

The Hybrid Genetic Algorithm for Solving Scheduling Problems in a Flexible Production System

Jebari Hakim

National School of Applied
Sciences Laboratory of
Information Technologies and
Communication Tangier,
Morocco

Rahali El Azzouzi Saida

National School of Applied
Sciences Laboratory of
Information Technologies and
Communication Tangier,
Morocco

Samadi Hassan

National School of Applied
Sciences Laboratory of
Information Technologies and
Communication Tangier,
Morocco

ABSTRACT

In a world, which goes quickly, the company is subjected to the market evolution. Also and to cope with it, the system of production is directed towards families of products and not a single type of product. This aptitude requires a great flexibility as well material as organizational.

The problems associated with FMS technology is relatively complexes compared to traditional production systems. This is the reason why the problems scheduling in these systems are NP complete. Therefore, there is no algorithm able to solve these problems exactly.

The objective of this work is to solve the problem of scheduling in a flexible production system by the adaptation of the genetic algorithm and the hybrid genetic algorithm - using the simple local search and the annealing simulate - in order to deduce the best Meta heuristic, which provides the best result of makespan.

General Terms

Scheduling, flexible production system, genetic algorithm.

Keywords

Scheduling, flexible production system, genetic algorithm, hybrid genetic algorithm, local search, annealing simulate.

1. INTRODUCTION

To increase the productivity by reducing the costs is, today, a major target in all the companies. The systems of production are characterized by their dynamism and their imprevision. The complex character of tasks are often subjected to constraints of time and requirement. Then, the system of production gives itself new objectives to be achieve.

In the area of industry where competition is fierce, large production units are becoming less and less competitive, because very few are adaptable to market variations. These units disappear gradually giving way to new industrial structures to different concepts: The Flexible Manufacturing Systems « FMS ». These systems are designed to produce small and medium quantities, at minimum cost, a variety of products with machine preparation times and changing tools minima. Their structures allow them to produce a very wide range of products and therefore be less subject to economic crashes.

The purpose of a flexible system of production is to achieve, not only in high productivity, but also with a great flexibility of production allowing him to follow the variations of the market.

The problem of selection of the optimal routing consists in determining the best routing while minimizing the

manufacturing costs by taking into account volume of production, the alternatives associated with the process of setting in production, as well as availability and the capacity of the machines.

2. THE FLEXIBLE PRODUCTION SYSTEMS

The competition between companies has led to seek to improve production while meeting delivery deadlines. To meet this goal, the production systems must be flexible and thus emerged the Flexible Production Systems (FMS).

The FMS are highly automated production systems and adapted for manufacturing small and medium series of products with different ranges and this at low cost. The literature offers various definitions of FMS. Retaining McCarthy and Liu (1993) which have the advantage of taking into account the process of production and storage.

A production system capable of producing different types of parts is consisting of CNC machines and an automated storage system connected by an automated handling system. The operation of the entire system is under the control and management of a computer system. [1]

A flexible production system is characterized by three main sets [2]:

- The means of production and storage.
- The transportation / shipping.
- The topology and the type of the guidance system of the transport system / handling.

The FMS are dynamic systems. They consist of a set of entities whose state can change in real time, for example, a resource that is capable of producing a piece can move to a non-productive state after the machining of the piece .

The dynamic is also realized by the interaction between these entities. To ensure the purpose of production, the production system must be provided in addition to the flexibility.

2.1. Flexibility of the Production Systems

In its general definition, flexibility reflects the ability of a system to change easily and cheaply to adapt to circumstances. Flexibility can be defined by all the properties and qualities of a manufacturing system that can withstand changes in the types of products and production capacity. These changes may be:

- **Internal:** such as equipment failures, failures of computer systems (management software), worker absenteeism, changes in production time ...

- **External:** as the change in the product design, product design complexity, variation in demand ...

To absorb the uncertainties caused by product design changes, the manufacturing system must be flexible and able to manufacture different types of products with a manufacturing time and minimum cost. Therefore, you could say that flexibility is the ability of manufacturing systems to meet both internal and external changes [3].

2.2. Classification of Flexible Systems of Production

There are many classifications of flexible production systems. The study of the literature shows that the classification criteria are varied for example:

- The workshop topology, the manufacturing method (cyclic / acyclic, unit / lot...) [4].
- The size and the complexity of the machines [5].
- The pieces flow models to treat [6].
- The operating characteristics and control [7].

