

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

**4,800**

Open access books available

**122,000**

International authors and editors

**135M**

Downloads

Our authors are among the

**154**

Countries delivered to

**TOP 1%**

most cited scientists

**12.2%**

Contributors from top 500 universities



**WEB OF SCIENCE™**

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.

For more information visit [www.intechopen.com](http://www.intechopen.com)



## Hybrid Job Shop and parallel machine scheduling problems: minimization of total tardiness criterion

Frédéric Dugardin, Hicham Chehade, Lionel Amodeo, Farouk Yalaoui  
and Christian Prins

*University of Technology of Troyes, ICD(Institut Charles Delaunay)  
France*

### 1. Introduction

Scheduling is a scientific domain concerning the allocation of limited tasks over time. The goal of scheduling is to maximize (or minimize) different criteria of a facility as makespan, occupation rate of a machine, total tardiness ... In this area, scientific community usually group the problem with, on one hand the system studied, defining the number of machines (one machine, parallel machine), the shop type (as Job shop, Open shop or Flow shop), the job characteristics (as pre-emption allowed or not, equal processing times or not) and so on. On the other hand scientists create these categories with the definition of objective function (it can be single criterion or multiple criteria). The main goal of this chapter is to present model and solution method for the total tardiness criterion concerning the Hybrid Job Shop (HJS) and Parallel Machine (PM) Scheduling Problem.

The total tardiness criterion seems to be the crux of the piece in a society where service levels become the central interest. Indeed, nowadays a product often undergoes different steps and then traverses different structures along the supply chain, this involve in general a due date at each step. This can be minimized as a single objective or as a part of a multi-objective case.

On the other hand, the structure of a hybrid job shop consists in two types of stages with single and parallel machines. That is why we propose to point out the parallel machine PM problem domain which can be used to solve the hybrid job shop scheduling system. This hybrid characteristic of a job shop is very common in industry because of two major factors: at first some operations are longer than other ones and secondly flexible factory. Indeed, if some operations too long; they can be accelerated by technical engineering but if it is not possible they must be parallelized to avoid bottlenecks. Another potential cause is the flexible factory: if a factory does many different jobs these jobs can perhaps pass through a central operation and so the latter must increase his efficiency.

This work is organized as follow: firstly a state of the art concerning PM is realized. The latter leads us to a the HJS problem where we summarize a state of the art on the minimization of the total tardiness and in a second step we present several results concerning efficient heuristic methods to solve the Hybrid Job Shop problem such as Genetic Algorithm or Ant Colony System algorithm. We also deal with multi-objective

Source: Multiprocessor Scheduling: Theory and Applications, Book edited by Eugene Levner, ISBN 978-3-902613-02-8, pp.436, December 2007, Itech Education and Publishing, Vienna, Austria

optimizations which include the minimization of total tardiness using the NSGA-II see Deb *et al.*, (2000). The Hybrid Job Shop Parallel Machine Scheduling rises in actual industrial facilities, indeed some of the results presented here have direct real application in a printing factory. Here the hybridization between the parallel machine stage and the single stage is provided by the printing and the winding operations which proceed with more jobs than cutting and packaging operations.

To put it in a nutshell, this chapter presents exact and approximate results useful to solve the Hybrid Job Shop problem with minimization of total tardiness.

## 2. Problem formulation

The hybrid job shop problems or the flexible job shop problem are various considered in this document can be shown using the classical notation  $HJS_m \mid \text{prec}, S_{sd}^m, r_j, d_j \mid \sum T_j$ . It can be formulated as follow:  $n$  jobs ( $j = 1, \dots, n$ ) have to be processed by  $m$  machines ( $i = 1, \dots, m$ ) of different types gathered in  $E$  groups. In this case two types of groups are considered: groups with single machines and groups with identical parallel machines.

Each job has a predetermined route that it has to follow through the job-shop. Only one operation for a job can be processed in a group. The maximal number of operations is equal to the number of groups. All the machines are available at the initial time 0. No order priority is assigned to the job.

The processing of job  $j$  on machine  $i$  is referred to as operation  $O_{i,j}$ , with processing time  $p_{i,j}$ . The processing times are known in advance. Job  $j$  has a due date  $d_j$  and a release date  $r_j$ , respectively, the last job operation completion time and the first job operation availability. No job can start before its release date and its processing should not exceed its due date. If operation  $O_{i,k}$  immediately succeeds operation  $O_{i,j}$  on machine  $i$ , a setup time  $S_{j,k}^i$  is incurred. Such setups are sequence dependent see Yalaoui (2003) and  $S_{j,k}^i$  need not be equal to  $S_{k,j}^i$ . Let  $C_j$  denote the completion time of job  $j$  and  $T_j = \max(C_j - d_j, 0)$  its tardiness. The objective is to find a schedule that minimizes the total tardiness  $T = \sum_{j=1}^n T_j$  in such a way that two jobs cannot be processed at the same time on the same machine. The splitting and the pre-emption of the operations are forbidden.

Table 1 shows an instance of the  $HJS_m \mid \text{prec}, S_{sd}^m, r_j, d_j \mid \sum T_j$  problem with four jobs and four machines  $4 \times 4$ .

| Job | $r_j$ | $d_j$ | Total processing time | Sequence | Processing time                                       |
|-----|-------|-------|-----------------------|----------|---|
| 1   | 0     | 35    | 27                    | 2-1-4-3  | $p_{2,1} = 4, p_{1,1} = 8, p_{4,1} = 10, p_{3,1} = 5$ |
| 2   | 0     | 22    | 11                    | 1-2-4    | $p_{1,2} = 2, p_{2,2} = 6, p_{4,2} = 3$               |
| 3   | 4     | 25    | 20                    | 1-2-4-3  | $p_{1,3} = 7, p_{2,3} = 5, p_{4,3} = 1, p_{3,3} = 7$  |
| 4   | 1     | 34    | 14                    | 4-3-1    | $p_{4,4} = 7, p_{3,4} = 4, p_{1,4} = 3$               |

Table 1. Example  $4 \times 4$

The  $HJS_m \mid \text{prec}, S_{sd}^m, r_j, d_j \mid \sum T_j$  problem, extends the classical job shop problem by the presence of identical parallel machines, by allowing for sequence dependent setup times between adjacent operations on any machine and the restriction of jobs arrival dates.

The classical job-shop problem,  $J_m || \gamma$ , is a well-known NP-hard combinatorial optimization one, see Garey and Johnson, (1979), which makes our problem a NP-hard problem too. The  $J_m || \gamma$  problem has been investigated by several researchers. It can be classified in two large families according to the objective function: minimizing makespan and minimizing tardiness.

### 3. State of the Art

#### 3.1 Parallel Machine

The Hybrid Job Shop is linked in some way to Parallel Machine Job Shop. Indeed as can be seen in the Figure 1, an Hybrid Job Shop is composed of different stages which can contain one single machine or parallel machines.

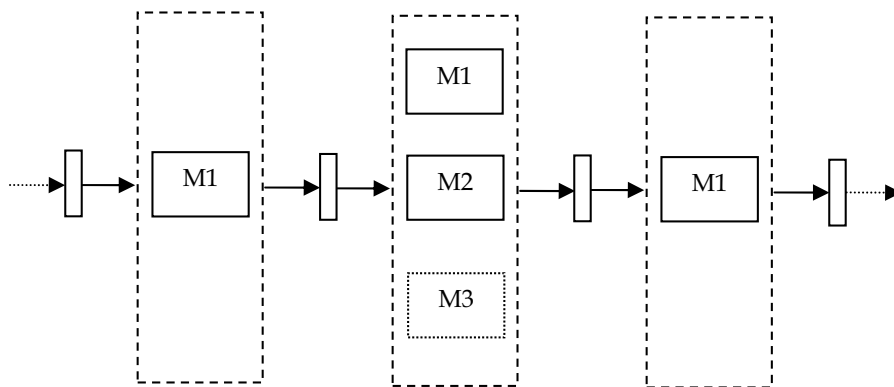


Figure 1. Example of Hybrid Job Shop

So this type of problem can be described as a sequence of parallel machine problem. Moreover the Parallel Job Shop problem has been widely studied especially for the minimization of the total tardiness.

