AN HETERARCHICAL MULTI-AGENT SYSTEM FOR THE INTEGRATION OF PROCESS PLANNING AND SCHEDULING USING OPERATION-BASED TIME-EXTENDED NEGOTIATION PROTOCOLS

Izabel Cristina Zattar, izabel.zattar@gmail.com
João Carlos E. Ferreira, jcarlos@emc.ufsc.br
Universidade Federal de Santa Catarina, Departamento de Engenharia Mecânica. GRIMA/GRUCON, Caixa Postal 476, 88040-900, Florianópolis, SC, Brazil. Tel.: (0xx48) 3721-9387 ramal 212 fax: (0xx48) 3721-7615

João Gabriel G. G. Rodrigues, gabriel.granado@gmail.com
Universidade Federal de Santa Catarina, Departamento de Engenharia Mecânica. GRIMA/GRUCON, Caixa Postal 476, 88040-900, Florianópolis, SC, Brazil. Tel.: (0xx48) 3721-9387 ramal 212 fax: (0xx48) 3721-7615

Carlos H. Barreto de Sousa, carlos.sousa@gmail.com
Universidade Federal de Santa Catarina, Departamento de Engenharia Mecânica. GRIMA/GRUCON, Caixa Postal 476, 88040-900, Florianópolis, SC, Brazil. Tel.: (0xx48) 3721-9387 ramal 212 fax: (0xx48) 3721-7615

It is proposed in this paper the on-line adaptation of process plan with alternatives, through the application of an operation-based time-extended negotiation protocol for decision-making about real-time routing of job orders of parts composed of machining operations in a job-shop environment. The protocol is modified from the contract net protocol to cater for the multiple tasks and many-to-many negotiations. The grouping of the machining operations enables reduction of setup times, resulting from the reduction of machines changes. For each part, all feasible routings are considered as alternative process plans, provided the different manufacturing times in each machine are taken into account. The time-extended negotiation period allows the visualization of all of the times involved in the manufacture of each part, including those times that are not considered in systems of this nature, such as the negotiation times among agents. Extensive experiments have been conducted in the system, and the performance measures, including routings, makespan and flow time, are compared with those obtained by the search technique based on the co-evolutionary algorithm.

***Keywords**: Integration of Planning and Scheduling, Negotiation Protocol, Agent Technology*

1. INTRODUCTION

Manufacturing process planning is the process of selecting and sequencing manufacturing processes such that they achieve one or more goals and satisfy a set of domain constraints. Manufacturing scheduling is the process of selecting a process plan and assigning manufacturing resources for specific time periods to the set of manufacturing processes in the plan. It is, in fact, an optimization process by which limited manufacturing resources are allocated over time among parallel and sequential activities (Shen et al., 2006). In traditional approaches process planning and scheduling functions are executed in a sequential way, and alternative resources are not usually considered. A large obstacle for the integration between process planning and production scheduling, in dynamic manufacturing environments, is the lack of flexibility for the analysis of alternate resources when allocating the jobs in the shop floor. In this phase, the process plan is treated as fixed, i.e. scheduling does not consider all the possible manufacturing combinations resulting from the use of alternate resources. Besides without the consideration of real-time machine workload and shop floor dynamics, the process plans determined offline during the planning stage are often out of touch with shop floor operations at the time of task execution. Therefore, there is a need for the integration of manufacturing process planning and scheduling activities for generating more realistic and effective plans to be used in the shop floor (Shen et al., 2006).

In spite of the advances in this area, it is observed that in many works that use the multi agent approach, they possess an appealing problem in the form of consideration of the activities that compose the scheduling of an order or a part. In (Usher, 2003), (Zhou et al., 2003) e (Wong et al., 2005) the features that compose each of the parts are treated in an independent way from each other, i.e. a single feature is negotiated at a time between the part and the resources. This kind of treatment may lead to an increase in the setup and queue times, resulting in longer flow times. This increase in the manufacturing times results from the many changes in the machines on which the parts are manufactured, but if the features are grouped based on the setup, it may improve the manufacturing and transport times.

This paper describes a multi agent system with a heterarchical structure for making decisions about the manufacture of parts composed of machining operations in a job shop layout, or similar kind of flexible manufacturing environments. The negotiation between the many agents present in the system is based on the grouping of machining operations with an extended period. Each job is announced, subdivided into operations, and later treated as the total sum of groups of machining operations that compose the process plan of the part. Also, in the proposed method alternative resources can carry out the manufacture of the parts, which increases flexibility in scheduling. The constraints related to the precedence between the machining operations are taken into account in this model, as well as the preparation time machine and fixturing setup times.

This paper presents a brief survey of the previous research in the areas of integration between process planning and production scheduling (section 2), and flexible process planning (section 3). The characteristics of the adopted model are described in section 4, with special emphasis on the grouping by operations and the formulation of the proposal time. The multiagent system is described in section 5, and finally some of the results of the implementation are shown in section 6. The conclusions are in section 7.

2. RELATED RESEARCH

The problem of integrating process planning and production scheduling has been under investigation in the last years, and many different approaches have been applied to accomplish that. The nonlinear process plan concept (NLPP, also called flexible process plan, alternative process plan, or multiple process plans) has been identified as a milestone for process planning and scheduling integration by almost all relevant publications (Shen et al, 2006). Recently many authors have suggested the multiagent systems (MAS) as an adequate approach for solving the process planning and scheduling integration problem. According to (Wang et al., 2003), there are two main reasons for the adequacy of MAS for solving this problem: (a) in the last ten years many steps were taken to improve and validate MAS, which resulted in an increase in flexibility, reuse capability, and scalability; (b) the development and popularity of the Java language, which reduced significantly the effort and time spent in the implementation of MAS.

