linear sequence of operations. Each operation requires an activity (e.g. milling, drilling,...) that can be achieved by one or more machines. Each machine has one or more competencies. An activity, which can be achieved by a machine is defined by a competency which gives a capability and cost coefficiency to this activity. A machine cannot have two competencies on the same activity, so the operation can be achieved on one machine to be chosen within a set of possible but not necessary equivalent machines. The processing time varies according to the capability of the chosen machine. The predicted time in the routing is one of the most performing machine that can process the operation. A machine can only execute one operation at a time.

We assume that there is no transport. No set-up time is considered or it can be integrated in the processing time of each operation. Once an operation has started on a machine, it will finish on the same one. We suppose that the number of disturbances is low, that the number of operations to schedule is finished, and that there is no closure time for the machines. The machines are either available, or in processing or in failure after a disturbance.

4.2. Adaptation of the model

For the previously described context, we associate customer agents to manufacturing order agents, and producer agents to machine agents. The objects in the environment are associated to the different operations to be achieved on the manufacturing orders. The set of objects associated to the routing followed by an MO constitutes the intervention domain of the concerned agent. An operation only concerns one MO: its associated object cannot be in two intervention domains of different MO agents. This is in perfect correlation with the definition of the model.

The intervention domain of a machine agent is defined by a set of objects associated to the operations requiring an activity performed by the machine. Given that several machines can achieve the same activity, some objects can belong to several intervention domains. This is also consistent with the definition of the model.

Taking account of the sequence of operations defined in the routing followed by an MO requires a slight modification of the definition of the function associated to the MO agent. The evaluation of this function in the beginning of cycle k for a given object does not only depend on the state of the environment at the end of cycle $k-1$, but also on the current state of the environment of cycle k. This constraint requires to update the state of objects of the intervention domain of an MO agent depending on the order defined by the associated routing.

This constraint does not modify the internal behaviour of the model and does not have any influence on the convergence.

The application of functions associated to the customer and producer agents for the MO and machine agents permits to identify different phases in their behaviour.

The MO agent first perceives the tendencies or propositions for the objects that are not definitively positioned from the environment. After this, it achieves a validation phase of required positions, with their perceived proposition. It schedules the operations, which have not been validated and finishes by influencing the associated objects with the evolved state operations.

The machine agent perceives the tendencies or required positions from the environment for the objects belonging to its intervention domain. It records the validated positions and schedules the tasks associated with the objects which are not definitely positioned. Finally, it influences these objects.

4.3. The MO agent validation phase

The MO agent validation phase consists in verifying each operation of the routing, which has not been validated if there exists a convenient proposition in comparison with the required position. The validation may be global or progressive. The global validation consists in validating all the operations of the routing not yet validated at the same time. If a global validation is not possible, a progressive validation is performed. The progressive method consists in validating the following operations by the sequence defined in the routing. This process ends when one operation cannot be validated. We have to note that even when the first operations have been validated in a progressive way, the MO agent can use a global validation of the other operations for the following cycle. This can be explained by a change of the chosen objective corresponding to taking account of the state of the environment by the MO agent.

The validation phase is based on the use of the sequential multi-graphs of propositions coming from the MO. In an MO agent multi-graph, each arc represents a proposition for one operation of the associated routing and each node represents the transition between two operations. The valuation of an arc is composed of a list of weights associated to the different possible objectives of the customer. The customers are usually interested in the cost and the delay or in any combination of these two factors. For the delay objective, which is the most commonly used, two parameters are used: starting date and due date. These parameters will be used as weights on the arcs of the graph. An additional weight must be introduced for each objective different from the delay objective. In Fig. 2 an MO agent multi-graph representing all the propositions for the n operations of its routing is shown. The valuation of the arc associated to a proposition k for operation j is a triplet (d_k^j, f_k^j, c_k^j), where d_k^j is the starting date of the proposition, f_k^j its end, c_k^j its cost. The MO agent records two propositions for operation 1, four for operation 2, two for operation i and three for operation n. The choice of a global validation is mandatory if at least one coherent path exists to satisfy the defined objective in the multi-graph.

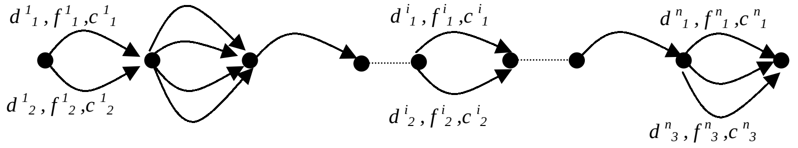

Fig. 2. The sequential multi-graph.

