Volume 2, Number 1, 2013

International Journal of Research in Industrial Engineering

journal homepage: www.nvlscience.com/index.php/ijrie

A Genetic Algorithm Coupled with Tabu Search for Bi-Objective Permutation Flow Shop

N. Shahsavari Pour[1], M.H. Abolhasani Ashkezari[2,*], H. Mohammadi Andargoli[2]

[1]Department of Industrial Management, Vali-e-Asr University, Rafsanjan, Iran

[2]Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Kerman, Iran

A R T I C L E   I N F O

Article history :
Received:
September 17, 2012
Revised:
December 25, 2012
Accepted:
January 21, 2013

Keywords :
Due Date, Flow shop scheduling problem, Genetic Algorithm, Makespan, Multi-Objective, Tabu search.

A B S T R A C T

Considering flow shop scheduling problem with more objectives, will help to make it more practical. For this purpose, we have intended both the makespan and total due date cost simultaneously. Total due date cost is included the sum of earliness and tardiness cost. In order to solve this problem, a genetic algorithm is developed. In this GA algorithm, to further explore in solution space a Tabu Search algorithm is used. Also in selecting the new population, is used the concept of elitism to increase the chance of choosing the best sequence.

To evaluate the performance of this algorithm and performing the experiments, it is coded in VBA. Experiments results and comparison with GA is indicated the high potential of this algorithm in solving the multi-objective problems.

1. Introduction

The first research in flow shop scheduling problem has been done by Johnson [1]. Johnson described an exact algorithm for the minimization of makespan for flow shop scheduling problem with n jobs and 2 machines. Several algorithms such as branch and bound have been proposed to achieve the exact solution for this problem [2-5]. Thornton and Hunsucker [6] have proposed a new heuristic method for flow shop scheduling problems with the minimization of makespan with multiple processors and no intermediate storage. Bouquard et al. [7] has presented various forms of flow shop scheduling problem to minimize makespan.

Recently many researchers have focused on the use of meta-heuristic algorithms to solve scheduling problems. (See [8-18]). Also, so many multi objective algorithms have been proposed. For example, Chang et al. [19] proposed two-phase subpopulation GA. In this approach simultaneously applies several subpopulations and assigns the weight for this subpopulation to explore the solution space uniformly. Hierarchical Fair Competition Model which divides the population into a hierarchical structure proposed by Hu et al. [20]. Mining Gene subpopulation GA that employs a mining gene technique based on the subpopulation genetic algorithm offered by Chang et al. [21]. Alves and Almeida [22] proposes a similar

*Corresponding author
E-mail address: Abolhasani.hossein@Yahoo.com

idea to the two-phase subpopulation genetic algorithm which proposed by Chang et al. [19]. Tan et al. [23] in their article are shown applying the immune algorithm to solve multi-objective problems. Two algorithms NSGA-II in Deb et al. [24] and SPEA-II in Zitzler et al. [25] are the best known algorithms for solving Multi-Objective scheduling problems. A multi-objective immune algorithm is proposed by Tavakkoli-Moghaddam et al. [26] for the flow shop problem and this algorithm is compared with a conventional multi-objective genetic algorithm, SPEA-II. Genetic algorithm integrated with artificial chromosomes for multi-objective flow shop scheduling problems is presented by Chang et al. [27]. In their article, an artificial chromosome generating mechanism is designed to reserve patterns of genes in elite chromosomes and to find possible better solutions.

In this paper, a flow shop scheduling problem is considered with two objectives. In order to solve this problem, a genetic algorithm is presented. In the proposed algorithm is used to elitism and tabu search algorithm to improve its performance. The following is a summary of the various parts of this article: In the next section we described the problem and its dimensions. The third section discusses the proposed algorithm and its features. In the fourth section, presents two examples and have examined the performance of the proposed algorithm. In the fifth section has been paid to conclusions of the paper.

2. Problem description

The flow shop scheduling problem (FSP) involves n jobs processed on m machines and if the sequences of processing jobs are the same for all machines, then the problem becomes permutation flow shop scheduling problem (PFSP). In this process, every job has some operations $O_{1,j}, O_{2,j}, \ldots, O_{m,j}$ and each operation has a non-negative processing time t_{ij}. The job operation $O_{i,j}$ must be only executed on machine i. A machine cannot perform more than one operation per time. Operation $O_{i,j}$ can be performed only after operation $O_{i-1,j}$ [28].

