

A new method coupling simulation and a hybrid metaheuristic to solve a multiobjective hybrid flowshop scheduling problem

Xiaohui Li, Hicham Chehade, Farouk Yalaoui, Lionel Amodeo

University of Technology of TroyesBP2060
Institut Charles Delaunay-LOSI, STMR UMR CNRS 6279
12 Rue Marie Curie, 10000, Troyes Cedex, France

Abstract

In this paper, we have studied a multiobjective hybrid flowshop scheduling problem where n independent jobs should be executed in a hybrid assembly line. The aim of our work is to optimize the makespan and the total tardiness of the whole production. A simulation based optimization algorithm is proposed here to solve this problem. It is a combination of the simulation software ARENA and the FLC-NSGA-II optimization method. The latter uses a fuzzy logic controller to adjust the crossover and the mutation probabilities, in order to enhance the research ability of the traditional NSGA-II algorithm. This method is first compared with the industrial solutions, and then with NSGA-II. The result shows the efficiency and the feasibility of our proposed method.

Keywords: Hybrid flowshop, Multiobjective optimization, Fuzzy logic, NSGA-II, ARENA, Simulation

1. Introduction

The hybrid flowshop scheduling problem consists of arranging a set of n independent jobs on a production line which is constructed by several stages of machines. Some stages may have only one machine, but at least one stage must have more than one machine in parallel. All jobs go through the stages of a machine with the same order. This sequence is not alterable. This kind of workshop is widely appeared in the manufacturing environment such as electronic and petrolic manufacturing. It has been widely studied in many works in the literature. The makespan is the most used criterion for the scheduling problem, such as in the works of [1], [2], [3], [4] and [5] where authors have concerned the minimization of the makespan for the hybrid flowshop scheduling problem. In addition to the makespan, other criteria may be also considered: the total flowtime [6], the maximum tardiness [7], the total completion times [8] and the total tardiness [9][10]. These works have concerned the single criterion optimization.

The real world scheduling problems are generally multiobjective optimization problems where several different criteria should be optimized simultaneously. Multiobjective optimization consists of finding a set of non-dominated solutions which optimize the two or more demanded objectives instead of only one optimal solution. This relationship is called the Pareto dominance relationship. Recently many Pareto based approaches are proposed to solve a multiobjective optimization problems such as the Multiple Objective Genetic Algorithm (MOGA) [11], the Niche Pareto Genetic Algorithm (NPGA) [12], the Non-dominated Sorting Genetic Algorithm (NSGA-II) [13], the Strength Pareto Evolutionary Algorithm (SPEA-II) [14], the Pareto Archived Evolution Strategy (PAES) [15], the Pareto Envelope based Selection Algorithm (PESA) [16]. To solve the multiobjective hybrid flowshop scheduling problem, Behnamian and Ghomi [17] have proposed a hybrid metaheuristic in 2009, and a multi phase covering the Pareto optimal front hybrid metaheuristic in 2010 [18]. Dugardin *et al.* [19] have proposed a Lorenz dominance based NSGA-II (L-NSGA) to solve a reentrant hybrid flowshop scheduling problem. Khalouli *et al.* [20] have proposed an ant colony method. Kahraman *et al.* [21] have proposed a parallel greedy algorithm. Karimi *et al.* [22] have developed a multi phase genetic algorithm and finally, Naderi *et al.* [23] have proposed an improved simulated annealing algorithm.

Due to the difficulty of the parameters settings of metaheuristics in general and therefore of genetic algorithms, we have developed a fuzzy logic controller to better set the probabilities of crossover and mutation. Fuzzy logic was first proposed by Zadeh [24] for system control. In the works of [25][26] and [27], the fuzzy logic is used to enhance genetic algorithms to solve a single objective problems. Lau *et al.* [28][29] have proposed the FL-GA, FL-NSGA2, FL-SPEA2 and FL-MICROGA to solve a VRP problem (vehicle routing problem). Yalaoui *et al.* [30] have developed the FLC-GA and FLC-NSGA-II for the reentrant scheduling problem. In our previous works [31][32], we have

developed FLC-NSGA-II and FLC-SPEA-II algorithms to solve a multiobjective parallel machines scheduling problem.