Table 1
Characteristics of the machines

Machine	Rule	Activity	Capability	Cost
1	FIFO	Milling	1	1
2	FIFO	Turning	1	1.7
		Cutting	1	1.7
3	FIFO	Drilling	1	1
		Turning	1.5	1
		Cutting	1.5	1

Table 2
Characteristics of the MOs

MO	Objective	Quantity	Order date	Due date	Max cost	Routing
1	Delay	1	2	26	—	1
2	Cost	1	1	—	24.5	2
3	Delay	1	0	9	—	3

A path is coherent if:

- each arc is associated with only one not validated operation, and each not validated operation is associated with only one arc and,
- for each couple of successive arcs with the respective propositions k for operation i and l for operation $i+1$, we have $d_l^{i+1} \geqslant f_k^i$.

A coherent path satisfies the objective of delay if the final date of the proposition associated to the last arc is inferior to the due date of the associated manufacturing order. It satisfies the cost objective if the sum of the costs of all the propositions of the path is inferior to the required cost.

5. Illustrative example

The considered shop is composed of three machines (Table 1). It can perform activities of milling, drilling, turning and cutting. Each machine can achieve several activities with different capabilities and different costs. The capabilities coefficient permits to adjust the predicted processing time for the operation. The indicated cost is based on the hourly cost of a machine. For an operation with a predicted processing time of twelve units and requiring the turning activity, it will be achieved by machine 2 after 12×1 units with a cost of 20.4 (12×1.7). This operation will be achieved by machine 3 after 18 (12×1.5) units with a cost of 18 (18×1).

Three MOs are to be achieved. The characteristics of the MOs are given in Table 2. We suppose that the objective of MOs 1 and 3 is to satisfy their due date, and the objective of MO 2 is to satisfy the cost objective. All solutions with a cost less than 24.5 are acceptable and equivalent for MO 2. If this cost constraint cannot be respected, MO 2 requires the best cost. We also suppose that the dispatching rule used for managing the queues is FIFO (first in first out).

Each MO uses a linear routing defined in Table 3. The operating times are defined for the most capable machines.

In order to simplify the presentation of our example, we shall focus on the successive states of the environment. For this, we represent the environment as a table where each line corresponds to the state of an object. Each object is characterised by its identifier, the required position of the concerned MO and the potential and effective proposition of

Table 3
Routings

Routing	Operation	Activity	Operating time
1	1	Milling	6
	2	Turning	8
	3	Drilling	5
2	1	Turning	8
	2	Cutting	7
3	1	Drilling	8.5

Table 4
Environment at the initial instant

Object	Wish	Effective propositions	Potential propositions
(1, 1, milling)	([0, 0], 0)	∅	∅
(1, 2, turning)	([0, 0], 0)	∅	∅
(1, 3, drilling)	([0, 0], 0)	∅	∅
(2, 1, turning)	([0, 0], 0)	∅	∅
(2, 2, cutting)	([0, 0], 0)	∅	∅
(3, 1, drilling)	([0, 0], 0)	∅	∅

Table 5
Environment at the beginning of cycle 1

Object	Wish	Effective propositions	Potential propositions
(1, 1, milling)	([2, 8], 0)	∅	∅
(1, 2, turning)	([8, 16], 0)	∅	∅
(1, 3, drilling)	([16, 21], 0)	∅	∅
(2, 1, turning)	([1, 9], 0)	∅	∅
(2, 2, cutting)	([9, 16], 0)	∅	∅
(3, 1, drilling)	([0, 8.5], 0)	∅	∅

the machine agents. The identifier is a triplet (n, t, k) where n is the number of the concerned MO agent, t the number of the associated operation and k the required activity. The format used for the propositions is the same as defined in the model description. The information on format concerns the time and the cost. In order to facilitate understanding, the best effective and potential propositions for each object are given in italics. The costs of the different propositions in the multi-graph are given in bold characters. At the beginning, the different objects are created within the environment with completely undefined co-ordinates (Table 4).

In order to satisfy the time criteria, the MO agents perform an infinite capacity scheduling of the different operations of the routing at the beginning of the first cycle. Then, the objects associated with these operations are influenced within the environment. (Table 5).

The tendencies of the environment are gathered by the machine agents. Machine agent 1 records the tendencies concerning object (1, 1, milling), machine agent 2 records the tendencies concerning (1, 2, turning), (2, 1, turning) and (2, 2, cutting) while machine agent 3 records tendencies for objects (1, 2, turning), (1, 3, drilling), (2, 1, turning), (2, 2, cutting) and (3, 1, drilling). Each machine agent performs a schedule for the different tasks according to the FIFO dispatching rule. Table 6 shows the state of the environment at the end of cycle 1 after collecting the influences of the machine agents. We note that objects (2, 1, turning), (1, 2 ,turning) and (2, 2, cutting) have been influenced by two different machine agents. The analysis of the propositions for these three objects shows the different capabilities of the machine agents.

