Genetic Algorithms in a Multi-Agent System

VACHER Jean-Philippe GALINHO Thierry LESAGE Franck CARDON Alain

PSI-LIRINSA
Insa de Rouen
Place Emile Blondel
F-76130 Mont-Saint-Aignan
Jean-Philippe.Vacher@insa-rouen.fr
Thierry.Galinho@insa-rouen.fr
Franck.Lesage@insa-rouen.fr

LIP6, UMR 7606
Paris VI, UPMC
4, Place Jussieu
F-75252 Paris VI
Alain.Cardon@lip6.fr

Abstract

Determining an optimal solution is almost impossible but trying to improve an existing solution is a way to lead to a better scheduling. We use a Multi-Agent System guided by a Multi-Objective Genetic Algorithm to find a balance point in the respect of a solution of the Pareto front. Of course, this solution isn't the best but allows a multi-criteria optimization. By crossover and mutation of agents, according to their fitness function, we improve an existing solution. Therefore, the construction of some system simulating living organisms or social systems, cannot be modelled using a strictly mechanical approach. They are typically adaptive and their behaviour is not regular. The multi-agent system must express radical characters, such as reification of emergence, property of controlled self-reproduction of groups of agents and not linear behaviour.

1. Introduction

Multi-agents systems (MAS), composed of agents having objectives to reach, may have for property the notion of agent creation. Nevertheless, this notion is simply based on cloning. Therefore, by supposing that this agent has relevant actions, its clone will provide only similar ones. So, it is possible that agents are relevant but yet less than their genitors. Thus, it is possible for agents that provided or done good actions could, by crossing, give birth to individuals whose characteristics would be superior [11,14].

However, by crossing, we do not forbid the generation of off-springs from several individuals and not only from two. In a general manner, agents are distributed entities, that communicate between them, to solve a problem in a co-operative manner. So, the use of the notion of evolution, by introducing evolutionary algorithms in order to simulate a Darwinian process, we think we will have the possibility of producing more efficient individuals.

Consequently, we plan to focus on the possible relationships between MAS and GA in order to define a new property of agents, and more generally, of MAS: **the notion of sexued reproduction**. However, it is necessary for us to define, as for the Genetic Algorithms, a function of selection, mutation and eventually a function of crossing.

2. The selection function

We can not use the classic selection function like the method of the roulette wheel, for it is proportional. Indeed, we would take an agent as a specific entity but without taking into account some exterior pressures on the agent: its environment and the system's emergence phenomena. Therefore, the selection function does not have only to consider actions of the agent ; these actions being good or bad, but communications between agents. We talk about senses (semantic or link) (Fig. 1).

By senses, we mean:

- The sense of communications with the other agents (links network)
- The semantics of the communication between two individuals.

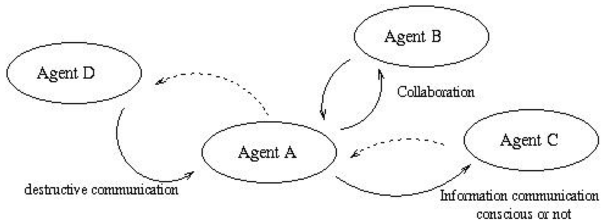

Figure 1. Agents in interaction with others agents

Therefore, our crossover process has to take into account the semantics of communication. But an agent, seen as a structure, is a compound entity made of the following elements: functions of communication, functions of action, functions of behaviour along with a local genetic patrimony.

Just as evolutionary algorithms simulate a Darwinian process, MAS can simulate the evolution of a nucleus or a group and by extension of an organization. Therefore, to define as unique, the cloning of the surviving of a kind is not logical. A social organization may not diversify and evolve by cloning: in all social organizations (human or animal), we have a crossover process that tends to preserve the natural inheritance but also to make it more powerful.

Therefore, our multi-agent system, by integrating this new concept of reproduction with crossing, will have to take into account these parameters. To achieve this, we can use a genetic algorithm switchboard, as defined by John HOLLAND [8,9] or David GOLDBERG [6] or an evolutionary strategy as defined by Thomas BÄCK [1] and Hans-Paul SCHEWFEL. Doing that, each agent (individual) will be characterized by a chain of bits whose length will correspond to a multiple of the number of parameters. This chain will correspond to a chromosome (Fig. 2) that will represent the structure of the agent.

$$\mathrm{STRUCTURE}_{\mathrm{agent\_1}} = (A1, \ldots, An, C1, \ldots, Cm, B1, \ldots, Bp)$$

Figure 2. Structure of an agent

Each character composing the agent will correspond to a numerical data making reference to a basis of rules.