The development of a mathematical modeling and the evaluation for the hybrid flowshop scheduling problem is very difficult especially when several additional constraints are considered and when the system has a stochastic nature. The ARENA software is a powerful tool to model and simulate the considered problem. In the works of [33] and [34], the authors have developed a simulation-based-optimization technique (the NSGA-II is coupled with the ARENA software) for a printing workshop and a hybrid flowshop respectively. The main contribution of this paper is that we have proposed a new method which couple the simulation software and the advanced metaheuristic for the first time. In this work, the FLC-NSGA-II algorithm is coupled with the ARENA environment, in order to have a better research ability. To show the advantages of this method, the simulation based FLC-NSGA-II algorithm is compared with the industrial solution and the classical NSGA-II algorithm.

The rest of this paper is organized as follows. In the next section, more details about the studied problem are presented. The fuzzy logic controlled genetic algorithm (FLC-NSGA-II) and the simulation based FLC-NSGA-II are presented in section 3 and section 4 respectively. Section 5 shows the measuring criteria and the experimental results. The conclusion and the perspectives are given in section 6.

2. Problem description

We have studied the same problem which is presented in our previous works [35][34]. In a hybrid flowshop scheduling problem, a set of n independent jobs should be executed in a workshop line which has f stages of machines and at least one stage must have more than one machine in parallel. We assume that no release dates of jobs are required. It means that all the customers orders arrive at time zero. Each job j consists of f parts of different operations which should be processed on f stages. The processing times of each operation of job j at stage t is defined as p_j^t . In addition, if job j has finished its work in the actual stage, a transfer time is required when it enters the next stage. This transfer time is decided by the distance between two machines. Some assumptions are presented as follows and they must be respected:

- All the jobs are available at time zero.
- All the machines in each stage are available at time zero.

- Each machine can execute only one job at the same time.
- Each job can be processed only once in each stage.
- Each job cannot be interrupted during its processing (The preemption is prohibited).
- Jobs are available for processing at a stage immediately after the finish at the previous stage.

In this work, two criteria are considered simultaneously to be optimized: the makespan and the total tardiness. The former presents the completion times of the whole production, and the latter presents the penalty of the production. The aim of our work is to find several non-dominated solutions which can optimize the considered objectives.

An assembly line from a real world case is considered in this work. This assembly line corresponds to a hybrid flowshop. It consists of three stages of machines. The first stage includes three parallel machines to fill the products in the bottles. The second stage has one machine for covering and labeling the bottles. Stage 3 has one machine to stock the bottles. The aim is to find several sequence solutions which enhance the production performances (the makespan and the total tardiness). This assembly line is presented in figure 1.

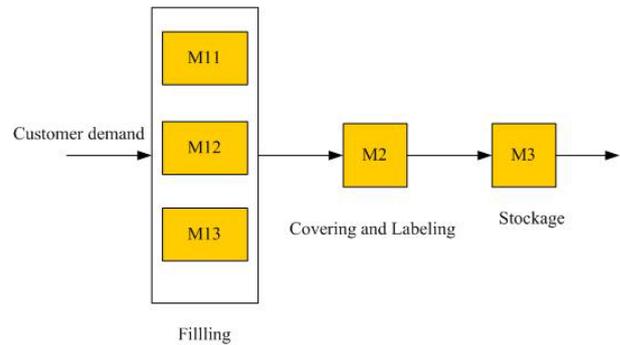


Figure 1: The studied assembly line[34]

3. Resolution methods

In this section, the classical NSGA-II and the fuzzy logic controlled NSGA-II (FLC-NSGA-II) are defined. The logic controller is applied to improve the parameters settings of the NSGA-II. Then the FLC-NSGA-II is coupled with the ARENA environment, this part is shown in the next section. The numerical results are given in section 5.