[2] Provides a simplified classification into four classes, this classification is done according to the operating characteristics and control of FMS:

2.2.1. Single Flexible Machine

SFM (Single Flexible Machine) is a production unit that contains computer-controlled machine with a unique ability to tool change, an element of handling and internal stock. The handling system in a SFM can be a robot or a device of change of pallets specialized. When a SFM is used like a component of a great system, its device of handling can be eliminated if its function can be realized by the handling systems of the great system.

2.2.2. Flexible Manufacturing Cell

An *FMC (Flexible Manufacturing Cell)* is a group of SFM connected by only one element of handling, with or without intermediate storage spaces located as starter or at exit of each SFM.

2.2.3. Multi-Computer Flexible Manufacturing System

An *MMFMS (Multi-Computer Flexible Manufacturing System)* is a group of SFM connected by an automated handling system, which contains two or several carriages. The systems of the type MMFMS are thus able to serve several machines simultaneously, with or without storage spaces intermediary, which can be in Input/output of each SFM.

2.2.4. Multi - Cell Flexible Manufacturing System

An *MCFMS (Multi - Cell Flexible Manufacturing System)* constituted of several FMC and possibly of SFM connected by an automatic transportation system.

3. SCHEDULING PROBLEMS IN A FLEXIBLE PRODUCTION SYSTEM JOB-SHOP

In a problem of scheduling of type flexible job-shop, a given operation being able to be realized by one or more resources, has one duration of treatment depend on the used resource. The relative constraints to this kind of problem are of standard precedence, temporal or disjunctive.

The problems of scheduling of workshops of type flexible job-shop constitute an extension of the problems job-shop and are, so even more difficult to solve [8].

The difficulty lies in choosing the best Meta heuristic for their resolution and for the determination of the best schedules in reasonable time possible, nearest to the optimal solution. Several authors have focused on solving the flexible job-shop scheduling problems.

Numerous approaches have been proposed for solving scheduling problems in flexible production systems. To this end, several classifications of these approaches exist in the literature. Kusiak and Ahn [9] distinguish three types of approaches, which are the approaches of operations research, the approaches based on artificial intelligence and the combination of both approaches. A classification proposed by Spano et al. [10] includes two categories: traditional approaches and approaches based on artificial intelligence. Indeed, the traditional approach can be divided into two categories: theoretical research on procedures optimization and experimental research. Theoretical research has focused on the development of mathematical models and optimal algorithms or sub optimal. Experimental research has focused on the priority rules and heuristics that can effectively solve scheduling problems. This classification is adopted by Chan and Chan [11] who were interested in the techniques of artificial intelligence, they have them classified into several categories which include in particular fuzzy logic, expert systems, neural networks, algorithms genetic but they ignored some approaches such as those based on multi-agent systems and other Meta heuristics such as ant colonies, the particle swarm...

Some authors have thought that the simulation can be regarded as a scheduling approach in FMS. Basnet and Mize [12] described the methodology of scheduling under different categories: mathematical programming, multi-criteria decisions, oriented heuristics, the theoretical verification, simulation and artificial intelligence. Joseph and Sridharan [13] reported that scheduling in FMS could be classified into analytical methods, heuristics, simulation, expert systems / artificial intelligence techniques and hybrid methods. The classification proposed by Balogun and Popplewell [14] distinguishes several categories based scheduling in FMS, combinatorial optimization, priority rules based on the simulation oriented heuristics, multi-criteria decision, hybrid methods and techniques artificial intelligence represented by several categories such as expert systems, neural networks, genetic algorithms and other techniques from artificial intelligence.

Other researchers are interested not only in dynamic scheduling problems, Suresh and Chaudhuri [15] distinguished three types of approaches: conventional (optimization, heuristics ...) approaches based on knowledge (e.g. expert systems ...) and distributed approaches such as multi-agent systems. Based on this classification and others (Shukla and Chen [16], Weirs Stoop [17], Brandimarte and Villa [18]), Ouelhadj and Petrovic [19] have developed a classification for dynamic scheduling in a FMS. Seven classes are then formed: heuristics, Meta heuristics, the knowledge based systems, fuzzy logic, neural networks, hybrid techniques and multi-agent systems.

Other classifications exist; for example, Letouzey [20] classified the methods for solving scheduling problems according to their way of solving the problem. Two categories are distinguished, direct methods solving the problem in a

single pass and iterative methods to produce a solution and then try to improve.

4. METHOD OF RESOLUTION PROPOSED

The goal of this work is to prepare a detailed study of method of adaptation of the Meta heuristic algorithms, which are the genetic algorithm and the hybrid genetic algorithm, for the resolution of the problem of scheduling in a flexible system of Job-Shop production.

