A Genetic Algorithm for Flexible Job Shop Scheduling

Imran A. Chaudhry, Abdul Munem Khan and Abid Ali Khan

Abstract— **Flexible Job Shop scheduling problem (FJSSP) is an important scheduling problem which has received considerable importance in the manufacturing domain. In this paper a genetic algorithm (GA) based scheduler is presented for flexible job shop problem to minimise makespan. The proposed approach implements a domain independent GA to solve this important class of problem. The scheduler is implemented in Microsoft Excel™ spreadsheet. The shop model is developed in the spreadsheet using the built in functions. Benchmark problems from the literature have been used to compare the performance of the proposed approach. The results show that the proposed approach is capable of achieving solutions comparable with earlier approaches used for the benchmark problems. It is also shown that the model can be easily customised to cater for any objective function without changing the basic GA routine thus making the proposed approach a robust and general purpose.**

Index Terms— **Flexible Job Shop, Scheduling, Genetic Algorithm (GA), Spreadsheet**

I. INTRODUCTION

Scheduling is an important activity in any manufacturing environment. It is a key factor for enhancing manufacturing productivity. The purpose of scheduling is to allocate resources to the jobs. Efficient schedules yield great savings in the overall manufacturing environment.

Flexible job shop scheduling is very important in the fields of combinatorial optimization and production management. Flexible job shop scheduling problem (FJSSP) is an extension of the classical job shop scheduling problem and is considered to be strongly NP-hard. In FJSSP same operation could be processed on more than one machine thus there are alternative machines available to process a particular job. FJSSP consists of two sub-problems. First is to assign each operation to a machine out of a set of capable machines, while the second deals with sequencing of the assigned operations on the machines.

In the recent years FJSSP has received considerable importance by the researchers due to the inherent complexity of the problem. Bruker and Schlie [1] were among the first to address the scheduling of jobs on multi-purpose machines. Since then many meta-heuristics have been developed to address this problem. Ozguven et al [2] and Chiang and Lin [3] provide a comprehensive review of the application of heuristics and mathematical approaches to FJSSP.

Imran A. Chaudhry, Abdul Munem Khan and Abid Ali Khan are with National University of Sciences & Technology, Islamabad. Pakistan. (Corresponding author: +92-300-4279895; e-mail: imranchaudhry-cae@nust.edu.pk, munem-cae@nust.edu.pk, khanabid@yahoo.com).

The aim of this paper is to present a spreadsheet based GA approach for a flexible job shop scheduling problem. We present a spreadsheet based general purpose genetic algorithm approach to minimise an objective function that is a combination of makespan, total workload and critical workload.

The rest of the paper is organised as follows: Section II defines flexible job shop scheduling problem (FJSSP) and assumptions. Section III gives the brief introduction of GA it application to flexible job shop scheduling problem and its operators i.e., crossover and mutation. Simulation results are given in Section IV. Finally, conclusions are presented in Section V.

II. DEFINITION OF FLEXIBLE JOB-SHOP SCHEDULING PROBLEM & ASSUMPTIONS

As compared to classical job shop where each job is required to be processed on only one machine, the allocation of tasks in FJSSP is more challenging as it requires a proper selection of a machine from a set of given machines to process each operation. In this research the FJSSP is composed of the following elements:

(1) Jobs. $J = \{J_1, J_2,\ldots\ldots\ldots,J_n\}$ is a set of n independent jobs to be scheduled. Each job J_i consists of a sequence $O_{i1}, O_{i2},\ldots\ldots\ldots,O_{ini}$ of operations to be performed one after the other according to a given sequence. All jobs are available at time 0.
(2) Machines. $M= \{M_1, M_2,\ldots\ldots\ldots, M_m\}$ is a set of m machines. Every machine processes only one operation at a time. All machines are available at time 0.
(3) Flexibility. A FJSSP is generally classified into two categories as follows:
    - Total flexibility: each operation can be processed on any of the M machines.
    - Partial Flexibility: each operation can be performed only on a subset of M machines.
(4) Other assumptions for the problem are as follows:
    - Every operation is processed on only one machine at a time.
    - The processing time of each operation is machine-dependent and machines are independent from each other.
    - Pre-emption of operations is not allowed, i.e., each operation once started must be completed without interruption.
    - Setup time required to setup a machine to process an operation is included in the job processing time of the job.

