A window time negotiation approach at the scheduling level inside supply chains

Marie-Claude Portmann, Zerouk Mouloua

A window time negotiation approach at the scheduling level inside supply chains

Portmann Marie-Claude
LORIA-INPL, Ecole des Mines de Nancy, Parc de Saurupt, CS 14234, 54042 Nancy Cedex, France,
portmann@loria.fr

Mouloua Zerouk
LORIA-INPL, Ecole des Mines de Nancy, Parc de Saurupt, CS 14234, 54042 Nancy Cedex, France,
mouloua@loria.fr

We consider a supply chain, which consists in a network of enterprises with independent decision centers. The finished products (or sub products) of the assembly enterprise are produced using components and/or sub products supplied by other enterprises or by external suppliers. We assume we are at the scheduling level and each enterprise builds its own schedules associated with its own production centers. As an operation can be performed only when the production center has received the necessary components and/or sub products, the schedules are dependent. This induces negotiations between decision centers of enterprises, which can be expressed in term of penalty functions associated with soft and hard release dates and due dates. At each negotiation point, hard release dates and due dates are considered as imperative constraints, while soft release dates and due dates define soft intervals and induce earliness and tardiness penalties. Both soft and hard constraints can be modified during the negotiation process. A global solution is searched by an iterative decomposition approach including alternatively bilateral negotiations of the soft and hard constraints between the production decision centers and just in time scheduling, minimizing the local total sum of penalties, built by approximation approaches. We assume that production centers are flow shop (linear production and/or assembly line). To solve each local just-in-time scheduling problem, we propose an approximation approach based on meta-heuristics, which explores the set of feasible and infeasible solutions, in which a solution is described by the job order on each machine (permutation of integers) and is evaluated using a “pert cost” algorithm. The infeasible solutions are evaluated by a lower bound of the number of non verified imperative constraints and the feasible solutions are evaluated by the minimal sum of penalties corresponding to the considered order. A semi-decentralized control is suggested to assume the negotiation convergence.

Keywords: scheduling, supply chain, just in time, negotiations, decomposition approach, “pert cost” algorithm, meta-heuristics

1. Introduction

We consider a supply chain, which consists in a network of enterprises with independent decision centers. In consequence, we assume each decision center can prepare its own schedules and negotiate release dates and due dates with their suppliers (upstream level of the supply chain) and their customers (downstream level of the supply chain). The decision system could have been totally decentralized, but in this case, there is no guarantee of convergence of the global negotiation process. In consequence, we choose a semi-decentralized decision system in which a common memory is shared and global rules are agreed at the network level in order to help the negotiations to converge (eventually to infeasible solutions of the initial global just in time scheduling problem, by authorization of unlimited increase of the finished product tardiness in the worst cases).

We consider the following production system. The finished products (or sub products) of the assembly enterprise are produced using components and/or sub products supplied by other enterprises or by external suppliers. We assume we are at the scheduling level and each enterprise builds its own schedules associated with its own production centers. We assume the transport between production centers is taken into account only by minimal time lags. As an operation on a product can be performed only when an enterprise has received the needed components and sub

products, the schedules are in fact dependent. This induces negotiations between enterprises, which can be expressed in term of soft and hard release dates and due dates. Hard release dates and due dates can be changed in order to increase or decrease the set of feasible solutions of each local partner. Soft release dates and due dates and their associated earliness and tardiness penalties can be real or only virtual delays and penalties in order to lead the whole network towards a win-win global solution. A global solution is searched by an iterative decomposition approach including alternatively bilateral negotiations of the soft and hard delays between the production centers and just in time scheduling minimizing the total sum of penalties inside each decision or production center. A semi-decentralized global negotiation approach will be suggested in section 2.
We assume that production centers are either flow shop (assembly line) or even job shop (this complicates only the just in time scheduling algorithms). To solve each local just-in-time scheduling problem associated with a given set of hard and soft release dates and due dates and a given set of penalty data, we propose an approximation approach based on meta-heuristics, which explores the set of feasible and infeasible solutions, in which a solution is described by the job order on each machine (permutation of integers) and is evaluated, using a polynomial "pert cost" algorithm. Section 3 presents the just in time scheduling problem and how the timing problem can be solved by a "pert cost" algorithm. Some experiments concerning the just in time scheduling problem are presented at the end of section 3.