Thereafter, one will proceed to make a comparison of the got results. The discounted objective of this comparison is to approach to Meta heuristic, the most appropriate to the resolution of the problem of scheduling.

4.1 Formulation of FJSP Problems

The problems of workshops of job-shop flexible can be formulated as follows [21]:

- Consider a set of n independent products to be realized on m machine $M_k, k = 1, 2, \dots, m$.
- Each product J_j consists of a sequence of n_j operations $O_{i,j}, i = 1, 2, \dots, n_j$ to run in a specific order,
- The execution of each operation i of a job J_j require a resource selected from a set of available machines,
- Each machine can carry out only one operation at a time,
- The assignment of an operation $O_{i,j}$ to a machine M_k involve the occupation of this machine during all the time of execution of the operation, noted $P_{i,j,k}$,
- Pre-emption is not authorized.
- Either $D^k(x)$, the completion date of the last operation realized on the machine k according to the scheduling x . The makespan $C_{\max}(x)$ of the scheduling x is defined by:

$$C_{\max}(x) = \max D^k(x) \quad 1 \leq k \leq m$$

The makespan is the total duration necessary to the manufacturing of all the products of the instance considered according to the fabrication order imposed by the scheduling x .

A solution that checks the constraints of the problem is said admissible. The main objective is to find an acceptable scheduling, which makes it possible to minimize the value of the makespan.

The FJSP presents two principal difficulties:

- The first relates to the assignment of each operation $O_{i,j}$ to a machine M_k ,
- The second is to calculate the start time $t_{i,j}$ and the end time $tf_{i,j}$ of the operation $O_{i,j}$.

4.2 Adaptation of Genetic Algorithm

The genetic algorithms result from a comparison between certain problems of optimization and the theory of the evolution of Darwin. This theory, supplemented recently by modern biology, puts in scene populations to be alive subjected to a natural selection and confronted with living conditions more or less favorable. The individuals are then forced to evolve from generation to generation, in order to

adapt to the medium under penalty of the extinction of the species.

Using a genetic algorithm requires the definition in advance of a search space whose basic elements are the chromosomes and a function defined on this space (fitness function) whose optimal value is assessed in relation with crossover operators and mutation operators chosen [22].

Five basic elements are necessary for the use of genetic algorithms [23]:

- A principle of coding of the elements of the population.
- A mechanism of generation of the initial population.
- A function to optimize, called fitness or function of evaluation of the individual.
- Operators allowing to diversify the population during generations and to explore the space of state.
- The parameters of dimensioning, represented by the size of the population, the full number of generations, or the criterion of stop, as well as the probabilities of application of crossover and mutation operators.

4.2.1 Proposed coding

A solution or an individual i , which composes this population, is coded by a matrix of sequence of operations, which defined the order of the operations to be performed by each machine as shown in Table I.

Table 1. Coding

		Operations		
Machines	M1	$O_{i,j}$
	M2	$O_{i',j'}$
	M3	$O_{i'',j''}$

The solutions constituting the initial population are randomly chosen in this study.

The algorithm for generating the initial population is the following:

```

Fix size  $N$  of the initial population
Generate a first solution
 $k=1$ 
While  $k \neq n$  Do
  Test the realizability
  If solution without cycle Then
    If non existing solution Then
      Solution  $K$  accepted
       $k = k + 1$ 
    If Not
      Refused solution
  End If

```

If Not

Refused solution

End If

Generate another solution.

End While

4.2.2 Proposed selection

The selection phase is to choose among the N individuals i of the current population the stronger individuals from which the next generation will be created. Either for a realizable schedule i , the value of the objective function f_i "Makespan" is calculated, to evaluate the long way on the disjunctive graph of the current problem. An evaluation function is associated with each solution to calculate its adaptive strength F_i . In the study case, the problem is to minimize the objective function $\min f(i)$, then the individual's strength i (fitness) can be simply expressed by $F_i = 1/f_i$. The individuals thus selected constitute an intermediate population.

The algorithm proposed is as follows:

For $i = 1$ to N **Do**

Generate an individual i (realizable scheduling)

Calculate Makespan_individual i

Calculer fitness_individual i

Storage

Calculate Probabilite_selection P_i

Generate a random number x , $x \in [0, 1]$

If $x < P_i$ **Then**

Select individual i

End If

End For

In this study, the method used is the selection by Rank (Rankine): assigning to each individual a rank R_i which depends on its fitness. The individuals of a population are then classified in a list according to the order ascending of their fitness. Each individual i is associated to the probability P_i to be selected. The selection will be proportional to their rank in the list of the population: $p_i = R_i / \sum_{i=1}^N R_i$.