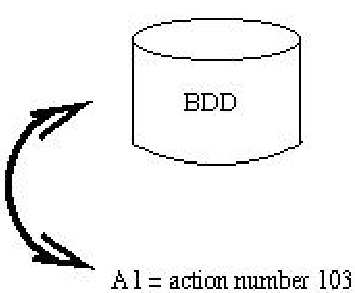

Figure 3. Data base for the knowledge of the agent

A_i, B_j, C_k will make reference to a base of address (Fig. 3). So the knowledge is a "infinite dimension", due to the fact that an agent only has a limited knowledge of its environment, the former only has, a priori, a finite knowledge. By finite knowledge, we suppose that it has a finished number of actions or knowledge available. Especially, at the level of rules of action, if one takes the set of placement rules described by Panwalkar [15], we have at most *n* rules, therefore by using assignment techniques commonly used in electronic and especially in the assignment memory, we can reserve a certain number of address corresponding to rules.

Therefore, for a binary rule coding (Fig. 5), we can use a coding on 10 bits, this manner, it is always possible to increase the knowledge to the level of our database.

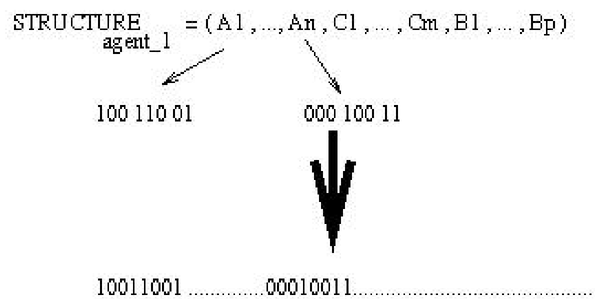

Figure 4. Binary representation of the structure of an agent

Nevertheless, the size of our chromosome is important, in order to reduce the place in memory, we plan to use the coding of Gray [19]. Thus, we can use genetic algorithms on MAS.

$$\begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{matrix}$$

The Gray matrix that can be used to code genes.

$$\begin{matrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{matrix}$$

The Gray matrix that can be used to decode genes.

3. The mutation function

The mutation will correspond to the change of a bit, thus, we can use switchboard operators.

$$1 \rightarrow 0 \text{ and } 0 \rightarrow 1$$

Our constraint, at the mutation level, corresponds to have a correspondence between the bits string and the database. Thus, by changing the value of one bit, we can introduce a new character. This will have a repercussion on the environment, but especially on its membership to a group. The communications it has been able to have with other elements of the group will be, incontestably, changed. For example, consider that the mutation introduces a certain aggressiveness at the agent level, then communications with the group are going to change and the group, consequently, will probably loose some of its social cohesion. Therefore, in order to avoid the too abrupt upset of the social balance that can exist between individuals composing a group and the organization itself, the mutation interventions by genetic algorithms will need to be weak. Nevertheless, we can consider that at the beginning, the simulation of the organization, as at the beginning of a civilization, progress was rapid enough. Therefore, at the beginning, we can introduce an important number of mutations. We will use as distribution, for the number of mutation by generation, a curve of parameters (α, β).

$$f : x \rightarrow \frac{\beta}{x^{\alpha}}$$

$$\beta \in R^{+} \text{ and } \alpha \in R^{+*}$$

Thus, by using this type of distribution, we introduce a lot if mutations at the beginning of the simulation and few at the end in order to avoid the breaking of the process of evolution by deeply modifying characteristics of chromosomes, therefore of individuals.

Too many mutation in the systems would inexorably set the seeds of chaos. We have previously seen a possible distribution. Nevertheless, by using a Gaussian distribution to determine the probability of mutation, we preserve the switchboard concerning the Genetic Algorithm.

4. The crossover operator

From an historical point of view, Genetic Algorithms [10] correspond to a random phenomenon, but the great difference compared to a classic random method is that here, we converge, step by step to an optimum (local or global) in the space of solutions [20]. Thus, we are not subject to hazard as we are in the former, totally random method.

A first crossing approach would be to consider an agent as a "pie chart" where each leaves corresponds to a character. By randomly choosing two cut points in our agent compared to a referential, we would exchange two parts to form new individuals.

