An Agent Based Architecture for Job-Shop Scheduling Problem Using the Spirit of Genetic Algorithm

CARDON Alain
LIP6 Paris VI
UPMC, Case 69
4, Place Jussieu
F-75252 Paris VI
Alain.Cardon@lip6.fr

GALINHO Thierry
LIH, Université du Havre
25, Rue Philippe Lebon, BP 1123
F-76063 Le Havre Cedex
galinho@univ-lehavre.fr

VACHER Jean-Philippe
LIH, Université du Havre
25, Rue Philippe Lebon, BP 1123
F-76063 Le Havre Cedex
Jean-Philippe.Vacher@poleia.lip6.fr

Abstract

In this paper, we investigate issues of coordination's agents in a distributed job-shop scheduling problem in which agents schedule potentially contentious activities asynchronously in parallel. Agents, in such a system, will in general have a limited view of the global state of resources and must exchange appropriate state's information with other agents in order to schedule effectively.
However, even given perfect instantaneous knowledge of other agents' resource requirements, agents may still not be able schedule effectively if they do not also model the possible actions of other agents and the effects of their own actions.
By crossovers and mutations, during the optimization of Job-Shop scheduling problems, agents with different granularities appear. Consequently, new tendances and behaviours are present in the Multi-Agent System. These ones take advantage of micro and / or macro heuristics obtained by genetic evolution. Then, new agents can contribute to work more efficiently on schedulings.

1 Introduction

In the Job-Shop Scheduling Problem (JSSP), Gantt diagram's optimization can be considered as a NP-Difficult problem [7]. Determining an optimal solution is almost impossible, but trying to improve the current solution is a way to lead to a better tasks repartition. Multiagent systems are often used for such problems, where a solution exists but is not easily calculable. We expect such systems to discover some solution in an emerging way, it is the reason why we used them. We use them to simulate the behaviour of entities that are going to collaborate to accomplish actions on the Gantt diagram so as to solve a given economic function. The ideal solution of such a problem is a point where each objective function corresponds to the best (minimum) possible value.
We present our job-shop scheduling problem, which is a simplificated version of FISIAS [11]. We present some results based on genetic algorithm. then, we present an agent model based on the contract net protocol to improve a solution corresponding to a Gantt diagram.

2 Job-Shop Scheduling problem

2.1 Definition

Scheduling is an essential function in production management. It is a difficult problem according to the number of calculations required to obtain a scheduling that optimises the chosen criterion [13]. In addition, there are many scheduling problems and various approaching methods have been proposed to solve some part of them. We are going to define our scheduling problem and to describe some existing problems, as well as the constraints that we are considered. Among of various definitions of the scheduling problem, we can highlight a common demoninator: it is the atomical task allocation with a minimum cost and in a reasonable time. We are in the field of discontinuous production, with the processing of small and average series. A scheduling problem exists:

- when a set of works (jobs) is to be processed,
- when this problem can be decomposed in tasks (operations),
- when the problem consists in the definition of the temporal task location and/or the manner to allocate them with the necessary resources.

[7] defines the scheduling problem of a discontinuous production: "The scheduling problem of the production consists in manufacturing at the same time, with the same resources, a set of different products". A scheduling determines what is going to be made, when, where and with what resources; given a set of tasks to accomplish, the scheduling problem consists in determining what operations have to be executed and to give dates and resources for these operations.

2.2 Our Job-Shop Scheduling Problem (JSSP)

Scheduling and planing are difficult problems with a long and varied history in the areas of operational research and artificial intelligence, and they continue to be active research areas. The scheduling problem which is subject to precedence and resources constraints is a NP-Difficult problem. It is then impossible to obtain an optimal solution satisfaying the real time constraint.
So heuristic algorithms are usually implemented to obtain a "good" solution instead of an optimal one [8]. Due to the number of the varieties of production processes and to the increasing rate of change of operational parameters characterizing the data to be processed (capacities of the resources, demands, etc.), it is becoming more and more difficult for management boards to achieve decision making.
The problem that we have chosen is the JSSP with M machines and N jobs because it is the most complex and the most often considered [7]. To determine the quality of the solution, a graphical interface has been developped. For our problem, the goal is the minimisation of delays for all jobs according to the "due dates" given by the manager according to other objectives. The choice of this objective has been made to obtain solutions very rapidly because calculations are very numerous with genetic algorithms. It is also because it is sufficient to compare these algorithms with other methods such as the gradient, the simulated annealing, etc. Genetic algorithm enable us to obtain good quality solution quickly and easily compared to other research techniques [6][4].