The Parallel Machine problem consists of scheduling  $N$  jobs on  $M$  different parallel machines without interruption. The goal here is to minimize the total tardiness. The parallel machine problem is known as NP-Hard, so the minimization of total tardiness in a parallel machine problem is also NP-Hard according to Koulamas C., (1994) and Yalaoui & Chu, (2002). Different reviews exist in the literature as Koulamas C., (1994) and Shim & Kim, (2007) and it appears that the one machine problem has been more studied than the multiple machine problems. On the other hand, one can also stress that the objective is mainly to minimize the makespan, the total flow time and more recently the minimization of the total tardiness. We now mention different interesting works for their heuristics or their problem.

In 1969 Pritsker *et al.*, (1969) have done the formulation with linear programming. Alidaee & Rosa, (1997) have proposed a method based on the modified due date method of Baker K.R. & Bertrand J.W., (1982). Other priority rules can be found in the work of Chen *et al.*, (1997). Koulamas has proposed the KPM to extend the PSK method of Panwalker *et al.*, (1993) to parallel machines problem, the former has proposed also a method based on Potts & Van Wassenhove, (1997) on the single machine problem and also an hybrid method with Simulated annealing Koulamas, (1997). Other authors were interested in this type of

problem, as Ho & Chang, (1991) with their traffic priority index or Dogramaci & Surkis, (1979) with different rules like Early Due Dates, Shortest Processing Time or Minislack. There is the work of Wilkerson & Irwin, (1979) and finally one must mention the Montagne's Ratio Method (Montagne, 1969).

We can also quote works of: Eom *et al.*, (2002), the tabu search based method of Armentano and Yamashita, (2000), the three phase method of Lee and Pinedo, (1997), the neural network method of Park & Kim, (1997), the work of Radhawa and Kuo, (1997) and also Guinet, (1995). And the former work of Arkin & Roundy, (1991), Luh *et al.*, (1990), Emmons & Pinedo, (1990) and Emmons, (1987).

More recently Armentano and de França Filho, (2007) have proposed a tabu-search with a self adaptive memory, Logendram *et al.*, (2007) proposed six efficient approach in order to take the best schedule, one can also mention the work of Mönch and Unbehaun, (2007) who compare their results to the best known lower bound. Anghinolfi and Paolucci, (2007) have proposed an algorithm based on tabu search, simulated annealing and variable neighbourhood search.

We have only cited heuristic approaches but some exact methods exist with tardiness as criteria as the work of Azizoglu & Kirca, (1998), Yalaoui & Chu, (2002), and Shim & Kim, (2007). The Branch and Bound method of Elmaghraby & Park, (1974), Barnes & Brennan, (1977), Shutten & Leussink, (1996) and Dessouky, (1998).

### 3.2 Parallel machine: useful results

One can mention different results which can be useful for a Hybrid Job Shop. Now, we propose a selection of properties and especially dominance ones from different authors. Assuming the following notations:

|                  |   |
|------------------|---|
| $J$              | set of jobs   |
| $M$              | set of machines   |
| $n$              | number of the jobs ( $n =  J $ )  |
| $m$              | number of the machines ( $m =  M $ )  |
| $p_i$            | processing time of job $i$  |
| $d_i$            | due date of job $i$   |
| $C_i(\sigma)$    | completion time of job $i$ in partial schedule $\sigma$                       |
| $T_i(\sigma)$    | tardiness of job $i$ in partial schedule $\sigma$                             |
| $\Gamma(\sigma)$ | completion time of the last job on machine $k$ in (partial) schedule $\sigma$ |
| $n_k(\sigma)$    | number of jobs assigned to the same machine, in partial schedule $\sigma$     |
| $S(\bullet)$     | set of jobs already included in partial schedule $\bullet$                    |

We will now enumerate the selection of dominance properties:

Proposition 1 (Azizoglu & Kirca, (1998)): *There exists an optimal schedule in which the number of jobs assigned to each machine does not exceed  $N$  such that:*

$$\sum_{i=1}^N p_{[i]} \leq A \quad \text{and} \quad \sum_{i=1}^{N+1} p_{[i]} \geq A \quad (1)$$

Where  $p_{[i]}$  is the processing time of the job with the  $i$ th shortest processing time.

Proposition 2 (Azizoglu & Kirca, (1998)): *If  $d_i \leq p_i$  for all jobs an SPT schedule is optimal.*

Proposition 3 (Azizoglu & Kirca, (1998)): *For any partial schedule  $\sigma$ , if  $d_i \leq p_i + \min_{k \in M} \Gamma_k(\sigma)$  for all  $i$  not in  $S(\sigma)$ , then it is better to schedule jobs after  $\sigma$  in an SPT order.*

Proposition 4 (Yalaoui & Chu, (2002)): For a partial schedule  $\sigma$  and any job  $i$  that is not included in  $\sigma$ , if there is another job  $j$  not included in  $\sigma$  that satisfies  $p_j \leq p_i$  and  $(p_i - p_j) (\max_{k \text{ in } M} \check{n}(\sigma) - 1) \leq \min\{d_i - p_i, \min_{k \text{ in } M} \Gamma_k(\sigma_{ik^*})\} - \max\{d_j - p_j, \min_{k \text{ in } M} \Gamma_k(\sigma)\}$ , where  $\check{n}(\sigma)$  denote the number of additional jobs that are schedule on machine  $k$  after partial schedule  $\sigma$  in an optimal schedule, then complete schedule  $\sigma_{ik^*}$  are dominated

Proposition 5 (Yalaoui & Chu, (2002)): For a partial schedule  $\sigma$  and any job  $i$  that is not included in  $\sigma$ , if there is another job  $j$  not included in  $\sigma$  that satisfies  $0 \leq p_j - p_i \leq \min_{r \neq S(\sigma)} p_r$ ,  $d_j \leq \min_{k \text{ in } M} \Gamma_k(\sigma) + p_j \leq d_i$  and  $(p_j - p_i) (\check{n}(\sigma) - 1) \leq \min\{d_i - p_i, \min_{k \text{ in } M} \Gamma_k(\sigma_{ik^*})\} - \min_{k \text{ in } M} \Gamma_k(\sigma)$ , then complete schedule  $\sigma_{ik^*}$  are dominated.

Proposition 6 (Shim & Kim, (2007)): For any schedule  $\sigma$  in which job  $i$  and job  $j$  are assigned to the same machine and job  $j$  precedes job  $i$ , there is a schedule that dominates  $\sigma$ , if at least one of the following three conditions holds:

1.  $p_i \leq p_j$  and  $d_i \leq \max(C_j(\sigma), d_j)$
2.  $d_i \leq d_j$  and  $C_i(\sigma) - p_i \leq d_j \leq C_i(\sigma)$
3.  $C_i(\sigma) \leq d_j$

### 3.3 Hybrid Job shop

Much of the research literature in job shop scheduling deals with pure job shop environments. However, currently most processes involve a hybrid of both the job shop and a flow shop with a combination of flexible and conventional machine tools.

In a classical job shop problem, the elementary product operations follow a completely ordered sequence according to the product to be manufactured. In some structures, each elementary operation may be carried out on several machines, from where, thanks to the versatility of the machines, a greater flexibility is obtained. We can talk about total flexibility if all the machines are able to carry out all the operations, otherwise, it is a partial flexibility. This is what we call the hybrid job shop or the flexible job shop.

This flexibility may also be applied to the flow shop problem leading then to the hybrid flow shop configuration. A hybrid flow shop is constituted of several stages or groups. Each stage is composed by a set of machines. The passing order in the stages for each part to be manufactured is the same one as in Gourgand *et al.*, (2001). In this work, we are particularly interested in the hybrid job shop scheduling problem.

The Hybrid Job Shop Problem (HJSP) is then an important extension of the classical job shop scheduling problem which allows an operation to be processed by any machine from a given set thus creating an additional complexity. The methodology is to assign each operation to a machine and to order the operations on the machines, such that the maximal completion time (makespan) of all operations or the total tardiness is minimized.

Many scheduling optimization problems have been studied in the research works dealing with complex industrial cases with flexibility. The hybrid job shop scheduling problem was one of those studies presented in the literature like Penz, (1996), Dauzere-Peres *et al.*, (1998), Xia and Wu (2005) and many others.