In spite of the advances in this area, it is observed that in many works that use the multi agent approach, a greater emphasis is on production scheduling, both predictive and reactive, while process planning is treated in a static way, i.e. it is determined before the part is released into production. It is also noticed that despite some authors use dynamic process planning in their approaches, there is a recurring problem related to the researches that study the integration of process planning and production scheduling, which is the consideration of the activities that compose the scheduling of an order or a part. In (Usher, 2003), the features that compose each of the parts are treated in an independent way from each other, i.e. a single feature is negotiated at a time between the part and the resources. This kind of treatment may lead to an increase in the setup and queue times, resulting in longer flow times. This increase in the manufacturing times results from the many changes in the machines on which the parts are manufactured, but if the features are grouped based on the setup, it may improve the manufacturing and transport times. These possible gains are investigated in this paper.

3. FLEXIBLE PROCESS PLANS

Making use of alternative process plans will create a new dimension for scheduling. Conventionally, scheduling tries to compromise job-machine conflicts while each job follows a fixed routing. This type of scheduling may be viewed as a planar, 2-dimensional scheduling, where job (or machine) and time are the two axes. Normally many features can be produced using a number of different methods (e.g., alternative setup, machine, tool configurations) while achieving the required quality specifications.

Operation flexibility relates to the possibility of performing an operation on alternative machines, with possibly distinct processing times and costs. This type is often called routing flexibility, (Lin and Solberg ,1991). Sequencing flexibility corresponds to the possibility of interchanging the sequence in which manufacturing operations required are performed. The first and the second types, respectively, involve alternative machines and alternative sequences, but operations to be performed are fixed. Processing flexibility is determined by the possibility of producing the same manufacturing feature with alternative operations or sequences of operations. Allowing for these flexibilities can provide better performance in mean flow time, throughput, and machine utilization (Lin and Solberg, 1991).

4. CHARACTERISTICS OF THE ADOPTED MODEL

The operation-based time-extended negotiation protocol is an adaptation of the protocol utilized by Usher (2003). Contrary to the typical duration of a negotiation process used in agent-based systems defined by how long it takes for the messages exchanged between the participating agents to be constructed, sent, and responses received, in the operation-based time-extended negotiation protocol the deadline corresponds to a fixed percentage of the expected time that will be required to setup and process the job on the current resource. According Usher (2003) by considering a definite time interval from the onset of negotiation to the response deadline, each resource can negotiate with multiple part agents simultaneously.

However, the protocol suggested by Usher (2003) has as limitations two equally important factors: (a) it does not provide any mechanism for grouping operations that compose a job, and although in that system multiple part agents are coordinated concurrently, each part agent can announce only one single task (or operation) at a time; (b) it considers the setup times independent of operation sequence.

For a better understanding of the setup, the nomenclature used in this paper is presented below. This representation of the variables was adapted from (Conway et al., 1967), and is used to describe both the sequencing problem and the proposed solution:

- i: index of the jobs to be processed by the shop; $1 \leq i \leq n$;
- j: index of the sequence of operations on a job; $1 \leq j \leq g_i$;
- g_i: the total number of operations on job i;
- $p_{i,m}$: amount of time required for resource m to perform the job i;
- $s_{i,m}$: total setup time of job i on resource m;
- $sp_{i,m}$: total machine setup time of job i on resource m. This value is independent of batch size;
- $sf_{i,m}$: represents the total fixture time of job i on resource m. This value is dependent of batch size;
- s_{ik}: represents the sequence-dependent time between jobs i and k.

4.1. Makespan with sequence-dependent setup times

For single machine scheduling problems with all release dates (r_i) equal zero and no sequence-dependent setup times, the makespan is independent of the sequence and equal to the sum of the processing times ($\Sigma p_{i,m}$). On the other hand, when there are sequence-dependent setup times, the makespan depends mostly on the schedule (Pinedo, 1995). In practice, setup times often have a special structure. Considering the following structure, two parameters are associated with the job, say a_i and b_i, and $s_{ik} = |a_k - b_i|$. This setup time structure can be described as follows: after the completion of job i the resource is left in state b_i, and to start job k the resource has to be brought into state a_k. The total setup time necessary for bringing the machine from state bi to a_k is proportional to the absolute difference between the two states (Pinedo, 1995). This state variable could be, for example, a measure of some setting of the machine, which in our case corresponds to the total setup time of job i on resource m, where $sp_{i,m} = sf_{i,m} + s_{ik}$. Certain jobs can have similar setups so that changing from one to another is simply a matter of adjusting stops and perhaps changing tools. Other jobs on the same machine could require an entirely different setup.

The use of sequence-dependent setup for single machine scheduling problems may be considered inadequate, but since the proposed system has a heterarchical nature (i.e. it does not consider a global objective), the machines that compose the shop can be considered independent. In this paper it is shown that the global makespan may be minimized through the reduction of makespan of each individual machine.

4.2. Grouping by operations

A multi-agent negotiation protocol is necessary for effectively coordinating the interactions between the part agents and the machine agents. In this paper the contract net protocol (Smith, 1980) is extended to support a multi-task, many-to-many negotiation. According to Wong et al. (2005) there have been lots of research efforts aiming to extend the original contract net protocol. Although in the work of Usher (2003) the contract-net based negotiation mechanism coordinates multiple job agents concurrently, each job agent can only announce a single task (or operation) at a time, announcing one task to multiple contractors, limiting to a single task announcement in each round of bidding. Figure 1 presents an example of the traditional negotiation mechanism in comparison with the mechanism based on the grouping by operations utilized in this work.