2. Global negotiation model

2.1. Global problem description

In the framework of decentralized or semi-decentralized decision system inside supply chain, several models have been proposed at the medium term level (Dudek and Stadtler, 2005), (Hurtubise et al, 2004). This proposition is in continuation of our own research works at the medium term planning (Ouzizi et al, 2006). At the medium term level, volumes associated with family of finished products, sub products and components are forecasted and production and transportation phases are planned in order to take into account the limited capacities of the resources. Negotiations can be organized between the independent partners in order to take into account their limited capacities and aggregated linear models can be used in order to compute each local planning. At the medium term level, hard and soft cumulated curves of component arrival or product delivery are negotiated between the supply chain partners and complementary resources can be bought. At the scheduling level, the forecasted medium term volumes on product family have been converted into firm orders of very precisely definite products. Each order is associated with one or several jobs (if lot sizing or splitting is used). Hence a job corresponds either to only one product or to an inseparable series of products, for which the detail schedule parameters are known: production centers to be visited, main sub products to be waited for (release date on the first machine of a given shop or further on another assembly machine), one or more production process inside each center (generalized job shop or hybrid flow shop). We assume here that external due dates have been promised to the final customers, which do not belong to the supply chain, and external release dates have been agreed with the external suppliers, most of them being imperative, due for example to transportation delays. We had to find the global schedule of the considered supply chain, while minimizing the total production cost and the total external penalties: win-win strategy inside the supply chain.

2.2. Global negotiation process

The ideas used at the medium term level, i.e. hard and soft cumulated curves of component arrival or product delivery with earliness and tardiness penalties negotiated between the supply chain partners, could be extended to the scheduling level replacing hard and soft cumulated curves by hard and soft operation windows. The main idea of the global negotiation process is also an extension of the process proposed in (Portmann, 1988) and (Chu et al, 1992). In these papers, the criterion considered was the sum of weighted tardiness (a regular criterion) and only active schedules associated with hard release dates were considered; in consequence, only hard release dates, but hard and soft due dates were used in this iterative splitting approximation approach, whose aim was to build a global schedule for a given production center, obtained by scheduling optimally cells and modifying hard release dates and hard and soft local due dates. This splitting approach gave very efficient solution methods.

The extended similar process is the following one illustrated by a small example on one product.

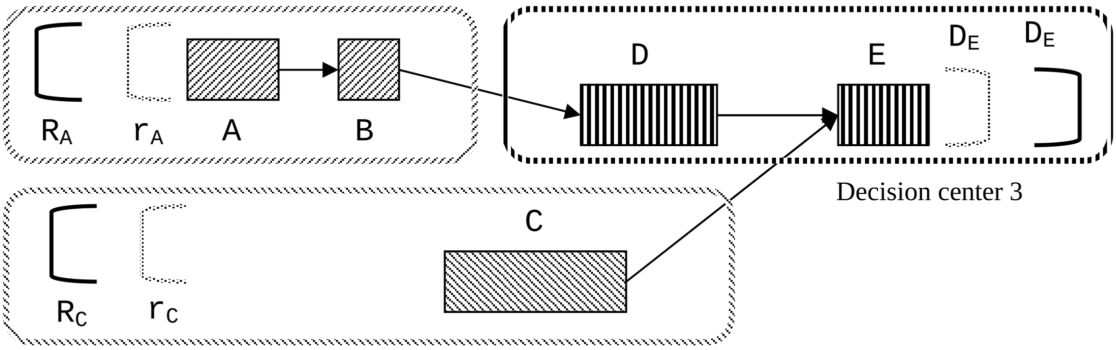

Figure 1: external constraints

In figure 1, soft and hard due dates associated with a given finished product order due to an external customer are denoted by d_E and D_E. The assembly of this finished product order needs two types of sub products used at two successive operations of the assembly line and sub products, made by two different supply chain partners, need themselves components from external suppliers, whose soft and hard release dates are respectively r_A, R_A, r_C and R_C.