3.1. NSGA-II

The NSGA-II algorithm has been developed by Deb *et al.* [13] as a fast and efficient multiobjective genetic algorithm. The aim of this algorithm is to find a set of non-dominated solutions based on the Pareto dominance relationship. The main

concept of the NSGA-II is the creation of an initial population, the selection of parents, the creation of children and the finding of non-dominated solutions. At the beginning, an initial population is randomly created. In each iteration, the following processes are executed: the offspring population (the population of children) is generated by the operations selection, crossover and mutation. Then, find the best solutions between the current population and offspring population to the current population of next iteration (all the solutions are ranked in different non-dominated set and a crowding distance is computed to rank the solutions in the same set). This process is repeated until the stopping criterion is satisfied. The classical one point crossover, the one point mutation and the tournament selection are applied in this work.

A multidimensional chromosome is applied to encode our problem. In each gene, the elements present the index of job, the machine index and the corresponding position respectively. An example of this chromosome is presented in table 1. The initial parameters values are set based on the work of Chehade *et al.* [33] where a good explanation of the parameters setting is presented. These parameters are as follows:

- Size of the initial population: 20
- Number of generations: 50
- Probability of crossover: 0.9
- Probability of mutation: 0.1

Table 1: Example of the chromosome

j	1	2	3	4	5
k^1	1	3	2	1	3
r^1	1	2	1	2	1
k^2	1	2	2	1	1
r^2	1	2	1	3	2

3.2. Fuzzy logic

Generally, several parameters should be set at the beginning of a genetic algorithm. However, the parameters settings is very difficult, since it is always based on the knowledge and the experience of the users. In genetic algorithms, the crossover and the mutation probabilities determine the convergence and the diversity of the results. Appropriate probabilities setting can produce better results and avoid too fast falling into local optimum. For these reasons, a fuzzy logic controller is proposed in this work to improve them throughout the iterations. In this study, the average objectives values of the solutions in the Pareto front and their diversity are considered as the input of the fuzzy logic controller. The output of the controller is the

variation of the two probabilities Δp_c and Δp_m . In the works of [28],[29],[30],[31] and [32], the authors have proposed a combination of the fuzzy logic and different genetic algorithms to solve different optimization problems.

The fuzzy logic was first proposed by Zadeh [24] for system control. The fuzzy logic controller consists of three parts: *fuzzification, decision making and defuzzification*. Two fuzzy logic controllers are proposed in this study to enhance the crossover probability and the mutation probability respectively. The details of the three parts are presented in our previous work [31], such as the settings of the membership functions, the linguistic terms and the decision tables.

3.3. Fuzzy logic based NSGA-II

The fuzzy logic controller is applied to change the crossover probability and the mutation probability of the NSGA-II, in order to enhance its search ability. The process is executed each ten consecutive generations, for the purpose of providing a sufficient respond time for the probabilities modifications. The average fitness values and the diversity of the solutions in the non-dominated set are taken in consideration as the input of fuzzy logic controller. Thus, the variation of probabilities are given as the output. The same parameters concerned above are used as the initial parameters. The main structure of the proposed algorithm is shown in figure 2 and algorithm 1.

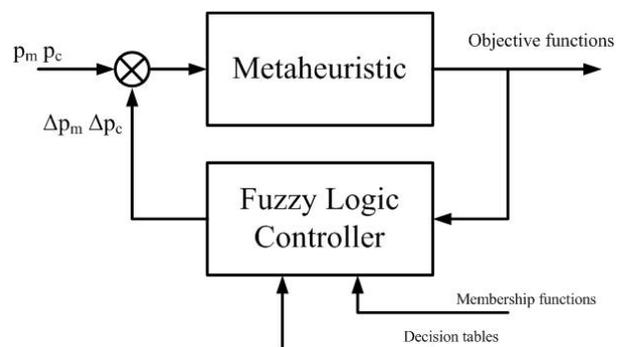


Figure 2: The structure of the FLC guided algorithm

Algorithm 1: Structure of the FLC-NSGA-II algorithm [31]