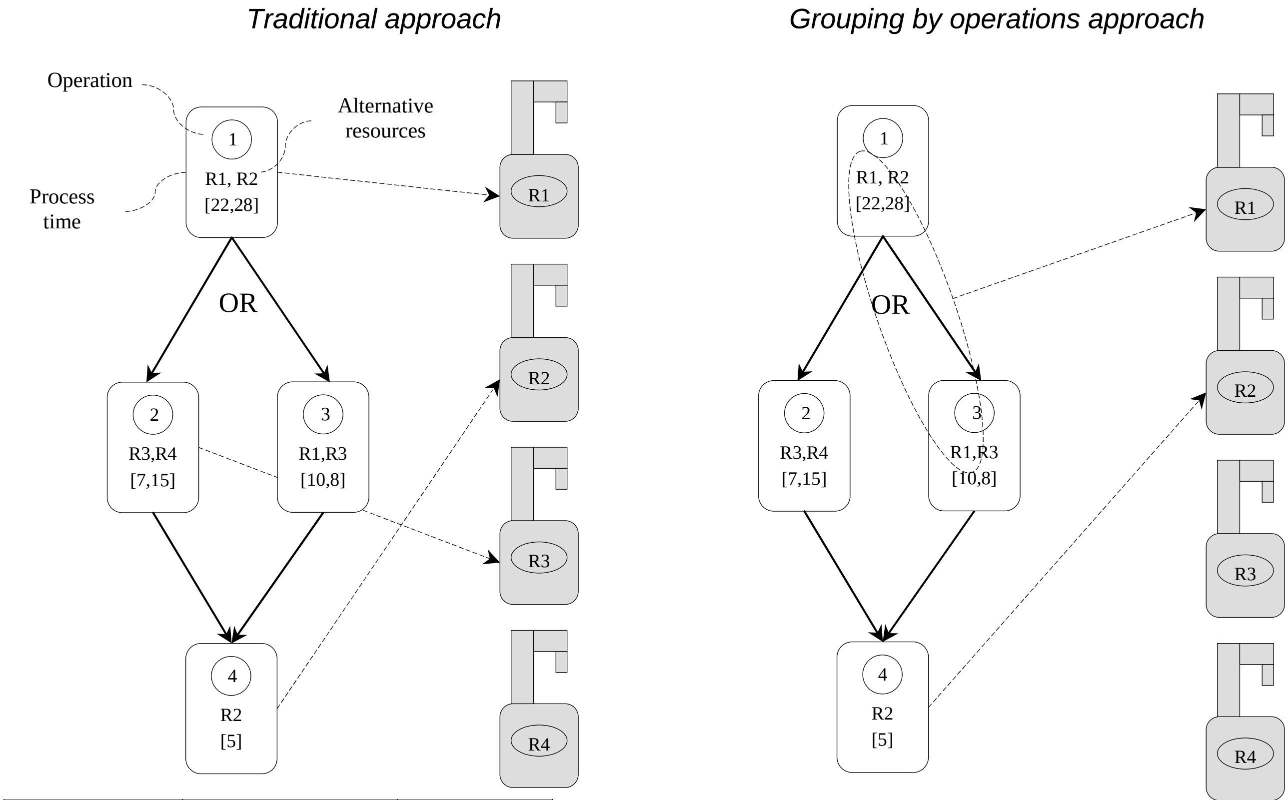
In the grouping by operations approach, whenever a new job agent is instantiated, all the operations that compose its process plan are evaluated simultaneously by all the resources available at the shop floor. Each resource agent analyses all the available process plans for that job, considering only the operations that it is capable to carry out. After this step, the resource agent performs the sum of the processing times of the operations for the job ($p_{i,m}$) and the sum of the total setup time ($s_{i,m}$). It is important to note that these sums will consider only the operations that can be executed by the resource, take into account the precedence relations between them. The obtained values, $p_{i,m}$ and $s_{i,m}$, will be used by the resource agent for elaborating the proposal time, as described in section 4.3.

In the traditional approach, where the times in the process plan correspond to the sum of the total processing and setup times of the part, a “short-sightedness” occurs when visualizing the process plan. For instance, considering that the criterion for choosing each resource is the shortest processing time of each operation, and that the batch size equals one, the chosen resources will be R1, R2 and R4. Thus, the total time will be:

$$(22 + 7 + 5) + (12 + 6 + 21) + (1 \text{ x } 1 + 1 \text{ x } 0.5 + 1 \text{ x } 0.3) = 74.8$$

On the other hand, by using the grouping by operations approach, and considering the makespan with sequence-dependent setup times, as shown in section 3.2, the chosen resources will be R1, R3 and R4, and the total time will be:

$$(22 + 10 + 5) + (12 + 21) + (1 \text{ x } 1 + 1 \text{ x } 0.3) = 71.3$$



	Total machine preparation time (spi,,m)	Fixture time (sfi,,m)
Resource 1	12	1
Resource 2	6	0.5
Resource 3	9	0.7
Resource 4	21	0.3

Figure 1. Traditional approach x grouping by operations approach

The time improvement occurs due to the reduction in the amount of machine changes, which decreases the number of necessary setups.