However, a problem happens, how do we set our referential? A priori, we can not set a permanent referential, because in this case, it supposes to consider an adjustable individual. So, an agent is an entity that has no facets. An agent is comparable to an individual part of an organization. Nevertheless, it is not possible to describe it as a physical individual (a man). Therefore this first approach is interesting but does not allow us to have entire satisfaction.

Knowing that agents have not all the same genetic patrimony, that is to say that they have no equal chromosome lengths and knowing that an agent has no facets, we can represent it as a toroïdal chain of bits.

- This representation does not suppose the intervention of the notion of facets of an agent.
- We can cross individual of different lengths [7].

It is always necessary to define a starting point for our chromosome in order to correctly exchange phenotypes. Which one do we choose?

In our system, an agent is composed of functions of action, knowledge and behaviours, that make a certain number of possible referentials. Therefore, the choice of a referential would be a problem, except by randomly choosing it. Thus, we can use this toroïdal representation. Among possible functions, what distribution must we use? In theory, no distribution is ideal, nevertheless, so, to continue with this circle scheme, we will use a circle distribution or gaussian method according to the probability.

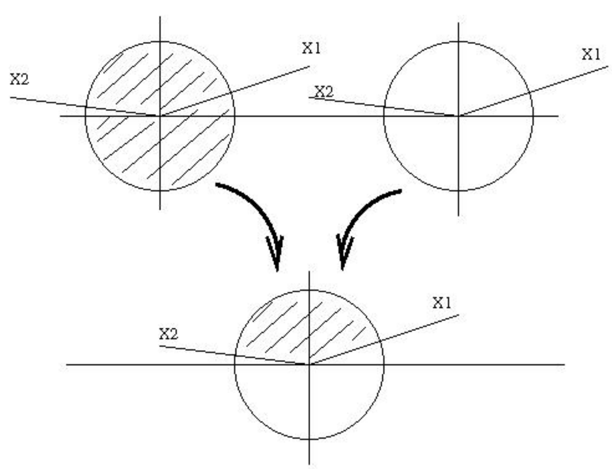

Figure 5. Crossover of the structure of an agent

Thus, it is possible to set a referential for the crosing. Though, the use of a simple crossing does always not give good results. Consequently, the use of a multiple crossings allows us to end to a more important mix. We will use the uniform crossover to always have viable individuals for our representation. However, it is always possible to use the crossover defined by David GOLDBERG such as the CX, OX and the PMX [13], that always give viable individuals.

4.1. Comparaison with other technics

In the article of Mertoguno & Lin [12], they use a Multi-Agent System to build an adaptive knowledge based system. This one represent this knowledge as a graph structure. Nevertheless, in our approach, the knowledge isn't represented as a graph but as a vector (such as a DNA). The main goal for our problem is not to have a distributed adaptive knowledge based but an only one knowledge based where all elementary heuristics are, to resolve our Job-Shop Scheduling Problem.

At the beginning and at the end of the evolution process of our system, the knowledge system is the same, it is only agents, components of the Gantt diagram, that change. By mutations and crossovers, agents are modified, in particular in their behavioural, tendencies, etc. It's the agents that are mutate, not the knowledge graph.

Therefore, in our approach, the main problem is that we cannot be sure that the system will stabilize at the end of the process. Agents can modified the Gantt diagram enormously.

5. The fitness function

In our case, it is necessary for us to optimize a Gantt diagram. Therefore, the last operation to undertake will have to correspond to the date of end minus the time of the task. It is necessary, therefore, to minimize the delay and the advance of the set of jobs.

The objective with an advance and a null delay is nearly impossible. In a general manner, we allow a certain delay or advance. Calculate the fitness of an agent, that is to say its impact on the Gantt. Of course, for the set of jobs, we can have a delay or a weak advance. Consequently, we no longer have a fitness function but many. We have as many objectives as we have jobs. Consequently, we have a case of "multi-objectives genetic algorithm". For this type of problems, we will use the basic concepts of the Multi-objective Optimization Problem (MOP).

5.1. Basic concepts and definitions

The fundamental difference between an optimization having a simple or multiple objectives is the idea of the definition of an optimal solution. The idea of optimality in the multi-objective case is a natural extension of what we have during an optimization for a unique objective.

A multi-objective optimization problem (MOP) can be defined as follows:

$$MOP : \min_{x \in X} f(x) \quad \text{where} \quad f(x) = (f_1(x), ..., f_n(x))$$

is a vector of n real values coming from objective functions, x is a vector of n variables of decision and

$$X = \left\{ x \middle| x \in R^m, g_k(x) \leq 0, k = 1, ..., m, and\, x \in S \right\}$$

is a set of possible solutions. $g_k(x)$ is a real function value representing the k^{th} constraint and S is a subset of R^m representing all the other forms of constraints. 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 MOP generally requires the identification of Pareto optimal solutions, a concept introduced by V. Pareto, a prominent Italian economist, at the end of the last century. 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.

