Meta-heuristics for real time routing selection in Flexible Manufacturing Systems (FMS)

M. Souier, A. Hassam, Z. Sari

Tlemcen Automatic Laboratory ,
Abou bekr Belkaïd University of Tlemcen,
PoBox 230, Tlemcen 13000, Algeria
{m_souier, a_hassam, z_sari@mail.univ-tlemcen.dz}

Abstract

Most studies in real-time Flexible Manufacturing System (FMS) scheduling and control area do not consider the effect of routing flexibility; their focus is typically on use of scheduling (i.e., dispatching) rules based on routing selection carried out prior to production. Such an approach is not applicable to random-type FMS, in which no knowledge about incoming part types is available prior to production. For such a scenario, parts can have alternative routings, even for parts of the same type. Thus, the control system of a random-type FMS requires the capability to adapt to the randomness in arrivals and other unexpected events in the system by effectively using operation and routing flexibility in real-time. In this chapter, the objective is to present a comparative study of a group of meta-heuristics, including tabu search (TS), ant colony optimization (ACO), genetic algorithms (GA), particle swarm optimization (PSO), electromagnetic meta-heuristic (EM), and simulated annealing (SA), against the Modified Dissimilarity Maximization Method (Modified DMM)**.** DMM **(Saygin and Kilic 1999)** is an alternative process plan selection method originally proposed for the routing selection in off-line scheduling of an FMS. In subsequent studies **(Saygin, Chen, and Singh, 2001) and (Saygin, Chen, and Singh, 2004)** DMM has been: (i) used as a real-time decision-making tool to select routings for the parts that are in the system, (ii) tested and benchmarked against First-in-First-out/First Available (FIFO/FA) and Equal Probability Loading (EPL). Based on the DMM model, a modified DMM **(Hassam and Sari 2007)** is developed for selection of alternative routings in real time in an FMS. Modified DMM method improves the performances of the FMS in terms of higher production rate, higher utilization rate of the machines and the material handling system.

8.1 Introduction

Nowadays, businesses are facing increased competition; pressure for higher variety of customized products, shorter lead times, higher quality, and lower cost due to competitors. Today, being flexibility in production and effectively managing it to be more competitive is more crucial than ever. Several decades after its conception, flexible manufacturing systems (FMS) still provide great flexibility. They provide various benefits, such as high resource utilization, high productivity, reduced work-in-progress, and many more.

In such systems, resource allocation decisions, process planning, and scheduling of operations are generally made dynamically and in a very short time almost in real-time, depending on the state of the production system (availability of resources, availability of the system handling, the presence of bottlenecks), the characteristics of the production plan (due date of manufacturing orders) and the production targets (production rate increase, reduce work in process).

The real-time scheduling of operations uses multiple approaches, such as selection of parts in buffers for immediate machining and selection of machines for a part among alternative machines by priority (i.e., scheduling or dispatching) rules, which is one of the simplest and most commonly used methods. These priority rules have been studied for many years. See **Saygin and Kilic, 1999** for an extensive literature survey.

Among the rules and methods of scheduling in real time, we can find the Dissimilarity Maximization Method (DMM) **(Saygin et *al.* 2001)**, **(Saygin and Kilic 2004)** which is a rule for selecting alternative routing in real time in an FMS based on its original version for off-line routing selection (**Saygin and Kilic, 1999**), and the Modified DMM **(Hassam and Sari 2007)** which is an improvement of the DMM rule in order to improve the performances of the production system. These methods use coefficients of dissimilarity between the machines to make decisions related to routings..

The scheduling problems in manufacturing systems are generally NP hard and there are not universal methods making it possible to solve all the cases effectively **(Garey and Johson 1979)**.

Meta-heuristics are the algorithms of the stochastic type aiming to solve a broad range of hard optimization problems, for which one does not know more effective traditional methods. Often inspired by analogies with reality, such as physics (simulated annealing, simulated diffusion,) biology (evolutionary algorithms, taboo search,) and ethnology (ant colony, swarm intelligence). They are generally of discrete origin, but can be adapted to the other types of problems and they share also the same disadvantages: difficulties of parameters adjustment and large computation time.

In this chapter, our interest is focused on a group of meta-heuristics, which include in particular the simulated annealing (SA), genetic algorithm (GA), taboo search (TS), ant colony algorithms (ACO) and particle swarm optimization (PSO) and electromagnetism like method (EM); we are going to present a comparative study between these meta-heuristics and the modified DMM rule.

We will see that the meta-heuristics are largely based on a common set of principles, which make it possible to design solution algorithms, the various regroupings of these principles lead thus to a large variety of meta-heuristics.

8.2 Literature review

The scheduling problems are usually NP hard. One of the first studies of the scheduling of FMS is the work of **(Nof *et al.,* 1979)** where they demonstrate the importance and effect of scheduling decisions on various performance measures of production systems. Traditional scheduling involves sequencing of operations and time allocation on their start and end times before the production starts. Traditional scheduling requires that production orders in terms of part types and their routings are known prior to production. On the other hand, real-time scheduling is carried out as a control activity, which involves real-time decision making in terms of selection of part types and their routings as parts come in to the production system. This category of scheduling problems involves various challenges, such as variable part arrival rates, unexpected breakdowns, need for synchronization of tool management, effective management of material handling systems, and lack of raw materials.

The factors listed above and many others, makes reordering required so that to avoid the increase in waiting time, the increase in work in process, the low utilization of machinery and equipment and possibly the degradation of the production system performances **(Wu and Wysk, 1989)**, **(Ishii and Muraki, 1996)**.

Several researchers propose different methods to provide maximum flexibility in real time scheduling in order to increase the performance of systems **(Saygin and Kilic, 1999) (Liu and MacCarthy, 1997)**, **(Saygin and Kilic, 1996)**. However, the real-time scheduling is always desirable but elusive goal **(Basnet and Mize, 1994)**, **(Shukla and Chen, 1996)**.

Consequently, establishing an integrated system for real-time scheduling and control that responds to changes in the state of the system is essential to improve the performance of a production system.

The control and real-time scheduling of flexible production systems have become a popular research area since the early 1980's, a period in which flexible production systems were adopted by the industrialized countries **(Saygin *et al*, 1995)**, **(Saygin and Kilic, 1997)**, **(Peng and Chen, 1998)**. But many studies in controlling and scheduling of FMS in real time do not take into account the flexibility of alternative routing **(Byrne and Chutima, 1997)**, **(Kazerooni *et al*, 1997)** and most studies that take into account this point , handle the problem of routing selection prior to the start of production **(Das and Nagendra, 1997)**, **(Cho and Wysk, 1995)**.

Scheduling rules have been studied by many researchers. The common conlcusion among these studies includes 1) results are dependent on the production system that has been studied, therefore cannot be generalized, 2) these rules are myopic in nature, therefore they lead to imperfect scheduling since they do not capture the relevant information at various levels of production systems **(Rachamadugu and Stecke, 1994)**, **(Gupta *et al.*, 1989), (Kouiss *et al.*, 1997)**.

The weakness of these scheduling rule-based approaches in handling real time scheduling in FMS has been the major driving force behind the development of new methods for alternative routing selection in real time.

8.3 Part Routing

8.3.1 General Job Shop Models:

In a job shop, each job has its own predetermined route to follow. The simplest job shop models assume that a job may be processed on a particular machine at most once on its route through the system (see Figure 8.1). In others, a job may visit a machine several times on its route through the system. These shops are said to be subject to recirculation, which increases the complexity of the model considerably. The routes of the jobs are order-specific and require recirculation. More general models assume a production environment that consists of a network of interconnected facilities with each facility being a (flexible) flow shop or a (flexible) job shop. At a higher level, supply chain managements also makes use of such planning and scheduling of networks for streamlining supply and demand among the business partners and end customers.