4.3. Formulation of proposal time

The proposal time is the sum of all the times considered by a resource agent for the elaboration of a proposal in response to a request made by a part agent. This proposal time indicates the time predicted to start manufacturing the job on the resource. Equations (1) to (4) represent the times that compose the proposal time:

$$\text{Proposal\_time} = \quad Tq_m + \left(\frac{\sum_{j=1}^{g_i} p_{i,m} + s_{i,m}}{g_i} \right) \tag{1}$$

$$Tq_m = \sum_{i=1}^{n} (q_i + c_i + w_i) \tag{2}$$

$$s_{i,m} = sp_{i,m} + sf_{i,m} * batch\_size \tag{3}$$

$$\text{Proposal\_time} = \quad \sum_{i=1}^{n} (q_i + c_i + w_i) + \left(\frac{\sum_{j=1}^{g_i} p_{i,j} + sp_{i,m} + sf_{j,m} * batch\_size}{g_i} \right) \tag{4}$$

where:

- Tq_m: queue time to carry out all manufacturable jobs on a resource. These jobs are already in the resource processing queue, but they have not yet started manufacturing at the instant of negotiation. If there are no jobs in the resource processing queue at the negotiation instant, then $Tq_m = 0$;
- q_i: resource queue time of job i;
- c_i: contract time. These orders have already contracted a resource, but have not yet arrived at the resource processing queue (for instance, they are still being manufactured at a previous resource);
- w_i: waiting time. It is the interval between the sending of the proposal for job execution by a resource agent, and the acceptance of the proposal by the part agent that is negotiating with the resource. If no jobs are in the waiting interval at the negotiation instant, then $w_i = 0$.

For a better understanding of the contract time (c_i), waiting time (w_i), and on resource queue time (q_i), which compose eq.(2-4), Fig. 2 presents an example illustrating the exchange of messages between three job agents *i1*, *i2* and *i3*, and three resource agents *R1*, *R2* and *Rn*. Each of the resource agents has an internal counter of the total queue time, *Tq1*, *Tq2* and *Tqn*, responsible for adding the total queue time (*Tqm*) that will later be used in the proposal.

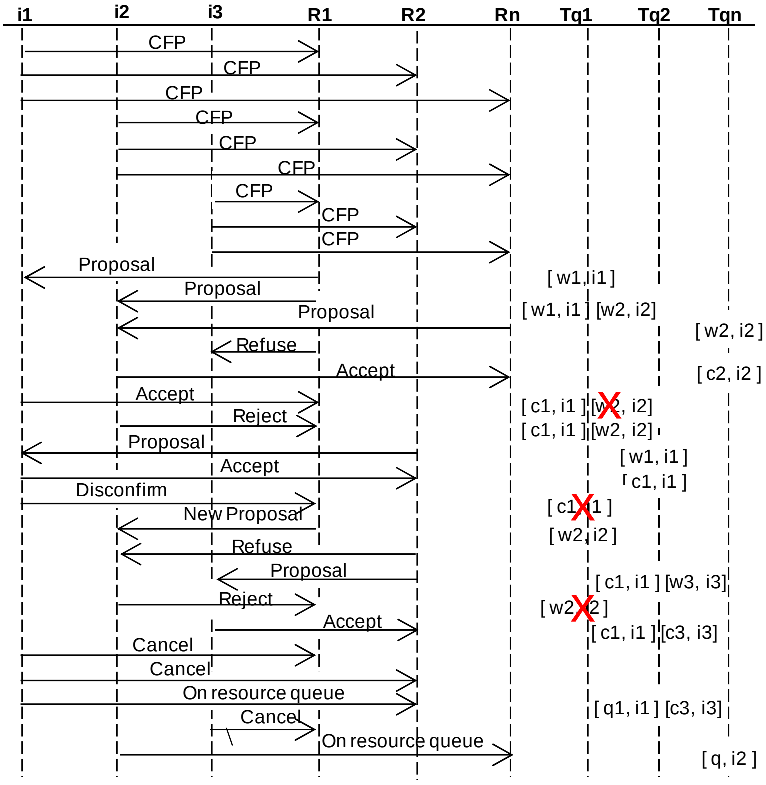


Figure 2. Negotiation between job agents and resource agents

The negotiation starts with job agents *i1*, *i2* and *i3*, which send a call for proposal (CFP), requesting a proposal time to resource agents *R1*, *R2* and *Rn*. As soon as resource *R1* sends a proposal to *i1*, the counting of the waiting time starts (w_i). This time is calculated in column *Tq1* until resource *R1* receives an accept or reject by *i1* referring to its proposal. In the case of a positive response (accept) by *i1*, the time will not be calculated as waiting time (w_i), and instead it will be considered as contract time ($c1$), remaining that way until job i1 is moved to the resource processing queue.

At this instant job *i1* sends to resource *R1* the message informing that it arrived at the resource queue. When resource *R1* receives this message, it considers the time related to *i1* as a portion of the resource queue time ($q1$).

If the resource proposal is rejected, as it occurs in the negotiation between *i2* and *R1*, where *R1* receives a "reject" of a proposal made to job *i2*, the waiting time ($w2$) is not considered as part of the resource negotiation time, and it is discarded.

4.4. Mechanism to compare the proposals

In order to characterize the dynamic scheduling environment, a mechanism that allows the renegotiation between resource agents and part agents even after a "reject" message by the part agent was created. This mechanism is triggered whenever an alteration occurs in the queue of jobs of the resource agent involved in the negotiation. The steps that compose this renegotiation are show in Fig. 3 and comprise the follow.