At the beginning of cycle 2, the MO agents compare the position of the objects with the wished positions. MO agent 3 performs a global validation. A convenient path in fact exists in the graph (Fig. 3c) which finishes at date 8,5 while the required due date is 9. In the graph of propositions of MO agent 2 (Fig. 3b), there are three time consistent paths. The global validation of objects (2, 1, turning) and (2, 2, cutting) is possible because two paths can be found which satisfy the cost objective. MO agent 2 chooses the first one (2, 1, turning) on machine 2 and (2, 2, cutting) on machine 3 with a cost of 24.1. This path is not the best because object (2, 1, turning) on machine 3 and (2, 2, cutting) on machine 3

Table 6
Environment at the end of cycle 1

Object	Wish	Effective propositions	Potential propositions
(1, 1, milling)	([2, 8], 0)	*([2, 8], 1)*	*([2, 8], 1)*
(1, 2, turning)	([8, 16], 0)	*([9, 17], 2)* ([20.5,32.5],3)	*([8, 16], 2)* ([8,20],3)
(1, 3, drilling)	([16, 21], 0)	*([43, 48], 3)*	*([16, 21], 3)*
(2, 1, turning)	([1, 9], 0)	*([1, 9], 2)* ([8.5,20.5],3)	*([1, 9], 2)* ([1,13],3)
(2, 2, cutting)	([9, 16], 0)	*([17, 24], 2)* ([32.5,43],3)	*([9, 16], 2)* ([9,19.5],3)
(3, 1, drilling)	([0, 8.5], 0)	*([0, 8.5], 3)*	*([0, 8.5], 3)*

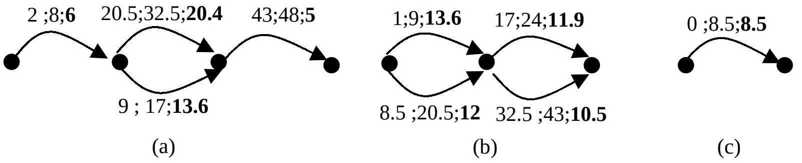

Fig. 3. Multi-graphs at the beginning of cycle 2.

Table 7
Environment at the beginning of cycle 2

Object	Wish	Effective propositions	Potential propositions
(1, 1, milling)	([2, 8], 1)	∅	∅
(1, 2, turning)	([8, 16], 0)	∅	∅
(1, 3, drilling)	([16, 21], 0)	∅	∅
(2, 1, turning)	([1, 9], 2)	∅	∅
(2, 2, cutting)	([32.5, 43], 3)	∅	∅
(3, 1, drilling)	([0, 8.5], 3)	∅	∅

Table 8
Environment at the end cycle 2

Object	Wish	Effective propositions	Potential propositions
(1, 1, milling)	([2, 8], 1)	∅	∅
(1, 2, turning)	([8, 16], 0)	*([9, 17], 2)* ([8.5,20.5],3)	*([9, 17], 2)* ([8.5,20.5],3)
(1, 3, drilling)	([16, 21], 0)	*([20.5, 25.5], 3)*	*([16, 21], 3)*
(2, 1, turning)	([1, 9], 2)	∅	∅
(2, 2, cutting)	([32.5, 43], 3)	∅	∅
(3, 1, drilling)	([0, 8.5], 3)	∅	∅

have a cost of 22.5. In the multi-graph of MO agent 1 (Fig. 3a) there is a time-consistent path, but a global validation cannot be performed because the delay objective is not satisfied. The required due date is 26 while the obtained finish date is 48. The progressive method permits to validate only the position of object (1, 1, milling). After the validation phase, the MO agents achieve a new infinite capacity schedule for the not validated operations. New influences are generated within the environment. The state of the new environment is given in Table 7.

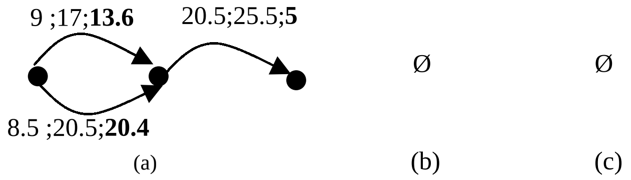

Fig. 4. Multi-graphs at the beginning of cycle 3.

Table 9
Environment at the beginning of cycle 3

Object	Wish	Effective propositions	Potential propositions
(1, 1, milling)	([2, 8], 1)	∅	∅
(1, 2, turning)	([9, 17], 2)	∅	∅
(1, 3, drilling)	([20.5, 25.5], 3)	∅	∅
(2, 1, turning)	([1, 9], 2)	∅	∅
(2, 2, cutting)	([32.5, 43], 3)	∅	∅
(3, 1, drilling)	([0, 8.5], 3)	∅	∅

After the activation of the machine agents by the supervisor agent, the machine agents collect the tendencies from the environment. Machine agent 1 records the final position of object (1, 1, milling). Machine agents 2 and 3 records the final position of objects (2, 1, turning) and (2, 2, cutting). Machine agent 3 registers the final position of object (3, 1, drilling). Machine agents 2 and 3 generate new propositions for the objects, which do not have final positions (Table 8). Then, the control is given to the MO agents.

