Comparative Study of Heuristics Algorithms in Solving Flexible Job Shop Scheduling Problem with Condition Based Maintenance

Yahong Zheng[1], Lian Lian[2], Khaled Mesghouni[3]

[1]School of Transport, Wuhan University of Technology (China)
[2]School of Transportation and Logistics, Dalian University of Technology (China)
[3]LAGIS, UMR CNRS 8219, Ecole Centrale de Lille (France)

yahong.zheng@live.cn, lilian591@gmail.com, khaled.mesghouni@ec-lille.fr

Abstract:

Purpose: This paper focuses on a classic optimization problem in operations research, the flexible job shop scheduling problem (FJSP), to discuss the method to deal with uncertainty in a manufacturing system.

Design/methodology/approach: In this paper, condition based maintenance (CBM), a kind of preventive maintenance, is suggested to reduce unavailability of machines. Different to the simultaneous scheduling algorithm (SSA) used in the previous article (Neale & Cameron,1979), an inserting algorithm (IA) is applied, in which firstly a pre-schedule is obtained through heuristic algorithm and then maintenance tasks are inserted into the pre-schedule scheme.

Findings: It is encouraging that a new better solution for an instance in benchmark of FJSP is obtained in this research. Moreover, factually SSA used in literature for solving normal FJSPPM (FJSP with PM) is not suitable for the dynamic FJSPPM. Through application in the benchmark of normal FJSPPM, it is found that although IA obtains inferior results compared to SSA used in literature, it performs much better in executing speed.

Originality/value: Different to traditional scheduling of FJSP, uncertainty of machines is taken into account, which increases the complexity of the problem. An inserting algorithm (IA)

is proposed to solve the dynamic scheduling problem. It is stated that the quality of the final result depends much on the quality of the pre-schedule obtained during the procedure of solving a normal FJSP. In order to find the best solution of FJSP, a comparative study of three heuristics is carried out, the integrated GA, ACO and ABC. In the comparative study, we find that GA performs best in the three heuristic algorithms. Meanwhile, a new better solution for an instance in benchmark of FJSP is obtained in this research.

1. Introduction

There is a lot of research which endeavors in the improvement of production (Wang & Liu, 2013). Among the various production manners, the job shop is a promising one used widely in practice. Job shop is a common and basic type of manufacturing system, especially in the production with multiple types and small quantity. The model of Job shop scheduling problem (JSSP) has become a very interesting research topic since 1960s. It was demonstrated NP-complete when m ≥ 2 (m is the number of machines) (Garey, Johnson & Sethi, 1976). Flexible job shop scheduling problem (FJSP), as an extension and generalization of JSSP, is more complicated. FJSP extends JSSP with alternative machine routings, by assuming that a machine is capable of performing more than one type of operation (Hussain & Joshi, 1998; Nasr & Elsayed, 1990; Najid, Dauzere-Peres & Zaidat, 2002; Thomalla, 2001).

It is known that the efficient and effective production is based on the consistent operation of machines. However, the unavailability of machines is a prevalent phenomenon in a manufacturing system. In practical manufacturing environment, there are always some uncertainties inducing unavailability of machines, such as unavailability of staff, operational errors of staff, machine breakdown, and so forth. Besides the final scrap, machine breakdown cannot be avoided, especially in term of modern machine, which is always with a sophisticated structure. Therefore, ensuring machines performing in good condition attracts many researchers and practitioners. Among them, preventive maintenance (PM) is widely studied.

PM is demonstrated effective to increase availability of machines in both practice and theory. However, there has not been much research on FSSP considering maintenance activities. We have referred several researchers working on JSSP integrated with preventive maintenance (PM) (Gao, Gen & Sun, 2006; Moradi, Ghomi & Zandieh, 2011; Lei, 2010; Wang & Yu, 2010). Other similar work studies the influence of maintenance activity in some other production modes, like parallel machines (Berrichi, Yalaoui, Amodeo & Mezghiche, 2010; Liao, Chen & Lin, 2006).