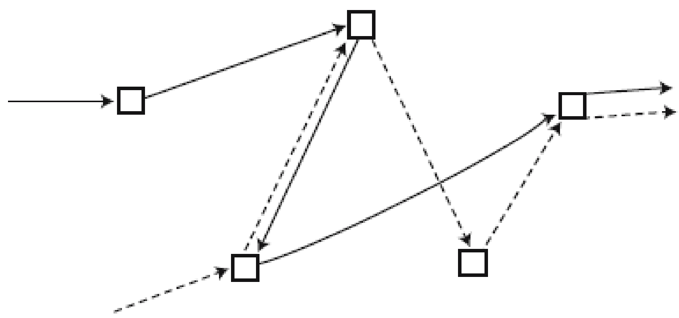


Fig8.1 Job shop.

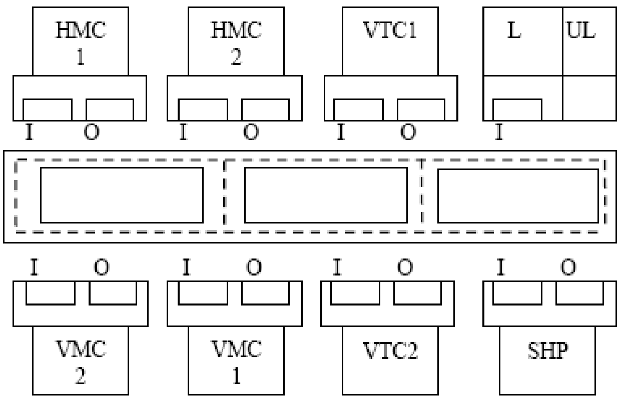
8.3.2 Simulation of an FMS Environment:

In order to compare meta-heuristics and the modified DMM model, we developed a simulation model of an FMS environment.

This system contains seven machines, a loading station, an unloading station, and one automated guided vehicle (AGV). Six different types of parts are considered for production in the system. The machines and stations are as follows:

- Two vertical milling machines (VMC).
- Two horizontal milling machines (HMC).
- Two vertical turning centres (VTC).
- One shaper (SHP).
- One loading station (L).
- One unloading station (UL).

Each machine has an input buffer and an output buffer. The loading station also contains an input buffer. The configuration of the FMS is given in figure 2:



HMC: Horizontal Machining centre.
VMC: Vertical Machining centre.
VTC: Vertical Turning centre.
SHP: Shaper.
L: Loading Station.
UL: Unloading Station.
I : Input Buffer.
O : Output Buffer.
- - AGV routes.

Fig.8.2 Configuration of the FMS.

The alternative routing and the processing time for each type of part are given in table 8.1:

Part type and Production RATIO	Routing (processing time)
A 17%	L – VTC1 (30) – VMC1 (20) - UL L – VTC1 (30) – VMC2 (20) - UL L – VTC2 (30) – VMC1 (20) - UL L – VTC2 (30) – VMC2 (20) - UL
B 17%	L – VTC1 (20) – SHP (1) – VMC1 (15)-UL L – VTC1 (20) – SHP (1) – VMC2 (15)- UL L – VTC2 (20) – SHP (1) – VMC1 (15) - UL L – VTC2 (20) – SHP (1) – VMC2 (15) - UL
C 17%	L – VTC1 (40) – VMC1 (25) - UL L – VTC1 (40) – VMC2 (25) - UL L – VTC2 (40) – VMC1 (25) - UL L – VTC2 (40) – VMC2 (25) - UL
D 21%	L – VTC1 (40) – SHP (1) – VTC1 (20) – HMC1 (35)–UL L – VTC1 (40) – SHP (1) – VTC1 (20) – HMC2 (35)–UL L – VTC1 (40) – SHP (1) – VTC2 (20) – HMC1 (35)–UL L – VTC1 (40) – SHP (1) – VTC2 (20) – HMC2 (35)–UL L – VTC2 (40) – SHP (1) – VTC1 (20) – HMC1 (35)–UL L – VTC2 (40) – SHP (1) – VTC1 (20) – HMC2 (35)–UL L – VTC2 (40) – SHP (1) – VTC2 (20) – HMC1 (35)– UL L – VTC2 (40) – SHP (1) – VTC2 (20) – HMC2 (35)–UL
E 20%	L – VTC1 (25) – SHP (1) – VTC1 (35) – HMC1 (50)–UL L – VTC1 (25) – SHP (1) – VTC1 (35) – HMC2 (50)–UL L – VTC1 (25) – SHP (1) – VTC2 (35) – HMC1 (50)–UL L – VTC1 (25) – SHP (1) – VTC2 (35) – HMC2 (50)–UL L – VTC2 (25) – SHP (1) – VTC1 (35) – HMC1 (50)–UL L – VTC2 (25) – SHP (1) – VTC1 (35) – HMC2 (50)–UL L – VTC2 (25) – SHP (1) – VTC2 (35) – HMC1 (50)–UL L – VTC2 (25) – SHP (1) – VTC2 (35) – HMC2 (50)–UL
F 8%	L –HMC1 (40) – UL L –HMC2 (40) – UL

Table 8.1 Alternative routings of part types.

The studied operations on the flexible production system are based on the following assumptions:

- The alternative routings of each type of part are known before the start of production.
- The AGV routes depend on the selected alternative routings in real time.
- The processing time is known.
- The processing time of an operation is the same on the alternatives machines identified for this operation.

Each machine can process one piece at a time.

8.4 DMM and modified DMM

8.4.1 DMM for Real Time Routing Selection

The original Dissimilarity Maximization Method (DMM) is a method for selecting alternative process plans developed by **(Saygin and Kilic, 1999)** for the selection of alternative routing to schedule off line FMS. It is a method inspired from the group technology. The DMM method has a reciprocal function in group technology, as it tends to maximize the dissimilarity coefficients instead of similarity coefficients.

This method selects routings for the parts in the system among their alternative routings where the total dissimilarity among the selected routings is maximized. Dissimilarity between two routings is defined in terms of machines that belong to each routing. The selection of a routing among alternative routings of each part is performed according to the maximization of the sum of the dissimilarity coefficients. This method was developed to reduce congestion and increase production rate in FMS.

Notations :

n : Number of parts.

q : Number of routings.

D_{ij} : dissimilarity between routings i and j.

C_{ij} = 1 if routing j belongs to the routings of part i. Otherwise, C_{ij} = 0

X_j = 1 if routing j is selected. Otherwise, X_j = 0

S_j: Sum of maximum dissimilarity.

The dissimilarity coefficient (dissimilarity of machine type) between two routing i and j is defined as follows **(Saygin and Kilic, 1999)**:

$$D_{ij} = \frac{\textit{Number of machine types that are not common in both routing i and j}}{\textit{Total number of machine types in both routing}} \quad (8.1)$$

For the selection of alternative routing we will maximize the total sum of dissimilarities between the routing as follows (Saygin and Kilic., 1999):

$$S_j = Max \sum_{i=1}^{q} \sum_{j=1}^{q} X_j D_{ij} \quad (8.2)$$

Subject to:

$$\sum_{j=1}^{q} C_{ij} X_j = 1 \quad for\ all\ parts\ i = 1,..,n \quad (8.3)$$

Equation (8.3) states that only one routing will be selected for each part.

$$\sum_{j=1}^{q} X_j = n \quad for\ all\ routings\ j = 1,..,q \quad (8.4)$$

Equation (8.4) states that the number of selected routings will be equal to the number of parts.