At the beginning of cycle 3, MO agent 1 identifies a convenient path with a finishing date of 25,5 in its graph of propositions (Fig. 4a). Since the required due date is 26, the MO agent 1 performs a global validation of the propositions. The objects (1, 2, turning) and (1, 3, drilling) are pushed to their final position (Table 9).

The machine agents record the last final positions and finish the cycle. Since all objects have a final position, the supervisor agent stops the scheduling process.

Note that, in this case, if MO agent 1 does not achieve a global validation at the beginning of cycle 3, then the scheduling process requires an additional cycle.

6. Reactivity of the model in the workshop context

A workshop is subject to many disturbances. That is why the predicted established plan has to take into account the occurrence of disturbances during exploitation. The plan has to be as robust as possible: it has to guarantee the respect of customers objectives as long as possible even in the presence of disturbances. It also has to take into account the stability of orders transmitted to the operators. We can identify some aptitudes to this type of reaction in the model. These aptitudes come from the degrees of freedom defined in the dynamic routings and also from the concept of activity. These degrees of freedom can be combined by the MO and the machine agents in order to achieve a reactive co-operation strategy.

6.1. Degrees of freedom based on dynamic routing

The definition of the SCEP model does not introduce precedence constraints to influence the objects belonging to the intervention domain of an MO agent. Then, in the workshop context we can use dynamic routings which can fully or partially integrate an open shop behaviour. A dynamic routing can be considered as a sequence of groups of operations. Each group may be permutable, sequential or synchronous. A permutable group is a flexible group used by the MO agents in order to easily take into account a partial unavailability of resources: in these groups the operations must be achieved sequentially, but the processing order is not a priori defined. The occurrence of disturbance which does not permit to process an operation in this group has for consequence the search of new compromises for the objects associated to the operations of the group. We record, if it is possible, several automatic permutations in the processing order.

6.2. Degrees of freedom based on activities

In the SCEP model, the objects belong to different intervention domains of machine agents. In a workshop context, this degree of freedom is represented by the concept of activity. The final position of an object results from a compromise between a machine agent, which can process the required activity and the concerned MO agent. Each disturbance occurring on the selected machine during the execution of the associated operation can be automatically processed, either through the search of a new compromise with another machine agent which can perform the required activity for this object or, by searching for several new compromises for the objects which are affected by a change in the position of the first perturbed operation. With this degree of freedom, we can take into account the total unavailability of a machine.

6.3. The reactive co-operation strategy

With these two types of degrees of freedom, a reactive co-operation strategy can be developed between the different agents in order to take into account the disturbances. In order to guarantee the stability of the orders released into the shop as long as possible, this strategy emphasises a local re-planning rather than a complete re-scheduling.

After the detection of a disturbance, the process consists in identifying the set of perturbed operations (SPO), the set of concerned machine agents (SCM) and the set of concerned MO agents (SMO). The perturbed operations are identified by a machine or an MO agent by propagation of the disturbance within its local plan. The concerned machine agents are the machine agents, which have been perturbed first and all the machine agents which can achieve the activities required by the perturbed operations. Considering these three sets, the following step consists in performing a new schedule of the affected operations. If the new placements for all these operations are consistent with the date constraint and the routing sequence constraint, then the reaction process is finished. If a coherent placement cannot be found for all the operations of the set SPO, the process continues firstly by adding the new perturbed operations into SPO, then the new concerned machine agents into SCM and finally the new concerned MO agents into SMO. Then, a new schedule is performed. The process is repeated as long as there is no coherent solution. In this method, we can identify several levels in the reaction process according to the number of operations of SPO. Each level corresponds to a tentative to take into account the disturbance. At each new tentative, the number of perturbed operations is more important. The last possible level corresponds to a full re-scheduling of all the operations. Whatever the scheduling type is (local or complete), the control is performed by the supervisor agent.

6.4. Example of reaction

The considered shop is composed of four machines (Table 10). It can achieve the four activities of milling (M), drilling (D), turning (T) and cutting (C). Each machine can perform one or several activities among these four activities: e.g. machine 3 performs the activities of turning and drilling. An activity can be achieved by different machines: e.g. the turning activity can be performed by machine 3 and machine 4. In order to simplify, we suppose that the machines, which can perform the same activity have the same capability. The dispatching rule on the machines is FIFO.

Five MOs are to be achieved. The characteristics of the MOs are defined in Table 11. We suppose that the objective of each MO is to satisfy its due date and we suppose that each MO is composed of one part.

Each MO is associated to a routing (Table 11). The processing times are given in Table 12.