Chen *et al.*, (1999) present a genetic algorithm to solve the flexible job-shop scheduling problem with a makespan criterion to be minimized. The chromosomes representing the problem solutions consist of two parts. The first part defines the routing policy and the second part the sequence of the operations on each machine. Genetic operators are introduced and used in the reproduction process of the algorithm. Numerical experiments show that the algorithm can find out high-quality schedules.



Gomes *et al.*, (2005) present an integer linear programming (ILP) model to schedule flexible job shop. The model considers groups of parallel homogeneous machines, limited intermediate buffers and negligible set-up effects. Orders consist of a number of discrete units to be produced and follow one of a given number of processing routes with a possibility of re-circulation. Good solution times are obtained using commercial mixed-integer linear programming (MILP) software to solve realistic examples of flexible job shops to optimality.

A genetic algorithm-based approach is also developed to solve the considered problem by Chan *et al.*, (2006). The authors try to solve iteratively a resource-constrained operations-machines assignment problem and flexible job-shop scheduling problem. In this connection, the flexibility embedded in the flexible shop floor, which is important to today's manufacturers, is quantified under different levels of resource availability.

Literature review shows that minimizing tardiness in hybrid job shop problems has been an essential criterion. It is the main objective of the work of Scrich *et al.*, (2004). Two heuristics based on Tabu Search are developed in this paper: a hierarchical procedure and a multiple start procedure. The procedures use dispatching rules to obtain an initial solution and then search for improved solutions in neighborhoods generated by the critical paths of the jobs in a disjunctive graph representation. Diversification strategies are also implemented and tested.

Alvarez-Valdez *et al.*, (2005) presented the design and implementation of a scheduling system in a glass factory aiming at minimizing tardiness by means of a heuristic algorithm. The structure basically corresponds to a flexible job-shop scheduling problem with some special characteristics. On the one hand, dealing with hot liquid glass imposes no-wait constraints on some operations. On the other hand, skilled workers performing some manual tasks are modeled as special machines. The system produces approximate solutions in very short computing times.

Minimizing tardiness in a hybrid job shop is one of the objectives in the work of Loukil *et al.*, (2007) that the authors tried to optimize. A simulated annealing is developed and many constraints are taken in consideration such as batch production; existence of two steps: production of several sub-products followed by the assembly of the final product; possible overlaps for the processing periods of two successive operations of a same job. At the end of the production step, different objectives are considered simultaneously: the makespan, the mean completion time, the maximal tardiness and the mean tardiness.

For our case study, two works have discussed the problem of minimizing tardiness in a hybrid job shop. The first was that of Nait Tahar *et al.*, (2004) by using a genetic algorithm. Only one criterion was taken into account which was the total tardiness. The results obtained showed that the genetic algorithm technique is effective for the resolution of this specific problem. Later, an ant colony optimization algorithm was developed by Nait Tahar *et al.*, (2005) in order to minimize the same criterion with sequence dependent setup times and release dates.

#### 4. Case study: industrial

In this section we will describe an industrial case of hybrid job-shop. Firstly we will describe the problem encountered by a company, and then we will develop three ways of solving the problem: one with a genetic algorithm, the second with a meta heuristic based on ant colony system and the third one with a non-dominated sorting genetic algorithm coupled with a simulation software.

The problem is located in the printing factory that could be described in Figure 2. This factory produces printed and unprinted roll from the raw material: paper, plastic film and ink are combined to produce a finish product. The plant employs 90 people to produce high quality packaging. It produces about 1500 different types of finished goods and delivers about 80 orders per week. During the process, Each product (job) is elaborated on a given sequence of machines. The tasks performed on these machines are called operation.

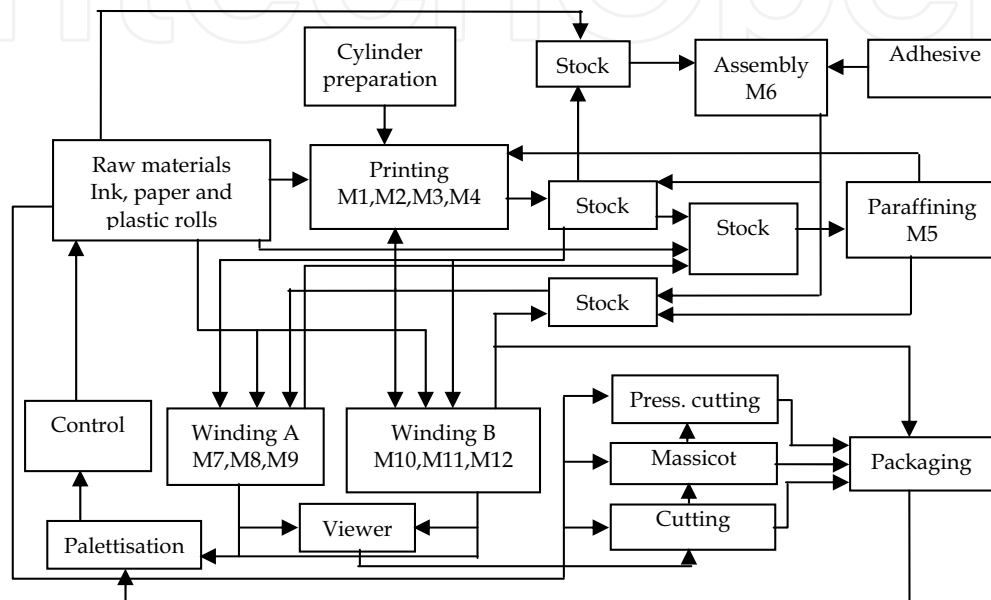


Figure 2. Structure of the printing factory, *Nait Tahar et al. (2004)*

As it appears, the factory structure shows an hybrid job shop structure, with some single machine stage (M5, M6) and multiple machines (M7, M8, M9, for instance) stage with identical parallel machines. Setup times are present: when a machine switch from one operation to another a “switching time” is required. The process is divided into four areas: printing, assembly, paraffining, winding and cutting. The process starts in the printing area where a drawing in one or more colours is reproduced on a paper, raw material. Two printing process can be used: photoengraving and flexography. The assembly combines two supports (printed or not) with a binder (adhesive) on their surface forming one. Paraffining put paraffin on the surface. Then the products reach the cutting and winding area. Finally the products are packaged, stored or shipped.

We now describe two methods used to solve this problem using Ant Colony System (ACS) based algorithm and Genetic Algorithm (GA).

Then a third method is presented dealing with a multi-objectif case resolution.

#### 4.1 Genetic Algorithm

The first method of *Nait Tahar et al., (2004)* uses a genetic algorithm to solve the problem. In a genetic algorithm the solution is represented in a chromosome. The first step is the modeling of the solution in a genetic way, each encoding is specific to one problem. We employ the following encoding with a matrix as it is shown in table 2.



| Machine | Operation1 | Operation2 | Operation3 | Operation4 |
|---------|------------|------------|------------|------------|
| 1       | 2,1,2,-    | 1,1,4,-    | 4,3,3,-    | 3,1,7,-    |
| 2       | 2,2,6,-    | 1,1,8,-    | 3,2,5,-    |            |
| 3       | 4,2,4,-    | 3,4,7,-    | 1,4,5,-    |            |
| 4       | 2,3,3,-    | 4,1,7,-    | 3,3,1,-    | 1,3,10,-   |

Table 2. Solutions encoding (example)

This encoding represents the scheduling in a table of  $m$  lines. Each line represents the operations to schedule in the form of  $n$  cells ( $n$  is the number of jobs). Each cell contains: the job number, the order of the operation in the manufacturing process, the processing time and the operation completion date. The representation of a solution considers the sequences for each job, a machine sequences and not a group sequences. The evaluation (fitness) of an individual is simply the total tardiness.

