Hybrid FMS Scheduling System Based on Petri Nets and advanced Meta Heuristics


Laboratory of  Automatics and Informatics of Guelma LAIG, University of Gulma, Algeria

Phone (213) 37 20 02 65    Fax : (213) 37 20 72 68
rhadiby@yahoo.fr
kechidas@yahoo.fr
tebbikh@yahoo.com

Abstract: This paper aims at presenting a hybrid FMS short-term scheduling problem. Scheduling problem, we are to study in this paper, is first introduced, and than a survey of the HFMS scheduling litterature is presented. Based on the art state of general scheduling algorithms, we present the Meta-heuristics; we have decided to apply for a given example of HFMS. That is the study of three advanced meta hauristics, Taboo searchTS, Genetic algorithms GA and simulated annealing SA. The HFMS model based on hierarchical Petri nets, helps to represent static and dynamic behaviour of the HFMS. This model is also used to design scheduling solutions. Hierarchical Petri nets model represents the whole process, and can be regarded as being made up a set of single timed colored Petri nets models. Each single model represents one process that is composed of many operations and tasks (manufacturing, transport, handling). TS, GA and SA are applied on each sub-model to make ressources allocation and to resolve conflicts on shared production ressources. A comparison between the three Meta-heuristics is made in simulation phasis to choose the best one to use in different kinds of HFMS.

Key words: Genetic algorithms GA, Hierarchical Petri nets, Hybrid flexible manufacturing systems “H.F.M.S”, Meta-heuristics, Modelling, Simulated annealing SA, Scheduling, Taboo search TS.

1 Problem description

Meta-heuristics have totally changed heuristics elaboration. Wether we begin by asking questions on characteristics and particularities of the problem to solve, before programming a specific method, meta-heuristics have inversed the process because the main algorithm of the method is given by the metaphore which has inspired the meta heuristic. Given a first heuristic, we search its amelioration by observing its weakness, according to the problem we have to solve.

In general, flexible manufacturing systems “FMS” are considered as discreet event dynamic systems “DEDS”. However, some characteristic variables are continuous, like tasks, jobs, evolution of temperature, of pressure and so forth. So, resulting FMS are called

Hybrid FMS because they can be seen as a set of some continuous phenomena depending on discrete events.

Their description is too difficult and complex. We have used hybrid hierarchicla Petri Nets, because of their capacity to depict continuous evolution on timed places and to represent discrete events on transitions. The main problem in an FMS is how to realize all the processes in an optimal time, with a minimal cost. The objective of scheduling system is to organize in time, realisation of interdependent tasks considering contraints on time, cost and ressources. Scheduling method must generally reconcile a static aspect related to the production planning, with a dynamic one, related to real time decisions considering the efficient state of tasks evolution. Several methods have been developed. They can be classified in three important classes :

- Exact optimization methods like graph theory , branch and bound method,…they have proved their limitation because of the important computing time they need to reach the optimal solution.
• Heuristic methods that the inconvenient consists in their particular and individual elaboration for a specific problem.

• Meta-heuristic methods that are iterative research process. They combine different intelligent concepts of known and general algorithms like genetic algorithms, Taboo search, priority rules method, stochastic descent method, neuronal networks, simulated annealing, and so forth...

We have developed two scheduling algorithms based on Taboo search (Ghoul 2002) and genetic algorithms (Ghoul et al 2004).

In this paper, we developpe a scheduling algorithm based on the "Simulated annealing " Meta-heuristic. The objective of the algorithm is to solve assignment problem of different tasks and jobs for each variable machine. The optimization criterion is "MAKESPAN" minimization (realization time of complete process) under "maximization of productivity" constraint.

A comparison between the three Meta-heuristics is given to evaluate each one and choose the best one to solve HFMS scheduling problem.

2 H.F.M.S modelling by hybrid hierarchical Petri nets

A large review of the literature has shown that hybrid flexible manufacturing systems has not been well studied, either in modelling or scheduling problems.

For these reasons, we have proposed some flexible models that can depict both discrete and continuous variables in an HFMS. Timed Petri nets and continuous Petri nets seem to be a complete modelling tool for HFMS. Hierarchical or well-formed Petri nets permit the decomposition of complex schedule into simple and elementary schedules according to different part types manufactured in the HFMS.

2.1 Well formed Petri nets (Atamna 1993)

"Well-formed" W.N’s have been precisely defined in order to combine the modelling power of the coloured PN’s with effective analysis methods. Model construction is obtained by the system intrinsic symmetries. The connection between the system symmetries and the model structure is got by introducing syntax in colours definition. In this way, we can automatically construct a Symbolic Reachability Graph that exploits symmetries for, in particular, reducing the state space size.