In this example, the routing, which is associated to MO 2, is composed of two groups of operations. The first group is sequential (type Seq.) and the second group is permutable (type Perm.). The other routings are only composed of sequential groups.

Table 10
Characteristics of the machines

Machine	1	2		3		4	
Dispatching rule	FIFO	FIFO		FIFO		FIFO	
Activity	M	D	C	T	D	T	C
Capability	1	1	1	1	1	1	1

Table 11
Characteristics of the MOs

MO	1	2	3	4	5
Quantity	1	1	1	1	1
Order date	0	1	0	6	1
Due date	27	30	25	20	15
Routing number	1	2	3	4	5

Table 12
Routings description

MO	1			2				3		4		5
Group	1	2		1	2			1	2	1	2	1
Type	Seq.	Seq.		Seq.	Perm.			Seq.	Seq.	Seq.	Seq.	Seq.
Operation	1	1	2	1	1	2	3	1	1	1	1	1
Activity	M	T	C	D	T	M	D	T	D	T	C	M
Time	7	4	5	5	6	6	8	5	5	5	7	3

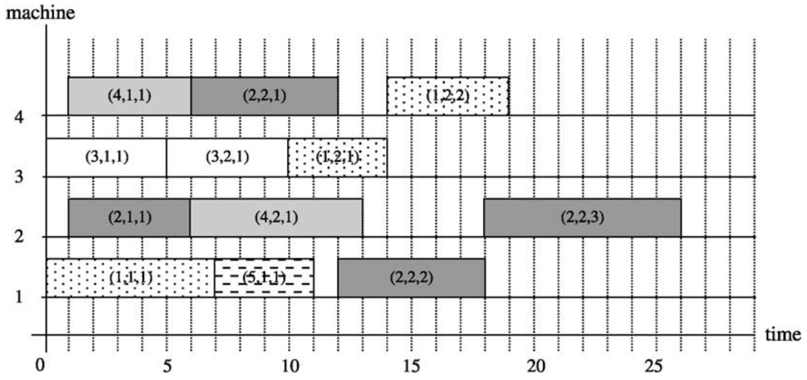

Fig. 5. Gantt chart after the initial scheduling.

Our objective here is to show the behaviour of the model when a disturbance occurs, and to show as to how the model uses the degrees of freedom provided by the concepts of activities and dynamic routing. In order to simplify the presentation, we use a Gantt chart where we can see the position of the different tasks on the machines at the end of the scheduling process and after taking into account a disturbance. We note (i,j,k) the task associated to operation k in group j of MO i.

After six cycles, the initial scheduling process is finished as described in Section 5. The result is presented in the Gantt chart of Fig. 5.

The detection of a disturbance on machine 4 during the execution of task (4,1,1) at time 3, with an estimated duration of 3 time units, causes an increase of the duration of task (4,1,1). Machine agent 4 identifies the perturbed operation (2,2,1) and MO agent 4 identifies the perturbed operation (4,2,1). Then, the set of identified perturbed operations is $\text{SPO} = \{(2,2,1),(4,2,1)\}$, the set of concerned MO agents is $\text{SMO} = \{2,4\}$ and the set of concerned machine agents is $\text{SCM} = \{2,4\}$. Since the required activities of operations (2,2,1) and (4,2,1) are turning and cutting, the set SCM must be completed with machine agent 3 which can achieve the turning activity: so $\text{SCM} = \{2,3,4\}$.

The Gantt chart of Fig. 6 shows the result of the first tentative for taking into account the disturbance. This first level of reaction is achieved by co-operation between the MO agents of the set SMO and the machine agents of the set SCM.

We note that the new placement of operation (4,2,1) is consistent but the placement for operation (2,2,1) is inconsistent because this new placement overlaps with the placement of operations (2,2,2) and (2,2,3). This incoherent schedule makes a second tentative necessary consisting first in an update of the three sets with the new perturbed

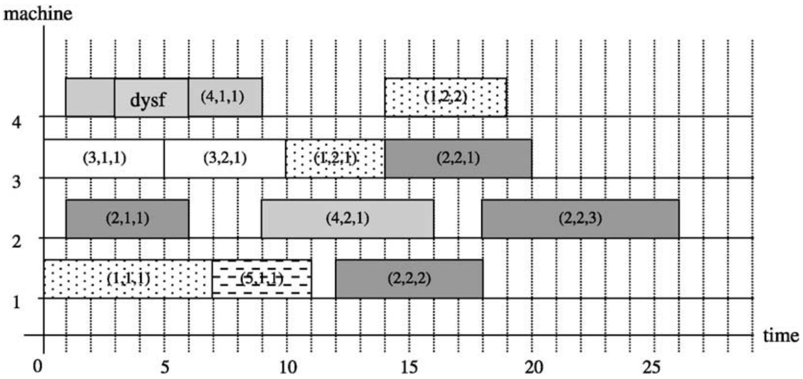

Fig. 6. Gantt chart after the first tentative of reaction.