(a) When sending a “reject” proposal, the part agent also needs to send the best proposal that it received until that moment of negotiation, i.e., the proposal that motivated its refusal;

(b) This proposal, which includes the information about the part that originated it, is stored temporarily by the resource agent;

(c) If an alteration occurs in the queue of jobs of the resource agent, such as an order cancellation, this resource will calculate new queue times, updating the proposal time of all of the jobs that are in its physical queue or negotiation queue;

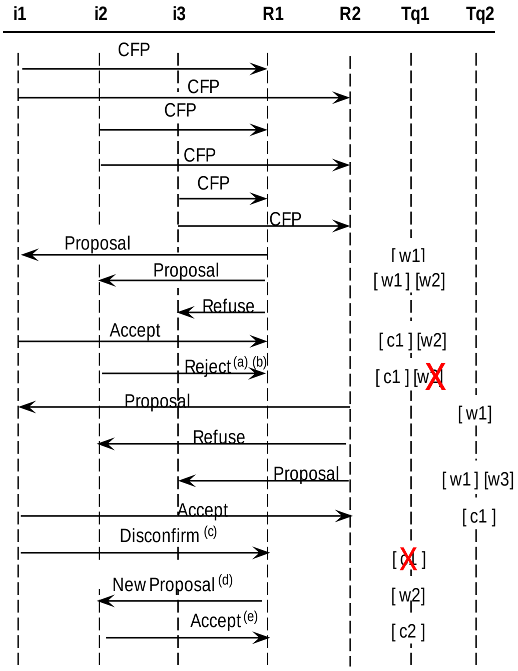


Figure 3. Mechanism to compare the proposals

(d) After calculating all the new times, the resource agent will analyze the proposals stored in item (b), comparing them with its new availability. In case the new proposal is better than the stored proposal, the new proposal is sent again to the part agent;

(e) Finally the part agent will analyze this new proposal, verifying if it will accept it or not.

5. CHARACTERISTICS OF THE MULTI AGENT SYSTEM

The system was developed in the Java language, according to standardization of FIPA, using the platform for the development of agents JADE (Jade, 2005), the development environment ECLIPSE, and the MySQL database. The program executes in the operating system Linux.

5.1. Agents that compose the system

The proposed model is composed by five different types of agents: Part Agent (PA), Resource Agent (RA), user Interface Agent (InA), Synchronizer Agent (SyA) e Server Agent (SvA). For the agent encapsulation two different approaches were used: functional decomposition (part agent, input order agent, synchronizer agent e server agent) e physical decomposition (resource agent). In the functional decomposition approach, software agents are used to encapsulate modules detailed to functions, such as order acquisition, process planning or scheduling. There are no explicit relationships between software agents and physical entities. In the physical decomposition approach, software agents are used to represent entities in the physical world, such as machines, parts, features (Shen et al., 2006). The agents that composed the system are described below:

- Part Agent (PA): it represents each one of the orders for the manufacture of a certain part with a programmed batch size. PA has the following functions: to search the database for information on the process plan of the part to be manufactured, and to determine dynamically the routing of its manufacture through negotiation with the RAs.

- Resource Agent (RA): it represents each physical resource on the shop floor. Starting with the database, this agent obtains information on the capability of the resource it represents, as well as its processing times. Their main functions are: to control the resource queue; to supervise its functioning, identifying the events that occur on it; and to accomplish the negotiation with the PA, elaborating proposals for the manufacture of the orders.
- User Interface Agent (InA): it is responsible for the interface with the system user. After being instantiated, this agent communicates with the server, and it retrieves the dates related to the parts and their respective process plans, allowing the user to elaborate its manufacturing order. An order can be composed of one or more parts and their respective quantities.
- Server Agent (SvA): its purpose is to implement a series of basic functionalities in the system, which include: to communicate with the InA, supplying information of the database; instantiate the PAs according to the request sent by the InA; instantiate with the Syncronizer Agent; to maintain a list of the orders under execution on the shop floor; and to send e-mail(s) to the users informing the output data about the manufacture of the part.
- Syncronizer Agent (SyA): its main objective is to characterize that the scheduling of the production orders does not obey any queue criterion or priority. Thus, the negotiation between PAs and RAs should happen, in theory, in a parallel and simultaneous way. In practice, if the system is running in a network, two messages cannot occupy the communication means at the same time. Besides, if the agents are running locally, their processing is scheduled by the operating system. In order to avoid those problems, an alternative was used to guarantee that the initialization of the negotiation happens in a random way, without benefitting any PA, with the use of the Synchronizer Agent (SyA). Whenever the PAs are instantiated by the SvA, the PAs wait for the synchronization message so that their life cycle. The SyA is then instantiated by the SvA. This instantiation is performed whenever the user sends a request at the beginning of simulation through the button Sync in the interface of InA. As soon as it is instanced, SyA communicates with the Agent Management System – AMS, in order to obtain a list of the agents present in the system, and then it sends a synchronization message to each one of them. Its implementation is relatively simple, as it has only one behavior. The agent remains blocked until the arrival of some message. That message is then interpreted, and an action is taken, in order to achieve the objectives listed above.

In order to facilitate the interaction with the user, after the manufacture of the order the user may visualize (through a Log Viewer) the flow times of the jobs in the shop. Whenever the software is executed, a Gantt chart is generated automatically, allowing the visualization of the jobs executed by each resource, with or without the display of their queues. The Log Viewer accesses the data created by the Part Agents for the construction of the Gantt chart. When clicking on a job in the graph, it is possible to obtain complementary information, such as the sequence of operations performed at that resource, the processing time, the queue time, etc. Figure 4 shows an order with 18 parts distributed in 15 resources. It also shows in details the queues in resources R8, R9, R10, R11 and R12.