Nb. of jobs	Nb. of operations	Nb. of machines	Calculation time
10	10	4	50sec.
50	5	4	9 min. 53 sec.
50	10	4	14 min. 27 sec.
100	10	10	40 min. 58 sec.
500	100	50	2 h. 24 min. 07 sec.

Figure 1: Benchmark's results.

2.3 Description of the Genetic Algorithm Used

The Genetic Algorithm, as for it, simulates the biological evolution process, in the senses of Darwin, of entities composing an organization. That is to say that less effective individuals are going to disappear to the profit of individuals better adapted. It appears therefore an elitist system, style " law of the strongest " that is established. The genetic algorithm used resumes the great lines described in works of D. Goldberg [4] and J. Holland [6]. Nevertheless, a Simple Genetic Algorithm (SGA) makes intervene only one function of performance. Here, we use the criterion of optimization of Pareto to establish our fitness function. We have a multicriterion optimization function. The result is a multi-objectif genetic algorithm.
The ideal solution of such a problem is a point where each objective function corresponds to the best (minimum) possible value. The ideal solution, in most cases, does not exist because of the contradictory nature, rather contradictory objective functions: compromises have to be done. A different concept of optimality has to be introduced. Solving a multiobjective problem generally requires the identification of Pareto optimal solutions. A solution is said Pareto optimal, or non dominated, if starting from that point in the design space, the value of any of the objective functions cannot be improved without deteriorating at least one of the others [3].

2.4 Results From The Genetic Algorithm

From our object modelisation, a genetic algorithm using the placing method has been developped [5]. This program uses the C++ language in order to use it on a SGI Origin 2000. Here are some benchmarks (See fig. 1). As the calculation time depends on the load of the computer, this time doesn't correspond to the CPU time. We can then determine heuristics to use with our dynamic model based on the contract-net protocol [12]. An other result coming from the simulation process is a set of graph giving the tardiness in function of the number of generations. We can see that the delay decreases rapidly. But it is not necesary to increase the number of generations to obtain a better result (Fig. 2).

2.5 Encountered problems

In our problem, the first level of modelisation was a static model. This one doesn't take into account the fact that we have interactions between machines and jobs. A job can be done only if the resource is free. On the other hand, the static model doesn't include the possible interaction between the workshop and the environment such as a strike, a machine failure, etc. In the static model, the placing obtained gives a solution corresponding to the schedule of all jobs. But, by negociations, the schedule can be built step by step.

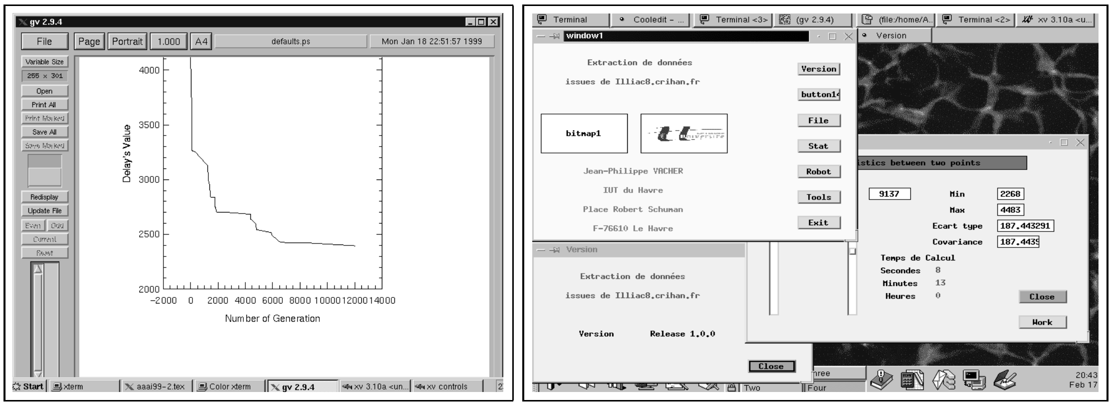

Figure 2: Tardiness' evolution and heuristics extraction programs

For example, if a job arrives too soon, the delay corresponding to the tardiness increase. Therefore, if we can change the schedule during the calculation process, we can improve the tardiness. It is one of our economic functions.