- Transportation of jobs from one machine is negligible and is included in the job processing time.

(5) Objective. The problem is to assign each operation to an appropriate machine (routing problem), and to sequence the operations on the machines (sequencing problem) in order to minimize *F(c)*, that is given by:

$$F(c) = 0.5\, F_1(c) + 0.2\, F_2(c) + 0.3\, F_3(c) \qquad (1)$$

where
- $F_1(c)$, the makespan or the maximum completion time of a schedule.
- $F_2(c)$, the total workload representing the total working time of all machines.
- $F_3(c)$, the critical machine workload, which is the most workload among all machines.

III. GENETIC ALGORITHMS

Genetic Algorithm (GA) is a stochastic optimization technique inspired by the nature and is similar to the evolution process in the biological world where the fittest survive to reproduce. GAs were first introduced by Holland [4] in 1975 at the University of Michigan. Initially a random population of solutions are generated. Two solutions are then selected as parents which mate with each other through the crossover operator resulting in the production of a child solution. If required the child solution may be modified to meet any constraints. At this stage the modified child solution is evaluated for fitness. If the child solution is fitter than the solutions already present in the population, it replaces the worst performing member of the population, otherwise it is itself discarded. Thus in every successive generation better solutions are produced till the time stopping criteria is reached. The flowchart of a generic GA is shown in Fig 1.

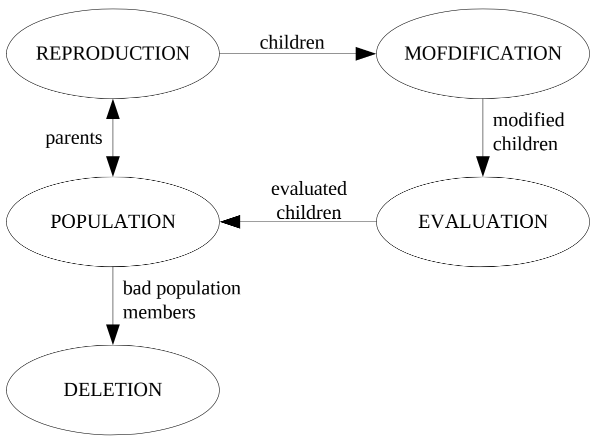

Fig. 1. Flowchart of a generic GA

Davis [5] was the first who suggested and demonstrated the application of GAs to a simple job shop scheduling problem. Since then a number of researchers have applied GAs to various types of manufacturing systems. Similarly there has been a growing trend towards the GA application to flexible job shop scheduling problem. Some of the recent applications of GAs to FJSSP are [6-21].

In this research we use a spreadsheet based commercial genetic algorithm Evolver [22] to solve the FJSSP to minimize *F(c)* as described in equation 1. The software functions as an add-in to the Microsoft Excel spreadsheet. The shop model in the spreadsheet is developed using the built-in functions of the Microsoft Excel. Scheduling rules and constraints are also defined in the spreadsheet model. Fig. 2 describes the Microsoft Excel - Evolver integration architecture. GA component generates population of solutions in the spreadsheet environment, which after crossover and mutation operations are passed back to the model as the fittest solution. Upon reaching the stopping criteria, the model passes the best solution back to the spreadsheet for storage and output.

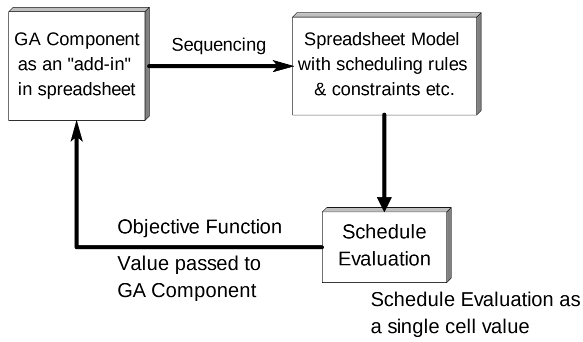

Fig 2: Microsoft Excel - Evolver integration architecture

A. Chromosome representation

The chromosome to represent the FJSSP forms of two parts. It is primarily a vector of integer numbers twice the length of total number of operations. The first part represents the list of all operations while the second part represents the machine associated with each operation appearing in the same order as are the operations and linked with each other in the spreadsheet model. The first part of the chromosome requires permutation representation where the sequence of the operations is varied keeping in view the precedence constraints. Any sequence of operations that does not satisfy the precedence constraints is automatically repaired by the GA. While for the second part a random number between 1 and number of available machines is generated for each related operation. We demonstrate this by a numerical example having 3-jobs to be processed on 3-machines. The example problem data is given in Table I.