6. IMPLEMENTATION AND EXPERIMENTS

In this paper, the performance of the proposed operation-based time-extended negotiation protocol is compared with the following approach: a symbiotic evolutionary algorithm (SEA) developed by Kim et al (2003a). They generated 18 parts with various combinations of flexibility levels. Each job consists of a minimum of 8 and a maximum of 22 operations. They constructed 24 test-bed problems with the 18 jobs. The number of jobs, the number of operations, and the job composition involved in each problem are listed in Tab. 1.

Table 1. Test-bed problems

Problem	Number of jobs	Job Number	Problem	Number of jobs	Job Number
1	6	1, 2, 3, 10, 11, 12	13	9	2, 3, 6, 9, 11, 12, 15, 17, 18
2	6	4, 5, 6, 13, 14, 15	14	9	1, 2, 4, 7, 8, 12, 15, 17, 18
3	6	7, 8, 9, 16, 17, 18	15	9	3, 5, 6, 9, 10, 11, 13, 14, 16
4	6	1, 4, 7, 10, 13, 16	16	12	1, 2, 3, 4, 5, 6, 10, 11, 12, 13, 14, 15
5	6	2, 5, 8, 11, 14, 17	17	12	4, 5, 6, 7, 8, 9, 13, 14, 15, 16, 17, 18
6	6	3, 6, 9, 12, 15, 18	18	12	1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17
7	6	1, 4, 8, 12, 15, 17	19	12	2, 3, 5, 6, 8, 9, 11, 12, 14, 15, 17, 18
8	6	2, 6, 7, 10, 14, 18	20	12	1, 2, 4, 6, 7, 8, 10, 12, 14, 15, 17, 18
9	6	3, 5, 9, 11, 13, 16	21	12	2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 16, 18
10	9	1, 2, 3, 5, 6, 10, 11, 12, 15	22	15	2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18
11	9	4, 7, 8, 9, 13, 14, 16, 17, 18	23	15	1, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18
12	9	1, 4, 5, 7, 8, 10, 13, 14, 16	24	18	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18

The complete set of data for all 24 test-bed problems and 18 parts, including the alternative process plans and the related data, is available in Kim et al (2003b) and not repeated here.

SEA is a co-evolutionary algorithm that can simultaneously deal with process planning and job shop scheduling in a flexible manufacturing environment. This approach is characterized by its ability to perform the effective and simultaneous search of the solution space formed by the two problems. In order to evaluate the performance of operation-based time-extended negotiation protocol, a number of experiments is conducted based on the test-bed problems provided in Kim et al (2003a).