All potential solutions to the MOP can thus be classified in dominated and non dominated (Pareto optimal) solutions, and the set of non dominated solution of an MOP is called Pareto front.

The first and most important step in solving a MOP is to find this set or a representative subset. Afterwards the decision maker's preference may be applied to choose the best compromise solution from the generated set.

The natural ordering of vector valued quantities is basic for Pareto optimality. To define the notion of domination let $f = (f_1, ..., f_n)$ and $g = (g_1, ..., g_m)$ be two real-valued vectors of n elements; f is partially smaller than g if:

$$\forall i \in 1, ..., n, \forall k \in 1, ..., m, f_i \leq g_k \quad \text{and} \quad \exists i: f_i < g_k$$

we note $f <_p g$

If $f <_p g$, we say that *f* dominates *g*. Consequently, a feasible solution x^* is said a Pareto optimal of the problem if and only if it does not exist another $x \in X$ such that $f(x) <_p f(x^*)$.

6. Development of Pareto Optimal Solutions

Two different strategies are effective in generating Pareto optimal solutions [4]. In the first strategy, an appropriate Scalar Optimization Problem (SOP) is set up in parametric form, so that the solution of the SOP with given values of the parameters, under certain conditions, belongs to the Pareto front; changing the parameters of the SOP leads the solution to move on the front. In the second one, the MOP is solved with a direct approach using the dominance criteria, so that a set of Pareto optimal solutions is developed simultaneously. The main advantage of the first strategy is that SOP are, generally, very well studied problems and many efficient methods are available to solve them.

On the other hand, some reduction strategies do not guarantee a complete equivalence between the original MOP and the resulting parametric SOP, when some conditions on the feasible set and on the objective functions are not satisfied. Consequently, some Pareto optimal solutions may never be discovered using this reduction approach. Furthermore, even when the reduction scheme allows a complete equivalence of the problems, it may be necessary to solve a great number of SOP to have a representative subset of the Pareto front. Two possible ways of defining the equivalent SOP are the Weighting Approach and the Constraint Approach.

6.1. Equivalent SOP 1: The Weighting Approach

Following the weighting approach, the MOP is put in correspondence with the following parametrized SOP.

$$P(w): \min_{x \in X} w^T f(x) = \sum_{j=1}^{n} w_j f_j(x)$$

where

$$w \in W = \left\{ \begin{array}{l} w \middle| w \in R^n, w_j(x) \geq 0, \\ j = 1,\dots,n, et \sum_{j=1}^{n} w_j = 1 \end{array} \right\}$$

the correspondence between the MOP and the SOP is subject to some rules. If x^0 is an optimal solution of $P(w^0)$ then it is also Pareto optimal if one of the two following conditions is verified:

- x^0 is the unique optimal solution of $P(w^0)$;
- w^0 is strictly positive.

This implies that at least some Pareto optimal solutions can be generated by solving *P(w)* for some properly chosen *w*, without any hypothesis on the convexity of *X* and *f(X).*

Instead, some convexity hypothesis are a necessary condition. Therefore, if both *X* and *f(*X) are convex, then for any given Pareto optimal solution x^*, it is possible to find a weight vector *w*, not necessary unique, such that x^* is a solution of *P(w).* Therefore, when these convexity assumptions are verified, all Pareto optimal solutions can, in theory, be found by varying *w* and solving *P(w)*, while, if they are not verified, some Pareto optimal solutions may never be discovered by this procedure.

6.2. Equivalent SOP 2: The Constraint Approach

The constraint approach is based on the following parametrized minimum problem:

$$P_k(\varepsilon): \min_{x \in X} f_k(x)$$

subject to

$$f_j(x) \leq \varepsilon_j,\ j = 1,\dots,n,\ and\ j \neq k$$

where

$$\varepsilon = (\varepsilon_1,\dots,\varepsilon_n)^T \in R^n$$

is the vector of parameters.