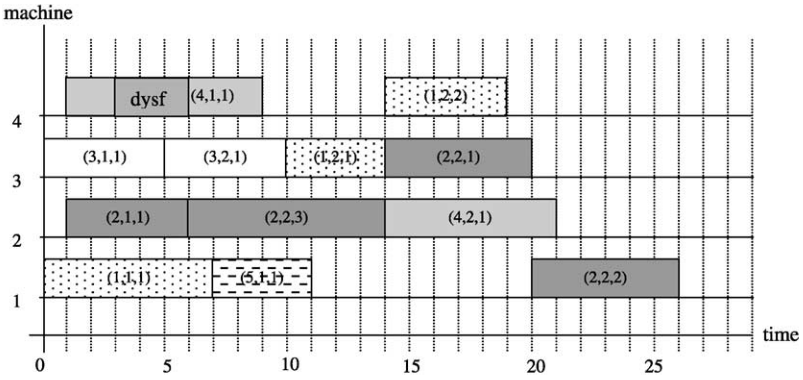

Fig. 7. Gantt chart after the second tentative of reaction.

operations (2,2,2), (2,2,3), the new concerned manufacturing order agents, and the new concerned machine agents. The new set of identified perturbed operations becomes SPO $= \{(2,2,1),(2,2,2),(2,2,3),(4,2,1)\}$, the new set of concerned MO agents is SMO $= \{2,4\}$ and the new set of concerned machine agents is SCM $= \{1,2,3,4\}$.

The Gantt chart of Fig. 7 shows the result of the second tentative for taking into account the disturbance. We note that the schedule is now coherent. A new processing orders are suggested for operations (2,2,1), (2,2,2) and (2,2,3) in comparison with the schedule shown in the Gantt chart of Fig. 5; moreover, a transfer of operation (2,2,1) from machine agent 4 to machine agent 3 has been performed.

7. Drawbacks and performance of this model

In classical methods the transport times are usually integrated in the routings and the estimated set-up times are integrated by the operations. This is possible because the used routing is deterministic. In the SCEP model, the routings are not deterministic and the transport times between two distinct machines in a workshop are most often not equal. These times cannot be integrated in these dynamic routings since operation n can be planned even if the machine that will process operation $n-1$ is not yet know. Similarly, the evaluation of an estimated set-up time requires the knowledge of the last performed operation on the concerned machine and the new operation to be performed. The MO agents cannot take into account these times during the scheduling phase because they do not know which machine will perform a given activity. These times cannot be integrated by the machine agents, which do not know from which machine a task to schedule comes. Taking account of the transport and the estimated machine set-up times in a SCEP model is a complex problem. As a matter of fact, the main problem is due to the need to combine an ad-hoc time horizon of decision-making with the distributed aspect of the model.

The SCEP model is implemented into a mock-up written in C++ language, which permits to test its performance. The proposed evaluation concerns the temporal and qualitative aspects. The objective of the temporal evaluation is to study the shop configuration influence on the processing time. The objective of the qualitative evaluation is to show the interest of the method in comparison with another distributed method based on the Contract Net model (Smith, 1980), which is often used in scheduling based on multi-agent systems.

7.1. The temporal evaluation

In order to be independent from the used computer equipment, here we use indicators like the number of cycles required to obtain a solution and the minimum, maximum or average number of scheduled operations per cycle. The number of cycles is a measurement of the processing time to obtain the solution. The processing time depends on the model complexity. This complexity is expressed in terms of number of activities, number of machines, number of operations, number of operations per activity, number of activities per machine, and so on. The minimum number of scheduled operations permits to verify the convergence of the model. The average number of scheduled operations per cycle is a measurement of the speed convergence of the model. The maximum number of scheduled operations per cycle is a measurement of the time horizon of decision-making of the MO agent. For the same workshop, we consider four configurations (A, B, C, D). Whatever the configuration, the proposed activities and manufacturing orders are the same. In configuration A, each machine performs only one activity and each activity is only possible on one machine. In configuration B, each machine can perform several activities, and each activity is only possible on one machine. In configuration C, each machine can perform several activities, and each activity is possible on several machines. In configuration D, each machine performs only one activity, and each activity is possible on several machines.

The analysis results show that whatever the problem complexity, the required number of cycles to obtain a solution varies linearly (Fig. 8b): the complexity has an influence only on the slope of the line. The analysis of Fig. 8b shows that sharing the working time of a machine between several activities increases the processing time. If this working time is reserved to one activity, the distribution of the activities on several machines reduces the processing time. The convergence speed is more important when the working time of the machines is not shared (Fig. 8c). Whatever the considered configuration, the time horizon of decision-making is not very different (Fig. 8d). It is a little better when the working time of the machines is shared. If this working time is reserved to one activity, then the distribution slightly increases the time horizon of decision-making. The value of the minimal number of operations per cycle having found a final position confirms that at each cycle, at least one operation is validated. The satisfaction of the due dates is better when the working time of the machines is not shared (Fig. 8a). If the working time of a machine is reserved to one activity, the distribution slightly improves this satisfaction.