4.2.3 Proposed crossover

The crossover operator is the largest in the AGs, it can efficiently explore the research space. The crossover operator applied to both individual parent1 and parent2 chosen from the selected population, generates two new solutions child1 and child2 by the combination of properties of parent 1 and parent 2.

The crossover algorithm is as follows:

Select randomly two parents indiv1 and indiv2

Select randomly a machine k from m

Cross:

For $i = 1$ to m **Do**

If $i \neq k$ **Then**

enf1 receives the same assignments that indiv1

enf2 receives the same assignments that indiv2

If not

enf1 receives the assignments of indiv2

enf2 receives the assignments of indiv1

End If

End For

Test the realizability

If enf1, enf2 without cycle **Then**

enf1 and enf2 accepted

If Not

enf1 and enf2 refused

End If

The method implemented in this study is to randomly choose a machine k and applying a cross between the sequences of operations associated with this machine only. The sequencing of operations on other machines is identical to the son and then tested realizability of obtained solutions, those who do not have cycle will be accepted.

4.2.4 Proposed mutation

The second important genetic operator is mutation; it comes in second place on the level of importance compared to the cross. A mutation is a disturbance introduced on the individual component to ensure diversity and expand the field of exploration.

The approach taken in this work consists in choosing randomly an individual and the machine must undergo this perturbation, and then randomly choosing two operations; the operation is to switch the order between them.

The algorithm of the mutation is as follows:

Select randomly an individual i

Select randomly a machine k from m

For $h = 1$ to m **do**

If $h \neq k$ **Then**

Indiv mutated receives the same assignments as indiv i

If Not

Select randomly two operations $O_{i,j}$ et $O_{i',j'}$

Swap the order of operations $O_{i,j}$ et $O_{i',j'}$

End If

End For

Test the realizability

If solution without cycle **Then**

Indiv mutated accepted

If Not

Indiv mutated refused

End If

4.2.5 Proposed replacement

This step represents the population of the next generation from the parents and children of the current generation. Its progeny replaces a fraction of the population in each

generation. The generation gap indicates the proportional parents replaced by children.

The generation gap indicates the proportional parents replaced by children.

The replacement algorithm is as follows:

Storing the child population

Storing the parents population

Replacement:

Select the N best individuals from children and from parents

4.2.6 The stop criterion

The process is stopped after a fixed number of generations, the best individuals of the population are then selected as solutions to the problem.

5. ADAPTATION OF HYBRID GENETIC ALGORITHM

The genetic algorithm has no evidence of convergence to the global optimum [24] and it suffers from the mechanism of selection centralized and overall which makes quickly a population homogeneous and can lead of this fact to a premature convergence.

Hybrid algorithms are introduced for the first time in 1989 by Moscato. They hybridize the genetic algorithms with local search methods.

These two methods are complementary because one makes it possible to detect good areas within the space of research whereas the other concentrates in an intensive way to explore these zones of the space of research [25] with an aim of the improvement of the performance of the genetic algorithm.

5.1 Descent Method

From a solution generated initially or found by heuristics for example, one can implement methods of descent. These methods articulate all around the same principle. To start from an existing solution, find a solution in the vicinity and to accept this solution if it is better than the current solution.

The algorithm below shows the skeleton of a descent method:

Initialization: find an initial solution X

Repeat

Search in the vicinity: Find a solution $x' \in N(x)$

If $f(x') < f(x)$ **Then** $x \leftarrow x'$

End If

Until $f(y) \geq f(x), y \in N(x)$

5.2 Population Management

The management of population is a procedure having for goal measurement of distance within the space of solutions and the control of the diversity of the population considered.

As shown in the algorithm below, a solution which does not have a sufficient distance for the population, is mutated randomly until it achieved this distance. Of course, other strategies are possible, like quite simply rejecting the solution [26].

Initialization: generate an initial population P of solutions

Fix the diversity parameter Δ

Repeat

Selection: Select two solutions p_1 et p_2

Crossover: Combine the two parent's solutions p_1 and p_2 to form solutions c_1 and c_2

Local search: apply a local search procedure on c_1 and c_2

For c any child **Do**

Until C does not satisfy the condition of addition,
($d_p(c) < \Delta$)

Do Mutation (c)

End Until

Delete the solution of population

Addition of solution: $p \leftarrow p \cup c$

End For

Update the diversity parameter Δ

Until satisfy a stopping criterion (fixed number of generation)

Δ is initialized with a low value and then increased over the iterations; this permits diversification in the end of the algorithm.