8.4.2 Modified DMM for Real Time Routing Selection:

The Modified DMM rule **(Hassam and Sari, 2007)** was developed based on the DMM rule mentioned earlier. The major motivation behind the modified DMM was twofold. For high arrival rate of parts and small buffer capacities, the production system is overloaded and yet the utilization rates of the machines and the material handling system are low.

These two factors affect the performance of the FMS. For this we propose the Modified DMM rule to overcome these problems. Our modification of DMM rule is intended to keep the same principle which depends on the maximization of dissimilarity coefficients for the selection of various routing, but by affecting several parts to a single routing.

So if all routes are selected by a part, the newly created part will choose among routing, the part will be delivered in the routing where the queue of the first machine of this routing, contains at least one free place.

8.4.3 ALGORITHM OF MODIFIED DMM RULE:

In this section we show the integration of Modified DMM rule in FMS for the selection of an alternative routing among routings available for each type of part. The parts arriving in the first have a higher priority following the FIFO rule, the other parts will wait in input or output queues of various machines or in the loading station. The modified rule will use the following algorithm for the selection of alternative routing in real time in a flexible production system.

Step 1: All routes are free (available) so X(i)=0.
Step 2: Calculation of dissimilarity coefficients D_{ij} (1).
Step 3: Creation (arrival) of parts.
Step 4: Condition: depending on the type of part tested:
If there's at least one free routing and at least one free place in the queue of the loading station.

Or

If all routes are busy and the input queue of the first machine of at least one routing contains at least one free place and this machine is not broken down.

Step 5: If the previous condition is not verified, the part is in a queue until the condition is verified.
Step 6: If the condition of step 4 is checked then we calculate the sum:

$$S(j) = \sum_{i=1}^{q} X(i)D(i, j) \tag{8.5}$$

Step 7: Test, if we find a maximum of S (j) (There is free routings).
Step 8: If the previous condition is checked then go to step 10.
Step 9: If the condition of step 7 is not checked, then select the routing where the input queue of its first machine contains at least one free place.
Step 10: Routing j selected according to Step 7 or step 9 is occupied, X(j)=1.
Step 11: Treatment of the part according to the selected routing j.
Step 12: At the end of treatment, routing becomes available again X(j)=0.
Step 13: Part leaves the system.

The cycle repeats itself from Step 3 to Step 11 every time a part arrives.

8.5 Meta-heuristics for job shop routing

8.5.1 Ant colony Optimization:

This meta-heuristic was introduced by **(Dorigo M 1992)** and was inspired by the studies on the real ant whose members are individually equipped with very limited faculties but can find the shortest path from a food source to their nest without visual cue. They are also capable of adapting to changes in the environment like the appearance of an unexpected obstacle on the initial path between the food source and the nest.

The first algorithm of this type of meta-heuristics was conceived to solve travelling salesman problem **(Dorigo M 1992)**. This algorithm principle is simple. When an ant *k* moves from city *i* to city *j*, it leaves a trail on the way. Moreover, it chooses the next city to be visited using a probability p_{ij}^{k} based on a compromise between the intensity of the trail $\tau_{ij}^{k}(t)$ and visibility η_{ij} that represents the reciprocal of the distance between i and *j*, the relative importance of the two elements is controlled by two parameters α and β.
Each ant *k* has a form of memory $tabu_k$ it points out the ordered list of the cities which have been already visited in order to force this one to form an acceptable solution. After a full run, each ant deposit a certain quantity of pheromone $\Delta\tau_{ij}^{k}(t)$ which depends on the quality of the solution found on the whole of its course. This algorithm has been adapted to our problem by replacing the city i by the part i and the city j by the routing j. For each part i, the choice of routing j is based on a compromise between the intensity of the trail $\tau_{ij}^{k}(t)$ and visibility η_{ij} (depends on the number of parts in the input buffer of the first machine of the routing and its load).
The relative importance of the two elements is always controlled by two coefficients α and β. If the full number of ants is m and the size of the loading station is n, a cycle is carried out when each m ants assigns n first parts of the infinite queue to routings j. After a full rotation (the assignment of all n first parts of the infinite queue to the routings by the ants), each ant leaves a certain quantity of pheromone $\Delta\tau_{ij}^{k}(t)$ which depends on the quality of the found solution on the whole of the selected routings for the parts.

Algorithm:

Step1: if there is a free place in the loading station then
Step2: For t = 1 to t_{max}
Step3: For each ant k = 1to m
Step4: Select randomly a routing for the first part of the infinite queue according to its type.
Step5: For each part i contents in the second place until the n^{th} place of the infinite queue
Step6: Select a routing i, among the possible routings according to a probability depending on the intensity of the trace and the number of the parts in the input queue of the first machine of this routing and its load.
Step7: End For.

Step8: Evaluation: of the objective function. (Produced loads of the routings)

Step9: Leave a track $\Delta \tau_{ij}^{k}(t)$ on the way $T^k(t)$ (for each routing j selected for part i by the ant k).

Step10: End For.

Step11: Evaporate the tracks and modify the intensities.

Step12: End For.

Step13: End if.

8.5.2 Simulated Annealing

The simulated annealing method was conceived by **(Kirkpatrick S et al 1983)**. It is a meta-heuristic inspired by a process used in metallurgy to obtain a well ordered solid state with minimal energy called annealing process.

This technique consists in carrying material at high temperature, then to lower this temperature slowly. This optimization method is based on works of **(Metropolis N et al 1953)** which allow describing the behavior of a system in thermodynamic equilibrium at a certain temperature. This technique transports the annealing process to the resolution of an optimization problem: the objective function to be minimized being the energy E of material. The temperature T is also introduced. From an initial solution at a temperature T, we generate another solution close in a random way. If this solution improves the objective function, this latter is automatically accepted. If it degrades the objective function, it can also be accepted according to a probability exp(-ΔE) were ΔE is the variation of the objective function, once thermodynamic equilibrium is reached one should lowers the front temperature slightly before implementing a new iteration.

Algorithm:

Step1: If there' is a free place in the loading station then

Step2: Build the initial state (assign n first parts to routings randomly).

Step3: Calculate the product of loads of the routings.

Step4: For t = 1,…, tmax

Step5: Modify the routings of certain parts among n first parts contained in the infinite queue.

Step6: Calculate the product of loads of the routings.

Step7: If the objective function is improved then this solution is accepted

Step8: End if

Step9: Else generate a random number

Step10: If this number is lower or equal to exp (- ΔE): (ΔE is the variation of the objective function) then this solution is accepted

Step11: End if

Step12: End if
Step13: End For.
Step14: End

8.5.3 Particle Swarms Optimization

PSO is a recent meta-heuristic approach proposed by Kennedy and Eberhart in **(Eberhart R.C and Kennedy J 1995)**. It is based on the metaphor of social interaction and communication, such as fish schooling and bird flocking when it is randomly searching for food in an area, where there is only one piece of food available and none of them knows where it is, but they can estimate how far it would be at each iteration. For this problem, the simplest strategy to find and get the food is to follow the bird known as the nearest one to the food.
In PSO, each single solution is called a particle, the group becomes a swarm (population) and the search space is the area to explore. Each particle has a fitness value calculated by a fitness function, and a velocity of flying towards the optimum. In the original version of PSO, all particles adjust theirs positions not only according to their own experience but also according to the experience of other particles, they fly across the problem space following the particle nearest to the optimum by two elastic forces. One attracts it to the best location so far encountered by the particle. The other attracts it with random magnitude to the best location encountered by any member of the swarm.