In this work, we suggest to apply condition based maintenance (CBM) to increase machine reliability in work procedure. CBM is a maintenance policy in which PM is triggered after identifying a symptom of impending failure with the aid of condition-monitoring techniques. The real-time attribute makes FJSP dynamic, for which we can mark this dynamic scheduling problem as DFJSPPM. In order to solve this problem, different to the simultaneous scheduling algorithm used in Kacem, Hammadi and Borne (2002), in which maintenance tasks and the existing normal jobs are scheduled simultaneously in an integrated procedure. Obviously, it is based on the prediction of maintenance tasks before the execution of jobs. This is not suitable for the proposed problem, in which the maintenance is not forecasted in advance. With the real-time attribution, the maintenance task emerges unexpectedly. Therefore, we propose an inserting algorithm (IA) to add PM into the pre-schedule scheme obtained in the normal problem of FJSP. In this case, the quality of the final result depends mostly on the pre-schedule scheme acquired in the solution of normal FJSP.

As the problem is rather innovative, there is no benchmark used to demonstrate the effectiveness of our proposed algorithm in the comparative study with other algorithms. We compare IA with a simulated scheduling algorithm (SSA) which is used in Neale and Cameron (1979). Meanwhile, comparative study of the three kinds of heuristic algorithms is carried out to obtain a better preschedule scheme.

The structure of the paper is as follows: in section 2, the problem of DFJSPPM is interpreted; solution approach is detailed in section 3; section 4 is an experiment for demonstrating the effects of the proposed approach; we conclude our work in section 5.

2. Problem statement

The theoretical basis of CBM is the P−F curve, which describes the deterioration process of the machine and can be referred in Zheng (2012). The principle of CBM is executing maintenance before breakdown occurs. The time period of executing maintenance is decided by the data analysis with the machine monitoring. The advantage of CBM is that it transforms the state of maintenance from passiveness to activeness. It gets rid of the disadvantage of the cyclic maintenance, which may induce inadequate maintenance or over maintenance. As CBM is based on condition monitoring, time windows of executing maintenance are not fixed and cannot be predicted. As monitoring is real-time, production system with maintenance is dynamic, without static periods or intervals before the execution of the whole work in job shop.

Take an example in Figure 1. Before some hidden failures (point P) are detected, maintenance tasks are not considered in scheduling problem. Manufacturing is executed according to a pre-schedule solution of the normal FJSP. When arriving point P, the maintenance tasks have to be considered to add into the work in progress. In our work, we use an Inserting Algorithm (IA) to

add PM to the pre-determined scheme. This is the stage to respond to the disruption of adding PM into the preschedule scheme.

Compared to the model of FJSP, elements of PM are added in FJSPPM as follows: There are Lj maintenance tasks which have to be processed on machine j during the planning horizon. The maintenance task corresponds to a predefined time window T, within which the starting time of the maintenance task can be moved. Likewise, some hypotheses are added:

(1) Maintenance task is considered as a special job; the duration of maintenance is concluded in makespan of jobs;

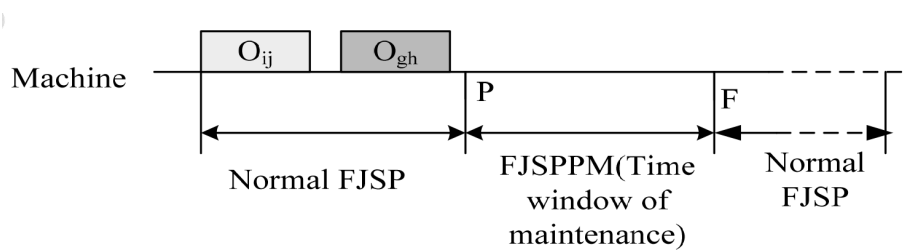(2) The machine is restored to good working condition after maintenance.

Figure 1. Dynamic FJSPPM with CBM

Because DFJSPPM depends on concrete problem, the first step of solving DFJSPPM is to specify the time windows of PM tasks. This step transfers the problem into FJSPPM. In the subsequent section comparative study with existing benchmarks of FJSPPM in literature is carried on to demonstrate our proposed algorithm.