5.3 Simulated Annealing

Starts by an acceptable initial solution and continues the exploration of the state space by performing minor disruptions to the current solution. If the neighbor solution obtained improves the criterion sought then it is retained. if on the contrary, it causes a degradation ΔE of the criterion, it is retained with a probability $e^{-\frac{\Delta E}{T}}$, where T is a parameter inversely proportional to the number of iterations. The diversification is thus privileged at the beginning of process, intensification being then dominant.

The simulated annealing able to provide satisfactory results as soon as one knows how to find a qualified initial solution and a way to explore neighboring solutions. The fact of keeping the less good solutions makes it possible to cover a larger space of research and to avoid a premature convergence towards a local optimum.

The algorithm is the following :

Select $s \in S$ solution and an initial temperature T

While a stopping criterion is satisfied **Do**

Choose Randomly $s' \in N(s)$

Generate a random real number R in [0.1]

If $r < p(t, s, s')$ **then** put $s \leftarrow s'$

Update T

End While

The function $p(t, s, s')$ is generally chosen as equal to the Boltzmann distribution $e^{-\frac{f(s)-f(s')}{T}}$. Thus,

- If $f(s') < f(s)$ then $e^{-\frac{f(s)-f(s')}{T}} > 1$, this means that $r < p(t, s, s')$ and therefore s' will be accepted.

- If T has a very high value then $e^{\frac{f(s)-f(s')}{T}} \cong 1$, and therefore it is almost certain to accept s'.
- If T has a very small value and if $f(s') > f(s)$ then $e^{\frac{f(s)-f(s')}{T}} \cong 0$ and so s' will probably be refused.

The hybrid genetic algorithm with the simulated annealing runs in two phases.

In the first one, the genetic algorithm randomly generates a set of initial solutions. It operates on the solutions by using the selection, crossover and mutation operators to produce eventually the better solutions. In the second one, after each generation, the genetic algorithm sends each solution to the simulated annealing to be improved. The best solutions obtained from simulated annealing are the genetic algorithm solutions for the next generation. The exchange between them continues until the stopping criterion is satisfied.

6. EXPERIMENTATION AND RESULTS

This article is interested to the case of production of metal parts for the automotive industry plant to simulate the operation of genetic algorithms and the hybrid genetic algorithm on the flexible job shop problem.

Five machines compose the workshop:

- Mach1: Cutting machine.
- Mach2: Thermoforming machine.
- Mach3: Welding machine.
- Mach4: Folding machine.
- Mach5: Punching machine.

One conveyor Conv for transferring parts between the machines, in both directions, and six robots:

- Rob in-out for loading and unloading of conveyor Conv.
- Rob1, Rob2, Rob3, Rob4, Rob5 for the supply of the machines Mach1, Mach2, Mach3, Mach4, Mach5.

The problem considered is the JSPF (5 x Njob), which represents the execution of N jobs on 5 machines, for each job j is associated 5 operations, operations are numbered by integers, the operative durations and the machines used to execute the operations are provided by the following functions:

$$\forall j = 1,2,3,4,5 \text{ Job}_j = \{O_{1,j}, O_{2,j}, O_{3,j}, O_{4,j}, O_{5,j}\},$$

$$\text{affect randomly } \{O_{1,j}, O_{2,j}, O_{3,j}, O_{4,j}, O_{5,j}\} \forall j = 1,2,3,4,5 \text{ to } \left\{ \begin{array}{l} \text{Mach}_1 \text{ OR } \text{Mach}_2 \text{ OR } \text{Mach}_3 \\ \text{OR } \text{Mach}_4 \text{ OR } \text{Mach}_5 \end{array} \right\}$$

$$\forall j = 1,2,3,4,5; \forall K = 1,2,3,4,5; \forall i = 1,2,3,4,5$$

$$\text{Duration}(O_{i,j}[\text{Mach}_k]) = \text{random}[1,20]$$

if there is an assignment to $O_{i,j}$ at $[\text{Mach}_k]$

These values are then stored in a file for their usage in three programs, they are expressed in minutes.

The parameters of the three programs are:

- The initial population size = 150
- Probability of crossover = 0.7
- Mutation probability = 0.01
- Number of generations = 500
- Initial temperature = 3
- Final temperature = 0.05
- Temperature decreasing rate = 2

Three algorithms were programmed in Java language and executed in Core™ i5 CPU with 2.1 GHz and 4 Go of RAM.