It has been obviously tenting to exploit the conciseness of the symbolic Reachability Graph, for also making a quantitative analysis. The stochastic well-formed coloured nets have then been introduced. From this model, we can directly construct an aggregated markov chain based on the symbolic Reachability Graph. However memoryless time distribution (exponential time, zero time) are only considered when it is not sufficient for the analysis of systems with the attribute “real time” (systems with time constraints, still called timed critical systems).

On the other hand, the “Stochastic Timed Petri Nets” have been defined. They deal with arbitrary time distributions (uniform, discrete and mixed distributions). An analysis methodology, based on the randomised state graph concept, has also been developed. As in PN’s, places of WN’s together with there marking play the role of describing the system state while transitions represent events that cause the state changes.

2.2 Continuous Petri nets (Lefebvre 2000)

They result from the timed Petri nets definition. In timed Petri nets, the marking $m_i(t)$ of a place $P_i$ is integer, however in continuous Petri nets, $m_i(t)$ is real. In timed Petri nets, a transition $T_i$ is fired instantly. In continuous Petri nets, it is fired continuously with a certain speed $V_j(t)$. A continuous Petri nets is characterised by a set of maximal speeds of its transitions.

$V_{\text{max}}=(V_{\text{max}}_j)_{j=1,...,m} \in \mathbb{R}^m$ (1)

Each transition $T_j$ is fired with real speed less or equal to its maximal firing speed.

$V(t)=(V_j(t))_{j=1,...,m} \in \mathbb{R}^m$ (2)

Is the set of firing speeds at time $t$.

The marking evolution in continuous Petri nets is defined by differential equation as follows:

$$\frac{dM(t)}{dt}=W.V(t)$$ (3)

When $W$ is the global incidence matrix of the Petri net.

2.3 HFMS modelling by hybrid hierarchical PN’s (Lefebvre 2000)

Hybrid systems are characterized by a set of discrete events and a continuous evolution according to these events.

Two monitoring structures are proposed depending of the signals that circulate between the system and the monitor.

![Figure 1. Discret / continuous monitoring](image-url)
In hybrid Petri nets modelling, discrete state variables are described by places of HPN. Discrete events are depicted on transitions. The system continuous evolution is defined by the marking continuous evolution. The first model is the most close one to the HFMS structure.

3 H F M S scheduling system based on Meta heuristics

3.1. Introduction (Ghoul et al 2002)

The production system studied is a flexible manufacturing cell composed of three machines denoted by M1, M2 and M3 and a robot R. IN represents a conveyer bringing the parts while OUT a conveyer for autogoint parts. Let for simplicity, all parts follow the same path from IN to M1, from M1 to M2, from M2 to M3 and from M3 to OUT.

The six types of products manufactured in the cell are given by the following:

A: IN $\Rightarrow$ M1 $\Rightarrow$ M2 $\Rightarrow$ M3 $\Rightarrow$ OUT $\Rightarrow$ IN.
B: IN $\Rightarrow$ M1 $\Rightarrow$ M2 $\Rightarrow$ OUT $\Rightarrow$ M2 $\Rightarrow$ M3 $\Rightarrow$ IN.
C: IN $\Rightarrow$ M1 $\Rightarrow$ M2 $\Rightarrow$ M3 $\Rightarrow$ M1 $\Rightarrow$ M2 $\Rightarrow$ M3 $\Rightarrow$ OUT $\Rightarrow$ IN.
D: IN $\Rightarrow$ M1 $\Rightarrow$ M2 $\Rightarrow$ M3 $\Rightarrow$ M1 $\Rightarrow$ M2 $\Rightarrow$ M3 $\Rightarrow$ M1 $\Rightarrow$ M2 $\Rightarrow$ M3 $\Rightarrow$ OUT $\Rightarrow$ IN.
E: IN $\Rightarrow$ M1 $\Rightarrow$ M2 $\Rightarrow$ M3 $\Rightarrow$ M1 $\Rightarrow$ M2 $\Rightarrow$ M3 $\Rightarrow$ M1 $\Rightarrow$ M2 $\Rightarrow$ M3 $\Rightarrow$ M1 $\Rightarrow$ M2 $\Rightarrow$ M3 $\Rightarrow$ IN.

$\Rightarrow$ : Means that the robot moves X to Y.
$\rightarrow$ : Means that the part is moved otherwise.

3.2. Model (Ghoul 2003)

The model is given by the figure below.