In this paper we have considered two objectives makespan and total due date cost $(ET_{\max})$ as objective functions of our problem. Assume permutation job set is $\{\pi_1, \pi_2, ..., \pi_n\}$, According to Yagmahan and Yenisey [29], completion times are shown as $C(\pi_i, j)$ then we have:

$$C(\pi_1,1) = t(\pi_1,1)$$
$$C(\pi_i,1) = C(\pi_{i-1},1) + t(\pi_i,1) \qquad i = 2,...,n$$
$$C(\pi_1, j) = C(\pi_1, j-1) + t(\pi_1, j) \qquad j = 2,...,m$$
$$C(\pi_i, j) = \max\{C(\pi_{i-1}, j), C(\pi_i, j-1)\} + t(\pi_i, j)\} \qquad i = 2,...,n \;\; ; j = 2,...,m$$

Then,

$$f_1 = Makespan = C(\pi_n, m),$$

According to notation in Moslehi et al. [30], in a flow shop, earliness (E_i) and tardiness (T_i) of each job i, maximum earliness $(E_{\max})$, maximum tardiness $(T_{\max})$, and the sum of maximum earliness and tardiness are considered as total due date cost $(ET_{\max})$.

$$E_{\max} = \max_{\forall i}\{\max\{0, d_i - t_{im}\}\} \qquad i = 2,...,n$$

$$T_{\max} = \max_{\forall i}\{\max\{0, t_{im} - d_i\}\} \qquad i = 2,...,n$$

And,

$$ET_{\max} = E_{\max} + T_{\max},$$

Then,

$$f_2 = ET_{\max} = Total \quad Duedate \quad Cost = \max_{\forall i}\{\max\{0, d_i - t_{im}\}\} + \max_{\forall i}\{\max\{0, t_{im} - d_i\}\}$$

$i = 2,...,n$

According to Shahsavari pour et al. [31], the fitness function is defined as Formula (1):

$$F_T = \left[w_{f_1} \times \frac{f_1 - f_{1_{\min}} + \gamma}{f_{1_{\max}} - f_{1_{\min}} + \gamma} \right] + \left[w_{f_2} \times \frac{f_2 - f_{2_{\min}} + \gamma}{f_{2_{\max}} - f_{2_{\min}} + \gamma} \right] \tag{1}$$

And, $w_{f_1} + w_{f_2} = 1 \quad and \quad 0 \le w_{f_1}, w_{f_2} \le 1$

where, w_{f_1}, w_{f_2} and are the planner-specified weight and indicate the relative importance of makespan and total due date cost. γ is a very small positive number in order to prevent dividing by zero in the fitness function. $f_{1_{\min}}$ and $f_{1_{\max}}$ are minimal and maximal values of makespan, $f_{2_{\min}}$ and $f_{2_{\max}}$ are minimal and maximal values of total due date cost in the current population.

3. Describing the presented GA

In this paper, a new genetic algorithm is proposed, which the following has been studied and explained the different features and parameters. First, to search deeper in each generation, some numbers of permutations are generated randomly and are added to total population for selecting in the next generation. By doing this procedure, are produced several new permutations in each generation. This increases the probability of finding an optimal answer for problem. In the second step, a tabu search algorithm (as described in the following) is generated, if they create better answers, the best of them is replaced by the first permutation. Thus, problem environment will checked more, and increases the probability of achieving the optimal solution. And finally, to avoid elimination of the top answers in every generation has used the concept of elitism. For this goal, in each generation some of the best sequences to be transferred directly to the roulette wheel in next generation. With this action with higher chance we will be able to keep the top answers in every generation.

Tabu search was introduced first by Glover [32] and then in the next articles paid to develop and introduce applications of this method [33-34]. This method is a sequential algorithm, that obtains a set of problem solutions, with successive moves from one solution x_n to another solution as x_{n+1} in its neighborhood $V(x_n)$. The movements are done with aim of reaching a good solution (optimal or near-optimal) and evaluate the objective function such $F(x)$ that must be minimized. A flow chart of algorithm is provided at Figure 1 and a pseudo code of this algorithm is shown below:

Pseudo-code for a simple tabu search (TS)

1. Set $n=1$, Select x_0 and set $x^* = x_0$.

2. Select x_i from $V(x_n)$.

2.1- If the move x_n x_i is on the tabu list set $x_{n+1} = x_n$ and go to 3

2.2- If the move x_n x_i is not on the tabu list set $x_{n+1} = x_i$,

Add the reverse move to the top of the tabu list

and delete the entry on the bottom.