The result obtained is the following:

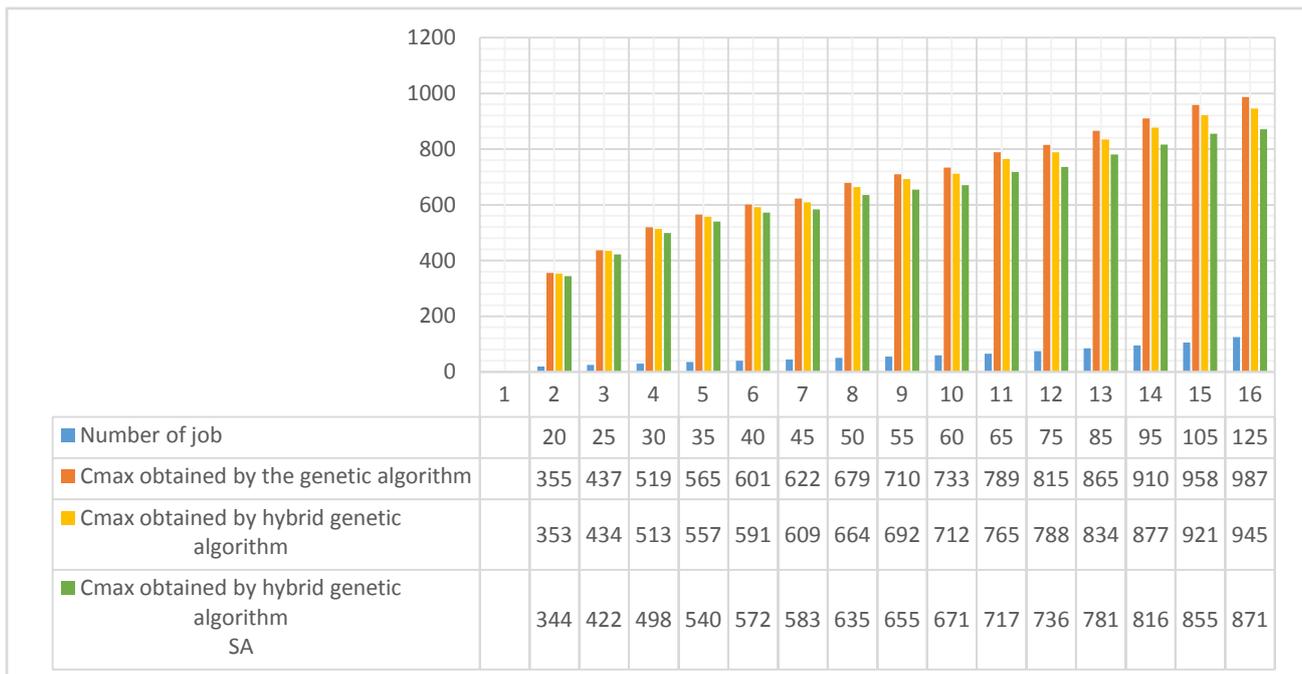


Fig 1: Final results

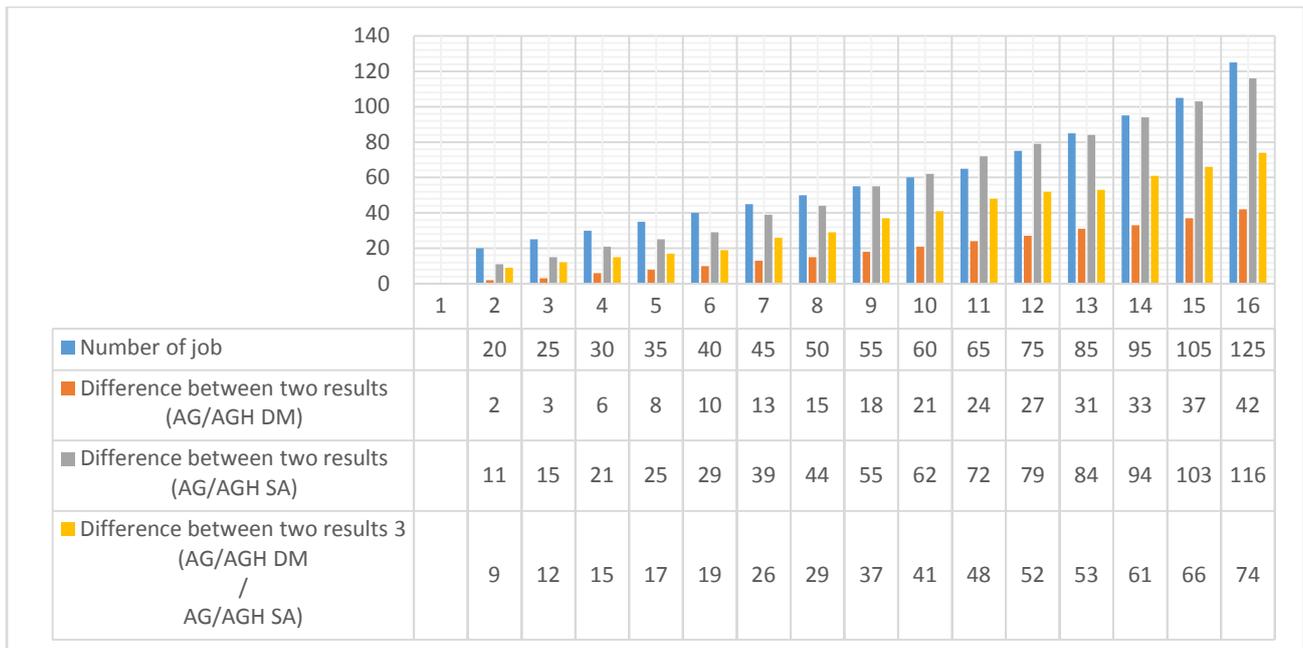


Fig 2: Comparison of the results

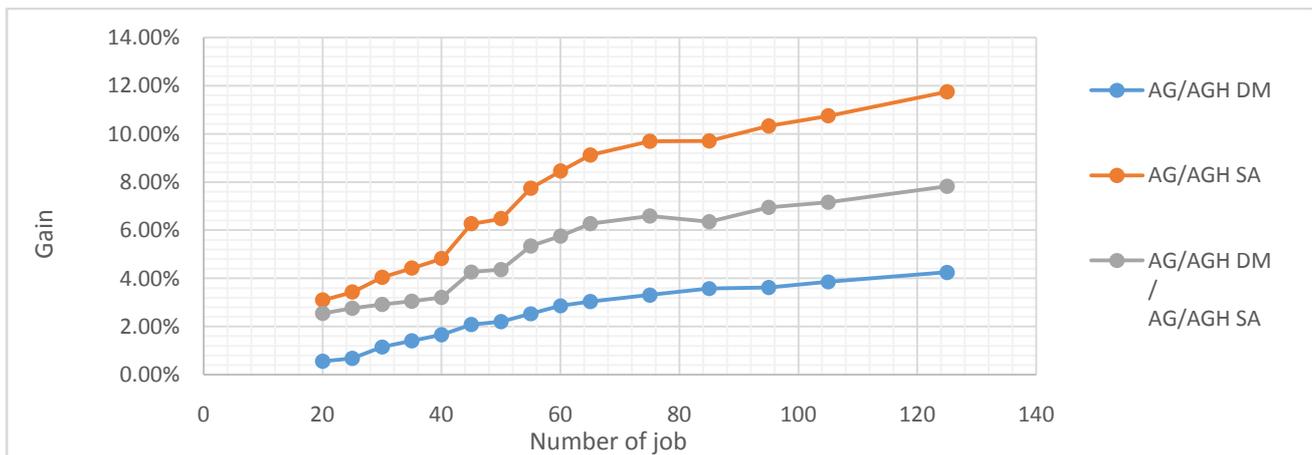


Fig 3: Comparison of the gain

The hybridation of the genetic algorithm with local search has improved the quality of obtained solutions by the genetic algorithm to solve the problem of scheduling in a flexible production.

The use of hybrid genetic algorithm with the simulated annealing has confirmed the capability of this method to achieve the best solution compared with the solution obtained by the hybrid genetic algorithm with the local search.

7. CONCLUSION

Scheduling in a flexible production system is an extremely difficult problem to solve. The Meta heuristic was adapted to solve it; the Meta heuristics are the genetic algorithm and the hybrid genetic algorithm with local search.

The hybrid approach between the genetic algorithms and the procedure of local search improve the solutions that the genetic algorithm can propose, the results of hybrid genetic algorithm with simulated annealing are better than the results of hybrid genetic algorithm with methods of descent and the management of the population and better than the result of genetic algorithm.

This work can spread out to study, to develop and to exploit other Meta heuristics; to compare their respective results in order to solve the problem of scheduling in a flexible production system.