If there exists only one decision center for the whole supply chain, then we have got only one just in time scheduling problem to be solved, in which production costs must be minimized while verifying the external hard release dates and due dates (if possible) and minimizing the total sum of earliness and tardiness external penalties.

While we consider a semi-decentralized control of independent decision centers, each decision center computes its own just in time scheduling and negotiates hard and soft release dates with the other partners of the supply chain with the aim of finding a global solution as good as possible, which will be a trade-off between the interests of the partners.

To organize the negotiations, we need initial soft and hard window constraints between the whole set of partners. They will be called "virtual"; because they are not imposed by the global just in time scheduling problem and they will change during the global negotiation process.

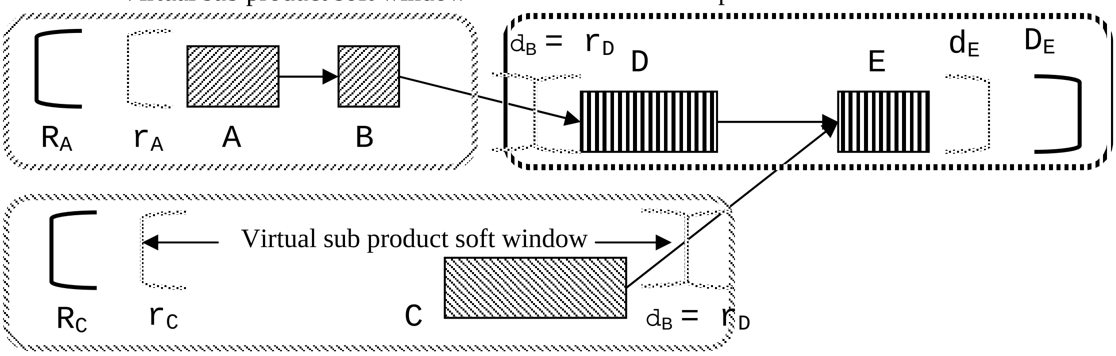

Figure 2: initial soft window constraints

First, similarly to (Portmann, 1988), we use the soft release dates and due dates of each sub product, of the final product and their processing times and we call a sharing function (for example, soft window lengths are proportional to the corresponding sum of processing times) in order to define “virtual” windows for each manufacturing part (see figure 2). It can be remarked that release and due dates are associated to any operation, whose predecessors and/or successors are in another decision centers (minimal time lags can be added in the model for transportation duration). Hard initial windows are also obtained by a given rule. For example, they can be very large at the beginning of the global process (each operation is left shifted on its own hard release date or on its previous left shifted operations for the hard release dates and each operation is right shifted on its own hard due date or on its following right shifted operations for the hard due dates) or the difference between the hard and soft window lengths of each sub product and/or product is computed by a function similar to the one use for the soft constraints (proportionality rule). Virtual penalties are attributed equally without any further information or individually increased when, for example, it is known that some machine of the production system is overloaded and upstream and downstream operations must respect as tight as possible the virtual given windows.

After this hard and soft windows attribution, the schedules of the various production centers become independent and an exact or approximation scheduling tool can be used in order to minimize the earliness and tardiness penalty sum while respecting the hard windows. Each schedule can be computed locally by the decision centers in real life applications. This initial schedule computation finishes the first step of the global process.