PSO originally designed for continuous optimization problems, but can be adapted to discrete problems like our problem of routing selection where the position of the particle is updated by the following equation proposed by Pan et al (Pan et al, 2005):

$$X_i(t) = c_2 \oplus F_3(c_1 \oplus F_2(w \oplus F_1(X_i(t-1)), p_i(t-1)), G(t-1)) \qquad (8.6)$$

The equation consists of three components: The first component is $\lambda_i(t) = w \oplus F_1(X_i(t-1))$ which represents the velocity of the particle. F_1 represents an operator which modifies the routing of some parts with the probability of w, a uniform random number r is generated between 0 and 1. If r is less than w then the F_1 is applied to generate a perturbed permutation of the particle by $\lambda_i(t) = F_1(X_i(t-1))$, otherwise current permutation is kept as $\lambda_i(t) = X_i(t-1)$.Of the same way, the second component which is cognition part of the particle $\delta_i(t) = c_1 \oplus F_2(\gamma_i(t), p_i(t-1))$ and the third component which is the social part of the particle $X_i(t) = c_2 \oplus F_3(\delta_i(t), G_i(t-1))$ have been modified where F_2 and F_3 represent the crossover with the probability C_1 and C_3 where p_i and G are the bests local and global.

Algorithm:

P_i, and G are the bests local and global.

n1: number of individuals

Step1: If n is the size of the queues.
Step2: If there is a free place in the loading station then
Step3: Initialize the population
Step4: Initialize the parameters
Step5: While (hasn't met stop criterion)
Step6: For i = 1 to n1 (for each particle x_i)
Step7: Calculate the products of the routings loads of this bird
Step8: If F (x_i) > F (Bl_i) then: update the best local Bl_i =xi
Step9: End if
Step10: If F (xi) > F (B_g) then: update the best global B_g = xi
Step11: End if
Step13: End for
Step14: For i = 1 to n1 (update the particle position)

$$X_i(t) = c_2 \oplus F_3\left(c_1 \oplus F_2\left(w \oplus F_1\left(X_i(t-1)\right), p_i(t-1)\right), G(t-1)\right)$$

Step15: End for
Step16: End while
Step17: End if
Step18: End if

8.5.4 Genetic algorithms

Genetic algorithms were proposed by **(Holland J.H 1975)**. They were inspired from the principles of natural genetics and the theory of evolution.
In a GA, each solution is stored in an artificial chromosome represented by a code. Each of these chromosomes is defined by two characteristics. The first is their genotype, which is the actual sequence which defines the chromosome. It is called like this because of the analogy with a genetic sequence in biology. The second is the phenotype, which is the decoded version of the genotype that determines the traits of the individual.
With each of the chromosomes, the parameters are decoded and evaluated by the fitness function to determine the quality of the phenotype. New candidates are generated gradually from a set of renewed populations by applying artificial genetic operators selected, after repeatedly using operators of crossover and mutation **(Goldberg, E.E 1989)**. Crossover is performed by taking two fit genotypes, choosing a place along the bit string, cutting each of them at that place and then connecting one string's left to the other string right and vice versa. This produces two new chromosomes, which are a combination of the two parents.

Reproduction is simply a matter of passing chromosomes which are judged to be above a certain fitness level through to the next generation and mutation is done by choosing bits randomly and swapping them.

Algorithm

Step1: If there' is a free place in the loading station then
Step2: Generate a random population.
Step3: While (hasn't meet stop criterion)
Step4: For each individual
Step5: Evaluate the fitness of this individual (the product of the routings loads).
Step6: If the objective function is higher than the best solution then update the best solution.
Step7: End for
Step8: Select the individuals for the reproduction (selection operator).
Step9: Apply the operator of crossing (we obtains a set of new individuals).
Step10: Apply the operator of mutation on the new individuals.
Step11: Constitute the new generation.
Step12: End while
Step13: End if

8.5.5 Taboo search

This method was formalized by **(Glover F. and Manual Laguana 1997)**. It is based on the use of mechanisms inspired by the human memory. An algorithm based on this meta-heuristic requires an initial solution and a neighborhood structure.

The principle of this meta-heuristic is simple: we generate an initial configuration and we proceed by transiting from one solution to another. The mechanism of passage of one configuration, called *s*, to the next one, called *t*, comprises two stages (Dréo et al, 2003):

- The first builds the set of the neighbours of s, i.e. the set of the accessible configurations in only one elementary movement of s, let V (s) be the set (or the subset) of these neighbours.
- The second evaluates the objective function *f* of the problem for each configuration belonging to *V* (*s*). The configuration *t*, which succeeds *s* in the series of the solutions built by the taboo method, is the configuration of *V* (*s*) in which f takes the minimal value. This configuration *t* is adopted even if it is worse than *s*; due to this characteristic the taboo method can avoid the trapping in the local minima.

To avoid to return to a retained configuration and generate a cycle in each iteration the taboo list (of length fixed or variable) that gave its name to the method contains *m* movements ($t \rightarrow s$), which are the opposite of the last *m* movements ($s \rightarrow t$) carried out.

Algorithm

Step1: If n is the capacity of the queues.
Step2: If there is a free place in the loading station then
Step3: Build the initial state S (assign n first parts of the infinite queue to routings in a random way)
Step4: Initialize the parameters
Step5: Calculate the products of the routings loads
Step6: While (hasn't meet stop criterion)
Step7: For t= 1,…, n_neighbors
Step8: Modify the routings of certain parts chosen randomly among n first parts of the infinite queue with no tabou's movements (modification of S).
Step9: Calculate the products of the routings loads.
Step10: If the objective function (produced the routings loads) is higher than the best solution, then update the best solution.
Step11: End if
Step12: End for
Step13: If T is the best of these neighbors then
Step14: Insert movement T->S in the tabou list.
Step15: S=T
Step16: End while
Step17: End if

8.5.6 Electromagnetism Like Method : EM

Electromagnetism-like algorithm is a population-based meta-heuristic which has been proposed by S. Birbil et al in 2003 (Birbil et Fang, 2003) to solve continuous problems effectively.
EM simulates the attraction-repulsion mechanism of electromagnetism theory which is based on Coulomb's law.
At this approach, each solution is characterized and updated by a charge and a force, the charge of each particle is relative of the objective function.

The generic pseudo-code for the EM which consists of five procedures (the initialisation of population, local search to explore the search space, calculation of the total force of each particle witch depends on the charges of the particles, moving along this force, and evaluation of the objective function before implementing a new iteration) is as follows:

1- initialize
2- while (hasn't met stop criterion) do
3- local search
4- calculate total force F
5- move particle by F

6- evaluate particles
7- End While

To solve our problem we used an hybrid Framework witch combines the EM algorithm with genetic operators proposed by (chen et al , 2007)

Algorithm:

m: size of the population.