3.3. Scheduling Algorithm

All the meta-heuristics are based on the same basic elements (neighborhood, memories, solution construction, choice strategy …). In the following, we present the most useful meta-heuristics in F.M.S scheduling problems.

3.4. Taboo search (Ghoul 2002)

This iterative and general method consists in a set of rules and mechanisms which role is to control and drive an intern heuristic. Its main particularity is the use of one or several memories, in order to keep the progress of each process. This memorized information will serve to forbid the exploration of a neighborhood already explored, and to avoid falling in a closed cycle that can conduct to a local optimum often different of the global one.

The Petri net-based modeling is used to describe the structure and the dynamic evolution of the F.M.S. The neighborhood of a solution is the set of all the reachable markings of a present state M (present marking).

- For each marking of the neighborhood, there is a set of fired transitions that represent challenges applied on the markings.
- All the research steps are memorized in the Taboo memory that contains all the transformations applied to the markings.
- The control mechanism allows determining the stop criteria of the research.
- In this example, the search is stopped when the
final marking is reached, that means the manufacturing process of the part type is achieved.

- Taboo memory is cyclic and bounded.

Let's have the present state \( M \), \( V(M) \) is the neighborhood of \( M \), we can reach \( M' \) from \( M \) by applying the evolution marking formula given in the following:

\[
M' = M + C(p,i) \times D
\]  

(4)

When \( C(p,i) \) is the incidence matrix and \( D \) the set of transitions fired by \( M \).

### 3.5. Genetic algorithms (Ghoul et al 2004]

#### 3.5.1 Definition

Genetic Algorithm was firstly invented by John Holland (1975) and his associates as the University of Michigan in the 1960. He is generally regarded as the father of GA although he did not give the process its name. GA is inspired by the mechanism of natural selection, a biological process in which the rule is "the fittest will survive ". It weeds out the bad and tends to produce more of the good individuals. Not only this produce more good solutions but butter and butter solutions. This is because it combines the best traits of parent "individuals" to produce superior "children". This combination operator is called crossover. The term GA derives from fact that individuals are represented as strings of bits analogous to chromosomes and genes. In addition to recombination by crossover, we also throw in random mutation of these bit-strings every so often. This prevents the GA from becoming trapped in finding good but non-optimal solution.

GA is inspired by the mechanism of natural selection where stronger individuals are likely to be the winners in a competing environment. GA uses a direct analogy of such a natural type of evolution. GA presumes a potential solution as an individual, which can be represented by a vector. This idea is familiar to biological applications, which can be termed as the genetic structure of chromosome. Throughout the genetic evolution, starting from a population of chromosomes, some fitter chromosomes tend to yield good quality offspring, and this means better solution to the problem.

#### 3.5.2 Chromosome syntax representation and population initialization

Representation of chromosomes is a crucial part of GA applications. In Holland's work, chromosomes are usually represented in binary strings. However, when problems are related to a real environment, binary encoding is not appropriate. Permutation representation and random keys representation are two of the most widely used methods to represent chromosome syntax for scheduling types of problems.

#### 3.5.3 Initial population generation

The population initialization technique used in this GA algorithm is a random real-number initialization. The procedure creates an initial population of GA filled with randomly generated real-number strings and permutation.

In our case, the initial population is composed of the 6 type parts A,B,C,d,E and F and the 4 resources M1, M2, M3 and R. Each type part is characterized by a set of variables (chromosomes) which are real numbers and can be: the type of the product, the job, number of failures on different resources, operating time of the following operations, residual operating time of each process, state of buffers and so forth.

### 3.5.4 Evaluation and fitness

An evaluation function plays a role similar to that which the environment plays in natural evolution, and it rates chromosome in terms of fitness. The multi-objective function of the F.M.S scheduling model represents the minimization of time spends in processing a number of jobs. Even the F.M.S is modeled by a high level Petri nets, the evaluation function is based on the specific Petri nets equation given in (4).

#### 3.5.5 Parent selection

The purpose of parent selection in GA is to choose parents and create offspring for subsequent generation in the population.

### 3.6. Simulated annealing

The method has been proposed under shape of the algorithm called "Métrropolis algorithm" that simulates energy modification of a material. The main idea consists in the analogy observed between a complex system optimization and a physical system behavior description.

This method is applied in several domains: operational research, imagery, production tools scheduling, and so forth. In this method, movements in research space are based on Boltzman distribution. This one, measure the probability of a system to be in a configuration \( C_i \) with an energy \( E(C_i) \), in a given temperature \( T \) in the space of configuration, which are defined by:

\[
\pi_i = \frac{\exp(-E(C_i)/K_T)}{\sum_{j \in U} \exp(-E(C_j)/K_T)}
\]

(5)

\( K \) is the boltzman constant \( K=1.3805.10^{-23}/k \).