The main advantage of this approach is that convexity assumptions are not required. Therefore all Pareto optimal solutions can always be discovered by solving the constraint problem $P_k(\varepsilon)$ for any *k.*

The correspondence between the MOP and the SOP is subject to the following rules:

If x^0 is an optimal solution of $P_k(\varepsilon^0)$, with ε^0 a vector for which $P_k(\varepsilon^0)$ is feasible, then x^0 is a non dominated solution of the MOP if one of the two following conditions occurs:

- x^0 is a unique solution of $P_k(\varepsilon^0)$ for some given *k* between *1* and *n.*

- x^0 is not unique, but solves $P_k\left(\varepsilon^0\right)$ for each and every *k=1,...,n*.

On the opposite, if x^* is a non dominated solution of the MOP, an ε^* can always be found such that x^* is the optimal solution of $P_k\left(\varepsilon^*\right)$ for each and every *k=1,...,n.*

In fact, this condition is verified when $\varepsilon_i = f_i\left(x^*\right)$ for all $i = 1, \ldots, n,\ i \neq k$.

6.3. Direct MOP Solution

Directly solving the MOP [4] has the advantage of finding a representative subset of the Pareto front in one shot; on the other hand, not many efficient methods exist which are capable of this approach. A genetic Algorithm using the dominance criteria to drive the evolution of the population is one of these methods. The characterizing feature of the multi-objective GA is thus the introduction of the Pareto criteria in the method used for individuals selection; by selecting individuals in the reproduction phase according to the domination criteria, a set of non dominated solutions can be developed. These are all possible alternative solutions to the problem, which meet the requirements at different level of compromise, and that approximate the Pareto front of the problem. In this way, the arbitrary choice regarding the weights to attribute to the different design criteria is avoided.

7. Going deeply in the relationships of GA and MAS

There are at least two major ways when one is wanting to merge GA and MAS. The first consists in having an outer algorithm which controls the reproduction and/or evolution of agents. This algorithms then sets rules for reproduction and selection according to so me optimal behaviour. However it has a major inconvenient: it can not deal with what we call adaptive systems [2]. This is why we must consider another way of using genetic tools along with MAS.

This way consists in considering that there is no given fitness function of any sort and to see agents as entirely autonomous genetic based entities. They must own their entire genetic code and must be only subjects to random mutations and choose their mates according to their own needs, much like in real life, that is.

Adaptive system primary goal is to adapt itself to its environment and has, strictly speaking, no problem to solve. The only problems that arise are those that are encountered when the system must find a way to adapt. The way its constitutive agents aggregation, struggle and more generally build its organization with time reveal the way the system is attuned to its environment. However, in the system we developed there is not such genetic tools and the agents merely perform cloning and alter their behaviour according to some rules and are constrained by the general emergence that is appearing in the system's agents organization.

We think it can be interesting to investigate the ways how genetic tools can bend the emergence of adaptive systems - in other words, their capacity to adapt. We are actually working on a model that will allow the reification of basic behaviour into agents with the aid of "comportemental DNA", it will code the way an agent will see its environment and its capacities to find agents that suit its needs in order to significantly alter a semantic characteristic in the system's emergence.

8. Objective of the MAS on the Gantt Diagram

By definition, MAS represent a subset emerging of the Artificial Intelligence that tend to put in evidence the two following principles:

- The complex system construction employing agent multiple,
- Mechanisms for the co-ordination of independent agent behaviours.

Nevertheless, this definition is not generally accepted in AI, for purposes contained in our article, we consider an agent as being an entity with objectives, actions to accomplish and areas of knowledge, which is situated in its environment.

However, the ability to consider the co-ordination of the autonomous agent behaviour is a new way among fields of the Distributed Artificial Intelligence (DAI). 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. Thus, **it appears the notion of group**. The objective of the MAS is to improve the Gantt diagram, therefore it invites to establish the notion of group corresponding to elementary entities having common grinds and physical sameness (same capacity of machine, etc.) or interdependence.

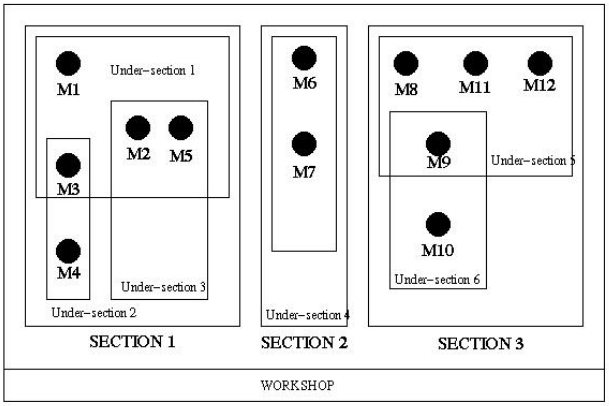

Figure 6. Description of our workshop

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 similarly or close nature.