An analysis is made at the end of each step of this splitting approach.
<u>Case 1:</u> If the external soft release dates and due dates are respected and any precedence constraint between two different production centers is respected, i.e. when an operation is late relatively to hard or soft “virtual” due dates, then we have the chance that the following operations are at least as late as it and when an operation is early relatively to its hard or soft “virtual” release date then the previous operations are sufficiently early for the global schedule to be feasible. It is the perfect case and the negotiation process can stop without any external negotiations.
<u>Case 2:</u> If external hard release dates and due dates are respected, but not any soft external windows, and nevertheless the schedules of the various production centers are compatible, then two ways can be followed before the next step, either to negotiate new external soft release dates and/or due dates or to increase the penalty associated with the external earliness and tardiness hoping to get a better set of feasible local schedules.
<u>Case 3:</u> If external hard windows are respected, but the set of schedules induce incompatibility between production centers, then the internal hard and soft windows must be modified in order to converge to a set of feasible schedules. In a semi-decentralized decision system approach, the order in which the windows are negotiated can be organized, going from supplier to customer (direct succession of schedules and windows modification) or from customer to supplier (retrograde succession of schedules and windows modification, knowing that the minimal hard release date must remain always positive). This organization is a way to assume the convergence of the global process (assuming that case 4 can enlarge the external hard windows to increase the set of feasible solutions).
<u>Case 4:</u> If external hard windows are not respected, it is a very delicate situation. The explanation could be that the whole global problem has no feasible solution, but it is impossible to verify if the problem is too big to use an exact algorithm to solve the centralized global just in time scheduling problem. The only way to continue is to relax slightly the hard final due dates and to begin a new negotiation process between the partners.
The convergence of the process can be assumed by the organization of the authorized negotiations (no loop in the negotiation process) and by the last assumption that hard final due dates can be relaxed. The efficiency of this global negotiation approach can only be verified by experimentation using simulation for modeling the human behavior inside the decision process.

3. Just in time scheduling problems

To complete this window negotiation process, we need a tool for solving the scheduling problem with hard and soft window inside each production center. We simplify here the scheduling problem assuming there are no production costs (such as inventory costs or overtime hours or sub contracting …) to be considered and we have only to minimize the sum of earliness and tardiness penalties.

If the local shop we consider is a flow shop and hard and soft release dates are associated to the first job operations, while hard and soft due dates are associated to the last job operations, then our problem can be denoted by $F/\overline{r_i}, r_i, d_i, \overline{d_i} / \Sigma\alpha_{r,i}E_i^r + \Sigma\beta_{r,I}T_i^r + \Sigma\alpha_{d,i}E_i^d + \Sigma\beta_{d,I}T_i^d$ in the classical $\alpha/\beta/\gamma$ notation where F means flow shop, $\overline{r_i}, r_i, d_i, \overline{d_i}$ are respectively the hard and soft release dates and due dates of the first and last operations of each job and $\alpha_{r,i}$, $\beta_{r,i}$, $\alpha_{d,i}$, $\beta_{d,i}$ are the earliness (α) or tardiness (β) penalty associated with the release date of the first operation ($_r$) and the due date of the last operation ($_d$). When we assume that release dates and/or due dates can be associated to any operations of the jobs, we can use a more general $\alpha/\beta/\gamma$ notation: J / $tmin_j$, t^*_j, $tmax_j$ / $\Sigma\alpha_jE_j+\Sigma\beta_jT_j$, where t^*_j is the ideal position of the starting time of the operation j, $tmin_j$ is its minimal authorized value (hard constraint for earliness) and $tmax_j$ is its maximal authorized value (hard constraint for tardiness).

We propose to use another approximation decomposition approach to solve this local problem. If we assume the operation sequence is given on each machine (without circuit for the job shop problem), then we have only to compute the optimal start time of each operation (i.e. where are put the idle times, because our criterion is not regular) and we use a meta-heuristic approach to explore the set of operation sequences on the machine in order to find a feasible solution as good as possible. We now present the timing scheduling problem when the sequence is given (by a chromosome) and the main feature of the meta-heuristic afterwards.