Since the encoding is chosen, we have to propose mutation and crossover operators. Three crossovers are known for the problems of sequencing: LOX (Linear Order Crossover), OX (Order Crossover) and X1 (One point crossover). We adopt X1 crossover with the studied problem for our encoding. For a parent  $P_1$  having a length  $t$ , a random position  $p$  ( $p < t$ ) is generated. To build the child  $E_1$ , the portion  $P_1$  between 1 and  $p$  inclusive is copied in  $E_1$  using the same positions. Then the portion of  $P_2$  between  $p$  which is not included and  $t$  is swept. Only the non present elements in  $E_1$  are copied. The missing elements in  $E_1$  are added after, from left to right. The construction of child  $E_2$  is identical, by permuting the role of  $P_1$  and  $P_2$ . A chromosome contains all the operations of the problem, and each operation is assigned to only one machine. To prevent a too fast convergence of the algorithm, a mutation is applied to the children with a weak rate. We tested two types of mutation named *mut-ch* and *mut-nb*. The first interchange two operations randomly selected from the busiest machine in the chromosome. *mut-nb* interchanges two operations from the machine having the most total tardiness.

The population stores a fixed number ( $T_{pop}$ ) of chromosomes in a table. These initial solutions are created randomly. For each machine belonging to a single machine group, a sequence is thus randomly generated. For the groups containing several units (identical parallel machines), the operation assignment and sequencing on each machine are also randomly done by balancing the work-load of the machine. For the selection we tested roulette technique and direct tournament. The genetic algorithm is an incremental (steady state) one : the new solutions immediately replace existing solutions in the population. Five procedure have been tested for the replacement: each parent is replaced by its children if there is improvement, the worst parent is replaced by the best descendant, the worst individual of the population is replaced by the better of the solutions, a randomly selected parent is replaced by a randomly selected child, the child replaces an individual chosen uniformly chosen under the median (incremental replacement). Our algorithm is tested with production data, coming from the network of the printing factory. These data are adapted to our algorithm, by creating instances of the same size as the randomly generated (25, 50, 100 jobs). We used a probability of 90% for the crossover and 10% for the mutation probability. Table 3 gives results from many instances, different columns show name of the instance, its size and its number of operations, moreover we can see the total tardiness of the industrial solution and the one with the solution given by the algorithm. Finally this table shows the improvement between industrial and genetic based solution.

| Instance | Size | Operations | Real total tardiness | GA total tardiness | Improvement in % |
|----------|------|------------|----------------------|--------------------|------------------|
| A1       | 25   | 133        | 1297.9               | 919.95             | 29.12            |
| A2       | 25   | 145        | 1330.1               | 915.91             | 31.14            |
| A3       | 25   | 161        | 1488.4               | 1106.33            | 25.67            |
| A4       | 25   | 155        | 1451.6               | 908.12             | 37.44            |
| A5       | 25   | 114        | 1237.4               | 809.75             | 34.56            |
| B1       | 50   | 339        | 2311.2               | 1268.62            | 45.11            |
| B2       | 50   | 297        | 2122.4               | 1229.72            | 42.06            |
| B3       | 50   | 308        | 2178.1               | 1432.75            | 34.22            |
| B4       | 50   | 325        | 2219.3               | 1350.67            | 39.14            |
| B5       | 50   | 341        | 2297.4               | 1335.02            | 41.89            |
| C1       | 100  | 637        | 4587.0               | 2366.89            | 48.40            |
| C2       | 100  | 524        | 4233.3               | 2279.63            | 46.15            |
| C3       | 100  | 612        | 4524.9               | 2308.15            | 48.99            |
| C4       | 100  | 688        | 4721.2               | 2307.72            | 51.12            |
| C5       | 100  | 653        | 4642.8               | 2512.68            | 45.88            |

Table 3. improvement of the industrial solution, Nait Tahar *et al.*, (2004)

The Genetic Algorithm has been coded in C on a 440 Mhz bi processors, it took near from 1000 seconds to get solutions. One can see that the improvement is important from 29 to 51% from the industrial solution use by the factory and based on the “Early Due Date” policy. We have improve significantly the industrial solution. We will see now how this solution can be improved with another meta-heuristic called Ant Colony System optimization.

#### 4.2 Ant Colony System

The ACS Nait Tahar *et al.*,(2005) attempts to solve the problem imitating the behaviour of ants searching food in the nature. Consider for instance the four-job example of table 1.

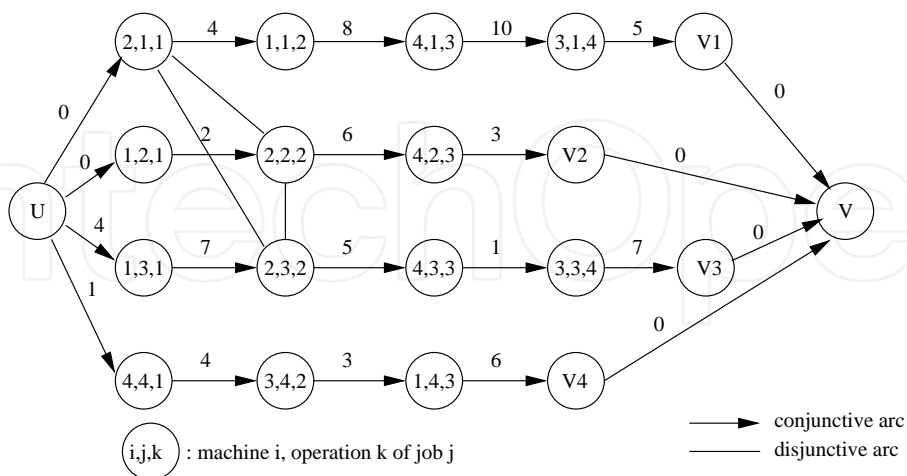


Figure 3. A disjunctive graph for 4x4 instance , Nait Tahar *et al.*, (2005)

We have to describe the sequence by a graph (see Figure 3) in order to apply an ant colony based algorithm. The former is a disjunctive graph where nodes are operations of the job on a certain machine, and the conjunctive arcs are weighted by the duration of operations and the arc connecting the node  $U$  corresponds to the release date  $r_j$ .

Thanks to this description we can apply an ant colony based algorithm sketched by the algorithm 4:

```

INITIALIZE
  Represent the problem by a weighted connected graph
  Set initial pheromone for every arc
REPEAT
  FOR each ant DO
    Randomly select a starting node
    REPEAT
      Choose the next node according to a node transition
      Update pheromone intensity on arc  $(a,b)$  using a local
      pheromone updating rule
    UNTIL a complete path from  $U$  to  $V$  is realized
  FOR each arc DO
    Update pheromone intensity using a global pheromone
    updating rule
  ENDFOR
ENDFOR
UNTIL satisfying stopping criterion
Output The global best path from  $U$  to  $V$  found

```

Algorithm 4. A skeleton of the Ant Colony System algorithm (Nait Tahar (2005))

A solution is a path from  $U$  to  $V$ . This path is built by an ant step by step, node by node. The principle is to simulate ants walking through the graph, at each node they have to choose one arc. The criterion for this choice is the probability of each arc to be taken: this probability grows with the number of ants which have traversed this arc. This mechanism is assumed by the pheromone lay down by each ant.

In this algorithm two things have to be precised:  $\tau_{a,b}$  the quantity of pheromone on the arc  $(a,b)$ , how the next node  $b$  is chosen, and finally how the pheromone quantity is updated on each arc.

Here is described how the pheromone quantities are determined:

$$\tau_{a,b} = \sum_{k=1}^n \Delta \tau_{a,b}^k \quad (2)$$

$$\Delta \tau_{a,b}^k = \begin{cases} \frac{Q}{W_k} & \text{If arcs } (a \rightarrow b) \text{ belongs to the path of ant } k \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

With  $W_k$  is the total tardiness of the arc selected by the ant  $k$ ,  $Q$  is a constant,  $n$  is the number of job generated and  $\tau_{a,b}$  is the quantity of pheromone at initial time. Now an ant can "walk" through the graph (i.e. it can build a path), partially guided by the pheromone:

$$b = \begin{cases} \arg \max_{b \in Succ(a)} \{ [\tau_{a,b}]^\alpha [\eta_{a,b}]^\beta \} & \text{if } \phi \leq \phi_0 \\ P_{a,b}^k & \text{otherwise} \end{cases} \quad (4)$$

$$P_{a,b}^k = \begin{cases} \frac{[\tau_{a,b}(t)]^\alpha [\eta_{a,b}]^\beta}{\sum_{j \in Succ(a)} [\tau_{a,j}(t)]^\alpha [\eta_{a,j}]^\beta} & \text{if } b \in Succ(a) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Where  $\eta_{a,b}$  is an estimate of desirability of the transition  $a,b$  according to the apparent tardiness cost (ATCS) heuristic, Lee *et al.*, (1997),  $Succ(a)$  is the set of adjacent nodes to  $a$ ,  $\Phi$  is a random number in  $[0,1]$ , and  $\Phi_0$  is a tuning parameter. Here we simulate the route of an ant  $k$  through the graph by a two-level decision making, Dorigo & Gambardella, (1997). At first there is a draw of  $\Phi$ : if  $\Phi$  is lower than  $\Phi_0$  then the next node visited by  $k$  will have the maximum value of  $[\tau_{a,b}]^\alpha [\eta_{a,b}]^\beta$ , otherwise the probability will be determined with the equation number (5).