Step1: If there is a free place in the loading station then
Step2: Generate a random population.
Step3: While (hasn't meet stop criterion)
Step4: For i= 1. , m (for each particle Xi)
Step5: Search with modifying the routings of parts among n first of the infinite queue.
Step6: Evaluate objective function.
Step7: If the objective function is higher than the best solution then update the best solution.
Step8: End for
Step9: Calculate of average of the objective functions (avg ← calcAvgObjectiveValues()).
Step10: For i= 1. , m
Step11: If I ≠ best and F (xi) > avg then
Step12: j = particle selected
Step13: Uniform crossing (xi, xj)
Step14: End if
Step15: Else If f(xi) < avg Then
Step16: Calculate force and move (xi)
Step17: End if
Step18: End for
Step19: End while
Step20: End if

8.6 Performance evaluation for routing selection method

8.6.1 Without presence of breakdown in the system:

8.6.1.1 Production rate:

Figure 8.3, shows that for a significant rate of arrival of the parts in the system (between 5 and 20 minutes), results obtained by Meta-heuristics are practically the same of these obtained with modified DMM method with a little advantage for Meta-heuristics. If the arrival rate is less or equal then 1/25 the production rate is practically the same for all methods. We can see that the results of PSO and GA meta-heuristics are the best among meta-heuristics.

Rate of arrival of the parts (1/min)	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
Modified DMM	99,99	99,99	99,98	99,71	84,47	60,73	41,67	21,15
ACO	99,99	99,99	99,99	99,99	90,52	64,54	43,2	21,51
TS	99,99	99,99	99,99	99,99	95,28	64,93	43,29	21,78
GA	99,99	99,99	99,99	99,99	98,44	71,08	47,25	23,7
PSO	99,99	99,99	99,99	99,99	99,22	69,33	46,1	23,12
SA	99,99	99,99	99,99	99,99	94,73	61,9	41,21	20,61
EM	99,99	99,99	99,99	99,99	98,52	66,94	44,75	22,42

Table8.2 Production rate for queue size=2.

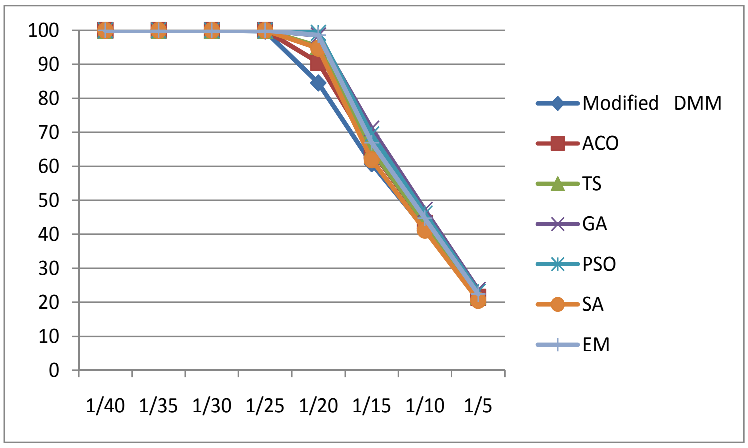


Fig.8.3 Production rate for queue size=2.

8.6.1.2 Machines utilization rate:

The utilization rate of the machines is a very significant criterion in the measurement of the performance, of a production system. The utilization rate for the machines VTC 1 and VTC2 is more significant for the Meta-heuristics and Modified DMM if the system is saturated (rate of arrival of parts is over than 1/25), the DMM Method performances are ameliorate if the parts arrives every 25 minutes if the rate is between 1/30 and 1/40 the utilization of the VTC machines is almost similar for all methods. Figure 8.4 show that GA is the best if one is interested on machines utilization rate for a rate of arrives of the parts superior than 1/20.

Rate of arrival of the parts (1/min)	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
Modified DMM	49.97	56.94	66.59	79.85	84.52	82.72	83.58	84.70
ACO	48.44	54.59	63.99	76.31	85.73	81.97	82.26	82.06
TS	48.24	54.425	63.72	76.43	90.36	84.73	84.75	85.27
GA	48.36	54.51	64.32	75.67	92.72	90.98	90.85	90.96
PSO	48.29	54.71	63.96	76.55	93.2	89.2	88.84	88.95
SA	48.59	54.74	64.16	76.2	89.42	80.49	80.41	80.47
EM	48.24	54.48	63.96	76.19	92.39	86.16	86.56	86.75

Table 8.3 VTC Machines utilization rate for queue size=2.

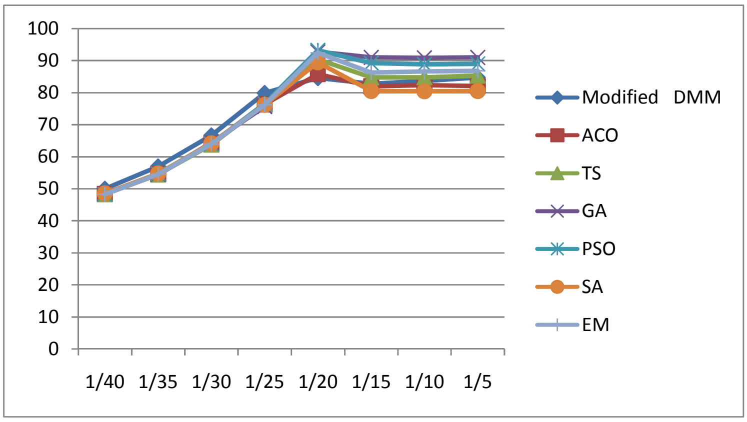


Fig.8.4 VTC Machines utilization rate for queue size=2 .

8.6.1.3 Material handling utilization rate

Figure 8.5 shows us that for a saturated system the ratio utilization of the AGV is practically the same for all methods. The results given by meta-heuristics are less good than those of the Modified DMM but these results are more stable because variations of values are very low, this is due to the high production rate and the increase in the use of the machines. The results of PSO and GA Meta-heuristics are little better than other meta-heuristics if the system is saturated.

Rate of arrival of the parts (1/min)	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
Modified DMM	14.98	17.46	21	27.31	30.26	29.16	30.19	30.44
ACO	17.47	19.58	22.89	27.36	30.64	29.37	29.46	29.41
TS	17.37	19.5	22.86	27.42	32.32	30.68	30.69	30.88
GA	17.43	19.54	23.16	27.04	33.07	32.69	32.67	32.68
PSO	17.40	19.64	22.98	27.48	33.20	32.00	31.97	31.98
SA	17.55	19.65	23.08	27.30	31.92	29.104	29.07	29.10
EM	17.37	19.52	22.98	27.29	32.89	31.12	31.2	31.26

Table 8.4 Material handling utilization rate for queue size=2.

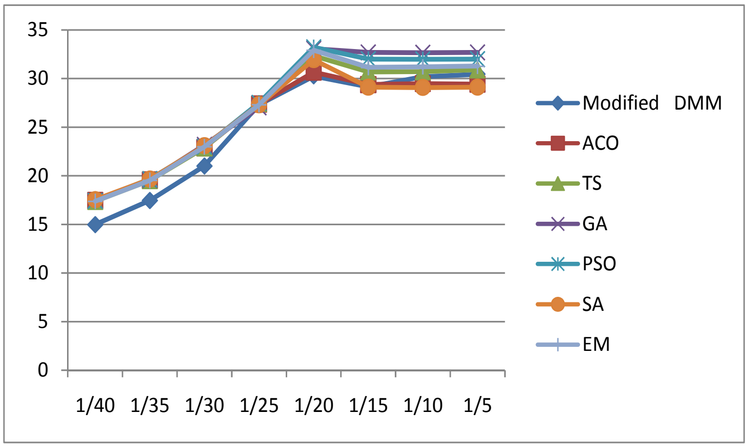


Fig.8.5 Material handling utilization rate for queue size=2.