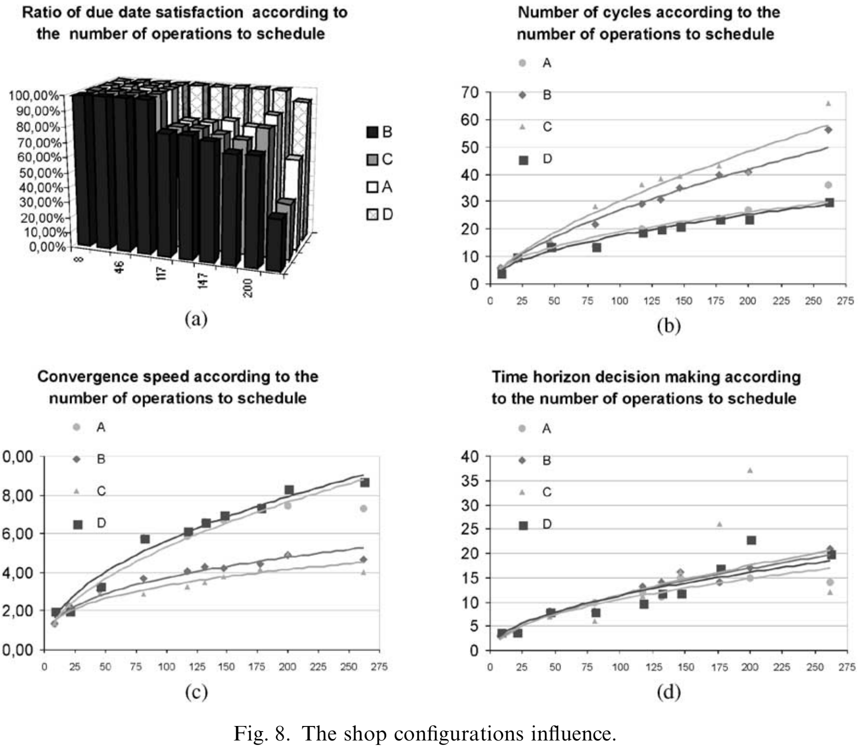

Fig. 8. The shop configurations influence.

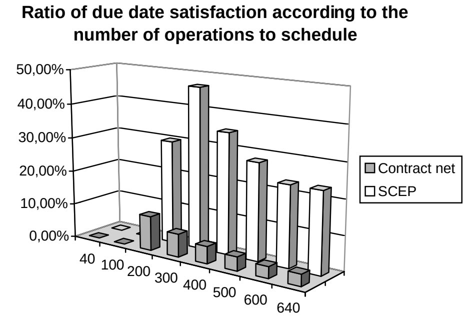

Fig. 9. SCEP/Contract net satisfaction rate.

7.2. Evaluation of the due dates satisfaction

The diagram of Fig. 9 shows the ratio of the due dates satisfaction according to the number of operations to schedule after using the SCEP model and a simulation of the Contract Net approach (Smith, 1980) obtained by a restriction of the SCEP model. Our study is based on a workshop, which is composed of eight machines being able to perform five activities. Five machines perform only one activity among the five possible activities. Each machine among the other three machines performs two activities. An activity can be performed by a maximum of two machines. Each manufacturing order has a routing with twenty operations.

Eight tests have been achieved. The obtained results show that the ratio of the due date satisfaction is most interesting with the SCEP model.

8. Conclusion

A new method based on the multi-agent techniques have been developed in this paper to address the problem of scheduling flexible manufacturing systems. This approach comes up to the customers' expectations using the most adapted production resources. It also takes into account the disturbances at different levels. Therefore, the SCEP model guarantees the correct stability of the transmitted orders to the operators. The selected architecture permits to consider several perspectives such as multi-sites scheduling, co-operation with other functions linked to the production management.

Several instances of the SCEP model have been geographically distributed; connected thanks to the CORBA bus with a hierarchical organisation (Jones and Saleh, 1990). The first results in this multi-site scheduling context are significant. However, these schedules do not consider the transport times yet. Developments are in progress in order to first integrate the transport times between the different sites, then the transport times between the machines on a site.

Other studies concerning the co-operation between the production function and the maintenance function are also in progress (Coudert et al., 1999, 2000). An SCEP model is used to solve the scheduling problem of production operations. Another SCEP is used to solve the scheduling problem of maintenance tasks on the machines of the production system. Both models perform their own scheduling solution while negotiating compromises with the other part. Furthermore, this approach may allow to deal with the scheduling problem of set-up tasks requiring particular resources, since set-up tasks can be considered like maintenance tasks.