And after making a choice for an arc, we have to update the pheromone according to the algorithm. We do this with the formula:

$$\tau_{a,b} = (1 - \theta)\tau_{a,b} + \theta\tau_0 \quad (6)$$

Where  $\theta$  ( $0 < \theta < 1$ ) is the local pheromone decay parameter and  $\tau_0$  is the initial amount of pheromone deposited on each arc. In our case we consider  $\tau_0 = (\sum_{j=1}^n T_j)_{EDD}^{-1}$  where  $(\sum_{j=1}^n T_j)_{EDD}$  is the total tardiness given by the Early Due Date. Once all of the ants have completed their path, the intensity of pheromone on each arc is update according to below:

$$\tau_{a,b} = (1 - \lambda)\tau_{a,b} + \lambda \sum \Delta \tau_{a,b}^k \quad (7)$$

Where  $\Delta \tau_{a,b}^k$  calculated by equation (2) is the pheromone currently laid by ant  $k$ , and  $\lambda$  is the evaporation rate of previous pheromone intensity ( $0 < \lambda < 1$ ).

Finally the authors compare this algorithm to Genetic Algorithm. In order to compare them to each other, the authors have tested these algorithms on 900 different instances, and they compare the computation time took by ACS and GA. This is possible with the use of  $Cycle^*$  determined by  $Cycle^* = PWI \times Cycle_{max}$  where  $Cycle_{max}$  is the stopping criterion of the algorithm and  $PWI$  is coefficient showing the weight of  $Cycle_{max}$  between the two methods, here we choose  $Cycle_{max} = 3000$  for the ACS and  $Cycle_{max} = 1000$  for the GA, these values represent the same amount of CPU time.

Finally we obtain better results with ACS than Genetic Algorithm. According to the Figure 5 it can be seen that this the Ant Colony System based algorithm improve its result at each iteration rather than the Genetic Algorithm does.

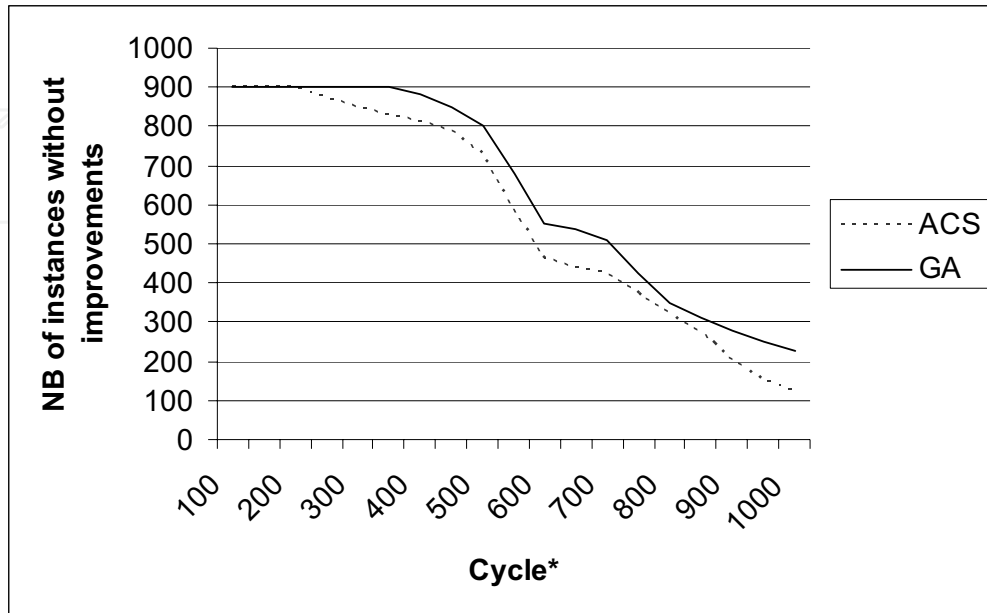


Figure 5. Comparison between ACS and GA (Nait Tahar *et al.*(2005)) on 900 instances

To conclude, we have introduced an interesting ant colony system for hybrid job shop scheduling problem with sequence dependent setup times and release dates to minimize the total tardiness, encountered in industrial situation. The ant colony method proved to be very efficient for randomly generated and real instances compared to a genetic algorithm.

#### 4.3 Non Dominated Sorting Genetic Algorithm

This part presents an optimization technique built by coupling the ARENA®, Kelton *et al.*, (2003) simulation software with a multi-objective optimizer based on the second version of a nondominated sorting genetic algorithm (NSGA-II) coded in Visual Basic for Application (VBA). This simulation-based-optimization technique is used to optimize the performances of the simulation model representing the considered workshop (the same study case adopted for the ACS and the GA) by testing new scheduling rules different from the only First In First Out (FIFO) rule which was adopted for the machines.

This work was developed in order to assess, by the means of a simulation software, the production system and to have a comprehensive tool in which the whole system's constraints will be handled as well as those of the logistical and handling system. These additional constraints have required a powerful simulation tool to manage them. In addition to that, the stochastic nature of some system's parameters (like the downtime of machines, the arrival times of products or others) makes analytical models very complicated or computationally intractable. That is why we have decided to use the simulation based optimization technique as it has been proved to be effective for such kind of applications.

Indeed, simulation is more and more used in today's industries with the aim of assessing their systems or to study the impact of changing system design parameters, Muhl *et al.*, (2003) and Sahlin *et al.*, (2004). ARENA®, developed by Systems Modelling Corporation, is

one of the softwares that can be used to model industrial systems in different domains like automobile, aeronautics as well as others like hospitals, banks, ...Kelton *et al.*, (2003).

While simulation makes it possible to test potential changes in an existing system without disturbing it or to evaluate the design of a new system without building it, simulation based optimization can be defined by coupling an optimization method with simulation in order to test many parameters that can maximize the performances of the simulated system, Hani *et al.*, (2006).

The coupling of heuristic methods with the ARENA® software, or other simulation software, was the subject of many works like Harmonosky (1995), Drake & Smith (1996) or others.

The second version of the non-dominated sorting genetic algorithm (NSGA-II) is a heuristic algorithm based on the genetic techniques applied by Goldberg (1989) and it was initially implemented by Deb *et al.*, (2000). It is based on the principle of the genetic algorithms by means of creating an initial population, selecting parents in order to get children and finally choose the best solution constructed from genes.

In addition to that, it consists on affecting fronts or groups to the proposed solutions. Front 1 contains the non-dominated solutions of the created population. Those individuals or solutions are then virtually removed from the population. We compare the remaining solutions and the next set of non-dominated solutions is assigned to front 2 and so on until that each individual of the population is affected to a front. Many works and researches have as a main subject the impact of NSGA algorithms on different optimization problems such as in Dolgui *et al.* (2005) for balancing and optimizing production lines or in Deb and Reddy (2003) and Deb *et al.* (2004).

Our model allows to simulate the production system with graphical animations starting from the exit of raw materials form the warehouse and until the exit of finished products while modeling their circulations which will be really done by the means of Auto Guided Vehicles (AGV). All the machines were modelled. In addition to the machines and handling systems' characteristics, the model contains as inputs:

- the workshop structure (production areas, warehouse and stock zones ...) on the scale,
- the different job sequences which guide the products forward between the servers,
- the simulation horizon (one year),
- the statistical law representing time between product arrivals.