If $F(x_i) \prec F(x^*)$, set $x^* = x_i$.

3. Set $n = n+1$.
Stop if stopping criteria are satisfied; otherwise go to 2.

---

4. Computational results

In this section, two examples are presented to evaluate the proposed method. To perform experiments we coded our algorithm in visual basic for applications (VBA) and run it in Intel T6570, 2.1 GHz with 2 GB RAM.

Example 1: At first is used from the paper Ho and Chang [35]. At this paper is presented an example with 4 machines and 5 jobs. The processing time of each job on each machine are given At Table 1. The objective is the minimization of makespan.

The best sequence obtained for this problem is (4, 2, 5, 1, 3) with the makespan = 213. We have implemented our algorithm 100 times with this processing time which the method presented in this paper has achieved to this number in 97% of the experiments. Table 2 shows the comparison of makespan value achieved by various articles.

Example 2: In this example we used the Reeves problems [36], which the test problems can be downloaded from OR-library web site (http://people.brunel.ac.uk/~mastjjb/jeb/info.html).

To find appropriate values for the parameters, algorithm was performed with different values, and the best values that obtained from the experiment are used as a basis for future experiments. Examples are including (20, 30, 50, 75) jobs and (5, 10, 15, 20) machines.

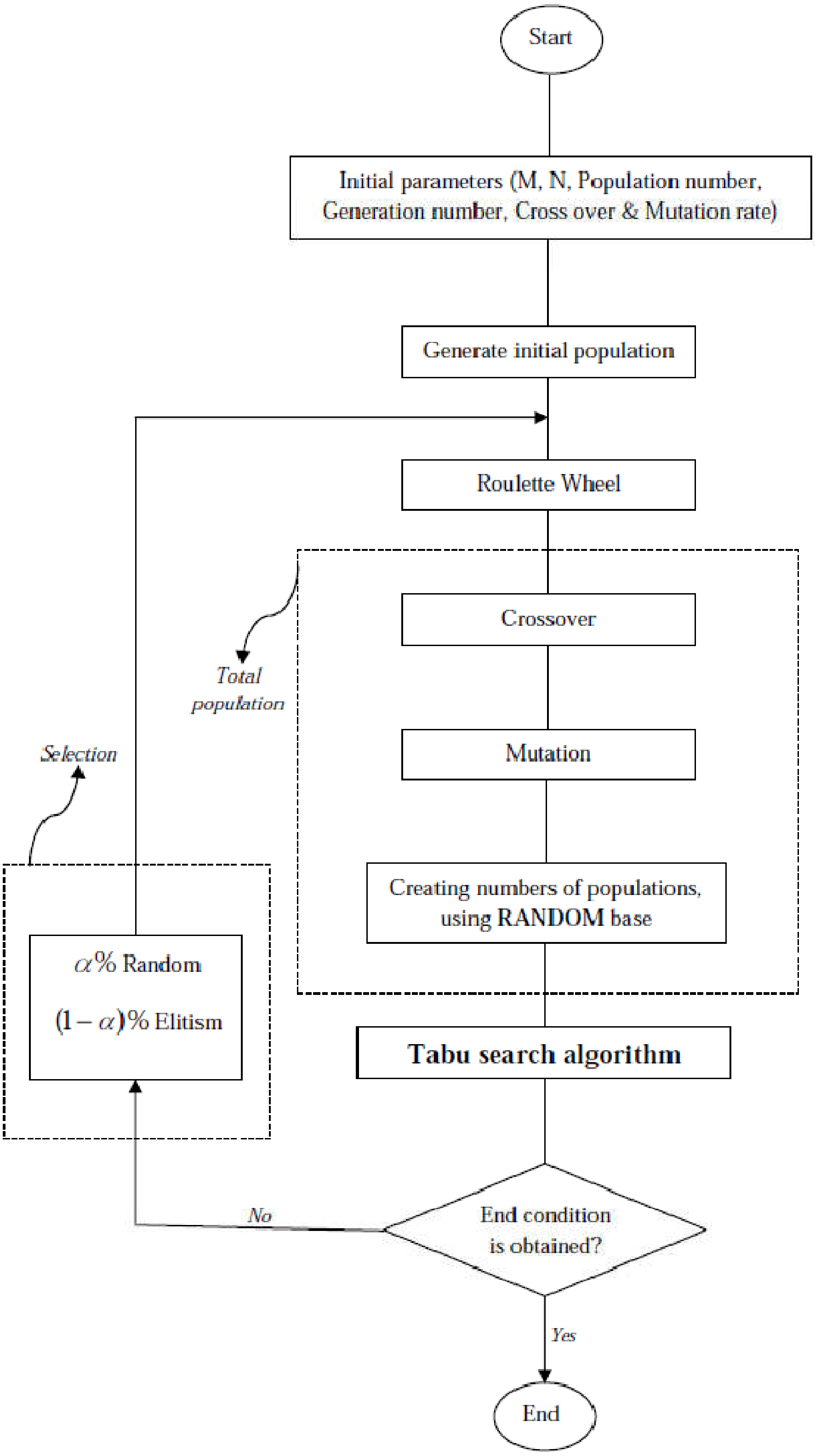

Figure 1. Flow chart of presented GA algorithm

The relative percentage increase (Δ) in any objective value for schedule T generated by any algorithm is given as (2). f_i' and f_i'' are objective value obtained by GA algorithm and proposed algorithm which finds after run program 200 times for each problem. After finding the minimum for any objective, results are repeated 20 times to find $\Delta(f_1)$ and $\Delta(f_2)$, and their averages are selected as the output. The relative percentage increase in total objective value (Δ_t) for schedule T generated is calculated as formula (3). Equal relative weighting is chosen to the makespan and total due date cost for total objective value, then $\alpha_1 = \alpha_2 = 0.5$. The results are proposed in Tables 3. Results of experiments show the superiority of the proposed approach.