Generate the initial population P_0 of size N
 Evaluate these solutions
 Sort these solutions by non-dominated front and crowding distance
WHILE stopping criterion is not satisfied **DO**
 Create the offspring population Q_t (with the operations of selection, crossover and mutation), evaluate all the solutions
 Compose the populations of parents and the children in $R_t = P_t \cup Q_t$
 Sort the solutions of R_t in different non dominated fronts F_i by the Pareto dominance
 $P_{t+1} = 0$
 $i=1$
 WHILE $|P_{t+1}| + |F_i| < N$ **DO**
 $P_{t+1} \leftarrow P_{t+1} \cup F_i$
 $i=i+1$
 END WHILE
 Rank the solutions of F_i by the crowding distance, add $N - |P_{t+1}|$ solutions in P_{t+1} by descending order of the crowding distance
 IF $t \bmod 10 = 0$ **THEN**
 The values of $f(t-1)$ - $f(t-2)$ and $d(t-1)$ are considered as the input of FLC, update the p_c and p_m with the FLC
 END IF
END WHILE

4. Simulation based optimization method

First, we present the modeling and the performances evaluation of the studied assembly line with ARENA. Then, the simulation based FLC-NSGA-II that we have used coupling ARENA with FLC-NSGA-II is proposed to find the non-dominated solutions.

4.1. Modeling and performances evaluations with ARENA

The model of the studied assembly line is realized with the ARENA simulation environment. ARENA is a powerful discrete event simulation software taking in consideration the diversity of its applications and the facility to be coupled with the external modules. It can be used in many area such as industrial systems and service firms (hospitals, restaurants or supermarkets for example) [33]. ARENA uses the SIMAN language where the users can create a model by placing the module (boxes of different shapes) that represent the processes or the logic functions. The main processes to simulate a model with ARENA are shown as follows: First, create the model by placing the modules; then set the parameters of the modules and the parameters of the system; finally, add some performances indicators in order to analyze the system performances and choose the final solution.

The same model of our previous work [34] is applied here. All the machines are located on a loop conveyor and each machine executes its corresponding mission. The used modules are presented in table 2, where we also show their functions.

Table 2: The list of used modules for the ARENA simulation model

Module	Function
Create	Add the entity
VBA	Fire a VBA event
Assign	Assign model status variables
Readwrite	Read data from a file(write data to a file)
Decide	Choose one way by a condition
Dispose	Entity exits the system
Station	Define a station corresponding to a physical or a logic station
Seize	Allocate units of one or more resources to an entity
Delay	Delay an entity by a special amount of time
Release	Release the resource that an entity has seized
Route	Transfer an entity to a special station
Leave	Transfer an entity to a station by a conveyor or other

Several priority rules can be applied in the simulation such as the FIFO (First In First Out), LIFO (Last In First Out), SPT (Shortest Processing Time), LPT (Longest Processing Time) and EDD (Earliest Due Date). For example, if we have applied the FIFO priority rule, it means that if an operation with a lower priority arrives at a non seized machine, it will be executed immediately. But these priority rules cannot ensure obtaining the optimal solutions, and it cannot correspond to the chromosome (the wanted sequence). For solving this problem, a special preemption structure shown in figure 3 is proposed in this work. In the decide module, if the arriving entity is not the one we want, it will reenter the queue until all the entities with higher priority have been passed. This special structure ensures that the treated sequence of jobs and the chromosome are the same.

4.2. Simulation based FLC-NSGA-II

In most of studies, the genetic algorithm uses a mathematical model to evaluate the objective function. However, this method has several weakness: the mathematical modeling is too professional for those who are not researchers in the field of operations research; several constraints cannot be considered such as the stochastic phenomena like machines failures; it is not obvious and easily adoptive to model. The method of mathematical model can be solved simply and quickly; the method of simulation is complex and slow, but many constraint can be considered in this method. On the other side, the general priority rules cannot ensure obtaining the optimal solutions. That is