In order to validate the simulation and to evaluate the production system, performances indicators were introduced in the model and they were compared to the real indicators adopted in the workshop. These indicators to optimize are:

1. The performance rate of each machine ( $g_1(k)$ ) (to be maximized)
2. The occupation rate of each machine ( $g_2(k)$ ) (to be maximized)
3. The total tardiness time resulting from processing all the jobs on a considered horizon of time  $T$  (to be minimized)

The multi objective optimization consists of finding the optimal objective function vector,  $g(k) = [g_1(k), g_2(k), T]$ , instead of a unique objective function. It aims at finding a compromise between the set of objectives.

We try to optimize those objectives by choosing the best priority for each queue of the considered machines. The multiplicity of choices could lead to results which are better in the case of using a unique rule as it was shown in the paper of Liu and Wu (2004). Until the beginning of this work, only one rule was tested: FIFO (First In First Out).



In our work, four priority rules were adopted to be tested as a first step: FIFO, LIFO (Last In First Out), SPT (Shortest Processing Time) and LPT (Longest Processing Time). The results obtained by the optimization algorithm will help us to make the final choice.

We present in this part the different properties of the developed algorithm.

A chromosome specifies the scheduling rule for each machine. The number of genes in the chromosomes of our algorithm is equal to the number of machines. The first step of our algorithm is then to create an initial population. The value of each gene  $k_i$  is generated randomly based on a uniform distribution  $U[K_{imin}, K_{imax}]$ .  $K_{imin}$  and  $K_{imax}$  are respectively the minimum and the maximum possible values of  $K_i$ . As we have four policies then  $K_{imin} = 1$  and  $K_{imax} = 4$ . The scheduling rules and the corresponding numbers are shown in table 4.

| Priority rule | Corresponding number |
|---------------|----------------------|
| FIFO          | 1                    |
| LIFO          | 2                    |
| SPT           | 3                    |
| LPT           | 4                    |

Table 4. Priority rules

The binary tournament technique is used to select the parents: two solutions are randomly selected and the best one becomes the first parent. This process is repeated to get the second parent. We choose the two-point crossover operation with a high probability and a very small point mutation probability.

The steps of the algorithm, as shown in Fig. 6, were inspired from the work of Deb *et al.* (2000). The overall structure of the NSGA-II algorithm is presented in Fig. 7. For more details about the algorithm, reader is referred to Chehade *et al.* (2007).

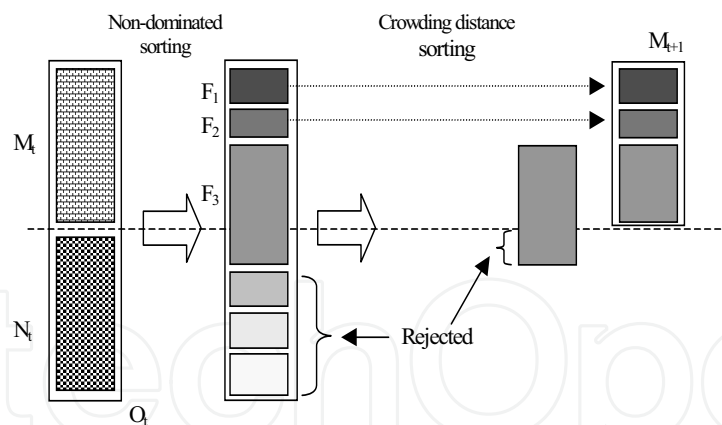


Figure 6. Steps of the NSGA-II algorithm (Deb *et al.*, 2000)

The simulation model and the optimization algorithm interact by means of a VBA procedure as the ARENA® software has a Visual Basic Editor. The coupling process works in the following way:

1. The algorithm starts by executing the first steps of the NSGA-II, to generate an initial population ( $M_t$ ) of size  $ns$

2. In order to calculate the fitness functions for the individuals of ( $M_t$ ), the simulation model is launched on  $ns$  iterations. Each iteration is supposed to calculate one fitness function corresponding to one individual (chromosome). The queues' priority rules of each machine are read directly from the chromosomes of the algorithm
3. Rank solutions in ( $M_t$ )
4. The algorithm creates then the offspring population  $N_t$  (of size  $ns$ ) by processing genetic functions (selection of parents, crossover, mutation)
5. The simulation model does  $ns$  new iterations in order to evaluate the chromosomes of the ( $N_t$ ) population. At this stage, we have now the population ( $O_t$ ) of size  $2ns$
6. Rank solutions in ( $O_t$ )
7. The algorithm executes its remaining steps (decomposition into fronts, crowding distance sorting). We have now the new parent population ( $M_{t+1}$ )
8. Repeat steps 4 to 7 till the stopping criteria is reached (which is in our case the number of generations)

```

Create the initial population  $M_t$  of size  $ns$ 
Evaluate the  $ns$  solutions using simulation
Sort  $M_t$  by non domination
Compute the crowding distance of each solution
REPEAT
Creation of the offspring population  $N_t$ : add  $n$  children at the end of  $M_t$  (with genetic
operators: selection, crossover and mutation of two parents) and evaluate each
solution by simulation
    Sort  $N_t$  by non domination
    Compute the crowding distance of each solution
    Sort the resulting population  $O_t$  of  $2*ns$  solutions by non-domination
     $M_{t+1} = 0$ ;
     $i = 1$ 
    WHILE  $|M_{t+1}| + |front(i)| \leq n$  do
        Add  $front(i)$  to  $M_{t+1}$ 
         $i = i + 1$ 
    END WHILE
     $missing = n - |M_{t+1}|$ 
    IF  $missing \neq 0$  THEN
        Sort the solutions by descending order of the crowding distance
        FOR  $j = 1$  to  $missing$  DO
            Add the  $j^{th}$  solution of  $front(i)$  to  $M_{t+1}$ 
        END FOR
         $P = M_{t+1}$ 
    END IF
UNTIL Stopping Criterion
    
```

Algorithm 7. Overall Structure of the NSGA-II algorithm (Deb *et al.*, 2000)

Table 5 shows a comparison between the real industrial data (RID) and the first results of simulation (SIM) initially get without applying the NSGA-II algorithm. It shows that a very

small gap for the three performances indicators was noticed which is a very good basis to realize later the optimization procedures.

Final results obtained after 100 generations showed that individuals of the last population are distributed on three fronts. Table 6 shows the optimization results which are compared to previous simulation results (SIM). The three objectives presented in the table are the performance rate of the machines (PR), the occupation rate of the machines (OR) and the total tardiness time (TT). It shows first the average result for each objective (OABF), the best (BIBF) and worst individual for each objective (WIBF), which gives an idea about the distribution of the individuals in this front. The numbers in brackets for OABF, BIBF and WIBF represent the difference between those parameters and the simulation results (SIM). The last row shows the standard deviation (STD) of those non-dominated individuals.

Table 6 shows results where each simulation iteration is set to cover a production horizon of ten years. We adopted a warm-up period of two years. The size of the initial population is 20, the number of generations is 100, the crossover probability is 0.9 and the mutation probability is 0.01. The average of the numerical results of the best front shows that the performance rate and occupation rate are improved by 6.28% and 12.7% respectively. As for the third indicator which is total tardiness, it is reduced by 48.3% on average.