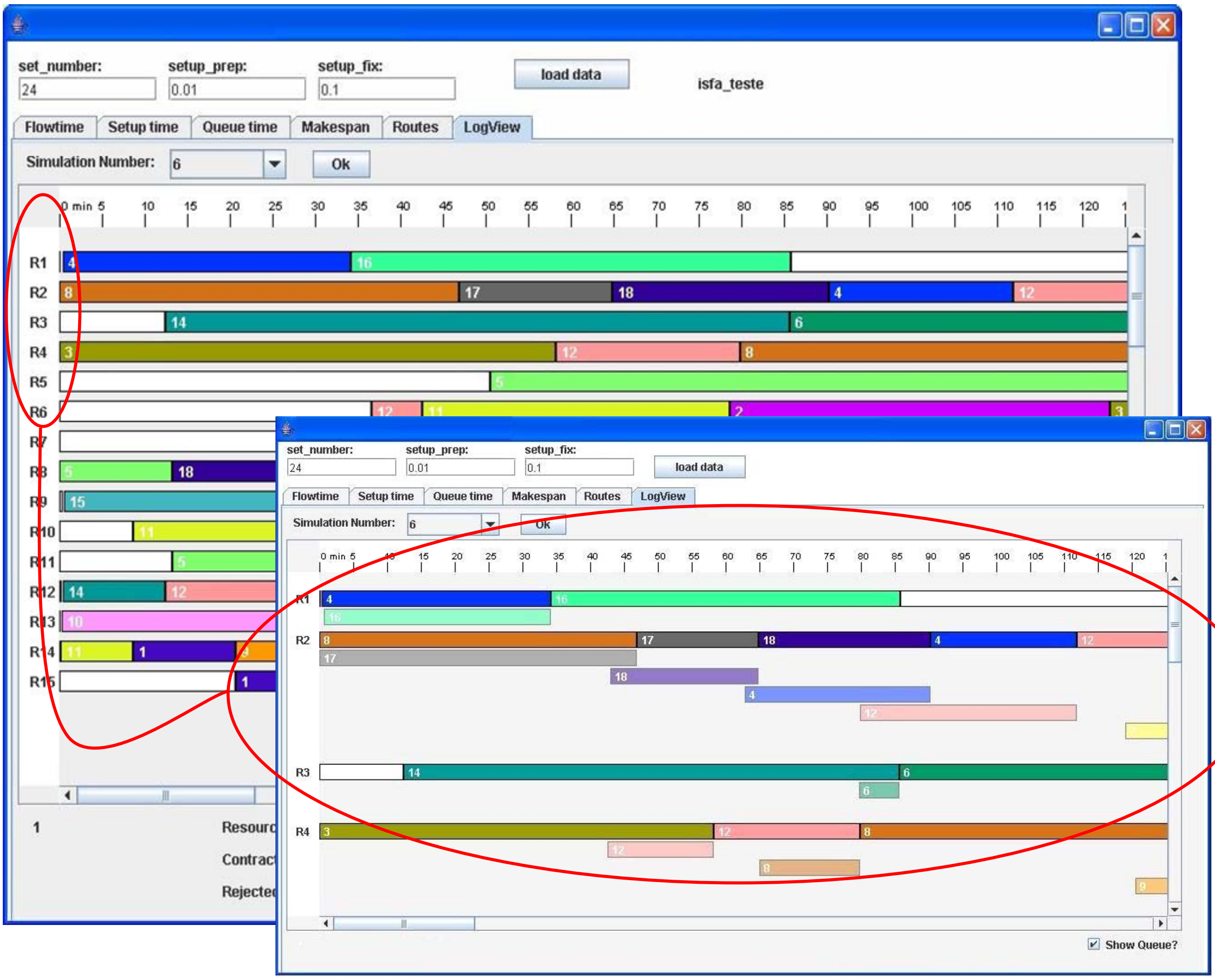

set_number: 24
setup_prep: 0.01
setup_fix: 0.1
isfa_teste
Simulation Number: 6

Figure 4. Gantt chart – it allows the visualization with and without the queues on the resources

6.1. Performance Comparison

The performance of the proposed operation-based time-extended negotiation protocol is compared with those three algorithms mentioned above. Table 2 shows the experimental results for mean makespan and flow time, respectively. The experiment for each problem is repeated 10 times for every test-bed problem. The average of this is reported in the tables. Since the test-bed proposed by Kim et al (2003b) does nor consider the setup times of the operations, it is necessary to carry out two performance comparisons in order to better characterize the nature o this investigation. Column Setup_0 in table 2 refers to the same conditions presented by Kim, i.e., the setup time is not considered. On the other hand, in column Setup_10_10, the total operation time used by Kim was divided in three parts: 80% processing time in the machine (pi,m); 10% machine setup (spi,m); 10% fixturing setup (sfj,m).

Table 3 reveals that, for several test-bed problems, the proposed operation-based time-extended negotiation protocol provides the best makespan performance among the compared algorithms. The global average obtained is also better than those generated by the other SEA algorithm used in the comparison. The cases in which the results for the makespan are worse than those attained by the SEA algorithm will be investigated in greater detail in the future, since in a preliminary analysis no dominant characteristic was found that could lead to a worse result. With regard to the flow time for all the given examples, the proposed operation-based time-extended negotiation protocol provides the best performance among the compared algorithms.

Table 2. Comparison of overall makespan

Problem	**SEA**		**Setup_0**		**Improved**	**Setup_10_10**		**Improved**
	Mean	s.d	Mean	s.d	**rate (%)**	Mean	s.d	**rate (%)**
1	437.6	10.9	458.28	10.16	-4.7	432.7	13.2	1.1
2	349.7	5.9	334.88	10.60	4.2	331.0	10.1	5.4
3	355.2	7.4	329.89	6.15	7.1	318.4	11.7	10.4
4	306.2	0.4	301.63	8.83	1.5	297.7	7.5	2.8
5	323.7	3.6	306.22	14.81	5.4	320.6	15.6	1.0
6	443.8	5.0	455.88	11.35	-2.7	434.5	11.3	2.1
7	372.4	1.3	348.54	8.82	6.4	352.8	18.4	5.3
8	348.3	5.7	337.68	8.01	3.0	327.3	10.0	6.0
9	434.9	9.8	463.47	9.01	-6.6	446.1	4.2	-2.6
10	456.5	10.8	467.42	13.43	-2.4	434.6	8.5	4.8
11	378.9	5.1	349.38	13.99	7.8	344.7	10.4	9.0
12	332.8	3.4	340.01	11.46	-2.2	336.1	29.0	-1.0
13	469.0	10.7	456.23	15.21	2.7	436.9	13.2	6.8
14	402.4	10.6	367.40	14.83	8.7	369.9	21.2	8.1
15	445.2	11.0	471.32	12.71	-5.9	457.3	8.2	-2.7
16	478.8	12.0	471.35	34.06	1.6	461.9	27.0	3.5
17	448.9	8.7	387.86	28.46	13.6	403.6	36.9	10.1
18	389.6	7.5	375.47	9.18	3.6	383.5	23.6	1.6
19	508.1	10.0	480.06	18.75	5.5	444.5	14.1	12.5
20	453.8	5.2	404.08	14.12	11.0	394.3	18.5	13.1
21	483.2	6.8	482.09	17.50	0.2	457.1	13.6	5.4
22	548.3	6.9	504.31	36.35	8.0	474.5	12.3	13.5
23	507.5	8.3	458.28	33.74	9.7	434.0	21.9	14.5
24	602.2	7.1	529.58	27.35	12.1	495.9	26.4	17.6
				Mean Improved rate (%) =	*3.7*		*Mean Improved rate (%) =*	*6.2*