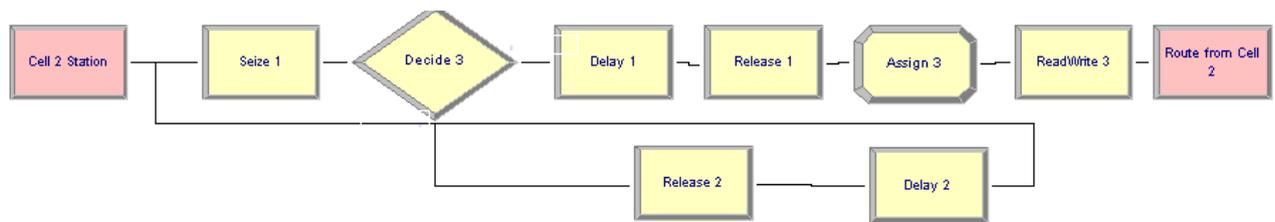


Figure 3: ARENA modules used to model a machine

why we propose the simulation based optimization method. The VBA (Visual Basic of Applications) tool is integrated in the ARENA environment. It provides a link between the simulation model and the optimization algorithm. The main conception of the ARENA based FLC-NSGA-II is presented in figure 4. At the beginning, the optimization algorithm provides the initial solutions to the simulation model, thus we can obtain the system performances (the makespan and the total tardiness) of each initial solution. Then, the simulation model sends these performances values to the optimization algorithm. The FLC-NSGA-II algorithm can compute several offspring solutions (the elitist of current solutions are always kept), and these solutions are considered as the input of the simulation model for the next iteration. At the same time, we improve the crossover probability and the mutation probability each ten consecutive iterations. This process is repeated until the stopping criterion is satisfied.

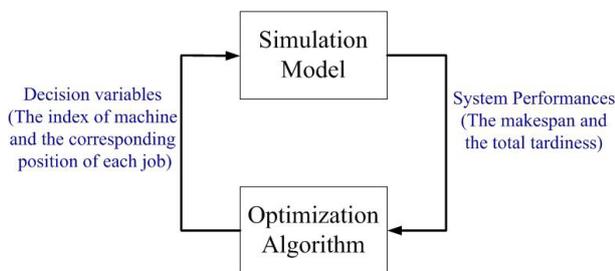


Figure 4: The main conception of simulation based optimization[34]

5. Experimental results

In this section, the optimization result of FLC-NSGA-II is compared with the industrial solution and the traditional NSGA-II. First, we present the measuring criteria for the comparison of the multiobjective optimization algorithms.

The stochastic nature of the model is applied in this work. The production times in the model is based on statistical laws by analyzing the real data from the assembly line. To solve this problem, the objective function is the means value of replications

where the simulation is repeated several times. For considering the system parameters setting: the number of replications is the product of the number of generation numbers, the size of two populations (current population and offspring population) and 10 (the replications for each evaluation); the simulation run time for each iteration is 20 hours; the warm up period of the system is set to zero.

5.1. Measuring criteria

To reach the optimization of the non-dominated set, it is necessary to get the following performances: the convergency, the diversity and the spread results [36]. In this work, three measuring criteria are applied to compare two non-dominated fronts: the number of non-dominated solutions, the measure of Zitzler[36] and the μ_d distance of Dugardin *et al.*[19]. In the measure of Zitzler, the value of $C1$ (respectively $C2$) measures the ratio of solutions in A (respectively B) which are dominated by at least one solution in B (respectively A). The μ_d distance is based on the Riise distance [37] which computes the distance between a solution on A and its orthogonal projection on B. The original Riise distance shows that if the set A is closer to the absolute optimal solutions than set B. In addition, the μ_d distance is the Riise distance divided by the the cross of the corresponding rectangle (see figure 5), it presents the proportional improvement to the diagonal of the rectangle.(see [19])

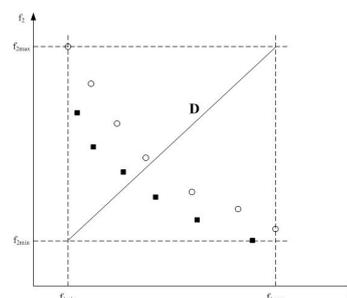


Figure 5: Distance between two fronts [19]

Table 3: The solutions objective values of the considered example

Index	FLC-NSGA-II		FIFO	
	C_{max}	$\sum T_j$	C_{max}	$\sum T_j$
1	42.06	103.18	-	-
2	42.70	99.35	-	-
3	42.81	96.50	-	-
4	43.05	94.46	-	-
5	-	-	43.15	135.24