8.6.1.4 Work in process

Figure 8.6 shows that if the rate of creation of the parts is greater or equal than 1/30, the number of parts that remains in the system if we use Modified DMM is higher than those of the Meta-heuristics. Therefore if the system is saturated the results given by meta-heuristics are better than those of Modified DMM. If the system is less saturated the results of work in process is draw nearer for all methods.

Rate of arrival of the parts (1/min)	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
Modified DMM	6.79	12.95	15.79	19.01	17.84	18.19	16.86	18.12
ACO	3.9	4.04	4.45	5.21	7.29	8.4	8.44	8.46
TS	3.89	4.03	4.47	5.26	7.98	8.4	8.4	8.41
GA	3.9	4.04	4.46	5.23	7.45	8.9	8.85	8.85
PSO	3.88	4.03	4.45	5.25	7.35	8.67	8.67	8.64
SA	3.91	4.03	4.46	5.22	7.11	7.95	7.95	7.95
EM	3.89	4.04	4.44	5.28	7.99	8.64	8.66	8.62

Table 8.5 Work in process for queue size=2.

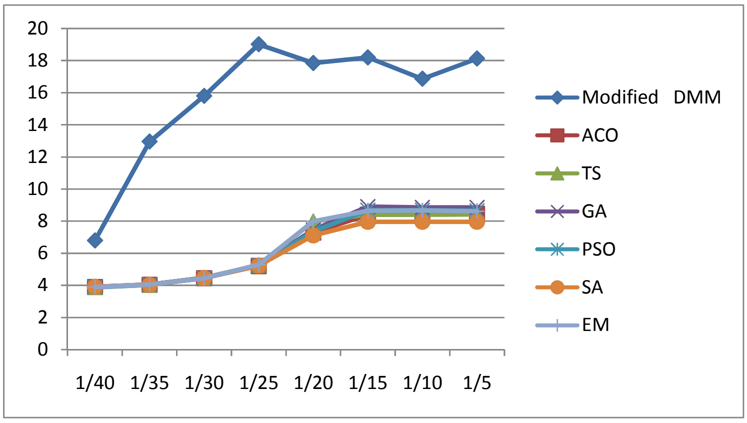


Fig 8.6 .Work in process for queue size=2.

8.6.1.5 Cycle time:

Figure 8.7, shows that if the rate of arrival of the parts is greater than 1/30, the cycle time of these parts in the system is better in the Meta-heuristics case than the ones in modified DMM. The results of cycle time given by meta-heuristics are practically the same. If the rate of arrival parts is between 1/30 and 1/40 we can see that the results of all methods are similar. These results show that the cycle times given by Meta-heuristics are better than those of Modified DMM in case of saturated system.

Rate of arrival of the parts (1/min)	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
Modified DMM	81,94	87,83	101,63	155,89	203,37	204,70	207,22	204,21
ACO	90,91	83,76	89,03	98,83	155,29	166,69	165,41	163,58
TS	91,49	80,37	86,24	98,02	135,97	171,79	174,84	173,94
GA	89,62	81,81	88,65	98,74	132,702	169,87	168,93	168,14
PSO	89,62	81,23	91,7	96,19	125,92	173,51	173,24	169,49
SA	95,33	83,81	90,34	99,59	135,51	170,44	168,95	168,15
EM	92,29	81,6	90,16	101,12	124,84	170,57	170,85	168,91

Table 8.6 Cycle time for queue size=2.

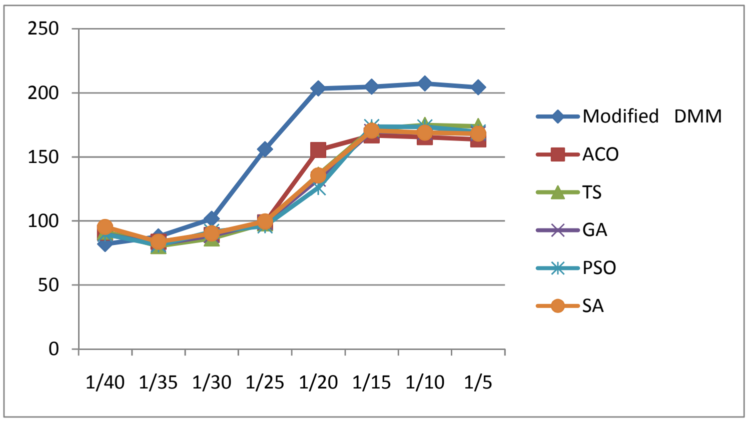


Fig8.7 .Cycle time for queue size=2.

8.6.2 With the presence of breakdown in the system:

8.6.2.1 Production rate:

The results in Figure 8.8 are similar to those in Figure 8.3, for a rate of arrival of the parts in the system between 5 and 25 minutes, results obtained by Meta-heuristics are better than those obtained in the modified DMM. GA and PSO meta-heuristic give the better results. If the arrival rate is less or equal then 1/30 the production rate is practically the same for all methods. We can conclude that in general even with the presence of breakdown in the system, the performances of meta-heuristics are the best.

Rate of arrival of the parts (1/min)	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
Modified DMM	99,99	99,97	99,62	85,02	66,70	50,91	34,29	17,04
ACO	99,99	99,99	99,99	99,98	84,47	63,13	42,11	21,13
TS	99,99	99,99	99,99	99,99	84,53	63,45	42,28	21,12
GA	99,99	99,99	99,99	99,99	92	69,05	46,22	23,12
PSO	99,99	99,99	99,99	99,99	90.04	67.5	44.99	22.61
SA	99,99	99,99	99,99	99,99	80,72	60,22	40,24	20,23
EM	99,99	99,99	99,99	99,99	89,78	65,61	43,76	21,96

Table 8.7 Production rate for queue size=2.

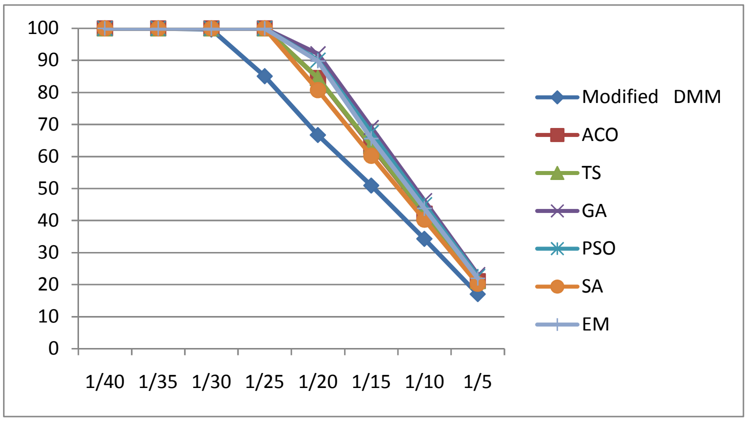


Fig8.8 .Production rate for queue size=2.

8.6.2.2 Machines utilization rate:

Even with the introduction of failures, the utilization rate for the machines VTC 1 and VTC2 is more significant for the Meta-heuristics when compared to Modified DMM if the system is saturated (rate of arrival of parts is over than 1/20). The

Modified DMM Method performances are ameliorated if the parts arrive every 25 minutes. Otherwise, the rate of utilization of the VTC machines is almost similar for all methods. Figure 8.9 show that PSO and GA meta-heuristics are the best if the system is saturated.