$$\Delta(f_i) = \left(\frac{f_i - \min(f_i', f_i'')}{\min(f_i', f_i'')}\right) * 100 \qquad i = 1,2 \tag{2}$$

$$\Delta_t = \alpha_1 \Delta(f_1) + \alpha_2 \Delta(f_2) \tag{3}$$

Table 1. Process times for the example

Job	M_1	M_2	M_3	M_4
J_1	31	41	25	30
J_2	19	55	3	34
J_3	23	42	27	6
J_4	13	22	14	13
J_5	33	5	57	19

Table 2. Comparison of different articles

Articles	**Permutation**	**Makespan**
Ho and Chang	(4, 2, 5, 1, 3)	213
Dannenbring	(5, 1, 2, 3, 4)	256
Gupta	(2, 1, 5, 3, 4)	251
Campbell et.al	(4, 2, 1, 5, 3)	246
Palmer	(5, 2, 4, 1, 3)	245
Hundal and Rajgopal	(4, 5, 2, 1, 3)	236
NEW APPROACH	(4, 2, 5, 1, 3)	213

Table 3. Result of comparison of GA with proposed algorithm

Problem number	$N \times M$	GA			Proposed approach		
		$\Delta(f_1)$	$\Delta(f_2)$	Δ_t	$\Delta(f_1)$	$\Delta(f_2)$	Δ_t
ReC01	20×5	0.79	1.73	1.26	0.37	0.29	0.33
ReC03	20×5	0.33	2.41	1.37	0.10	0.81	0.46
ReC05	20×5	0.91	1.09	1.00	0.49	0.20	0.35
ReC07	20×10	1.71	3.19	2.45	0.29	0.59	0.44
ReC09	20×10	0.70	1.61	1.16	0.92	0.18	0.55
ReC11	20×10	1.12	2.99	2.06	0.42	0.69	0.56
ReC13	20×15	1.01	0.99	1.00	0.18	1.91	1.05
ReC15	20×15	1.79	1.81	1.80	0.30	1.01	0.66
ReC17	20×15	1.42	2.78	2.10	0.68	0.92	0.80
ReC19	30×10	1.29	3.10	2.20	1.99	0.81	1.40
ReC21	30×10	1.82	2.61	2.22	0.38	0.30	0.34
ReC23	30×10	1.19	2.88	2.04	0.22	1.29	0.76
ReC25	30×15	1.92	1.30	1.61	0.81	1.04	0.93
ReC27	30×15	2.92	3.39	3.16	0.49	0.81	0.65
ReC29	30×15	1.28	1.29	1.29	1.39	1.30	1.35
ReC31	50×10	1.88	2.71	2.30	1.20	1.61	1.41
ReC33	50×10	2.18	1.06	1.62	1.16	0.72	0.94
ReC35	50×10	1.76	2.47	2.12	0.55	1.04	0.80
ReC37	75×20	2.77	1.17	1.97	0.82	0.69	0.76
ReC39	75×20	1.30	3.81	2.56	1.79	1.17	1.48
ReC41	75×20	1.81	2.91	2.36	1.20	0.92	1.06
Average		1.519	2.252	**1.886**	0.750	0.871	**0.811**