5.2. Numerical results

First, the proposed method is compared with the general priority rule FIFO (First In First Out) which are used in industrial production. The comparison result is given in figure 6 where 50 jobs are to be executed on the studied assembly line. Table 3 shows the objective values of the obtained solutions of the FLC-NSGA-II algorithm and the FIFO rules. The first column shows the solution index, solutions one to four present the solutions obtained by the FLC-NSGA-II algorithm and solution five is obtained by the FIFO rule. We can observe that the advantage of FLC-NSGA-II is obvious. The values of each objective of the FLC-NSGA-II front have enhanced the performance. The best and the worst values of the makespan are reduced by 2.59% and 0.2% respectively, and reduced by 43.1% and 31.1% for those of total tardiness.

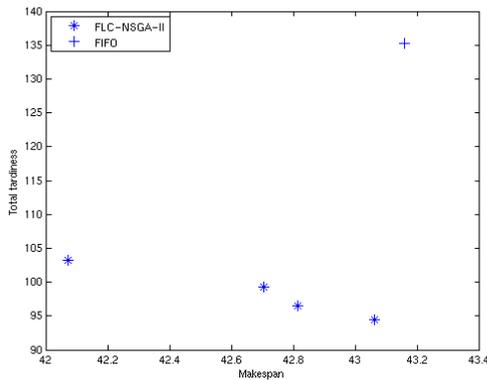


Figure 6: The comparison between FLC-NSGA-II and priority rules

In a previous work [34], we have proved the advantage of ARENA based NSGA-II by the comparison with industrial solutions and an exact method. In this work, we compare the FLC-NSGA-II algorithm with the original NSGA-II algorithm by coupling with ARENA. In addition, we consider a stochastic case instead of the deterministic case of [34]. We test 10 instances where 50 jobs should

be executed in the assembly line, all instances have different jobs data configurations. Table 4 shows the comparison results. The value of μ_d distance, the value of C1 and C2, the number of non-dominated solutions are shown in this table.

Firstly, The first line of table 4 is explained. In this problem, 50 jobs should be processed on the assembly line. The value of μ_d is equal to -0.020 shows that the obtained front of FLC-NSGA-II dominates the NSGA-II front. $C1 = 0$ and $C2 = 1$ show that there are none solutions in FLC-NSGA-II front that are dominated by at least one solution from NSGA-II front, and both solutions in NSGA-II are dominated by those of FLC-NSGA-II front. The FLC-NSGA-II fronts have more non-dominated solutions than NSGA-II fronts, corresponding the values of n_{s1} and n_{s2} . In second, the instance six is explained. The two Pareto fronts are very closed, the μ_d distance is approximately equal to zero. But one solution on the FLC-NSGA-II front is dominated by one solution on the NSGA-II front, hence the value of C1 is equal to 0.5. The value of C2 is equal to zero means that none solution on the NSGA-II front is dominated by the FLC-NSGA-II. The results of table 4 show the advantages of FLC-NSGA-II, since the results of FLC-NSGA-II are better than those of NSGA-II in almost all cases.

Table 4: Comparison results between FLC-NSGA-II and NSGA-II

index	μ_d	C1	C2	n_{s1}	n_{s2}
1	-0.020	0	1	3	2
2	-0.021	0	1	4	2
3	-0.001	0	0.667	4	3
4	-0.020	0	1	4	4
5	-0.007	0	0.5	2	2
6	0	0.5	0	2	3
7	-0.007	0	0.667	3	3
8	-0.025	0	1	3	2
9	-0.017	0	1	3	3
10	0.025	0	1	4	3

6. Conclusion

In this paper, we deal with a multiobjective hybrid flowshop scheduling problem where the makespan and the total tardiness are to be minimized at once. A simulation-based optimization method is proposed to solve this problem, which is a coupling of a fuzzy optimization algorithm FLC-NSGA-II and a simulation software ARENA. The computing results show the advantages of the proposed method by the comparisons with FIFO rule and the NSGA-II algorithm. It proves that the fuzzy

logic controller can enhance the search ability of the genetic algorithm. The ARENA based FLC-NSGA-II is efficient to solve multiobjective optimization problems with the simulation of the production. In our future works, it is interesting to develop another hybrid metaheuristic with a faster search ability, in order to decrease the computing time. A powerful exact method is also considered, in order to compare the results for larger instances.