TABLE I. AN EXAMPLE WITH THREE JOBS AND 8 OPERATIONS AND THREE MACHINES

Jobs	1			2			3	
Operations	11	12	13	21	22	23	31	32
Available Machines	M_1-M_2-M_3	M_1-M_2	M_1-M_2-M_3	M_1-M_2-M_3	M_1-M_3	M_1-M_2-M_3	M_1-M_2-M_3	M_2-M_3

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

In Table 1, operation 11 and 12 represent the first and second operations of job 1 respectively and so on. Fig 3 depicts a solution for this problem.

Operations								Machine assigned							
J_1	J_2	J_2	J_1	J_3	J_3	J_2	J_1	M_1	M_2	M_3	M_2	M_1	M_3	M_2	M_1
11	21	22	12	31	32	23	13	1	2	3	2	1	3	2	1

Fig. 3.An example of solution representation

Each operation block in the first part of the chromosome is linked to the corresponding machine assignment block in the spreadsheet. For example, job 1-operation 1 represented by 11 is to be processed on machine 1, similarly job 3-operation 2 represented by 32 is to be processed on machine 3. The presented example is a case of partial flexibility where some of the operations can be processed on all available machines, while some can only be processed on only a subset of machines. In case of a partial flexibility, the problem is converted into a total flexibility by assigning a higher processing time to the machines that cannot process a particular operation. For example, job 1-operation 2 (denoted by 12) cannot be processed on machine 3; hence a processing time of 99 would be assigned for operation 12 to be processed on machine 3.

B. Crossover Operator

Crossover operator is an integral component of GA, where two chromosomes (parents) are combined together (mate) to produce a new chromosome (offspring or child solution). The idea behind crossover operator is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution according to a user-defined crossover probability.

For the first part of the chromosome, i.e., operations, order crossover [23] is used as it preserves the order of the genes within the chromosome without violating the precedence constraints. While for machine assignment uniform crossover is implemented. In this case a random number is generated between 1 and total number of available machines.

C. Mutation Operator

Mutation operation maintains diversity in the solutions and prevents the population from stagnating at any local optima. For the operations block of the chromosome, the mutation is performed by randomly swapping the positions of two tasks without violating the precedence constraints, while for the machine assignment block each block or variable is looked at individually and a random number among available machines is generated and replaced accordingly depending on the mutation rate.

IV. SIMULATION RESULTS

To illustrate the effectiveness of the proposed approach, two benchmark problems have been taken from Kacem [24] to optimize the objective function given in equation 1. These problems have already been solved by a number of researchers. The first problem is a 4-job 5-machine flexible job shop with total flexibility having a total of 12 operations. The second problem is a partially flexible medium size problem with 8-jobs and 8-machines with a total of 27 operations. The requisite data for both the problems are given in Table II and II respectively.

TABLE II. JOB DATA FOR 4-JOB 5-MACHINE FLEXIBLE JOB SHOP (12 OPERATIONS)

Job	**Job 1**			**Job 2**			**Job 3**			**Job 4**		
Operation	***1***	***2***	***3***	***1***	***2***	***3***	***1***	***2***	***3***	**1**	***2***	***3***
M_1	2	5	4	2	5	4	9	6	2	4	1	5
M_2	5	4	5	5	6	5	8	1	5	5	5	1
M_3	4	5	5	4	9	4	6	2	4	2	2	2
M_4	1	7	4	7	8	54	7	5	2	1	4	1
M_5	2	5	5	8	5	5	9	4	4	5	12	2

TABLE III. JOB DATA FOR 8-JOB 8-MACHINE FLEXIBLE JOB SHOP (27 OPERATIONS)