3.1. Timing scheduling problem (i.e. chromosome evaluation)

We have to solve the $F_{seq}/\overline{r_i}, r_i, d_i, \overline{d_i} / \Sigma\alpha_{r,i}E_i^r + \Sigma\beta_{r,I}T_i^r + \Sigma\alpha_{d,i}E_i^d + \Sigma\beta_{d,I}T_i^d$ scheduling problem or the J_{seq} / $tmin_j$, t^*_j, $tmax_j$ / $\Sigma\alpha_jE_j+\Sigma\beta_jT_j$, where "seq" means the sequences on the machine are given by a chromosome. For job shop problem, some chromosomes can be unacceptable whatever could be the hard windows because of circuit in the precedence graph, while this cannot happen for flow shop or single machine problem. We assume the meta-heuristic used is able to build only chromosome with no circuit in the precedence graph (we have not this problem in our first experiments because we consider only flow shops). When considering only the hard windows ($\overline{r_i}$, $\overline{d_i}$) or [$tmin_j$, $tmax_j$], for fixed sequence on the resources, our problem is simply a "PERT" problem, which could be polynomially solved by computing the left shifted schedule ($\lambda_{k,i}$) or (λ_j) and the right shifted schedule ($\lambda'_{k,i}$) or (λ'_j) and verifying that for each operation k of each job i (denoted j) we have $\lambda_j \leq \lambda'_j$. We use the number of times this inequality is not verified to measure the infeasibility of the chromosomes relatively to the hard windows, but we keep this infeasible chromosome with very bad fitness as current solution of simulated annealing or inside the genetic algorithm population.

We now assume a chromosome is totally feasible (no circuit due to the sequences on the machine and there exists at least one solution verifying the hard windows) and we want to find each operation starting time of a feasible schedule in order to minimize the total sum of earliness and tardiness penalty. This problem can be modeled as a "pert cost" scheduling problem as follows. s is the source and t the sink of the "pert cost" extended graph CG, where each operation, assumed to be non preemptive can be represented by its beginning event. The duration of each operation is added to the value of the arc issued from this event. The operation sequence of the job and on the machine is represented by usual precedence constraints between the operations.

We denote by H the greatest hard due date (or the greatest $tmax_j$), H is the length of the considered horizon. The figure 3 gives the sub graph representing earliness or tardiness penalty for any operation. OP_j is the beginning of the operation j, t^*_j is its ideal position. If x_i is equal to 0, then the operation j is ideally placed. If x_i is positive, then the event is early and if it is negative then the event is late. We considered that the arc issued from s is compressed and induce a cost $\alpha_j.x_j$ when x_j is positive and that the arc arriving in t is compressed and induce a cost $-\beta_j.x_j$ when x_j is negative. It is important to remark than the length of the path from s to t through BS_j, OP_j and ES_j is exactly equal to H, the length of the horizon and the values x_j will be chosen so that the maximum path in the complete precedence graph will also be equal to H, and in this case, any operation j is placed on the time axis by knowing the value of x_j. $t^*_j - x_j$ must remain greater than $tmin_j$ and smaller than $tmax_j$.

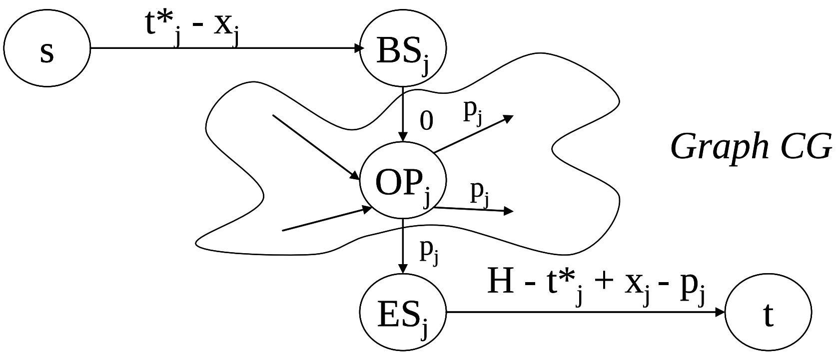

Figure 3: precedence constraints linked to earliness and tardiness penalties

When points of the graph CG are in series with the same set of successors or predecessors, we can delete the beginning of the point series and add the values of the delete arcs to the last arc kept. The data of a particular instance corresponding to a flow shop with two machines and earliness and tardiness cost relatively to the release dates for the first machine and the due dates for the second machine are given in table 1 and the simplified corresponding "pert cost" extended graph is given in the figure 4.

i	p_1	p_2	$\overline{r}$	r	α_r	β_r	α_d	β_d	d	$\overline{d}$
1	5	2	0	10	2	1	1	2	20	30
2	2	3	3	9	3	2	1	3	15	30
3	4	5	1	7	3	0	0	5	18	30
4	4	2	5	9	4	3	2	3	25	30
5	3	1	3	5	2	1	1	2	22	30