7. Acknowledgment

This research was supported by the CRCA (Conseil Régional de Champagne Ardenne) and FEDER (Fonds Européen de Développement Régional in France).

References

- [1] T.S. Arthanari and K.G. Ramamurthy. An extension of two machines sequencing. *Opsearch*, 8:10–22, 1971.
- [2] S.A. Brah and K.G. Hunsucker. Branch and bound algorithm for the flow shop with multiple processors. *European Journal of Operation Research*, 51:88–99, 1991.
- [3] B. Chen. Analysis of classes of heuristics for scheduling research. *Journal of Operation Research*, 46:234–244, 1995.
- [4] A. Guinet, M.M. Solomon, P.K. Kedia, and A. Dussauchoy. A computational study of heuristics for two stage flexible flowshops. *International Journal of Production Research*, 34:1399–1415, 1996.
- [5] A. Vignier, P. Commandeur, and C. Proust. New lower bound for the hybrid flowshop scheduling problem. In *IEEE 6th International Conference on Emerging Technologies and Factory Automation Proceeding*, pages 446–451, 1997.
- [6] C. Rajendran and D. Chaudhuri. A multi-stage parallel-processor flowshop problem with minimum flowtime. *European Journal of Operations Research*, 57:111–122, 1992.
- [7] A. Vignier, J.C. Billaut, C. Proust, and V. Tkindt. Resolution of some 2-stage hybrid flowshop scheduling problems. In *IEEE 6th International Conference of Systems, Man and Cybernetics, Information Intelligence and Systems*, volume 4, pages 2934–2941, 1996.
- [8] A. Vignier, D. Dardilhac, D. Dezalay, and C. Proust. A branch and bound approach to minimize the total completion time in a k-stage hybrid flowshop. In *Proceedings 1996 IEEE Conference on Emerging Technologies and Factory Automation*, volume 1, pages 215–220, 1996.
- [9] S.W. Choi, Y.D. Kim, and G.C. Lee. Minimizing total tardiness of orders with reentrant lots in a hybrid flowshop. *International Journal of Production Research*, 43:2049–2067, 2005.
- [10] G.C. Lee and Y.D. Kim. A branch-and-bound algorithm for a two-stage hybrid flowshop scheduling problem. *International Journal of Production Research*, 42:4731–4743, 2004.
- [11] C. M. Fonseca and P.J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993.
- [12] J. Horn, N. Nafpliotis, and D.E. Goldberg. A niched pareto genetic algorithm for multi-objective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, volume 1 of *IEEE World Congress on Computational Computation*, pages 82–87, 1994.
- [13] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization : Nsga-ii. In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, 2000.
- [14] E. Zitzler, M. Laumanns, and L. Thiele. Spea2 : Improving the strength pareto evolutionary algorithm. In *Proceedings EUROGEN 2001 Evolutionary Methods for Design, Optimization and Control With Applications to Industrial Problems*, Athens, Greece, September 2001.
- [15] J.D. Knowles and D.W. Corne. Approximating the nondominated front using the pareto archived evolution strategie. *Evolutionary Computation*, 8(2):149–172, 2000.
- [16] D.W. Corne, J.D. Knowles, and M.J. Oates. The pareto envelope based selection algorithm for multiobjective optimization. In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 839–848, 2000.
- [17] J. Behnamian, S.M.T. Fatemi Ghomi, and M. Zandieh. A multi-phase covering pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic. *Expert System with Applications*, 36:11057–11069, 2009.
- [18] J. Behnamian and S.M.T. Fatemi Ghomi. Hybrid flowshop scheduling with machine and resource-dependent processing times. *Applied Mathematical Modelling*, 2010. (Article in press).
- [19] F. Dugardin, F. Yalaoui, and L. Amodeo. New multi-objective method to solve re-entrant hybrid flow shop scheduling problem. *European Journal of Operation Research*, 203(22-31), 2010.
- [20] S. Khalouli, F. Ghedjati, and A. Hamzaoui. A