	Job 1			**Job 2**				**Job 3**			**Job 4**			**Job 5**				**Job 6**			**Job 7**			**Job 8**			
	1	***2***	***3***	***1***	***2***	***3***	***4***	***1***	***2***	***3***	***1***	***2***	***3***	***1***	***2***	***3***	***4***	***1***	***2***	***3***	***1***	***2***	***3***	***1***	***2***	***3***	***4***
M1	5	10	-	5	-	-	10	10	-	1	3	12	4	3	10	-	11	6	11	10	5	-	-	2	7	9	9
M2	3	-	10	7	8	10	8	-	10	4	1	11	6	6	-	9	9	7	-	5	4	9	8	8	4	9	-
M3	5	5	-	3	5	-	9	-	6	5	6	7	2	7	7	8	-	1	9	9	2	-	9	5	7	-	3
M4	3	8	5	9	2	5	6	7	4	6	5	8	10	8	4	7	6	4	9	10	6	9	3	9	8	8	7
M5	3	3	6	8	6	6	4	6	8	-	9	10	3	9	9	4	7	6	9	11	7	11	8	-	9	5	1
M6	-	9	2	-	7	4	7	5	9	10	7	5	9	-	8	2	5	9	7	-	-	9	6	4	-	6	5
M7	10	9	4	9	10	1	-	2	10	-	8	6	5	10	6	7	3	-	6	10	10	10	-	-	10	7	8
M8	9	6	5	-	9	7	-	4	-	7	4	9	7	-	-	-	6	10	4	-	-	5	10	10	-	1	-

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Results of both the problems obtained by the proposed approach were compared with other meta-heuristics already found in the literature. The comparison of the proposed approach has been done with 14 different heuristics as mentioned below. The comparative results are given in Table IV. From Table IV we can see that the solution obtained by the proposed approach is better than or equal to the other approaches.

(1) A1: Temporal Decomposition approach [24]
(2) A2: AL+CGA heuristic [25]
(3) A3: Approach by localization [25]
(4) A4: Controlled GA [25]
(5) A5: Simulation model by Xing et al. [26]
(6) A6: PSO+SA algorithm [27]
(7) A7: Heuristic SPT [28]
(8) A8: moGA algorithm [28]
(9) A9: Multiobjective Artificial Immune Algorithm [29]
(10) A10: Hybrid Multiobjective Evolutionary Approach [30]
(11) A11: Pareto-based discrete artificial bee colony algorithm [31]
(12) A12: MOGA [32]
(13) A13: MOEA + GLS [33]
(14) A14: MOPSO + LS [34]

As stated earlier the proposed approach is a general purpose approach which could be used for any objective function without changing the model or the GA routine. In order to demonstrate this feature, the model was run for an added feature of release dates whereby meaning that all jobs are not available at $t=0$, thus the jobs may arrive in the shop at different times. The only modification done in the model was to cater for the release time of each job so that the starting time of the job can be delayed to the release time. The release date constraints considered are given as follows:

(1) Instance 4 × 5: $r_1 = 3$, $r_2= 5$, $r_3= 1$, $r_4= 6$.
(2) Instance 8 × 8: $r_1= 2$, $r_2= 5$, $r_3= 8$, $r_4= 3$, $r_5= 1$, $r_6= 5$, $r_7= 7$, $r_8= 0$.

Both the problems that have been attempted earlier were re-run with the added constraint of release time. The comparison of results obtained by the proposed approach for problems with release time with other heuristics is presented in Table V. The Gantt charts for best solution obtained by proposed approach for 8-job 8-machine problem without and with release times are given in Fig. 4 and 5 respectively.

For problems with release time also the solutions produced by the proposed approach is equal to or better than the earlier approaches for both the problems.

V. CONCLUSION

In this paper we presented a spreadsheet based domain independent GA approach for flexible job shop scheduling problem. It has been demonstrated that the proposed approach is general purpose and can be adopted for any objective function without changing the basic GA routine. This has been shown by optimizing two different functions. The performance of proposed approach is compared with fourteen other meta-heuristics. The results show that the proposed approach finds solutions that are equal to or superior than the earlier approaches. Similarly the model can also be customized to cater for any change in the shop or addition of constraints. Spreadsheet environment also enables a user to carry out what-if analysis.