The Mathematical model of the problem is as follows:

Notations used:

- Indices

  i, h: Index of jobs, *i, h = 1, 2, · · · , n*;

  j: Index of machines, *j = 1, 2, · · · ,m*;

  k, g: Index of operation sequence, *k, g =1, 2,· · · , n_i*;

  l: Index of maintenance tasks, *l = 1, 2, · · · , L_j*.

- Parameter

s_n: Total number of j obs;

m: Total number of machines;

n_i: Total number of operations of Job i;

L_j: Total number of maintenance tasks on Machine j;

O_{ik}: The kth operation of Job i;

PM_{jl}: l^{th} potential maintenance on Machine j;

p: Maximum number of maintenances needed on one machine;

A_{ik}: The set of available machines for Operation O_{ik};

t_{ikj}: Processing time of O_{ik} on Machine j;

CM_j: Unit processing cost on machine j;

p_{jl}: Duration of Maintenance task PM_{jl};

$[t_{Ejl}, t_{Ljl}]$: Time window associated with maintenance, where t_{Ejl} is the earliest starting time, and t_{Ljl} is the latest starting time of Maintenance task PM_{jl}.

- Decision variables

$x_{ikj} = \{1(\text{if machine } j \text{ is selected for Operation Oik}), 0(\text{otherwise})\}$

c_{ik}: Completion time of the Operation O_{ik};

y_{jl}: Completion time of Maintenance task Pm_{jl}.

Different to the classic model of FJSP with makespan of jobs as objective function, we add processing time of PM. Constraints of PM tasks are surely added. Our model is given as follows:

$$\min F_1 = \max(\max_{1 \leq i \leq n} c_{ini}, \max_{1 \leq j \leq m} y_{jLj}) \tag{1}$$

s.t.

$$[(c_{hg} - c_{ik} - t_{hgj})x_{hgj}x_{ikj} \geq 0] \vee [(c_{ik} - c_{hg} - t_{ikj})x_{hgj}x_{ikj} \geq 0], \forall(i, k), (h, g), j \qquad (2)$$

$$[(y_{jl} - c_{ik} - p_{jl})x_{ikj} \geq 0] \vee [(c_{ik} - y_{jl} - t_{ikj})x_{ikj} \geq 0], \forall(i, k), (j, l) \qquad (3)$$

$$x_{ikj} = 1, j \in A_{ik}, \forall i, k \qquad (4)$$

$$t_{Ejl} \leq y_{jl} \leq t_{Ljl}, \forall l, j \qquad (5)$$

$$x_{ikj} \in [0, 1], \forall I, k, j \qquad (6)$$

$$c_{ik} \geq 0, \forall i, k \qquad (7)$$

$$y_{jl} \geq 0, \forall j, l \qquad (8)$$

Objective function (1) is minimization of makespan, which is the max value between the latest completion time of job and that of maintenance. Inequality (2) ensures no-overlapping constraints between operations on the same machine. In- equality (3) ensures no-overlapping constraints between preventive maintenance tasks and operations on the same machine. Equation (4) states that only one machine should be selected from the set of available machines for each operation. Inequality (5) states that the preventive maintenance tasks have to be executed within their time windows. Inequalities (7) and (8) ensure the feasibility of the two variables.

3. Solution approach

The solution procedure of FJSPPM is decomposed into two stages: stage1 to solve FJSP for a pre-schedule scheme; stage 2 to insert PM tasks to the pre-schedule solution with an inserting algorithm (IA). For solving FJSP, three heuristic algorithms are applied to obtain the best solution, genetic algorithm (GA), ant colony optimization algorithm (ACO) and artificial bee colony algorithm (ABC).

3.1. Integrated Genetic Algorithm for FJSP

For the application of GA to solve FJSP, an integrated GA is proposed, in which the two sub problems, machine assignment and sequence scheduling, are integrated in one GA procedure.

The overall structure of integrated GA is described as follows:

(1) Coding: the genes of chromosome represent machine assignment of operations, with size (n, n_i);

(2) Initial population: It is initialized that machine for each operation is randomly selected from the set of available machines, A_{ij}, to guarantee its feasibility.