3 Actions on the Gantt Diagram

The JSSP does not admit a computable solution, so the use of multiagent systems for the solving of such a problem seems reasonable. Multiagent systems research is concerned with the behavior of a set of agents that co-operate in order to solve a problem [9]. In Multi-Agent System (MAS), the agents are seen as little problem solvers that co-operatively work in order to solve a problem far beyond their individual abilities. Here, we consider an agent as an entity with goals, actions to accomplish and areas of knowledge, which is situated in its environment. Therefore, because of the knowledge of agents, rules of actions, ..., the MAS will have for principal objective to group agents having similar behaviours to elaborate strategies to the jobs level, jobs of jobs, machines, machines of machines, etc. Indeed, the problem of conflicts between agents is a major concern in MAS research [1] [9]. The objective of the MAS is to improve the Gantt diagram, therefore it leads us to establish the notion of group corresponding to elementary entities having common grinds and physical sameness (same capacity of machine, etc.) or interdependence.

We will use the notion of zone for the roundup of entities on the Gantt diagram while we will speak about the notion of group for the roundup of entities of similar or close nature. Since, agents have to intervene on groups and elementary entities, the MAS will be then composed of micro and meta-agents. It is therefore important, for this evolution, to introduce agents having an evolving character: the meta-agents of evolution. These meta-agents' function will be to make this organization evolving by means of a genetic algorithm establishing a sexued reproduction of agents. It is interesting to note that, traditionally, agents only clone themselves. But here, we use a genetic algorithm for the physical evolution of the agents. It appears, in the course of the evolution, different agents granularities. We have therefore micro and meta-agents that are going to intervene, according to their granularity, on an entity or a group, by passing by intermediate levels. Thus, agents having a meta-knowledge are going be able to intervene on the macro-entities (groups) as well as on some zones of the Gantt diagram. Thus, there is a distributed agent system being able to mutate and hosting agents able to achieve a cross-over based

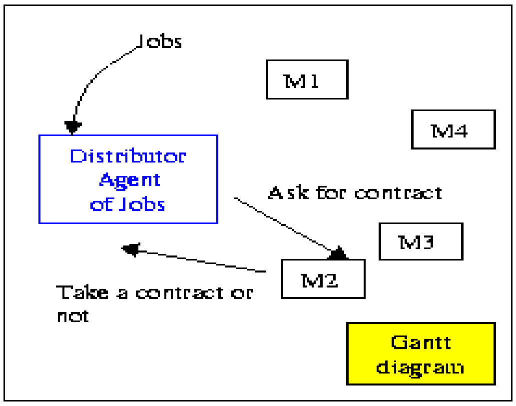

Figure 3: Representation of the universe

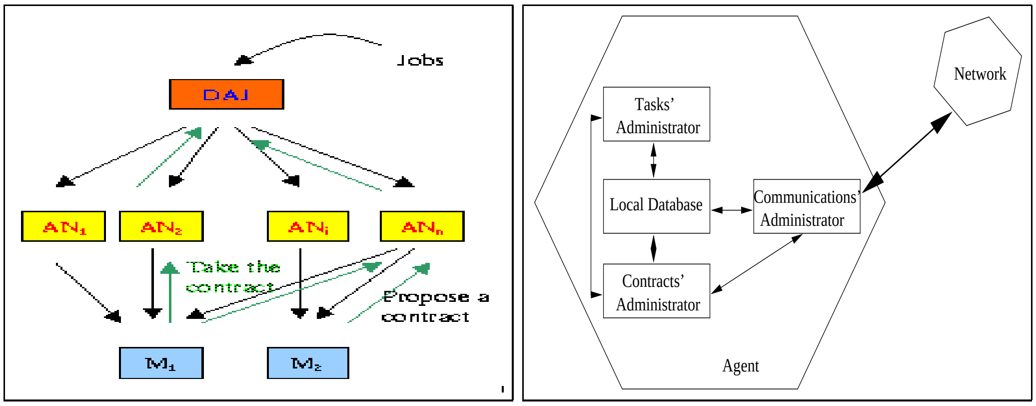

Figure 4: First representation of the Contract Net

reproudction between them.