Agents have to intervene on groups and elementary entities, the MAS will be then composed with micro and meta-agents. It is therefore important, for this evolution, to introduce agents having a character : the meta-agents of evolution. These meta-agents will have therefore as function to make evolve this organization by means a Genetic Algorithm establishing the sexued reproduction of agents. It is necessary to note that, traditionally, agents have as unique possibility only the cloning. But here, we use Genetic Algorithm for the physical evolution of agents. It appears therefore, in the course of the evolution, different sizes of agents: we will speak about agent's granularity. We have therefore micro and the meta-agents that are going to intervene, according to their size, 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 (group) as well as on some zones of the Gantt diagram. It appears therefore a distributed agent system being able to mutate and cross between them.

9. The Multi-dimesion Transformation of a Gantt Diagram

The representation of a planning under the form of a Gantt diagram in two dimensions does not allow to define the global characteristics or general of the former. By the former, we hearing the notion of quality, the respect of the master plan, etc. An operation, constituent of a job, makes emerge only the characteristics places (delay, advance, etc). Nevertheless, these local characters do not make show local information, nevertheless capital, aiming to obtain from global evaluations from predictions (they as global) of the commercial service. The roundup of operations by family, by taking account a macro-nomenclature, to which are made correspond the macro-programs for the realization of tasks corresponding to a family, is not visible. General manner, given the nature NP -Difficult of organizations, we plan to improve a Gantt diagram, according to an economic function. In order that, it is necessary us to transform the evaluated Gantt in a dynamic system whose:

- views to the level place, correspondent to a structure such that the share, the operation, ... and
- views to the level global, correspondent, they also to a structure (Gantt, etc).

These views are in interaction unite them by report to others by the dynamic function intermediary, which correspond to the evolution of an organization. The idea is to end to an organization of manipulated elements. This organization is in tension, that is to say that some elements "react" by putting in obviousness the fact that they not contribute to an improvement of the Gantt diagram. The goal of our system is "to slacken" it, to end to a global improvement of the Gantt without arrive to a "rupture". By rupture, we hear the fact that the Gantt no longer goes from the whole to satisfy production needs and to take account master plans of the enterprise.

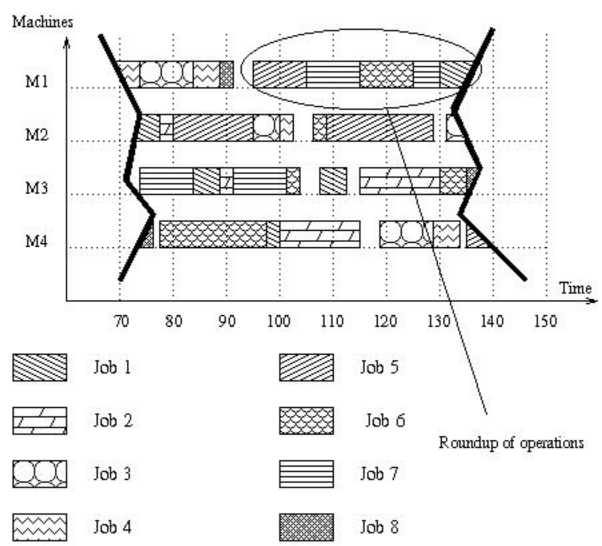

Figure 7. Roundup of operations on the Gantt Diagram after the simulation

To make this, we plan to realize a coloured transformation of the Gantt diagram to obtain a multi-dimensional representation of the former. We have chosen as formalism the decomposition of the spectrum of the white light to represent the n-uplet **Advance -- Delay - Priority**. Thus, by playing on nuances, we can represent all possible cases n-uplet. However, some characteristics are not visible but have inevitably to appear in the coding (family of product, etc) so as to realize the crosscheck to determine strategies to implement for multi-agent systems, composed of global agents (of trends) and local agents (operation on one or a group of tasks), could improve globally the Gantt diagram, while satisfying to respect of the population. Thereby, the radiation of an operation corresponds to its ability to make show information clearly: activity to realize, time of cycle, machine to use, etc. A 2-D Gantt diagram represents a discreet environment because of the presence of holes, consequently, it is not possible to consider as continuous space. Topologies, generally defined, make reference to a continuous environment. Our objective is to show that connexities ordinarily defined in a discreet space are equivalent to the connexities of a continuous space. That will allow us to define continuous totalities from totality discreet. A general manner, that returns to make a decomposition of R^n.