8. REFERENCES

- [1] MacCarthy B.L. et Liu J, Anew Classification Scheme For Flexible Manufacturing Systems. International Journal of Production Research, 31(2), 299-309, 1993.
- [2] Brauner N.et Castagna P., Les Systèmes Flexibles de Production. Journal Européen des Systèmes Automatisés, 2004.
- [3] Adamou, M., Contribution à la modélisation en vue de la conduite des systèmes flexibles d'assemblage à l'aide des réseaux de Petri orientés objet. Thèse de doctorat, Université de Franche-Comté, 1997
- [4] Rembold U., Nnajy B., Storr A., Computer Integrated Manufacturing and Engineering. Addison-Wesley, 1993.
- [5] Kusiak A., Flexible manufacturing systems: a structural approach. Int. J. of Pro. Res, vol. 23, p. 1057-1073, 1985.

- [6] Browne J., Dubois D., Rathmill K., Sethi S., Stecke E., Classification of Flexible manufacturing systems. *FMS Magazine*, vol. 2, p. 114-117, 1984.
- [7] Maccarthy B., Liu J., A New Classification Scheme For Flexible Manufacturing Systems. *International Journal of Production Research*, vol. 31, no 2, p. 299-309, 1993.
- [8] R. Garey et D.S. Johnson, « Computers and Intractability: A guide to the theory of P-completeness ». Editions Freeman and Co, 1979.
- [9] Kusiak, A. and Ahn, J. (1992). Intelligent scheduling of automated machining systems. *International Journal of Computer Integrated Manufacturing Systems*, 5(1), 3-14.
- [10] Spano, M. R., O'Grady, P. J. and Young, R. E. (1993). The design of flexible manufacturing systems. *Computers in Industry*, 21, 185-198.
- [11] Chan, F.T.S. and Chan, H. K. (2004). A comprehensive survey and future trend of simulation study on FMS scheduling. *Journal of Intelligent Manufacturing*, 15(1), 87-102.
- [12] Basnet, C. and Mize, J. H. (1994). Scheduling and control of flexible manufacturing systems: a critical review. *International Journal of Computer Integrated Manufacturing*, 7(6), 340-355.
- [13] Joseph, O.A and Sridharan, R. (2012). Effects of flexibility and scheduling decisions on the performance of an FMS: simulation modelling and analysis. *International Journal of Production Research*, 50(7), 2058-2078.
- [14] Balogun, O.O., Popplewell, K. (1999). Toward the integration of flexible manufacturing system scheduling. *International Journal of Production Research*, 37(15), 3399-3428.
- [15] Suresh, V. and Chaudhuri, D. (1993). Dynamic scheduling a survey of research. *International Journal of Production Economics*, 32(1), 53-63.
- [16] Shukla, C. S. and Chen, F. F. (1996). The state of the art in intelligent real-time FMS control: a comprehensive survey. *Journal of Intelligent Manufacturing*, 7(6), 441-455.
- [17] Stoop, P. P. M. and Weirs, V. C. S. (1996). The complexity of scheduling in practice. *International Journal of Operations and Production management*, 16(10), 37-53.
- [18] Brandimarte, P. and Villa, A. (1999). Modelling manufacturing systems: from aggregate planning to real-time control. Springer Berlin.
- [19] Ouelhadj, D. and Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12 (4), 417-431.
- [20] Letouzey, A. (2001). Ordonnement interactif basé sur des indicateurs : Applications à la gestion de commandes incertaines et à l'affectation des opérateurs. Thèse de doctorat, L'Institut National Polytechnique de Toulouse, France.
- [21] Chu, C., Proth J. M. « L'ordonnement et ses applications » série : sciences de l'ingénieur, Collection organisation industrielle, Masson, 1996.
- [22] S.K. Iyer et B. Saxena, « Improved genetic algorithm for the permutation flowshop scheduling problem ». *Computers and Operations Research*, vol. 31, pp. 593-606, 2004.
- [23] N. Durand, J.M. Alliot et J. Noailles, « Algorithmes génétiques : un croisement pour les problèmes partiellement séparables ». Journées Evolution Artificielle Francophones, JEAFF, Toulouse, 1994.
- [24] Hoos H.H. and Stützle T., *Stochastic local search: Foundations and applications*. Morgan Kaufmann, 2004.
- [25] Moscato P., *New Ideas in Optimization*, chapter Memetic Algorithms: A Short Introduction. McGraw-Hill, 1999.
- [26] Sörensen K., Sevaux M., MA|PM: memetic algorithms with population management, aUniversity of Antwerp, Faculty of Applied Economics, Prinsstraat 13, B-2000 Antwerp, Belgium bUniversity of Valenciennes, CNRS, UMR 8530, LAMIH-SP, Le Mont Houy-Bat Jonas 2, F-59313 Valenciennes cedex 9, France, 2004.