Table 1: precedence constraints linked to earliness and tardiness penalties

It is a particular "pert cost" problem in which you can only pay to decrease the duration of the arcs adjacent to the source s or to the sink t. You can solve it polynomially using a dual approach. But due to the particular shape of the graph, the primal algorithm (using alternatively "pert" algorithm and search of minimal cut on the critical sub graph), when using the maximal compression associated to each cut, is probably polynomial (we do not prove it formally until now). Any way, the minimal sum of earliness and tardiness penalty can be computed for each chromosome by using a "pert cost" algorithm and this sum is used to compute the fitness of each chromosome.

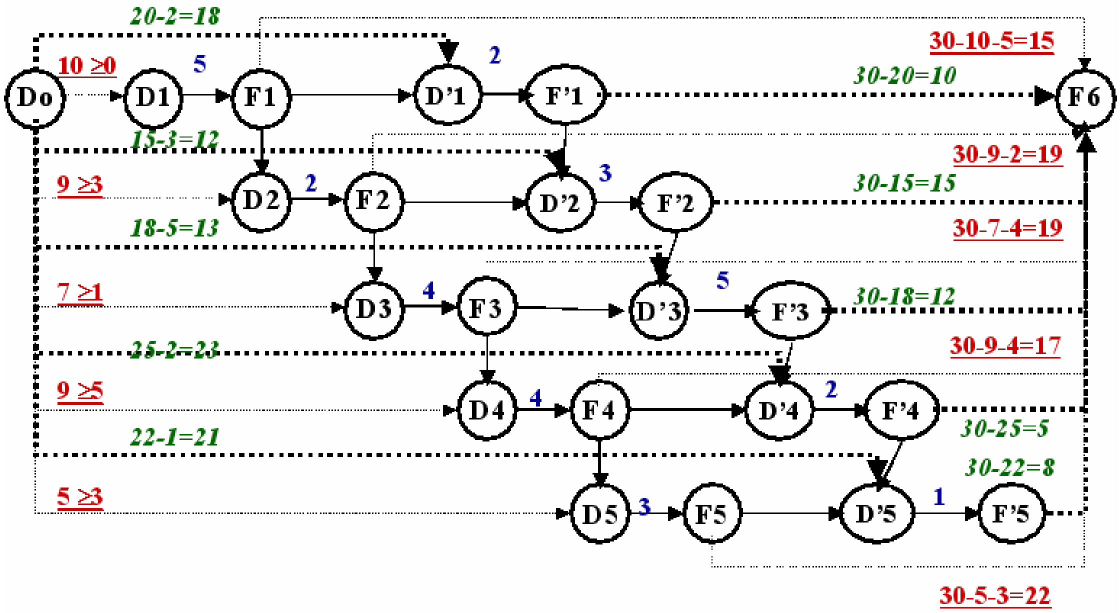

Figure 4: simplified extended "pert cost" for a flow shop problem

3.2. Meta-heuristic approaches

We already used and published papers using meta-heuristic and more specifically genetic algorithms for which we analyze specifically the performance of the permutation cross-over operators for various criteria (Djerid and Portmann, 2000), (Portmann and Vignier, 2000). Nevertheless a rapid experimentation using the simple genetic algorithm (SGA) of (Goldberg, 1989) provides us with deceiving results and our genetic algorithm needs to be improved. On another way, simulated annealing is very rapid to test and only to show meta-heuristic could be a good approach for our problem we use it also in the following results.