Table 3. Comparison of overall flow time

Problem	**SEA**		**Setup_0**		**Improved**	**Setup_10_10**		**Improved**
	Mean	s.d	Mean	s.d	**rate (%)**	Mean	s.d	**rate (%)**
1	318.9	3.7	302.39	5.97	5.2	300.5	5.6	5.8
2	287.7	4.7	273.23	2.93	5.0	266.0	3.2	7.5
3	304.8	4.3	285.54	5.71	6.3	278.9	5.3	8.5
4	251.3	4.8	246.84	1.84	1.8	245.4	4.4	2.4
5	280.3	3.2	259.31	5.58	7.5	261.5	6.1	6.7
6	384.7	5.7	353.02	6.55	8.2	343.5	4.8	10.7
7	314.1	2.6	297.37	4.21	5.3	291.6	7.6	7.2
8	295.2	5.0	281.22	5.13	4.7	278.3	7.1	5.7
9	298.9	7.0	286.58	2.75	4.1	282.8	7.4	5.4
10	349.2	6.1	313.20	5.07	10.3	309.1	6.9	11.5
11	312.9	7.6	288.18	5.47	7.9	285.2	5.1	8.8
12	279.6	4.7	267.84	4.42	4.2	261.9	6.1	6.3
13	387.0	7.1	335.63	5.50	13.3	324.8	4.9	16.1
14	346.9	8.5	317.62	5.63	8.4	317.9	4.8	8.3
15	316.1	6.2	292.66	3.54	7.4	286.7	7.1	9.3
16	359.7	4.3	318.60	7.42	11.4	316.8	5.0	11.9
17	364.7	4.7	313.46	4.99	14.1	306.2	4.7	16.0
18	322.5	6.4	286.44	9.15	11.2	284.8	7.8	11.7
19	406.4	4.6	336.41	6.41	17.2	339.4	5.5	16.5
20	372.0	5.7	324.78	4.79	12.7	323.1	8.5	13.1
21	365.4	8.2	323.23	6.47	11.5	305.2	7.4	16.5
22	417.8	5.8	360.68	9.48	13.7	352.2	11.9	15.7
23	404.7	5.1	347.28	12.86	14.2	334.4	11.5	17.4
24	452.9	7.5	391.77	12.54	13.5	387.4	12.7	14.5
				Mean Improved rate (%) =	9.1		*Mean Improved rate (%) =*	10.6

7. CONCLUSION

The system proposed in this paper uses a heterarchical multiagent model that allows the dynamic process planning while reducing makespan and flow time through the reduction of the setup time between the jobs. In order to reach this objective, an operation-based time-extended negotiation protocol was used.

One of the most significant contributions to the efficacy of the proposed operation-based time-extended negotiation protocol is the use of flexible process plans that can be verified step by step during the sequencing and routing of jobs, which allows the resources group the operations that they are capable of manufacturing, reducing the machine setup time. This grouping allows the reduction of both the makespan and the flow time, and this is due to the reduction in the number of machine changes on which the jobs are manufactured. This shows that the simplification of the scheduling problem in a job shop layout through the inclusion of setup times in the total processing time of the machines may result in an incorrect analysis of the problem.

As a future work, an analysis of the influence of the setup times in the reduction of makespan and flow time will be carried out. This analysis will be based on the gradual increase of the contribution of the machine setup time. A mechanism will also be created for the analysis of the relationship among the several flexibilities of process plan in relation to the lot size and setup times.

8. REFERENCES

Benjaafar S, Ramakrishnan R., 1996, "Modelling, measurement and evaluation of sequencing flexibility in manufacturing systems", International Journal of Production Research, Vol. 3, pp. 1195–220.

Conway R. W., Maxwell W. L., Miller L. W., 1967, "Theory of scheduling", Dover Publications, 294 p.

http:// www.fipa.org/.

Jade Administrator's Guide, 2005. 18 September 2006 < http://jade.tilab.com/doc/index.html>.

Jade Programmer's Guide, 2005, 18 September 2006 < http://jade.tilab.com/doc/index.html>.

Kim YK, Park K, Ko J., 2003a, "A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling", Computers & Operations Research, No. 30, pp. 1151-1171.

Kim Y. A set of data for the integration of process planning and job shop scheduling, 2003b. 13 September 2006 <http://syslab.chonnam.ac.kr/links/data-pp&s.doc>.

Lin YJ, Solberg JJ., 1991, "Effectiveness of flexible routing control", The International Journal of Flexible Manufacturing Systems, No. 3, pp. 189–211.

Pinedo M., 1995, "Scheduling, Theory, Algorithms, and Systems", Prentice Hall, New Jersey, 586 p.

Shen W, Wang L, Hao Q., 2006, "Agent-Based Distributed Manufacturing Process Planning and Scheduling: A State-of-Art Survey", IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews, No. 36, pp. 563-571.

Smith R. G., 1980, "The contract net protocol: high level communication and control in a distributed problem solver", IEE Transactions on Computers, No. 29, pp.1104-1113.

Tan W., Khoshnevis B., 2000, "Integration of process planning and scheduling – a review", Journal of Intelligent Manufacturing, Vol.11, No. 1, pp. 51-63.

Usher J., 2003, "Negotiation-based routing in job shops via collaborative agents", Journal of Intelligent Manufacturing No. 14, pp. 485-558.

Wang Y. H., Yin C. W., Zhang Y., 2003, "A multi-agent and distributed ruler based approach to production scheduling of agile manufacturing systems", International Journal of Computer Integrated Manufacturing, Vol.16, No.2, pp. 81-92.

Wong T. N., et al., 2005, "Dynamic shop floor scheduling in multi-agent manufacturing systems", Expert Systems with Applications, pp. 1-9.

Zhou Z. D., et al., 2003, "A multi-agent based agile scheduling model for a virtual manufacturing environment", International Journal of Advanced Manufacturing Technology, Vol. 21, pp. 980-984.

9. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.