From the transformation of the Gantt diagram in discreet multi-dimension images, we can regroup identical information between them. By transforming our Gannt diagram in multi-dimension image, we can have a more global vision of the system on which agents will be able to intervene.

10. Heuristics granularity in a multi-agent system for our models

In production management, Gantt diagram's optimization can be considered as NP-Difficult problem. 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. Therefore, we use Multi-Agents Systems (M.A.S.) [5]. These simulate the behaviour of entities that are going to collaborate to accomplish actions on the Gantt with view to better resolving the given economic function. Component agents of the M.A.S. [2] can be:

- Local agents whose actions result of "simple" heuristics acting on a well known task (permutation of tasks in case of due date, measure of the algebraic tardiness of a task, etc.).
- Global agents whose actions are the result of heuristics, more global that can be extracted from Gantt diagram (too many holes, a lot of job witch are late, etc.) referring to aeras. These agents have an a priori knowledge of the environment, they can determine a quality for our diagram: good, worst, ...

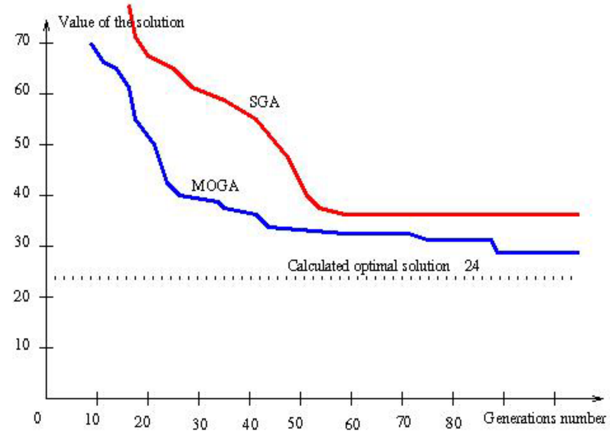

Figure 8. Convergence velocity for our problem

- Therefore glogal agents contain meta-heuristics corresponding to actions to consider according to the meaning that agents have from the Gantt diagram [16]. By opposition, local agents own "simple" or "combinated" actions heuristics. The problem is to bridge the gap between actions of local agents (local heuristics) and global agents (global heuristics). During the co-operation process of agents, they are developed and left their marks in the environment. The resolution of the optimization problem is done by the agent's evolution.

This evolution can be obtained from different handles:

- a mutation of the behaviour of some agents, these being obtained by evolutionary algorithm,
- a pool of agents inducing the emergence of agents of intermediate granularities (intermediate heuristics) between local agents and global agents.

Thus, by operating of local agent pools to form global agents or by refining global agents to obtain quasi minimal agents, we will end at the creation of agent of intermediate granularities. However, this mutation of the behaviour does not have to be made in totally random maner, it must take different communications into account that can exist. Therefore, this mutation will have to correspond to a character or to an emergent tendency of the action of agents [17]. This agents reproduction by evolutionary algorithm will not have to take crossover from two parents into account but from two or more parents. The crossbreeding will not have to take local functions of agents into account but senses resulting from the agregation of agents in a group with a group action function. Consequently, new agents, with an intermediate granularity, will have functions resulting from parents but, they will have senses, tendencies and news visions for actions to accomplish on the Gantt.

11. Conclusion

We have seen in this communication the way in which we have chosen to represent the problem at the level of workshop and at the level of jobs. This representation is only for a Job-Shop problem with M machines and N jobs.

In Job-Shop production scheduling, Gantt diagram's optimization can be considered as NP-Difficult problem. Determining an optimal solution is almost impossible, but trying to improve an existent solution is the way to lead to a better tasks repartition. Therefore, we use Multi-agent Systems (M.A.S.). It simulates the behaviour of entities that are going to collaborate to accomplish actions on the Gantt diagram so as to resolve the given economic function. Multi-agent systems include cognitive agents whose the behaviour tends to satisfy one or some objectives taking into account some constraints of facilities and their proper valuations. Normally, agent creation is simply based on cloning. Thus, it is possible for agents that provided or done good actions, that they could give birth to

individuals whose characteristics would be superior by crossing. So, we use the notion of evolution, by introducing evolutionary algorithms such as the Genetic Algorithms so as to simulate a Darwinian process.