(3) Offspring generation: provide diversification in the population and avoids being stuck in local solution. In the crossover procedure, we use entire row or column crossover, to keep feasibility of individuals. In mutation procedure, individual and its mutation position is selected randomly and within the mutation probability. In order to ensure feasibility of solutions, the mutated value is also in the set of available machines.

(4) Fitness evaluation: Makespan is used as the fitness value, obtained in the decoding procedure, in which scheduling algorithm is used to get sequence scheduling. Operations on the same machine are sequenced according to the position of occurrence.

(5) Sequence scheduling: Firstly, according to processing sequence to schedule all operations of all jobs, e.g. the first operation of each job is firstly scheduled, and then the second operation, and so on, until all operations are scheduled.

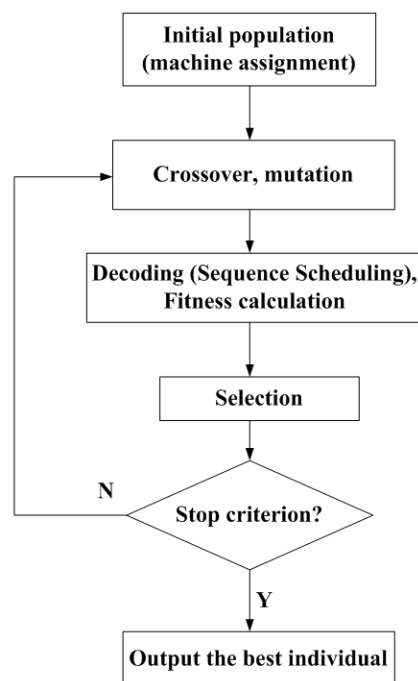The procedure of integrated GA is interpreted in Figure 2.

Figure 2. Flowchart of integrated GA for FJSP

3.2. Ant Colony Optimization Algorithm for FJSP

In this approach, the two sub problems, machine assignment and sequence scheduling of FJSP, are integrated as well. It is assumed that the ant colony deposes two kinds of pheromone, τ_{mijk} and $\tau_{iji'j'}$, to control the two sub problems respectively. The operation of the pheromone is interpreted as follows:

(1) Machine assignment

Machine assignment is executed in the first stage. An ant chooses machine k for operation O_{ij} with probability:

$$P_{ijk} = \frac{\tau\_m_{ijk}^{\alpha\_m} \cdot \eta_{ijk}^{\beta\_m}}{\sum_{k=1}^{A_{ij}} \tau\_m_{ijk}^{\alpha\_m} \cdot \eta_{ijk}^{\beta\_m}} \qquad (9)$$

Where α_m and β_m are parameters to control weights of pheromone fitness and heuristic respectively. η_{ijk} represents the rule of choosing machines. $\eta_{ijk}=1/T_{ijk}$ is the processing time of operation O_{ij} on machine k. A_{ij} is the set of available machines for operation O_{ij}.

(2) Sequence scheduling

Sequence scheduling is executed after machine assignment. Take an example of 3 jobs and 3 machines. Machine assignment is obtained from the preceding stage: operations O_{13}, O_{32}, O_{33} and O_{23} will be executed on machine 1; O_{11}, O_{31} and O_{22} will be executed on machine 2; O_{21} and O_{12} will be executed on machine 3. Set A includes the available jobs which wait to be executed, initially, A = {O_{11}, O_{21}, O_{31}}. Set Tabu represents the operations which have already been scheduled, initially, Tabu = {0}. The first operation is chosen from the set A randomly. Here, we pick up O_{11} for example. Simultaneously, update Set A and Tabu: A = {O_{12}, O_{21}, O_{31}}, Tabu = {O_{11}}.After choosing the first operation of one job randomly, the ant chooses the next operation in accordance with probability:

$$P_{iji'j'} = \frac{\tau_{iji'j'}^{\alpha} \cdot \eta_{iji'j'}^{\beta}}{\sum_{i=1}^{n} \sum_{j=1}^{n_i} \tau_{iji'j'}^{\alpha} \cdot \eta_{iji'j'}^{\beta}} \qquad (10)$$

Where, α and β are two parameters to control the weights of fitness of pheromone and heuristic separately, $\eta_{iji'j'}$ is to represent the rule of sequence scheduling.