TABLE IV. COMPARISON OF RESULTS OF PROPOSED APPROACH WITH OTHER HEURISTICS

Problem Instance	Objective	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	Proposed GA Approach
4 × 5	$F_1(c)$	-	16	-	-	12	-	-	-	-	-	12	12	-	-	12
	$F_2(c)$	-	35	-	-	32	-	-	-	-	-	32	32	-	-	32
	$F_3(c)$	-	9	-	-	8	-	-	-	-	-	8	8	-	-	8
	$F(c)$	-	17.7	-	-	14.8	-	-	-	-	-	14.8	14.8	-	-	14.8
8 × 8	$F_1(c)$	19	16	16	16	14	15	19	15	16	15	14	15	14	14	14
	$F_2(c)$	91	75	75	77	77	75	91	73	73	75	77	75	77	77	77
	$F_3(c)$	19	14	13	11	12	12	16	14	13	12	12	12	12	12	12
	$F(c)$	33.4	27.2	26.9	26.7	26	26.1	32.5	26.3	26.5	26.10	26	26.10	26	26	26

TABLE V. COMPARISON OF RESULTS WITH RELEASE DATE

Problem Instance	Objective	A2	A10	A11	A13	A14	Proposed GA Approach
4 × 5	$F_1(c)$	18	16	16	16	16	16
	$F_2(c)$	33	33	33	33	33	33
	$F_3(c)$	7	7	7	7	7	7
	$F(c)$	17.7	16.7	16.7	16.7	16.7	16.7
8 × 8	$F_1(c)$	-	20	20	-	-	19
	$F_2(c)$	-	73	75	-	-	73
	$F_3(c)$	-	13	12	-	-	13
	$F(c)$	-	28.5	28.6	-	-	28