- meta-heuristic approach to solve a jit scheduling problem in hybrid flow shop. *Engineering Applications of Artificial Intelligence*, 23:765–771, 2010.
- [21] C. Kahraman, O. Engin, I. Kaya, and R. Elif Öztürk. Multiprocessor task scheduling in multistage hybrid flow-shop: A parallel greedy algorithm approach. *Applied Soft Computing*, 10:1293–1300, 2010.
- [22] N. Karimi, M. Zandieh, and H.R. Karamooz. Bi-objective group scheduling in hybrid flexible flowshop: A multi-phase approach. *Expert System with Applications*, 37:4024–4032, 2010.
- [23] B. Naderi, M. Zandieh, and V. Roshanaei. Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness. *International Journal of Advanced Manufacturing Technology*, 41:1186–1198, 2009.
- [24] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
- [25] R.T.F.A. King, B. Radha, and H.C.S. Rughooputh. A fuzzy logic controlled genetic algorithm for optimal electrical distribution network reconfiguration. In *Proceedings of 2004 IEEE international conference on networking, sensing and control*, pages 577–582, 2004.
- [26] A.A. Lofti and F.H. Kashani. Bandwidth optimization of the e-shaped microstrip antenna using the genetic algorithm based on fuzzy decision making. In *Proceeding of 2004 IEEE antennas and propagation society international symposium*, pages 2333–2336, 2004.
- [27] Y.H. Song, H.S. Wang, P.Y. Wang, and A.T. Johns. Environmental/economic dispatch using fuzzy logic controlled genetic algorithms. *IEE Proceeding of Generation Transmission and Distribution*, 144(4):377–382, 1997.
- [28] H.C.W. Lau, T.M. Chan, W.T. Tsui, and G.T.S. HO. Cost optimization of the supply chain network using genetic algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 99(1), 2009.
- [29] H.C.W. Lau, T.M. Chan, W.T. Tsui, F.T.S. Chan, G.T.S. Ho, and K.L. Choy. A fuzzy guided multi-objective evolutionary algorithm model for solving transportation problem. *Expert Systems with Applications*, 36:8255–8268, 2009.
- [30] N. Yalaoui, F. Dugardin, F. Yalaoui, L. Amodeo, and H. Mahdi. *Production Engineering and Management under Fuzziness*, volume 252, chapter Fuzzy Project Scheduling, pages 143–170.
- [31] X. Li, F. Yalaoui, L. Amodeo, and H. Chehade. Metaheuristics and exact methods to solve a multiobjective parallel machines scheduling problem. *Journal of Intelligent Manufacturing*, July 2010. (Accepted, Available online).
- [32] X. Li, H. Chehade, F. Yalaoui, and L. Amodeo. Fuzzy logic controller based multiobjective metaheuristics to solve a parallel machines scheduling problem. *Journal of Multiple-Valued Logic and Soft Computing*, 2011. Accepted.
- [33] H. Chehade, F. Yalaoui, and L. Amodeo. A simulation based optimization method applied to a printing workshop case. *Advances In Management*, 1(3):42–49, 2008.
- [34] X. Li, H. Chehade, F. Yalaoui, and L. Amodeo. Simulation-based-optimization to solve a multiobjective hybrid flowshop scheduling problem. In *International Conference on Industrial Engineering and Systems Management 2011*, Metz, France, 2011. Accepted.
- [35] X. Li, H. Chehade, F. Yalaoui, and L. Amodeo. Lorenz dominance based metaheuristic to solve a hybrid flowshop scheduling problem with sequence dependent setup times. In *2011 International Conference on Communications, Computing and Control Applications (CCCA’11)*, 2011.
- [36] E. Zitzler and L. Thiele. Multi-objective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.
- [37] A. Riise. Comparing genetic algorithms and tabu search for multi-objective optimization. In *In Abstract conference proceedings*, page 29, Edinburgh, UK, July 2002.