3.3. Artificial Bee Colony Algorithm for FJSP

In the traditional ABC algorithm, there are three groups of bees in artificial bee colony: employed bees, onlooker bees and scout bees. These three kinds of bees undertake mutually and cooperate closely. In this paper, the application of the ABC algorithm to solve FJSP is with the same method of coding, initial population, fitness evaluation and decoding procedure as that in our proposed integrated GA mentioned previously. Meanwhile, the crossover and mutation operators in Wang, Zhou, Xu, Wang and Liu (2012) are used to generate new neighbour food sources in the employed bee phase. Moreover, the local search strategy based on the critical path is employed to improve the convergence speed in the onlooker bee phase.

3.4. Inserting algorithm for adding PM

Based on the pre-schedule obtained in the procedure of solving FJSP, Inserting Algorithm (IA) is used to add PM tasks in the pre-schedule. In IA, PM tasks are inserted to idle intervals of jobs after all jobs have been scheduled. We aim to make full utilization of the idle intervals, which always exist and are unavoidable in scheduling. IA is described as follows:

(1) Effective idle time intervals need to be found from left to right on the scheduling sequence of each machine, in the range of time windows of PM task.

(2) If the maximal effective idle time interval cannot satisfy the duration of the PM task, PM task is performed at the beginning of the maximal idle time interval, and then all its posterior sequences of operations are delayed.

4. Numerical experiments

To verify the effectiveness of our approach for FJSPPM, a benchmark in Neale and Cameron (1979) is applied, in which the example of FJSP is cited from Xia and Wu (2005). The three instances are as follows: J8M8 is an example for FJSP with partial flexibility, 8 jobs with 27 operations processed on 8 machines; J10M10 is an instance of total flexibility, 10 jobs with 30 operations processed on 10 machines; J15M10, instance of total flexibility, 15 jobs with 56 operations processed on 10 machines. All our experiments are tested on a computer with Intel (R) Core (TM) 2 CPU (2.66GHz 2.67 GHz), programs are realized in Matlab R2008a. The results

obtained in subsequent experiments through executing the program 100 times to get an average value.

4.1. Test of proposed algorithms applied for FJSP

We compare the performance of our proposed three approaches with optimal solutions obtained in Xia and Wu (2005) and Neale and Cameron (1979). Results from our approach and best solutions in literature are illustrated in Table 1. PSO + SA: in Xia and Wu (2005), a hybrid approach of Simulated Annealing and Particle Swarm Optimization was used for FJSP. hGA: GA with neighborhood search in Neale and Cameron(1979). From Table 1, we can see that our proposed approach, integrated GA, performs better than the two approaches in literature in small scale instance. For instance of large scale, it is also competitive. It is exiting that through integrated IA, a better solution for the instance J8M8 is obtained, which is shown in Figure 3. As well, we can see that integrated IA performs much better than our proposed algorithms, ACO and ABC. The parameters used in ACO, integrated GA and ABC are interpreted in Table 2, Table 3 and Table 4 respectively.

	J8M8	J10M10	J15M10
Integrated GA	14	7	12
ACO	15	7	16
ABC	16	8	15
SA+PSO	15	7	12
hGA	15	7	11

Table 1. Result of integrated GA for FJSP

Parameters	Value
Number of ants (M_ant)	400
Generation (NC_max)	0.9
Space range parameter (y)	10
Pheromone evaporation rate (ρ)	0.1
Control parameter of pheromone of operation (α)	1
Control parameter of heuristic of operation (β)	2
Control parameter of pheromone of machine (α_m)	1
Control parameter of heuristic of machine (β_m)	2
Strength of pheromone (Q)	10

Table 2. Parameter of ACO

Parameters	Value
Population size (N)	2000
Generation (M)	200
Crossover probability (Pc)	0.9
Mutation probability (Pm)	0.1