5. Conclusions

This paper presented an efficient algorithm for multi-objective problems. A bi-objective flow shop scheduling problem was considered to verify the performance of this algorithm. The makespan and total due date cost were this two objectives and Total due date cost was included the sum of earliness and tardiness cost. The genetic algorithms were presented for solving this problem involved some features that would help to improve its performance. In this algorithm, in order to improve local searching, the tabu search algorithm was used. The results of implementing these algorithms in the computer and comparison of them with genetic algorithm, showed better performance for presented algorithm. In future, this algorithm can be applied to other multi-objective problems.

Acknowledgment

Special thanks to dear Dr. Shahsavari pour for his honestly and valuable efforts in training and enhance the knowledge of his students.

References

[1] Johnson, S.M. (1954), Optimal two- and three-stage production schedules with setup times included, *Naval Research Logistics Quarterly*, Vol. 1, No. 1, pp. 61–68.

[2] Ashour, S. (1970), An experimental investigation and comparative evaluation of flow shop sequencing techniques, *Operations Research*, Vol. 18, No. 3, pp. 541–549.

[3] Baker, K.R. (1975), A comparative study of flow shop algorithms, *Operations Research*, Vol. 23, pp. 62–73.

[4] Ignall, E. and Schrage, L. (1965), Application of the branch and bound technique to some flow shop scheduling problems, *Operations Research*, Vol. 13, No. 3, pp. 400–412.

[5] McMahon, G.B. and Burton, P. (1967), Flow shop scheduling with branch and bound method, *Operations Research*, Vol. 15, pp. 473–481.

[6] Thornton, H.W. and Hunsucker, J.L. (2004), A new heuristic for minimal makespan in flow shops with multiple processors and no intermediate storage, *European Journal of Operational Research*, Vol. 152, pp. 96–114.

[7] Bouquard, J.L., Billaut, J.C., Kubzin, M.A. and Strusevich, V.A. (2005), Two-machine flow shop scheduling problems with no-wait jobs, *Operations Research Letters*, Vol. 33, No. 3, pp. 255–262.

[8] Eksioglu, B., Eksioglu, S.D. and Jain, P. (2008), A tabu search algorithm for the flow shop scheduling problem with changing neighborhoods, *Computers and Industrial Engineering*, Vol. 54, pp. 1–11.

[9] Abolhasani Ashkezari, M.H., Shahsavari Pour N. and Mohammadi Andargoli, H. (2012), An ant colony system for solving fuzzy flow shop scheduling problem, *International Journal of Engineering and Technology*, Vol. 1, No. 2, pp. 44-57.

[10] Tseng, L.Y. and Lin, Y.T. (2009), A hybrid genetic local search algorithm for the permutation flow shop scheduling problem, *European Journal of Operational Research*, Vol. 198, pp. 84–92.

[11] Zhang, Y., Li, X. and Wang, Q. (2009), Hybrid genetic algorithm for permutation flow shop scheduling problems with total flow time minimization, *European Journal of Operational Research*, Vol. 196, pp. 869–876.

[12] Zheng, T., and Yamashiro, M. (2010), Solving flow shop scheduling problems by quantum differential evolutionary algorithm, *International Journal of Advanced Manufacturing Technology*, Vol. 49, pp. 643–662.

[13] Zobolas, G.I., Tarantilis, C.D. and Ioannou, G. (2009), Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm, *Computers and Operations Research*, Vol. 36, pp. 1249–1267.

[14] Jarboui, B., Ibrahim, S., Siarry, P. and Rebai, A. (2008), A combinatorial particle swarm optimization for solving permutation flow shop problems, *Computers and Industrial Engineering*, Vol. 54, pp. 526–538.

[15] Kuo, I.H., Horng, S.J., Kao, T.W., Lin, T.L., Lee, C.L., Terano, T. and Pan, Y. (2009), An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model, *Expert Systems with Applications*, Vol. 36, No. 3, pp. 7027–7032.

[16] Lian, Z., Gu, X. and Jiao, B. (2008), A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan, Chaos, Solitons and Fractals, Vol. 35, pp. 851–861.

[17] Zhang, C., Ning, J. and Ouyang, D. (2010), A hybrid alternate two phases particle swarm optimization algorithm for flow shop scheduling problem, *Computers and Industrial Engineering*, Vol. 58, pp. 1–11.