As a consequence, the algorithm has showed that it has considerable improvements on the performances of the model.

|                             | RID  | SIM  | GAP (%) |
|-----------------------------|------|------|---------|
| <b>Performance rate (%)</b> | 56   | 57.2 | 2       |
| <b>Occupation rate (%)</b>  | 75   | 74.4 | 0.8     |
| <b>Total tardiness</b>      | 53.6 | 53.1 | 0.9     |

Table 5. Simulation results compared to real industrial data

|             | PR(%)          | OR(%)         | TT(hours)      |
|-------------|----------------|---------------|----------------|
| <b>SIM</b>  | 57.2           | 74.4          | 53.1           |
| <b>OABF</b> | 63.48 (+6.28%) | 87.1 (+12.7%) | 27.6 (-48.3%)  |
| <b>BIBF</b> | 68.9 (+11.7%)  | 94.8 (+20.4%) | 19.9 (-62.53%) |
| <b>WIBF</b> | 57.8 (+0.6%)   | 76.9 (+2.5%)  | 36.2 (-31.8%)  |
| <b>STD</b>  | 4.63           | 6.96          | 6.15           |

Table 6. Optimization results

## 5. Conclusion

In this chapter we have presented different results useful for scheduling tasks through a hybrid job shop system. At first we have dealt with the parallel machine job shop since its structure is near from the multi processors stages of a Hybrid Job Shop. Then we have presented some theoretical results and their application in the industry. We have developed some examples of modeling industrial lines for a genetic application or an ant colony system application. After this step of modeling, the result show real improvements of the minimization of the total tardiness in an industrial case. These results could be very useful in the semiconductor manufacturing or in the paper industries since the Hybrid Job Shop structure seem to be common in this area.

## 6. References

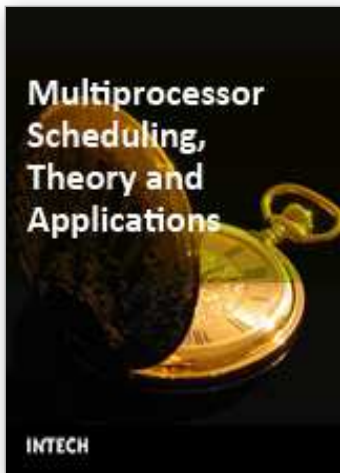
- Alidaee, B. & Rosa, D. (1997). Scheduling parallel machines to minimize the total weighted and un-weighted tardiness. *Computers and Operations Research*, 24, 4, 1997, pp. 775-788.
- Alvarez-Valdes, R., Fuertes, A., Tamarit, J.M., Giménez, G. & Ramos, R. (2005). A heuristic to schedule flexible job-shop in a glass factory. *European Journal of Operational Research*, 165, 2, 2005, pp. 525 - 534.
- Anghinolfi, D. & Paolucci, M. (2007). Parallel machine total tardiness scheduling with a new hybrid metaheuristic approach. *Computers & Operations Research*, 34, 11, November 2007, pp. 3471-3490.
- Arkin, E.M. & Roundy, R.O. (1991). Weighted tardiness scheduling on parallel machines with proportional weights. *Operations Research*, 39, 11, 1991, pp. 64-81.
- Armentano, V.A. & de França Filho, M.F. (2007). Minimizing total tardiness in parallel machine scheduling with setup times: An adaptive memory-based GRASP approach. *European Journal of Operational Research*, 183, 1, November 2007, pp. 100-114.
- Armentano, V.A. & Yamashita, D.S. (2000). Tabu search for scheduling identical parallel machines to minimize mean tardiness. *Journal of intelligent manufacturing*, 11, 4, 2000, pp. 453-460.
- Azizoglu, M. & Kirca, O. (1998). Tardiness minimization on parallel machines. *International Journal of Production Economics*, 55, 1998, pp. 163-168.
- Baker, K.R. & Bertrand, J.W. (1982). A dynamic priority rule for sequencing against due-date. *Journal of Operational Management*, 3, 1982, pp. 37-42.
- Barnes, J.W. & Brennan, J.J. (1977). An improved algorithm for independent jobs to reduce the mean finishing time. *AIIE Transactions*, 17, 1977, pp. 382-387.
- Chan, F.; Wong, T. & Chan, L. (2006). Flexible job-shop scheduling problem under resource constraints. *International Journal of Production Research*, 44, 11, 2006, pp. 2071 - 2089.
- Cehade, H., Amodeo, L., & Yalaoui, F. (2006). Simulation based optimization of a printing workshop using NSGA-II. *META'06*, n°576 (CDROM Proceedings), November 2006, Hammamet.
- Chen, K., Wong, J.S. & Ho, J.C. (1997). A heuristic algorithm to minimize tardiness for parallel machines. *Proceedings of ISMM International Conference*, pp. 118-121, Anaheim, 1997.
- Chen, H.; Ihlow, J. & Lehmann, C. (1999). A genetic algorithm for flexible job-shop scheduling. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2, 1999, pp. 1120 - 1125.
- Dauzère-Pérès, S.; Roux, J. & Lasserre, J.B. (1998). Multi-resource shop scheduling with resource flexibility. *European Journal of Operational Research*, 107, 1998, pp. 289 - 305.
- Deb K., Agrawal S.; Pratab, A. & Meyarivan, T. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pp. 849 - 858, 2000, Paris, France.
- Deb, K. & Reddy, A.R. (2003). Reliable classification of two-class cancer data using evolutionary algorithms. *Biosystems*, 72, 2003, pp. 111 - 129.

- Deb, K., Mitra, K.; Dewri, R. & Majumdar, S. (2004). Towards a better understanding of the epoxy-polymerization process using multi-objective evolutionary computation. *Chemical engineering science*, 59, 2004, pp. 4261 – 4277.
- Dessouky, M.I. (1998). Scheduling identical Jobs with unequal ready times on parallel machines to minimize the total lateness. *Computers and Industrial Engineering*, 34, 1998, pp. 793-806.
- Dogramaci, A. & Surkis, J. (1979). Evaluation of a heuristics for scheduling independent jobs on parallel identical processors. *Management Science*, 25, 1979, pp. 1208-1216.
- Dolgui, A., Makdessian, L. & Yalaoui, F. (2005). Deux méthodes d'optimisation multicritère pour l'aide la conception d'une ligne de production. *Paper presented at the 6<sup>th</sup> conf. on industrial engineering*, Juin 2005, Besancon.
- Dorigo, M. & Gambardella L.M. (1997). Ant colonies fort the travelling salesman problem. *Biosystems*, 43, 1997, pp. 73-81.
- Drake, G. & Smith, J. (1996). Simulation system for real-time planning, scheduling, and control. *In Proceedings of WSC96*, December 1996, Coronado.
- ElMaghraby, S.E. & Park, S.H. (1974). Scheduling jobs on a number of identical machines. *AIIE Transactions*, 6, 1974, pp. 1-13
- Emmons, H. (1987). Scheduling to a common due date on parallel processors. *Naval Research Logistics*, 34, 1987, pp. 803-810
- Emmons, H. & Pinedo, M. (1990). Scheduling stochastic jobs with due dates on parallel machines. *European Journal of operational research*, 47, 1990, pp. 49-55
- Eom, D.H., Shin, H.J., Kwun, I.H., Shim, J.K. & Kim, S.S. (2002). Scheduling jobs on parallel machines with sequence-dependent setup time. *International Journal of Advanced Manufacturing technology*, 19, 2002, pp. 926-932.
- Goldberg, D.E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Kluwer Academic Publishers, ISBN: 978-0201157673, Boston.
- Gomes, M., Barbosa-Póvoa, A. & Novais, A. (2005). Optimal scheduling for flexible job shop operation. *International Journal of Production Research*, 43, 11, 2005, pp. 2323 – 2353.
- Gourgand, M., Grangeon, N. & Norre, S. (2001). Modèle générique orienté objet du flow shop hybride hierarchies. *Proceedings of MOSIM'01*, pp. 583 – 590, April 2001, Troyes.
- Garey, M. & Johnson, D. (1979). *Computers and intractability: A guide to the theory of NP completeness*. Freeman, ISBN 978-0716710455, San Francisco.
- Guinet, A. (1995). Scheduling independent jobs on parallel machines to minimize tardiness criteria. *Journal of Intelligent Manufacturing*, 6, 1995, pp. 95-103
- Hani, Y., Chehade, H., Amodeo, L. & Yalaoui, F. (2006). Simulation based optimization of a train maintenance facility model using genetic algorithms. *In Proceedings of the IEEE/SSSM'06*, October 2006, Troyes.
- Harmonosky, C. (1995). Simulation based real-time scheduling: review of recent developments. *In Proceedings of WSC95*, December 1995, Arlington.
- Ho, J.C. & Chang, Y.-L. (1991). Heuristics for minimizing mean tardiness for m parallel machine. *Naval Research Logistics*, 38, 1991, pp. 367-381
- Kelton, W., Sadowski, R., & Sturrock, D. (2003). *Simulation with Arena*. McGraw- Hill, ISBN: 978-0072919813, New York.