Table 3. Parameter of integrated GA

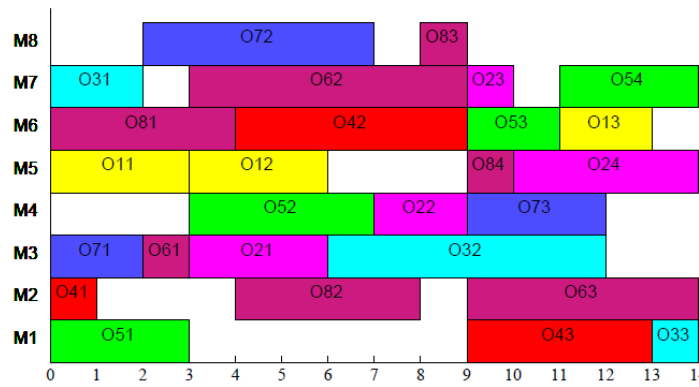| Parameters	Value
Number of bees	2000
Generation	1000
limit	5

Table 4. Parameter of ABC

Figure 3. Gantt chart of best solution in J8M8

4.2. Test of inserting algorithm to add PM

The two approaches, IA and SSA are compared to the three examples in Neale and Cameron (1979). The three examples of FJSPPM are as follows: J8M8PM is with one maintenance task on each machine; J10M10PM with one maintenance task on each machine; J15M10PM, with two maintenance tasks on each machine. Results are illustrated in Table 5. The value obtained by hGA with SSA is in Neale and Cameron (1979). Corresponding to the pre-schedule scheme of J8M8 obtained above, the result of solving J8M8PM is shown in Figure 4.

		J8M8PM	J10M10PM	J15M19PM
Integrated GA with SSA	makespan executive time	17 541.10	8 556.31	12 915.72
Integrated GA with IA	makespan executive time	18 251.91	9 252.56	13 387.15
HGA with SSA	makespan	17	8	12

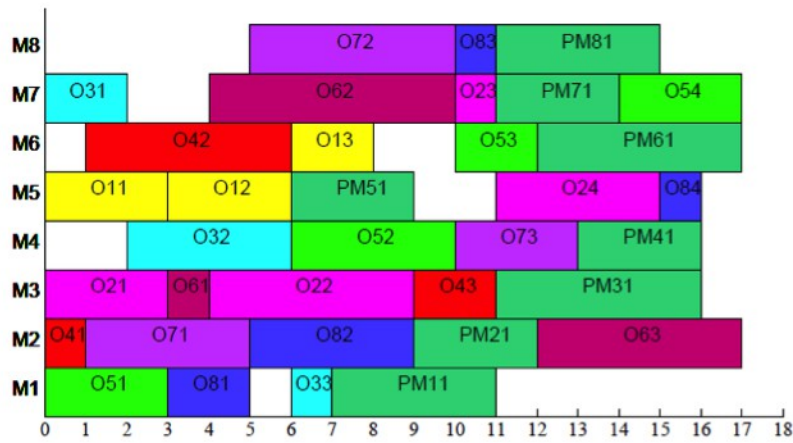Table 5. Results of approach applied onFJSPPM

Figure 4. Gantt chart of best solution in J8M8PM

5. Conclusions

Unavailability of machines, as a main form of uncertainty in manufacturing system, has attracted much attention of many researchers and practitioners. In our study of uncertainty in a manufacturing system, we concentrate on the flexible job scheduling problem. In order to ensure availability of machines in FJSP, condition based maintenance, a kind of preventive maintenance, is suggested to add into a pre-schedule scheme obtained through solving normal FJSP, which forms the proposed inserting algorithm. It is found that the quality of the final result of scheduling depends on the pre-schedule scheme acquired in the stage of solving normal FJSP. In order to find the best solution of FJSP, a comparative study of three heuristics is carried out, the integrated GA, ACO and ABC. In the comparative study, it is encouraging that a new best solution for an instance of FJSP, J8M8 is obtained. To demonstrate the effectiveness of the proposed IA, the benchmark of normal FJSPPM is applied. Although IA obtains inferior results compared to SSA used in literature, it performs much better in executing speed. With IA, we used vide internal between jobs. It is a module independent from the module of solving FJSP. We can easily change either of them to improve the solution. This advantage will be useful when the scheduling problem needed to connect to other management system, like ERP. Moreover, factually SSA in literature for solving normal FJSPPM is not suitable for the dynamic FJSPPM. In the future research, improving of IA is expected, as well the heuristic algorithm for solving normal FJSP.