“ - ” represents results not available / attempted by the heuristic

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

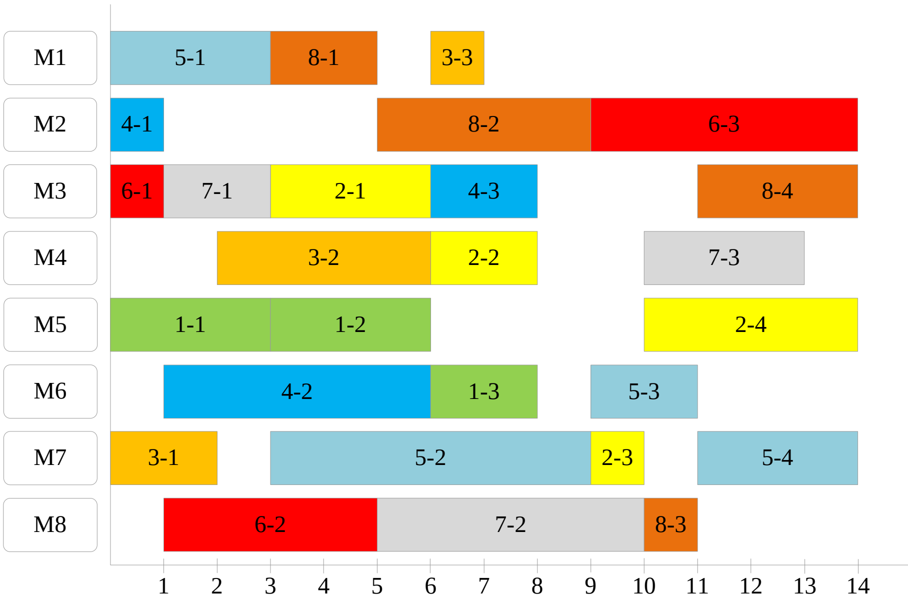

Fig. 4. Gantt chart for 8-job 8-machine problem without ready times

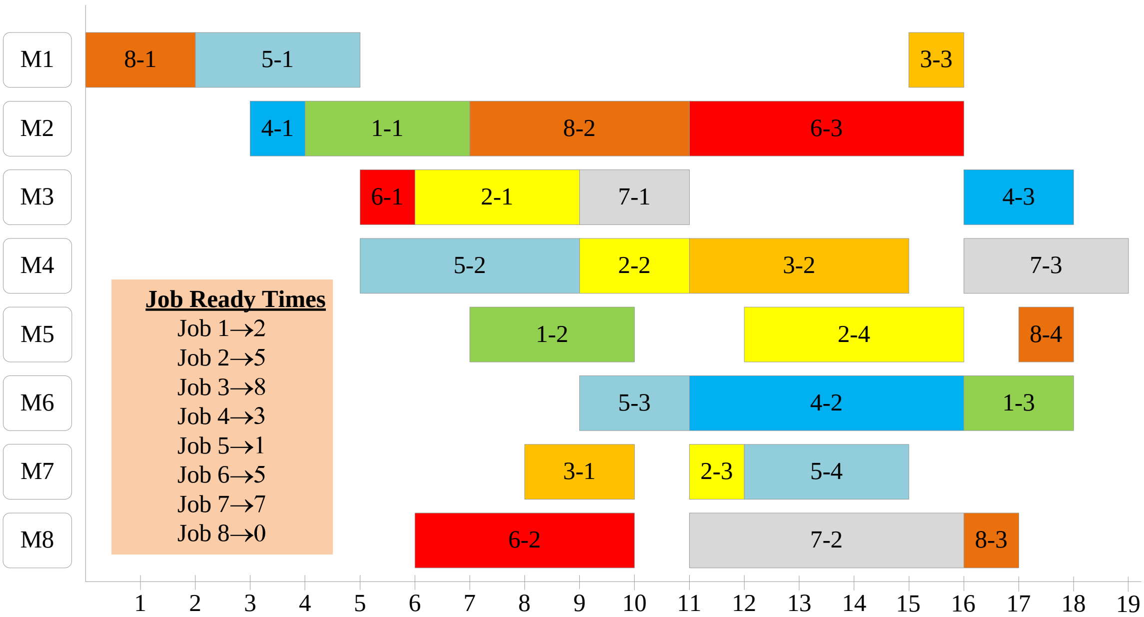

Fig. 5. Gantt chart for 8-job 8-machine problem with ready times

REFERENCES

[1] P. Bruker and R. Schlie, "Job shop scheduling with multi-purpose machines", *Computing*, vol. 45, pp. 369-375, 1990.

[2] C. Ozguven, Y. Yavuz and L. Ozbakır, "Mixed integer goal programming models for the flexible job-shop scheduling problems with separable and non-separable sequence dependent setup times", *Applied Mathematical Modelling*, vol. 36, pp. 846–858, 2012

[3] T.-C. Chiang and H.-J. Lin, "Flexible job shop scheduling using a multi-objective memetic algorithm", in D.-S. Huang et al. (Eds.): *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, Lecture Notes in Computer Science*, vol. 6839, 2012, pp 49-56.

[4] J. H. Holland, *Adaptation in natural and artificial systems*, Ann Arbor, University of Michigan Press, USA, 1975.

[5] L. Davis, "Job shop scheduling with genetic algorithms", In J. J. Grefenstette (ed) *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, Morgan Kaufmann, San Mateo, 1985, pp. 136-140.

[6] F. Pezzella, G. Morganti and G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem", *Computers & Operations Research*, vol. 35, no. 10, pp. 3202-3212, 2008.

[7] D. Lei, "A genetic algorithm for flexible job shop scheduling with fuzzy processing time", *International Journal of Production Research*, vol. 48, no. 10, pp. 2995-3013, 2010.

[8] L. De Giovanni and F. Pezzella, "An improved genetic algorithm for the distributed and flexible job-shop scheduling problem", *European Journal of Operational Research*, vol. 200, no. 2, pp. 395-408, 2010.

[9] C. Gutiérrez and I. García-Magariño, "Modular design of a hybrid genetic algorithm for a flexible job–shop scheduling problem", *Knowledge-Based Systems*, vol. 24, no. 1, pp. 102-112, 2011.

[10] L.-N. Xing, Y.-W. Chen and K.-W. Yang, "Multi-population interactive coevolutionary algorithm for flexible job shop scheduling problems", *Computational Optimization and Applications*, vol. 48, no. 1, pp. 139-155, 2011.

[11] N. Al-Hinai and T. Y. ElMekkawy, "An efficient hybridized genetic algorithm architecture for the flexible job shop scheduling problem", *Flexible Services and Manufacturing Journal*, vol. 23, no. 1, pp. 64-85, 2011.

[12] G. Zhang, L. Gao and Y. Shi, "An effective genetic algorithm for the flexible job-shop scheduling problem", *Expert Systems with Applications*, vol. 38, no. 4, pp. 3563-3573, 2011.

[13] N. Al-Hinai and T.Y. ElMekkawy, "Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid

genetic algorithm", *International Journal of Production Economics*, vol. 132, no. 2, pp. 279-291, 2011.