4 A Contract Net Based Approach

The contract net is a protocol for the resolution of distributed problem, conceived by [12]. The main objective of this protocol being the negociation, it proceeds by allocation of tasks to a set of problem solvers and uses the concept of negotiation to grant contracts. The basis architecture contains nodes having a chief and carrying roles.

4.1 The Contract Net Procotol

Contract net based systems represent a concept that can be used to establish mechanisms of cooperation between agents. A contract net consists of a number of nodes that are represented by individual agents. By analogy to a sale session, suspended sub-tasks are openly proposed to auctions to which each node can reply according to the interest that it has for this sub-task. The stage of task attribution represents a process in which all the nodes (agents) are associated. The idea is to use the available resources and the existent knowledge of agents as efficiently as possible; that is to say by allocation of one sub-task to the agent that is the most apt to operate at a given moment.

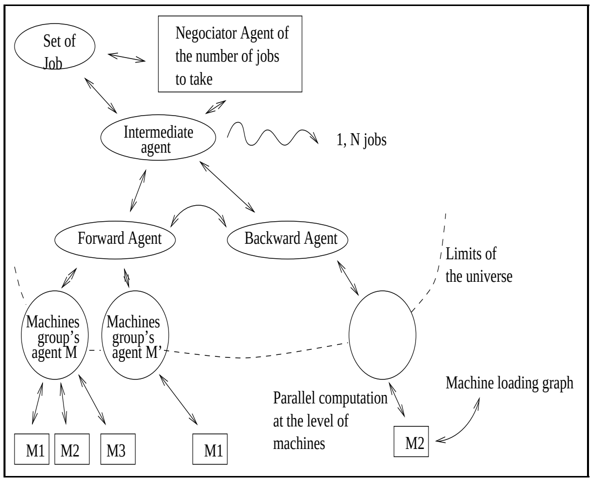

Figure 5: Contract Net structure between the different level representation

The contract net protocol forms the skeleton of our system. It is defined by a language between the node that can be understood by the set of agents. The communication between agents is always based on the notion of an "acceptance message". This specificity defines the exact role of an agent.

4.2 The different types of agents

The local database contains the basis knowledge of the associated nodes and also information on the progression state of the cooperative negotiation as well as the state of the resolution process. The task of the communication' manager is to establish communications with others agents. It is the unique component of a node that is in direct relationship with the system. Especially, it insures the reception and sends messages.
The contracts manager's jobs is to examine the "auctionned" task, the compliance with the contract and its ending. In the other words, the contracts manager provides the coordination of all agents. The tasks administrator is responsible for the progresion state of a process and the results of a task assignation on a given agent. It receives the problem that needs to be solved from the contract manager. It uses the local database in order to find a solution and gives it back to the contract manager.
The job of a contract net based system often begins with a problem division stage. After that, the problem to be solved is being divided in a set of sub-probelms.A special agent, the manager, assigns a tasks to a sub-problem. The manager issues a public offer, called a contract, for each sub-problem to be solved according to the scheme defined by R. Smith [12]. Because of the distributed nature of the problem, we chose an agent based modelisation, a simplificated version of FISIAS [11]. The constituting elements of the environment (the universe) can be the machines (and groups of machines), the jobs (and groups of jobs), the Gantt diagram (an attribute) and distributor agent of jobs (DAJ). Machines can be seen as agents whose task is to perform the job's operation at a given time. However, the machine can also be seen as a purely reactive agent, which, according to the job, can reply: *"I can do it or not"*.