References

Archimede, B., 1998. Application of a multi-agent scheduling method for the processing of dynamic routings. IMACS Multiconference on Computational Engineering in Systems Applications: CESA'98, Nabeul-Hammamet, Tunisie, 1–4 April.

Archimede, B., Coudert, T., 1998. A multi-agent scheduling approach for the flexible manufacturing systems. IFAC Workshop on Distributed Computer Control Systems: DCCS'98, Come, Italy.

Baker, A., 1998. A survey of factory control algorithms which can be implemented in a multi agent heterarchy. Journal of Manufacturing Systems 17 (4), 297–320.

Coudert, T., Archimède, B., Grabot, B., 1999. Un système multi-agents pour la coopération production/maintenance. In: Habchi, A. and Haurat, A. (Eds.), Proceedings of the Second Conference on Modélisation et SIMulation: MOSIM'99. SCS International Publications, Annecy, France.

Coudert, T., Grabot, B., Archimède, B., Integration of maintenance constraints in scheduling: fuzzy modelling and multi-agent approach. In: Camarinha-Matos, L.M., Afsarmanesh, H., Erbe, H.H., (Eds.), Advances in Networked Enterprises: Virtual Organisations, Balanced Automation and Systems Integration. Proceedings of the Fourth IEEE/IFIP Conference on Balanced Automation Systems (BASYS'2000). Kluwer Academic Publishers, Berlin, Germany, pp. 297–304.

Ferber, J., 1989. Eco problem solving: how to solve problem by interactions. Proceedings of the Ninth Workshop on Distributed Artificial Intelligence, Rosario Resort, Eastsound, Washington, pp. 113–128.

Ferber, J., 1999. Multi-Agent Systems: an Introduction to Distributed Artificial Intelligence. Addison-Wesley, Reading, MA.

Jones, A., Saleh, A., 1990. A multi level/ multi layer architecture for intelligent shop floor control. International Journal of Computer Integrated Manufacturing 3 (1), 60–70.

Krothapalli, N.K.C., Deshmukh, A.V., 1999. Design of negotiation protocols for multi-agent manufacturing systems. International journal of production research 37 (7), 1601–1624.

Lin, G.Y., Solberg, J.J., 1994. Autonomous control for open manufacturing systems. In: Joshi S.B. and Smith J.S. (Eds.), Computer Control of Manufacturing Systems (Research and development). Chapman & Hall, London.

Ouelhadj, D., Hanachi, C., Bouzouia, B., 1998. Kara Hassan F., Hadj Mohamed N., Multi-agent system for dynamic control in manufacturing cells. IFAC Workshop, Ninth Symposium on Information Control in Manufacturing, Nancy–Metz, France, pp. 153–158.

Saad, A., Kawamura, K., Biswas, G., 1997. Performance evaluation of Contract Net-based heterarchical scheduling for flexible manufacturing systems (special issue on Intelligent Manufacturing Planning and Shopfloor Control). International Journal of Automation and Soft Computing 3 (3), 229–248.

Sohier, C., Denis, B., Lesage, J.J., 1998. Eco-problem solving for the adaptive control of production systems: the Casper project. IFAC Workshop, Ninth Symposium on Information Control in Manufacturing, Nancy–Metz, France, pp. 65–70.

Smith, R.G., 1980. The contract net protocol: high level communication and control in a distributed problem solver. IEEE Transactions on Computer 29 (12), 1104–1113.

Tranvouez, E., Espinasse, B., Chirac, J. P., 1998. A multi-agent based scheduling system: a co-operative and reactive approach. IFAC Workshop, Ninth Symposium on Information Control in Manufacturing, Nancy–Metz, France, pp. 147–152.

Wooldridge, M., Jennings, N.R., 1995. Intelligent agents: theory and practice. The Knowledge Engineering Review 10 (2), 115–152.

Bernard Archimède was born in France in 1959. He received his Ph.D. degree in computer science from the University Bordeaux I in 1991. Since 1992, he has been researcher in the Department Production Automatisée of the Laboratoire Génie de Production of the national school of engineers at Tarbes. He is also Assistant Professor of the Ecole Nationale d'Ingénieurs de Tarbes where he gives lectures on computer science, simulation and computer integrated manufacturing systems. Now, he has broad interest in planning distributed architectures and multi-agents systems.

Thierry Coudert received a PhD from the National polytechnic Institute of Toulouse (France) in 2000. He is now working as a post-doc at National Engineering school of Tarbes (ENIT-France). His research activities are carried out at the Laboratoire Génie de Production. His work mainly concern production activity control, distributed scheduling and multi-agents systems.