[14] T.-C. Chiang and H.-J. Lin, "Flexible job shop scheduling using a multiobjective memetic algorithm", in D.-S. Huang et al. (Eds) *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence Lecture Notes in Computer Science*, vol. 6839, 2012, pp 49-56

[15] S.Habib, A. Rahmati, M. Zandieh and M. Yazdani, "Developing two multi-objective evolutionary algorithms for the multi-objective flexible job shop scheduling problem", *The International Journal of Advanced Manufacturing Technology*, Online First™, DOI: 10.1007/s00170-012-4051-1, 2012.

[16] M. R. Amin-Naseri and A. J. Afshari, "A hybrid genetic algorithm for integrated process planning and scheduling problem with precedence constraints", *The International Journal of Advanced Manufacturing Technology*, vol. 59, no. 1-4, pp. 273-287, 2012.

[17] S. Habib, A. Rahmati and M. Zandieh, "A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem", *The International Journal of Advanced Manufacturing Technology*, vol. 58, no. 9-12, pp. 1115-1129, 2012.

[18] J. F. Wang, B. Q. Du and H. M. Ding, "A genetic algorithm for the flexible job-shop scheduling problem" in G. Shen and X. Huang(Eds.), *Advanced Research on Computer Science and Information Engineering Communications in Computer and Information Science*, vol. 152, 2011, pp 332-339.

[19] J. C. Chen, C.-C. Wu, C.-W. Chen and K.-H. Chen, "Flexible job shop scheduling with parallel machines using genetic algorithm and grouping genetic algorithm", *Expert Systems with Applications*, vol. 39, no. 11, pp. 10016-10021, 2012.

[20] D. Lei, "Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling", *Applied Soft Computing*, vol. 12, no. 8, pp. 2237-2245, 2012.

[21] Q. Zhang, H. Manier and M.-A. Manier, "A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times", *Computers & Operations Research*, vol. 39, no. 7, pp. 1713-1723, 2012.

[22] Evolver: the genetic algorithm super solver, Palisade Corporation, New York, USA, 1998.

[23] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

[24] I. Kacem, S. Hammadi and P. Borne, "Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems" *IEEE Transactions on Systems, Man & Cybernetics, Part C*, vol. 32. No. 1, pp. 1-13, 2002.

[25] I. Kacem, S. Hammadi and P. Borne, "Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic", *Mathematics and Computers in Simulation*, vol. 60, no, 3-5, pp. 245-276, 2002.

[26] L.-N. Xing, Y.-W. Chen and K.-W. Yang, "Multi-objective flexible job shop schedule: Design and evaluation by simulation modelling", *Applied Soft Computing*, vol. 9, no. 1, pp. 362-376, 2009.

[27] W. Xia and Z. Wu, "An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems" *Computers & Industrial Engineering*, vol. 48, no. 3, pp. 409-425, 2005.

[28] H. Zhang and M. Gen, "Multistage-based genetic algorithm for flexible job-shop scheduling problem", *Journal of Complexity International*, vol. 11, pp. 223-232, 2005.

[29] Z. Davarzani, M.-R Akbarzadeh-T and N. Khairdoost, "Multiobjective artificial immune algorithm for flexible job shop scheduling problem", *International Journal of Hybrid Information Technology*, vol. 5, vol. 3, pp. 75-88, 2012.

[30] J. Xiong, X. Tan, K.-W. Yang, L.-N. Xing and Y.-W. Chen, "A hybrid multiobjective evolutionary approach for flexible job-shop scheduling problems", *Mathematical Problems in Engineering*, vol. 2012, Article ID 478981, 27 pages, 2012.

[31] J. Q. Li, Q. K. Pan and K. Z. Gao, "Pareto-based discrete artificial bee colony algorithm for multiobjective flexible job shop scheduling problems", *The International Journal of Advanced Manufacturing Technology*, vol. 55, no. 9-12, pp. 1159–1169, 2011.

[32] X. Wang, L. Gao, C. Zhang and X. Shao, "A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem", *The International Journal of Advanced Manufacturing Technology*, vol. 51, no. 5–8, pp. 757–767, 2010.

[33] N. B. Ho and J. C. Tay, "Solving multiple-objective flexible job shop problems by evolution and local search", *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 38, no. 5, pp. 674–685, 2008.

[34] G. Moslehi and M. Mahnam, "A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search", *International Journal of Production Economics*, vol. 129, no. 1, pp. 14–22, 2011.