In this expression, KT shows that when the temperature is high, all states have the same probability. It means that most of configurations are reachable. When temperature is low, states with high energy become less probable then those with low energy. Temperature have not equivalent in optimization problems. It is only a control parameter that indicates the system configuration. Energy represents the evaluation function or the fitness. The metropolis criterion is used to decide if the new configuration gives an acceptable variation of the fitness value.
4 Simulation results

4.1. Taboo search
(Card[T] is the Taboo memory length)

<table>
<thead>
<tr>
<th>Iterations Nb</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card[T]</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Stopping criteria</td>
<td>10</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Mean processing time</td>
<td>711</td>
<td>711</td>
<td>710</td>
</tr>
<tr>
<td>Mean productivity</td>
<td>1.14</td>
<td>1.26</td>
<td>1.18</td>
</tr>
</tbody>
</table>

4.2. Genetic algorithms

Cycles number: 10

<table>
<thead>
<tr>
<th>Machines Nb</th>
<th>Full flexibility</th>
<th>Partial flexibility</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Mean processing time</td>
<td>774</td>
<td>520</td>
</tr>
<tr>
<td>Mean productivity</td>
<td>0.69</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Cycles number: 15

<table>
<thead>
<tr>
<th>Machines Nb</th>
<th>Full flexibility</th>
<th>Partial flexibility</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Mean processing time</td>
<td>762</td>
<td>515</td>
</tr>
<tr>
<td>Mean productivity</td>
<td>0.76</td>
<td>1.26</td>
</tr>
</tbody>
</table>

Cycles number: 20

<table>
<thead>
<tr>
<th>Machines Nb</th>
<th>Full flexibility</th>
<th>Partial flexibility</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Mean processing time</td>
<td>736</td>
<td>504</td>
</tr>
<tr>
<td>Mean productivity</td>
<td>0.66</td>
<td>1.32</td>
</tr>
</tbody>
</table>

4.3. Simulated Annealing

<table>
<thead>
<tr>
<th>Mean productivity</th>
<th>Without failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>cycles Nbr</td>
<td>Deterministic</td>
</tr>
<tr>
<td>10</td>
<td>1.28</td>
</tr>
<tr>
<td>15</td>
<td>1.79</td>
</tr>
<tr>
<td>25</td>
<td>1.91</td>
</tr>
</tbody>
</table>

5 Comments

✓ The Taboo search method has given some good results. However, several problems have to be resolved. In some cases, the Taboo method prevents the system to reach an interesting solution because it is too restrictive. To solve this problem we use an Aspiration criterion which idea consists in removing the taboo statute associated to a transformation if this one permits to compute a best solution. Otherwise, when the research process browses a branch on a long period, we can vary this research by stopping research and beginning a new one on another random generated solution.

✓ In genetic algorithm method, if we compare simulation results of the two kinds of functioning (full and partial flexibility), we can do some remarks:
1. The genetic algorithm is a rather adaptive algorithm with any change of his environment like failure on machines, market requirements…
2. If cycles number increases, all the parameters improve. Indeed, the space of research increases and allows the algorithm to find the global optimum.
3. When the shop is full flexible, results are better than them obtained with partial flexibility. When a machine is broken down, operations are reallocated to another machine that can realize them. Then conflicts number increases and waiting states become more important, so that will increase the makespan. In absence of a row material, corresponding to a part type, ratios will be modified; consequently, the productivity will be reduced.

The optimal sequence research by the GA have given some important results compared with the other methods like taboo search. Some problems have to be considered, specially a best adaptation of genetic operators according to the characteristics of H.F.M.S.

✓ With the simulated annealing meta-heuristic, the evaluation function has an important effect. Productivity is best when the evaluation function is well chosen. The inconvenient of this method is the great probability to fall in a local optimum. The role of temperature parameter is to show the process evolution. More the process is advanced; less we accept a solution with a high cost. But, at the beginning, acceptance of a bad solution permits us to completely explore the solution space in order to reach the global optimum.
6 Conclusion

In this paper, we have attempted to realize a flexible scheduling algorithm, which takes account all types of events, variables, and machines. Different kinds of evolution, parallel evolution, deterministic, stochastic, full flexible, half flexible, with failure, without failure, with failure and repair..

We can see that the algorithms are adaptable because they can recover the lost time spending in repair and reach the maximal productivity.

7 References