[18] Zhang, J., Zhang, C. and Liang, S. (2010), The circular discrete particle swarm optimization algorithm for flow shop scheduling problem, *Expert Systems with Applications*, Vol. 37, pp. 5827–5834.

[19] Chang, P.C., Chen, S.H. and Lin, K.L. (2005), Two-phase sub population genetic algorithm for parallel machine-scheduling problem, *Expert Systems with Applications*, Vol. 29, No. 3, pp. 705–712.

[20] Hu, J., Goodman, E., Seo, K., Fan, Z. and Rosenberg, R. (2005), The hierarchical fair competition framework for sustainable evolutionary algorithms, *Evolutionary Computation*, Vol. 13, No. 2, pp. 241–277.

[21] Chang, P.C., Chen, S.H. and Liu, C.H. (2007), Sub-population genetic algorithm with mining gene structures for flow shop scheduling problems, *Expert Systems with Applications*, Vol. 33, No. 3, pp. 762–771.

[22] Alves, M.J. and Almeida, M. (2007), MOTGA: A multi-objective Tchebycheff based genetic algorithm for the multidimensional knapsack problem, *Computers and Operations Research*, Vol. 34, No. 11, pp. 3458–3470.

[23] Tan, K.C., Goh, C.K., Mamun, A.A. and Ei, E.Z. (2008), An evolutionary artificial immune system for multi-objective optimization, *European Journal of Operational Research*, Vol. 187, No. 2, pp. 371–392.

[24] Deb, K., AmritPratap, S.A. and Meyarivan, T. (2000), A fast and elitist multi objective genetic algorithm-NSGA-II, in: *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pp. 849–858.

[25] Zitzler, E., Laumanns, M. and Bleuler, S. (2004), A tutorial on evolutionary multi objective optimization, in: *Proceedings of The Workshop on Multiple Objective Metaheuristics*, Berlin, Germany. Springer-Verlag, pp. 3–37.

[26] Tavakkoli-Moghaddam, R., Rahimi-Vahed, A.R. and Mirzaei, A.H. (2007), Solving a bi-criteria permutation flow shop problem using immune algorithm, in: *Proceedings of the First IEEE Symposium on Computational Intelligence*, Honolulu, Hawaii, Vol. 1, pp. 49-56.

[27] Chang, P.C., Chen, S.H., Fan, C.Y. and Chan, C.L. (2008), A Genetic algorithm integrated with artificial chromosomes for multi-objective flow shop scheduling problems, *Applied Mathematics and Computation*, Vol. 205, pp. 550–561.

[28] Deng, G. and Gu, X. (2012), A hybrid discrete differential evolution algorithm for the no-idle permutation flow shop scheduling problem with makespan criterion, *Computers and Operations Research*, Vol. 39, pp. 2152–2160.

[29] Yagmahan B. and Yenisey, M.M. (2008), Ant colony optimization for multi-objective flow shop scheduling problem, *Computers and Industrial Engineering*, Vol. 54, pp. 411–420.

[30] Moslehi, G., Mirzaee, M., Vasei, M., Modarres, M. and Azaron, A. (2009), Two-machine flow shop scheduling to minimize the sum of maximum earliness and tardiness, *International Journal of Production Economics*, Vol. 122, pp. 763–773.

[31] Shahsavari pour, N., Modarres, M., Tavakkoli-Moghaddam R. and Najafi, E. (2010), Optimizing a multi-objectives Time-Cost-Quality trade-off problem by a new hybrid genetic algorithm, *World Applied Sciences Journal*, Vol. 10, No. 3, pp. 355-363.

[32] Glover, F. (1986), Future Paths for Integer Programming and Links to Artificial Intelligence, *Computers and Operations Research*, Vol. 13, No. 5, pp. 533-549.

[33] Glover, F. (1989), Tabu Search - Part 1, *ORSA Journal on Computing*, Vol. 1, No. 2, pp. 190–206.

[34] Glover, F. (1990), Tabu Search - Part 2, *ORSA Journal on Computing*, Vol. 2, No. 1, pp. 4–32.

[35] Ho, J.C. and Chang, Y.L. (1991), A new heuristic for the n-job, M-machine flow-shop problem, *European Journal of Operational Research*, Vol. 52, pp. 194- 202.

[36] Reeves, C.R. (1995), A genetic algorithm for flow shop sequencing, *Computers and Operations Research*, Vol. 22, pp. 5–13.