Consequently, we have proposed to put in evidence the use of spirit of Genetic Algorithms into evolution in a M.A.S. We plan to focus on the possible relationships between MAS and GA in order to define a new property of agents, and more generally, of MAS: **the notion of sexued reproduction**. In our case, it is necessary for us to optimize a Gantt diagram. Therefore, the last operation to undertake will have to correspond to the due date minus the time of the task. It is necessary to minimize the delay and the advance of the set of jobs. The objective with an advance and a null delay is nearly impossible. We must calculate the fitness of an agent, that is to say its impact on the Gantt. Of course, for the set of jobs, we can have a delay or a weak advance. Consequently, we haven't a fitness function but many. We have as many objectives as we have jobs. So why, we have a case of "multi-objective genetic algorithm".

The ideal solution of this 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. Solving a multi-objective problem requires the identification of Pareto optimal solutions. During the optimization process, it appears different agents with different granularities.

Therefore, glogal agents contain meta-heuristics corresponding to actions to consider according to the resources that agents have from the Gantt diagram; local agents own "simple" or "combinated" actions heuristics. The problem is to bridge the gap between actions of local agents (local heuristics) and global agents (global heuristics). During the co-operation process of agents, they are developed and left their marks in the environment. New agents, with an intermediate granularity, will have functions coming from parents but, they will have senses, tendencies and news visions for actions to accomplish on the Gantt. Communications between global and local agents, due to their actions, manage the appearance of agents with an intermediate granularity and the global optimization in Job-Shop Scheduling Problems.

Communications between global and local agents, due to their actions, manage the appearance of agents of intermediate granularity and the global optimization in production scheduling.

12. References

[1] T. Bäck, "Self-Adaptation in Genetic Algorithms", *ECAL-92*, 1992, pp. 263-271.

[2] A. Cardon, and F. Lesage, "An Interpretation Process of Communication Between Actors in a Distributed System for Crisis Management", *Proceedings of the Symposium of Informatic Economics*, 1997.

[3] A. Cardon, and S. Durand, "A Model of Crisis Management System Including Mental Representations", *AAAI Spring Symposium*, Stanford University, USA, 1997.

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

[5] J. Ferber, *Les systèmes multi-agents, Vers une intelligence collective*, Ed. InterEditions, Paris, 1995, pp. 1-66.

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

[7] D. E. Goldberg, K. Deb, and Bradley Korb, "Messy Genetic Algorithms Revisited: Studies in Mixed Size and Scale", *Complex Systems*, 1990, 4, pp. 415-444.

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

[9] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications in biology, control and artificial intelligence*, Complex adaptive systems, MIT Press, Cambridge, MA, 1992.

[10] J. H. Holland, "Genetic Algorithms", *Scientific American*, 1992, 260 (9), pp. 44-51.

[11] C. G. Langton, "Artificial Life: the proceedings of an interdisciplinary workshop on the synthesis and simulation of living systems", Addison-Wesley, Redwood City, CA, 1989.

[12] J.S. Mertoguno, and W. Lin, "Distributed knowledge-base: adaptative multi-agents approach", *Proceeding IEEE International Joint Symposia on Intelligence and Systems*, IEEE Comput. Soc. Press, 1996, 1, pp.334-340.

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

[14] R. Pfeifer and P. Verschure, "Distributed adaptive control: a paradigm for designing autonomous agents", *ECAL-92* [18], MIT Press, Cambridge, MA, 1992, pp. 21-30.

[15] S. Panwalkar, "A Survey of Scheduling Rules", *Operations Research*, 1977, 25(1), pp. 45-62.

[16] JP. Vacher, A. Cardon, and T. Galinho, " Genetic Algorithms in a Multi-Agent System", *Proceeding of the EUROGEN'97 Conference*, Trieste, Italy, 1997, Web site : http://eurogen.cira.it.

[17] JP. Vacher, A. Cardon, and T. Galinho, "Heuristics Granularity in a Multi-Agents System (MAS): Application to the Production Management", *Proceeding of the YOR10 conference*, Guildford, UK, 1998.

[18] F. J. Varela, and P. Bourgine, *ECAL-92: Towards a Pratice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, MIT Press, Cambridge, MA, 1992.

[19] D. Whitley, A Genetic Algorithm Tutorial, *Colorado State University*, 1993.

[20] D. L. Whitley, "An Executable Model of a Simple Genetic Algorithm", *FOGA2* [21], 1992.

[21] D. L. Whitley, *FOGA2: Foundations of genetic algorithms 2*", Morgan Kaufmann, San Mateo, California, 1993.