Rate of arrival of the parts (1/min)	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
Modified DMM	49,88	56,94	66,78	77,16	77,77	78,13	78,50	78,75
ACO	48,29	54,58	63,9	76,13	80,44	80,2	80,24	80,5
TS	48,40	54,65	63,955	76,03	82,75	82,8	82,76	82,75
GA	48,3	54,90	64,04	76,1	88,135	88,44	88,63	88,67
PSO	48.64	54.8	64.26	76.45	86.71	86.63	86.61	86.9
SA	48,23	54,47	64,25	76,2	78,73	78,35	78,53	79,84
EM	48,30	54,60	64,01	75,95	86,16	84,6	84,65	84,96

Table 8.8 VTC Machines utilization rate for queue size=2.

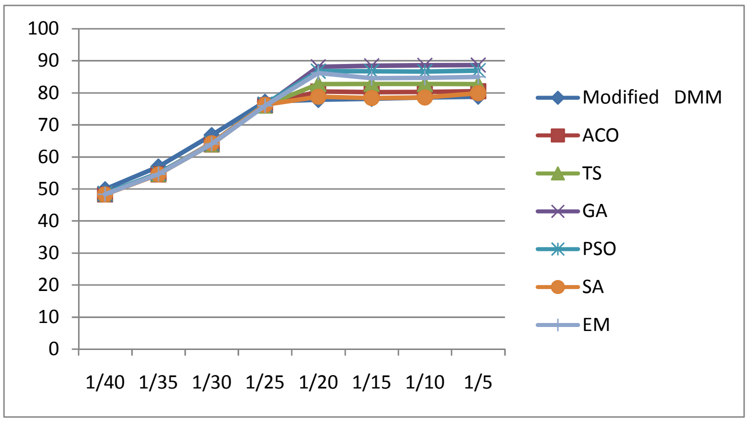


Fig8.9 .VTC Machines utilization rate for queue size=2.

8.6.2.3 Material handling utilization rate:

Figure 8.10 shows us that for a saturated system the ratio utilization of the AGV is more significant for meta-heuristics than modified DMM rule. If the rate of arrival parts is less than 1/25 all of the methods give practically the same performances. The results of PSO and GA Meta-heuristics are little better than other meta-heuristics if the system is saturated.

Rate of arrival of the parts (1/min)	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
Modified DMM	14,98	17,55	21,19	24,28	24,07	24,54	24,67	24,54
ACO	17,39	19,57	22,95	27,27	28,82	28,74	28,75	28,84
TS	17,45	19,61	22,97	27,21	29,96	29,98	29,96	29,97
GA	17,402	19,73	23,02	27,25	31,78	31,79	31,83	31,85
PSO	17.57	19.68	23.13	27.42	31.2	31.16	31.15	31.24
SA	17,37	19,52	23,13	27,302	28,46	28,33	28,309	28,88
EM	17,404	19,58	23,008	27,17	30,96	30,49	30,51	30,62

Table 8.9 Material handling utilization rate for queue size=2.

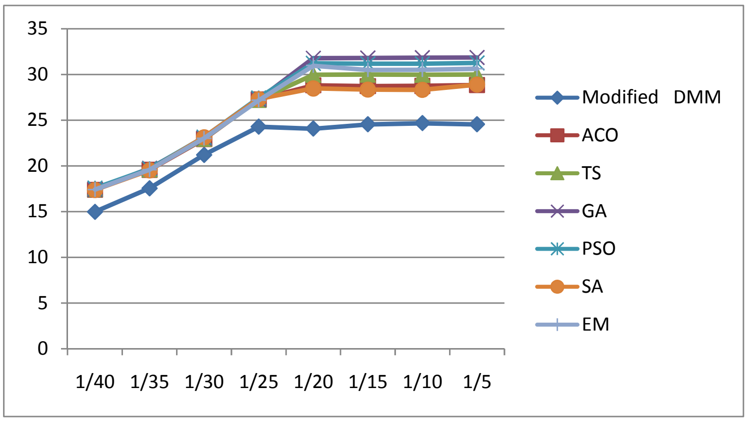


Fig8.10 .Material handling utilization rate for queue size=2.

8.6.2.4 Work in process:

Figure 8.11 shows that if the rate of arrival of the parts is greater or equal than 1/30, the number of parts that remains in the system if we use Modified DMM rule is higher than those of meta-heuristics. In this case the results of all meta-heuristics are the same practically. If the system is not saturated the results given by meta-heuristics and Modified DMM rule are very similar.

Rate of arrival of the parts (1/min)	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
Modified DMM	4.55	7.79	28.62	29.82	30.51	28.04	28.64	27.2
ACO	3.95	4.09	4.55	5.4	8.46	8.43	8.43	8.46
TS	3.94	4.1	4.54	5.36	8.41	8.4	8.41	8.39
GA	3.92	4.1	4.58	5.34	8.83	8.87	8.85	8.81
PSO	3.94	4.09	4.55	5.39	8.67	8.68	8.68	8.7
SA	3.93	4.1	4.54	5.44	7.95	7.94	7.99	8.05
EM	3.93	4.09	4.56	5.41	8.65	8.61	8.61	8.6

Table 8.10 .Work in process for queue size=2

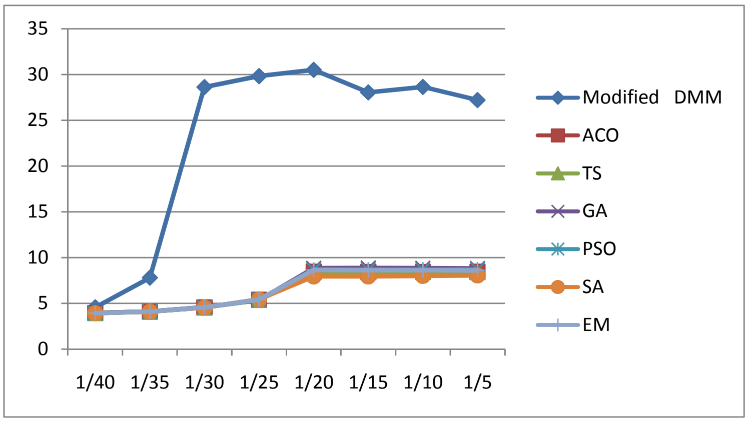


Fig8.11 .Work in process for queue size=2.

8.6.2.5 Cycle time:

Figure 8.12, shows that if the rate of arrival of the parts is greater than 1/30, the cycle time of these parts in the system is better in the Meta-heuristics case than the ones in modified DMM. The results of cycle time given by meta-heuristics are practically the same. If the rate of arrival parts is between 1/30 and 1/40 we can see that the results of all methods are similar. These results show that the cycle times given by Meta-heuristics are better than those of Modified DMM in case of saturated system even with presence of breakdowns.

Rate of arrival of the parts (1/min)	1/40	1/35	1/30	1/25	1/20	1/15	1/10	1/5
Modified DMM	91,10	99,13	119,99	325,16	353,34	346,81	339,85	341,13
ACO	97,59	82,67	90,14	104,02	169,87	169,89	167,36	175,42
TS	94,59	85,92	91,91	103,42	177,106	178,4	173,76	176,05
GA	94,27	85,76	93,3	104,61	170,8	174,606	175,23	176,25
PSO	95.5	84.9	93.2	103.2	170.8	175.1	174.6	173.9
SA	93,07	86,26	93,58	99,45	172,33	175,78	175,08	173,36
EM	92,78	83,37	91,26	102,16	174,09	172,91	175,84	169,91

Table 8.11 Cycle time for queue size=2.