Acknowledgement

References

Berrichi, A., Yalaoui, F., Amodeo, L. & Mezghiche, M. (2010). Bi-Objective Ant Colony Optimization approach to optimize production and maintenance scheduling. *Computers & Operations Research*, 37, 1584-1596. http://dx.doi.org/10.1016/j.cor.2009.11.017

Gao, J., Gen, M., & Sun, L. (2006). Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm. *Journal of Intelligent Manufacturing*, 17, 493-507. http://dx.doi.org/10.1007/s10845-005-0021-x

Garey, M.R., Johnson, D.S., & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of operations research,* 1(2), 117-129. http://dx.doi.org/10.1287/moor.1.2.117

Hussain, M.F., & Joshi, S.B. (1998). A genetic algorithm for job shop scheduling problems with alternate routing. *IEEE International Conference on Systems, Man, and Cybernetics,* 3, 2225-2230.

Kacem, I., Hammadi, S., & Borne, P. (2002). Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* 32, 1-13. http://dx.doi.org/10.1109/TSMCC.2002.1009117

Lei, D. (2010). Fuzzy job shop scheduling problem with availability constraints. *Computers & Industrial Engineering*, 58, 610-617. http://dx.doi.org/10.1016/j.cie.2010.01.002

Liao, C.J., Chen, C.M., & Lin, C.H. (2006). Minimizing makespan for two parallel machines with job limit on each availability interval. *Journal of the Operational Research Society*, 58, 938-947. http://dx.doi.org/10.1057/palgrave.jors.2602209

Moradi, E., Ghomi, S.M.T., & Zandieh, M. (2011). Bi-objective optimization research on integrated fixed time interval preventive maintenance and production for scheduling flexible job-shop problem. *Expert Systems with Applications*, 38(6), 7169-7178. http://dx.doi.org/10.1016/j.eswa.2010.12.043

Najid, N.M., Dauzere-Peres, S., & Zaidat, A. (2002). A modified simulated annealing method for flexible job shop scheduling problem. *IEEE International Conference on Systems, Man and Cybernetics*, 5, 6-15.

Nasr, N., & Elsayed, E.A. (1990). Job shop scheduling with alternative machines. *The international journal of production research*, 28, 1595-1609. http://dx.doi.org/10.1080/00207549008942818

Neale, M., & Cameron, J.R. (1979). *Committee for Tero technology: A guide to the condition monitoring of machinery.* London: HMSO Publications, 89-96.

Thomalla, C.S. (2001). Job shop scheduling with alternative process plans. *International Journal of Production Economics*, 74, 125-134. http://dx.doi.org/10.1016/S0925-5273(01)00119-0

Wang C., & Liu, X.B. (2013). Integrated production planning and control: A multi-objective optimization model. *Journal of Industrial Engineering and Management*, 6(4), 815-830. http://dx.doi.org/10.3926/jiem.771

Wang, L., Zhou, G., Xu, Y., Wang, S., & Liu, M. (2012). An effective artificial bee colony algorithm for the flexible job-shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 60(1-4), 303-315. http://dx.doi.org/10.1007/s00170-011-3610-1

Wang, S., & Yu, J. (2010). An effective heuristic for flexible job-shop scheduling problem with maintenance activities. *Computers & Industrial Engineering*, 59, 436-447. http://dx.doi.org/10.1016/j.cie.2010.05.016

Xia, W., & Wu, Z. (2005). An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, 48, 409-425. http://dx.doi.org/10.1016/j.cie.2005.01.018

Zheng, Y. (2012). *Supply chain management under availability & uncertainty constraints.* Lille: Ecole Centrale de Lille. PhD thesis, 136.