3.3. First experimentation for the just in time scheduling problem

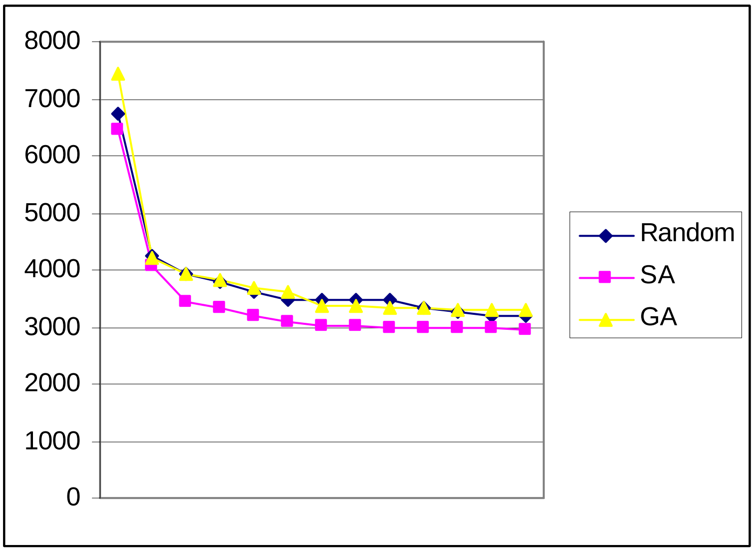

Figure 5: comparison of approximation approach

The experiments were made on generated flow shop examples with 10 jobs and 5 machines. For each instance, three algorithms were used: random generation of a sequence (Random), simulated annealing (SA) and simple algorithm genetic (GA). To see the rapidity to find solution as good as possible, we compute the function BEST(Algo, k) which gives the value of the best found feasible solution after having evaluated k sequences. To eliminate the fact that hazard could be more or less favorable, we execute each algorithm 5 times on the same instance and we keep the mean value, which provides us with MEAN-BEST(Algo,k) and we present here in the figure 5 the average on four instances. Only 300 feasible solutions analyzed by “pert cost”, where generated for each execution of the algorithm. Bad results obtained by the genetic algorithm can be due to the fact that the initial population is randomly built and it is possible, with only 300 sequences studied among 10!, SA or GA have not yet converged. We will continue our experimentation with longer computational time and with improvement of our very simple current meta-heuristic scheme.

4. Conclusion

We propose here a new decomposition approach to organize the negotiation between various independent partners of a supply chain at the scheduling level using hard and soft windows as entity of negotiation. This decomposition can also be used as an approximation approach for building a just in time solution in presence of only one decision center and several production centers, as an extension of the method proposed in (Chu et al, 1992). The interest of this last proposition is that it can be compared with other methods using experimentations, while the model with several decision centers and human decision models must be validated using simulation. Another important problem will be to convince the industrial decision makers and managers of the interest of the proposed models.

References

[1] Chu C., Portmann M.C., Proth J.M. (1992) A splitting up approach to simplify job-shop scheduling problems. *International Journal of Production Research*, **30**-4, 859-870.

[2] Djerid L. Portmann M.C. (2000) How to Keep Good Schemata Using Cross-over Operators for Permutation Problems. *International Transactions in Operational Research* **7**-6, 637-651.

[3] Dudek, G. and H. Stadtler (2005) Negotiation-based collaborative planning between supply chains partners. *European Journal of Operational Research*, **163**, 668–687.

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

[5] Hendel Y., Sourd F. (2006) An improved earliness–tardiness timing algorithm. *Computers & Operations Research*, In Press.

[6] Hoogeven J.A., Van de Velde S.L. (2001) Scheduling with target start times. *European Journal of Operational Research*, **129**, 87-94.

[7] Hurtubise S., Olivier C., Gharbi A., Planning tools for managing the supply chain (2004) Computers & Industrial Engineering, **46**, 763–779

[8] Ouzizi L., Anciaux D., Portmann M-C., Vernadat F. (2006) A model for co-operative planning within a virtual enterprise. *International journal of Computer Integrated Manufacturing*. **19**-3, 248-263.

[9] Portmann M.C. (1988) Méthodes de décompositions spatiales et temporelles en ordonnancement de la production. *RAIRO-APII*, **22**-5, 439-451.

[10] Portmann M.C., Vignier A. (2000) Performances' study on crossover operators keeping good schemata for some scheduling problems, Proceedings of GECCO’2000, 331-338,

[11] Sourd F. (2005). Punctuality and idleness in just-in-time scheduling. *European Journal of Operational Research*, **167**-3, 739-751.