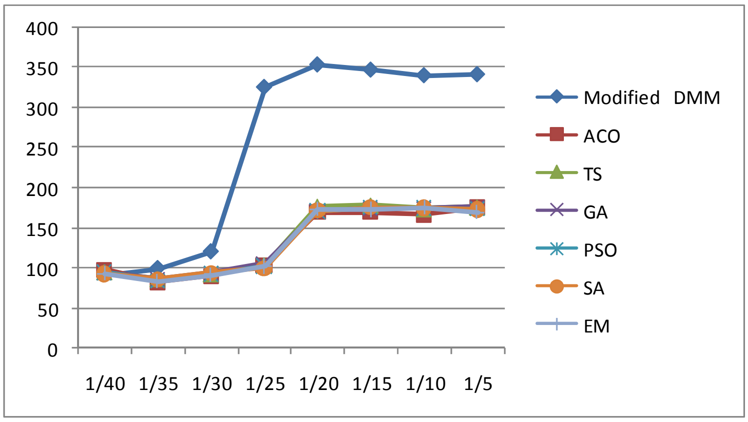


Fig8.12 .Cycle time for queue size=2.

8.7 Conclusion

In this paper we presented a number of meta-heuristics and compared their performances with a method of selection of alternative routing in real time namely: modified Dissimilarity Maximization Method.
For each rule, we notice that the simulation results without breakdowns are better than those with breakdown, which is predictable since breakdown lowers the performance of the system. Results obtained showed that all meta-heuristics gave results better than modified DMM and clearly increased the performances of the system for a saturated production system and high rate of creation of parts because they increase the production rate and the utilization rate of machines and the utilization of AGV. We can remark the improvement of the performances concerning the cycle time and the rate of the work-in-process of the system if we use the

meta-heuristics contrary to modified DMM which can't improve these performances.
Results showed that PSO (Particle Swarms Optimization) and GA (Genetic Algorithms) gives the best results practically in all cases. If the production system is not overloaded all methods give the same results practically.

References

Basnet C, Mize J. H (1994) Scheduling and control of flexible manufacturing systems: a critical review, International journal of Computer Integrated Manufacturing, 7(6), 340-355

Birbil, S.I. and Fang, S.(2003). 'An Electromagnetism like Mechanism for Global Optimization', Journal of Global Optimization, vol 25, pp.263–282.

Byrne M. D, Chutima P (1997) Real-time operational control of an FMS with full routing flexibility, International journal of Production Economic, 51, 109-113

Garey M.R, Johson D.S (1979)"Computers and intractability a guide of the theory of NP-completeness", W.H. Freeman and company, San Francisco

Chen, S. Chang, P. Chan, C. and Mani. (2007).' A Hybrid Electromagnetism-Like Algorithm for Single Machine Scheduling Problem', Book chapter in Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, Volume 4682/2007

Cho H, Wysk R. A (1995) Intelligent workstation controller for computer-integrated manufacturing : problems and models, Journal of Manufcturing Systems, 14(4), 252-263

Das S. K, Nagendra P (1997) Selection of routes in a flexible manufacturing facility, International Journal of Production Economics, 48, 237-247

Dorigo M (1992) Optimization, learning and natural algorithms, Italy, PHD thesis, DEI, Politecnico di Milano

Dréo J, Pétrowski A, Siarry P, and Thaillard E. (2003). 'Métaheuristiques pour l'optimisation difficile', Eyrolles .

Eberhart R.C, Kennedy J (1995) New optimizer using particle swarm theory, Proceedings of the 6th International Symposium on Micro Machine and Human Science, 39–43

Glover F, Manual Laguana, (1997) Tabu Search. Article was principally adapted from the book « Tabu Search » of Glover, Laguana 1997, Kluwer Academics publishers.

Goldberg E.E (1989), Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Reading, MA (1989).

Gupta Y. P, Gupta, M. C, Bector, C. R (1989) A review of scheduling rules in flexible manufacturing systems, International Journal of Computer Integrated Manufacturing, 2, 356-377.

Hassam A, and Sari Z, (2007), Real-Time Selection of Process Plan in Flexible Manufacturing Systems: Simulation Study, International Conference on Industrial Engineering and Systems Management, BEIJING - CHINA

Holland J.H (1975), Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor.

Ishii N, Muraki M (1996) A process-variability-based on-line scheduling system in multi product batch process', Computing in Chemical Engineering, 20, 217-234.

Kazerooni A, Chan F. T. S, Abhary K (1997) A fuzzy integrated decision-making support system for scheduling of FMS using simulation, Computer Integrated Manufacturing Systems, 10 (1), 27-34.

Kirkpatrick S, Gelatt C. D, Vecchi M. P (1983) Optimization by Simulated Annealing, Science Number 4598, volume 220, 4598, 671-680.

Kouiss K. Pierreval H, Mebarki, N. (1997) Using multi-agent architecture in FMS for dynamic scheduling, Journal of intelligent Manufacturing, 8, 41-47.

Liu J, MacCarthy B. L (1997) A goal MILP model for FMS scheduling, European journal of operational research, 100, 441-453.

N.Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, Equations of state calculations by fast computing machines, Journal of Chemical Physics, 21:1087–1091, 1953.

Nof, S. Barash, M. Solberg, J. (1979) 'Operational control of item flow in versatile manufacturing system', International journal of production research, 17, pp.479-489.

Pan, QK. Tasgetiren, M.F. and Liang, Y.C. (2005) 'A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem with makespan criterion', Proceedings of the international workshop on UK planning and scheduling special interest group, UK PLANSIG2005. City University, London, pp. 31–41.

Peng, C. Chen F.F., (1998) 'Real-time control and scheduling of flexible manufacturing systems : a simulation based ordinal optimization approach', International Journal of Advenced Manufacturing Technology, 14 (10), pp.775-786.

Rachamadugu, R. and Stecke, K. E. (1994) 'Classification and review of FMS scheduling procedures', Production Planning and Control, 5, pp.2-20.

Saygin, C.. Kilic, S.E., (2004) 'Dissimilarity Maximization Method for Real-time Routing of Parts in Random Flexible Manufacturing Systems', International Journal of Flexible Manufacturing, Systems, 16(2), pp.169-182.

Saygin, C. Chen, F.F. Singh, J. (2001) 'Real-Time Manipulation of Alternative Routings in Flexible Manufacturing Systems: A simulation Study', International Journal of Advanced Manufacturing Technology, 18, pp.755-763.

Saygin, C., Kilic, S.E. (1999) 'Integrating flexible manufacturing systems with scheduling in flexible manufacturing system', International Journal of Advanced Manufacturing Technology, 15(4), pp.268-280.

Saygin, C., Kilic, S.E. (1997) 'Scheduling of flexible manufacturing system', MicroCAD 97 Conference, University of Miskolc, Hungary, vol. H, pp.19-23.

Saygin, C., Kilic, S.E. (1996) 'Effect of flexible process plans on performance of flexible manufacturing systems', proceedings of 7th International DAAM symposium, Vienna, Austria, pp.393-394.

Saygin, C., Kilic, S.E., Toth, T. and Erdelyi, F. (1995) 'On scheduling approaches of flexible manufacturing systems: gap between theory and practice', Selected paper – Postprint volume of the 3rd IFAC/IFIP/IFORS Workshop – Intelligent Manufacturing Systems 95, Pergamon/Elsevier Science, pp. 61-66.

Shukla, C. S., Chen, F. F. (1996) 'The state of the art in intelligent real-time FMS control: a comprehensive survey', Journal of intelligent Manufacturing, 7, 441-455.

Wu, S. Y. D., Wysk, R. A. (1989) 'An application of discrete event simulation to on-line control and scheduling in flexible manufacturing', International journal of Production Research, 27, 1603-1623.