Then, we can use the contract-net defined, by R. Smith [12], whose manager, the DAJ will propose the placement of a job on a machine through the use of a deleguated negociation agent for this job (Negociating Agent). According to the information given by a

machine, the NA will establish a contract between a machine Mi and the DAJ. But the DAJ can always propose a job to several machines in order to achieve some competition between machine agents. However, the DAJ can break this contract at any time in case of incapacity of the agent to comply with the contract.
A machine can accept many contracts corresponding to the same job, but negociated by different NA which have different strategies. Then, the contract ideology is *"the strongest wins"* since a contract can be broken by both ends. However, we can propose two types of agents: a foreward placing and a backward placing agent. Moreover, there are some workshop constraints since we have generic machines, groups of machines and specialized machines. We can consider that we have "group of machines" agents which will propose the different jobs to the machines they represent. At this level, we can operate a parallel computation on the machines and so, on the Gantt diagram.
Nevertheless, for the moment, we consider that the DAJ can only process a job. An intermediate agent in charge of chosing the number of jobs can be proposed.

5 Going Deeply in the Relationships of GA and MAS

The use of GA in MAS is the beginning of what can be an interesting research area. There are clearly two kind of approaches, the first is centralized, in other words, some of the genetic is outside the agent. The function of selection is a good example of such out-of-the-agent feature [13].
However, we believe that if one wants to completely merge the genetic approach and MAS, we must make the agent a completely autonomous genetic entity. By that we mean that not only the genetic patrimony has to be "onboard" but also the functions of selection and crossing. An agent must choose which other agent it wants to reproduce with [13]. The location of the function of mutation is not clearly known since it is caused by the possible exposure to external events coming from the environment and during the genetic code replication phase. If we also introduce the notion of motivated behavior for agents [2] we go deeply in the artificial life problematics. The genetic autonomy and the notion of motivation for an agent may lead to a drastically new kind of emergence phenomenon (different social behavior, auto-referring evaluation process, ...) in self-organizing multi-agent systems. It is certainly a difficult task but it may set the seeds of a prolific approach concerning artificial life.

6 Conclusion

Determining an optimal solution is almost impossible, but trying to improve an existent solution is the way to lead to a tasks repartition which is better. During the simulation process, agents granularity appears with the mutation behaviour introduced by GA [10]. At the end of the simulation, communications between global and local agents, due to their actions, lead to the appearance of agents of intermediate granularity and the global optimization in production scheduling. This communication reflects the genetic integration in a multiagent system.
The distributed vision of a scheduling problem of Job-Shop allows to end to a more realistic vision because each agent has a vision of its environment. That is to say that they are capable to react to a punctual problem just as a foreman in its workshop. They are going to react immediately following the breakdown of a machine. Similarly, it is going to take decisions, as agents, that are going to have a repercussion on the environment. Consequently, the representation by a multi-agent system using a contract net is a close

vision of the reality that can be easily transcribed in an industrial organization. The development corresponding to our original dynamic approach is in progress.

References

[1] A.H. Bond and L. Gasser. Readings in distributed artificial intelligence. 1988.

[2] A. Cardon and F. Lesage. Toward adaptive information systems: considering concern and intentionality. In *Proceedings of KAW98*, 1998.

[3] V. Chankong et Y. Y. Haines. *Multiobjective decision making: theory and methodology*, volume 8 of *North Holland series in system science and engineering*. North Holland, New York, 1983.

[4] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[5] T. Haynes. On–line adaptation of search via knowledge reuse. In John R. Koza, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max Garzon, Hitoshi Iba, and Rick L. Riolo, editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 156–161, 1997.

[6] J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[7] P. Chretienne J. Carlier. *Problèmes d'ordonnancement modélisation / complexité / algorithmes*. Ed. Masson, 1988.

[8] J. Kingdon J. Stender, E. Hillebrand, editor. *Genetic Algorithms in Optimisation, Simulation and Modelling*, volume 23 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, Amsterdam, 1994.

[9] N.R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. 1998.

[10] Z. Michalewicz. *Genetic algorithms + data structures = evolution programs*. Springer, Berlin, 1992.

[11] JP. Pécuchet. Fisias, fast interactive system for intelligent automated scheduling. *Revue Automatique et Productique Appliquées, Hermes*, 2(4):23–38, 1989.

[12] Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, December 1980.

[13] JP. Vacher, A. Cardon, and T. Galinho. Genetic algorithms in a multi-agent system. In *Proceeding of the EUROGEN'97 Conference*, 1997. Web site : http://eurogen.cira.it.