- Koulamas, C. (1994). The total tardiness problem: review and extension. *Operations Research*, 42, 1994, pp. 764-775.
- Koulamas, C. (1997). Decomposition and hybrid simulated annealing heuristics for the parallel-machine total tardiness problem. *Naval Research Logistics*, 44, 1997, pp. 109-125
- Lawler, E.L. (1964). On scheduling problems with deferral costs. *Management Science*, 11, 1964, pp. 280-288.
- Lee, Y.H. & Pinedo, M. (1997). Scheduling jobs on parallel machine with sequence-dependent setup. *European Journal of Operational Research*, 100, 1997, pp. 464-474
- Lee, Y.H., Bhaskaran, K. & Pinedo, M. (1997). A heuristic to minimize the total weighted tardiness with sequence dependent setups. *IIE transactions*, 29, 1997, pp. 45-52
- Liu, M. & Wu, C. (2004). Genetic algorithm using sequence rule chain for multiobjective optimization in re-entrant micro-electronic production line. *Robotics and Computer Integrated Manufacturing*, 20, 2004, pp. 225 - 236.
- Logendran, R., McDonell, B., & Smucker, B. (2007). Scheduling unrelated parallel machines with sequence-dependent setups. *Computers & Operations Research*, 34, 11, November 2007, pp. 3420-3438.
- Loukil, T., Teghem, J. & Fortemps, P. (2007). A multi-objective production scheduling case study solved by simulated annealing. *European Journal of Operational Research*, 179, 3, 2007, pp. 709 - 722.
- Luh, P.B., Hoiomt, D.J., Max, E., & Pattipati, K.R. (1990). Schedule generation and reconfiguration for parallel machines. *IEEE Transactions on Robotics and Automation*, 6, 1990, pp. 687-696.
- Mönch, L. & Unbehaun, R. (2007). Decomposition heuristics for minimizing earliness-tardiness on parallel burn-in ovens with a common due date. *Computers & Operations Research*, 34, 11, November 2007, pp. 3380-3396.
- Montagne, E.R.Jr. (1969). Sequencing with time delay costs. Arizona State University, Engineering Research Bulletin, 1969, Technical report 5.
- Muhl, E.; Charpentier, P. & Chaxel, F. (2003). Optimization of physical flows in an automotive manufacturing plant: some experiments and issues. *Engineering Applications of Artificial Intelligence*, 16, 2003, pp. 293 - 305.
- Nait Tahar, D., Yalaoui, F., Amodeo, L. & Chu, C. (2004). Hybrid job-shop scheduling in a printing workshop, *MOSIM'04*, pp. 775-762, Ecole des Mines de Nantes-France, September 2004, Nantes.
- Nait Tahar, D., Yalaoui, F., Amodeo, L. & Chu, C. (2005). An ant colony system minimizing total tardiness for hybrid job shop scheduling problem with sequence dependent setup times and release dates. *International Conference on Industrial Engineering and Systems Management 2005*, n°10 (CDROM Proceedings), May 2005, Marrakech.
- Panwalker, S.S., Smith, M.L. & Koulamas, C. (1993). A heuristic for the single machine total tardiness problem, *European Journal of Operational Research*, 70, 1993, pp. 304-310.
- Park, M.W. & Kim, Y.D. (1997) Search heuristic for a parallel machine scheduling problem with ready time and due dates. *Computers and Industrial Engineering*, 33, 1997, pp. 793-796.
- Penz, B. (1996). Une méthode heuristique pour la résolution des problèmes de job-shop flexible. *Journal européen des systèmes automatisés*, 30, 6, 1996, pp. 767 - 780.



- Potts, C.N. & Van Wassenhove, L.N. (1997). Single machine tardiness sequencing heuristics. *IIE Transactions*, 23, 1997, pp. 346-354.
- Pritsker, A.A.B., Walters, L.J., & Wolf, P.M. (1969). Multi-project scheduling with limited resources: A zero-one programming approach. *Management Science*, 16, 1969, pp. 93-108
- Radhawa, S.U. & Kuo, C.H. (1997) Evaluating scheduling heuristics for non-identical parallel processors. *International Journal of Production Research*, 35, 1997, pp. 969-981.
- Sahlin, P., Eriksson, L., Grozman, P., Johnsson, H., Shapovalov, A. & Vuolle, M. (2004). Whole building simulation with symbolic DAE equations and general purpose solvers. *Building and Environment*, 39, 2004, pp. 949 - 958.
- Scrich, C., Armentano, V. & Laguna, M. (2004). Tardiness minimization in a flexible job shop: A tabu search approach. *Journal of Intelligent Manufacturing*, 15, 1, 2004, pp. 103 - 115.
- Shim, A.-O. & Kim, Y.-D. (2007). Scheduling on parallel identical machine to minimize total tardiness. *European Journal of Operational Research*, 177, 2007, pp. 135-146.
- Shutten, J.M.J. & Leussink, R.A.M. (1996). Parallel machine scheduling with release dates, due dates and family setup times. *International Journal of Production Economics*, 46, 47, 1996, pp. 119-125.
- Wilkerson, L.J. & Irwin, J.D. (1971). An improved algorithm for scheduling independent jobs. *AIIE Transactions*, 3, 1971, pp. 239-245.
- Xia, W. & Wu, Z. (2005). An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, 48, 2, 2005, pp. 409 - 425.
- Yalaoui, F. & Chu, C. (2002). Parallel machine scheduling to minimize total tardiness. *International Journal of Production Economics*, 76, 3, 2002, pp. 265-279
- Yalaoui, F. & Chu, C. (2003). An efficient heuristic approach for parallel machine scheduling with job splitting and sequence dependent setup times. *IIE Transactions*, 35, 2, 2003, pp. 183 - 190.



## **Multiprocessor Scheduling, Theory and Applications**

Edited by Eugene Levner

ISBN 978-3-902613-02-8

Hard cover, 436 pages

**Publisher** I-Tech Education and Publishing

**Published online** 01, December, 2007

**Published in print edition** December, 2007

A major goal of the book is to continue a good tradition - to bring together reputable researchers from different countries in order to provide a comprehensive coverage of advanced and modern topics in scheduling not yet reflected by other books. The virtual consortium of the authors has been created by using electronic exchanges; it comprises 50 authors from 18 different countries who have submitted 23 contributions to this collective product. In this sense, the volume can be added to a bookshelf with similar collective publications in scheduling, started by Coffman (1976) and successfully continued by Chretienne et al. (1995), Gutin and Punnen (2002), and Leung (2004). This volume contains four major parts that cover the following directions: the state of the art in theory and algorithms for classical and non-standard scheduling problems; new exact optimization algorithms, approximation algorithms with performance guarantees, heuristics and metaheuristics; novel models and approaches to scheduling; and, last but not least, several real-life applications and case studies.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Frederic Dugardin, Hicham Chehade, Lionel Amodeo, Farouk Yalaoui and Christian Prins (2007). Hybrid Job Shop and Parallel Machine Scheduling Problems: Minimization of Total Tardiness Criterion, Multiprocessor Scheduling, Theory and Applications, Eugene Levner (Ed.), ISBN: 978-3-902613-02-8, InTech, Available from: [http://www.intechopen.com/books/multiprocessor\\_scheduling\\_theory\\_and\\_applications/hybrid\\_job\\_shop\\_and\\_parallel\\_machine\\_scheduling\\_problems\\_\\_minimization\\_of\\_total\\_tardiness\\_criterion](http://www.intechopen.com/books/multiprocessor_scheduling_theory_and_applications/hybrid_job_shop_and_parallel_machine_scheduling_problems__minimization_of_total_tardiness_criterion)

# **INTECH**

open science | open minds

### **InTech Europe**

University Campus STeP Ri

Slavka Krautzeka 83/A

51000 Rijeka, Croatia

Phone: +385 (51) 770 447

Fax: +385 (51) 686 166

[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai

No.65, Yan An Road (West), Shanghai, 200040, China

中国上海市延安西路65号上海国际贵都大饭店办公楼405单元

Phone: +86-21-62489820

Fax: +86-21-62489821

© 2007 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen