Single Machine Preemptive Scheduling by Hybridized Meta-Heuristic Approach

N. Javadian
Department of Industrial Engineering, Mazandaran University of Science and Technology, P.O.Box 734, Babol, Iran
E-mail: nijavadian@ustmb.ac.ir

J. Rezaeian
Department of Industrial Engineering, Mazandaran University of Science and Technology, P.O.Box 734, Babol, Iran
E-mail: j_rezaeian @ustmb.ac.ir

H. Khorshidian
Department of Industrial Engineering, Mazandaran University of Science and Technology, P.O.Box 734, Babol, Iran
E-mail: h.khorshidian@ustmb.ac.ir

K. Rahmani
Department of Industrial Engineering, Mazandaran University of Science and Technology, P.O.Box 734, Babol, Iran
E-mail: k.rahmani@ustmb.ac.ir

***Abstract*—this paper is investigated the bi-criteria scheduling problem of minimizing the sum of the earliness and tardiness penalties on a single machine while the preemption is allowed. The problem which is known to be NP-hard is compatible with the concepts of just-in-Time (JIT) production. While the complexity of the problem is high, therefore, meta-heuristic algorithms are highly applied for such problems. Imperialist competitive algorithm (ICA) is a novel global search heuristic that uses imperialistic competition as a source of inspiration. Two meta-heuristic algorithms such as ICA and hybrid algorithm of ICA and genetic algorithm are applied. The computational results show that the performance of the hybrid algorithm is much better than ICA in finding the best solutions and in execution speed.**

Keywords- Single machine scheduling; Just-in-time; Preemption; Hybrid algorithm; Imperialist competitive algorithm; Genetic algorithm

I. INTRODUCTION

This paper considers a single machine scheduling problem with earliness and tardiness costs. Also, existence of preemption and machine idle time in sequencing are allowed. The non-preemptive single machine scheduling problem with earliness and tardiness costs has been regarded as an NP-hard problem which has been widely studied in recent years [6]. Also, several exact algorithms and heuristics were proposed [2], [4] and [7]. An overview of existing and new results on preemption consideration for single machine scheduling problems is presented in [3]. Recently, a neighborhood search (NS) is applied in the JIT single machine scheduling problem with preemption [5].

Imperialist competitive algorithm (ICA) is a novel global search heuristic that uses imperialism and imperialistic competition process as a source of inspiration. The term imperialist competitive algorithm was introduced by [1]. In this paper, two meta-heuristic algorithms based on JIT scheduling problem (JITSP) are presented: ICA and hybrid algorithm of ICA and genetic algorithm. The computational results show that the performance of the hybrid algorithm is much better than ICA in finding the best solutions and also in execution speed. Our computational experiments demonstrated that the hybridized approach yields excellent results.

II. PROBLEM DEFINITION

The problem can be stated as follows. A set of n independent jobs $\{j_1, j_2, ..., j_n\}$ has to be scheduled on a single machine which can handle at most one job at a point of time. The machine is assumed to be continuously available and breakdown is not occurred.

There is a due date d_i for each job j_i, which is to be processed on a single machine. Each of the n jobs is available for processing at time zero and has a determinate processing time p_i, earliness weight α_i, and tardiness weight β_i.

III. PROPOSED IMPREIALIST COMPETITIVE ALGORITHM

A. *The representation scheme*

In this procedure, the country divided to H equal periods, where H is obtained by following formula:

$$H = \max\{\sum_{i=1}^{n} p_i, d_{\max}\} + n \ . \quad (1)$$

Number of periods that assigned to job j_i is equal to the corresponding processing time. In this way, remained periods of country which are not assigned, are considered as idle periods. Each period is presented by the city and each number in country is presented by the value of the city. At first, a primal sequence (primal country) is produced that the value of cities are randomly distributed between 1 and H, here 9, and then a decoding method is implemented. By decoding method we transformed numbers between 1 and the p_1 to job j_1. In the same way, we decoded numbers between p_1+1 and p_1+ p_2 here 4 and 5 to job j_2. If number of jobs is more, the procedure will go on in the same way.

978-1-61284-486-2/11/$26.00 ©2011 IEEE

B. The initial population

To start the optimization algorithm we produce the initial population of size N_{pop}. We select N_{imp} of the most powerful countries to form the initial empires. The remaining N_{col} of the population will be the colonies each belongs to an empire. Then, we have two types of countries: imperialist and colony. All the colonies of initial population consists primal sequences that the quantity of these sequences is equivalent to the number of countries.

Each city of country is denoted by q_i.

$$\text{country} = (q_1, q_2, q_3, \ldots, q_H) \quad (2)$$

C. Power evaluation

The cost of a country is obtained by evaluating the cost function f at the variables $(q_1, q_2, q_3, \ldots, q_H)$. Then $\text{cost} = f(q_1, q_2, q_3, \ldots, q_H)$

According to the cost function, power of countries must be obtained by calculating two parameters, start time and completion time.

By calculating these parameters, earliness and tardiness of each job is obtained by formulas (3) and (4):

$$E_i = \max\{0, d_i - p_i - S_i\} \quad (3)$$

$$T_i = \max\{0, C_i - d_i\} \quad (4)$$

Finally, objective function of each country is calculated by E_i, T_i in formula (5).

$$\text{Cost function} = \sum_{i=1}^{n} (\alpha_i E_i + \beta_i T_i) \quad (5)$$

In such JIT problems that aim to minimize penalized earliness-tardiness, power of countries is different from their cost. We need to set an upper bound equal to a large number (M) for calculating power of countries. Formula (6) provides power functions.

$$\text{Power function} = M - \text{Cost function} \quad (6)$$

D. Division strategy

To form the initial empires, the colonies are divided among imperialists based on their power. That is the initial number of colonies of an empire should be directly related to its power. To divide the colonies among imperialists, the normalized cost of an imperialist is defined by formula (7).

$$Q_v = \max\{q_z\} - q_v \quad (7)$$

Where q_v is the cost of vth imperialist and Q_v is its normalized cost. The normalized power of each imperialist is defined by formula (8).

$$G_v = \left| \frac{Q_v}{\sum_{z=1}^{N_{imp}} Q_z} \right| \quad (8)$$

In addition, the normalized power of an imperialist is the portion of colonies that should be possessed by the imperialist. Then, the initial number of colonies of an empire will be obtained by formula (9).

$$N.Q_v = round\left\{G_v.(N_{col})\right\} \quad (9)$$

Where $N.Q_v$ is the initial number of colonies of vth empire and N_{col} is the number of colonies. For dividing the colonies, $N.Q_v$ of the colonies are chose randomly and are allocated to each imperialist. These colonies along with the imperialist will form vth empire. According to this procedure, bigger empires have greater number of colonies while weaker ones have less.

E. Moving the colonies of an empire toward the imperialist

Imperialists countries started to improve their colonies. We have simulated moving all the colonies toward the imperialist. The direction of the movement is the vector from colony to imperialist.

F. Competition strategies

In applying the ICA, two types of competition are seen. The first one is competition between imperialist and colony in which four countries must be considered: two offspring by crossover operator, one offspring by mutation operator and the selected colony. Values of objective function of these countries must be compared. If one of the three offspring has a lower cost than the selected colony, that offspring is replaced by the selected colony. Otherwise, crossover and mutation operators are repeated.

During moving toward the imperialist, a colony may reach to a position with lower cost than the imperialist. In such a case, the imperialist moves to the position of that colony and vice versa. Then algorithm will continue by the imperialist in a new position and then colonies start moving toward this position.

The second one is Competition between two imperialist countries. Total power of an empire is largely affected by the power of imperialist country. But the power of the colonies of an empire has an effect, albeit negligible, on the total power of that empire. This fact is modeled by defining the total cost in formula (10).

$$T.Q_v = \text{Cost}(\text{imperialist}_v) + \xi \,\text{mean}\left\{\text{Cost}(\text{colonies of empire}_v)\right\} \quad (10)$$

Where $T.Q_v$ is the total cost of the vth empire and ξ is a positive number which is considered to be less than 1.

A little value for ξ causes the total power of the empire to be determined by just the imperialist and increasing it will enhance the role of the colonies in determining the total power of an empire.

Indeed, all empires try to take possession of colonies of other empires and control them. This imperialistic competition gradually brings about a decrease in the power of weaker empires and an increase in the power of more powerful ones. This competition is modeled by just picking one of the weakest colonies of the weakest empires and making a competition among all empires to posses this colony. The modeled imperialistic competition based on their total power works as follows. Each of empires will have a probability of taking possession of the mentioned colonies.

The competition starts with finding the possession probability of each empire based on its total power. The normalized total cost is simply obtained by $N.T.Q._v = \max_z \{T.Q._z\} - T.Q._v$ where $T.Q._v$ and $N.T.Q._v$ are respectively total cost and normalized total cost of *v*th empire. The possession probability of each empire is given by formula (11).

$$G_{G_v} = \left| \frac{N.T.Q._v}{\sum_{z=1}^{N_{imp}} N.T.Q._z} \right| \quad (11)$$

To divide the mentioned colonies among empires based on the possession probability of them, the vector G is formed as $G = [G_{q_1}, G_{q_2}, \ldots, G_{q_{N_{imp}}}]$.

Then, a vector with the same size as G whose elements are uniformly distributed random numbers is considered.

$$Y = \left[y_1, y_2, y_3, \ldots y_{N_{imp}} \right] \qquad y_1, y_2, y_3, \ldots y_{N_{imp}} \sim U(0,1)$$

Then, vector A is formed by simply subtracting Y from G.

$$A = G - Y = [A_1, A_2, A_3, \ldots A_{N_{imp}}] =$$

$$[G_{u_1} - y_1, G_{u_2} - y_2, G_{u_3} - y_3, \ldots G_{u_{N_{imp}}} - y_{N_{imp}}]$$

Referring to vector A we will hand the mentioned colonies to an empire whose relevant index in A is maximum.

G. Eliminating the powerless empires

Powerless empires will destroy in the imperialistic competition and the colonies will be divided among remained empires. In modeling collapse mechanism, different criteria can be assumed for concerning an empire powerless. In our implementation, we consider that an empire is collapsed and eliminate it when it loses all of its colonies.

H. Stopping criterion

After a while, all the empires expect the most powerful one will collapse and all the colonies will be under the control of this unique empire.

IV. THE HYBRID ALGORITHM OF ICA AND GA

The performance of the proposed hybrid algorithm can be divided into two sections. The first part is producing an appropriate solution (in fact, the first part is an input for the second one). The task of the second part is improving the population into an optimal sequence. Since ICA make more searches on different sequences of JITSP due to its high execution speed, it is used as a good tool for producing the appropriate population (a good first solution) for GA. On the other hand, GA can be a good tool for improving the population into an optimal sequence due to its accuracy in finding global optimum solutions.

V. COMPUTATIONAL RESULTS

A. Data generation

In this section, performance of the proposed algorithms for JITSP is analyzed. To present the efficiency of the algorithms, problems with different sizes are considered. The small size problems are associated with 30 and 40 jobs, medium size with 50 and 60 jobs, and large size with 80 and 90 jobs. In order to compare the algorithms, 18 examples are designed and the best, average and the worst results were submitted after 5 runs for each of them. There are 3 instances for each problem size.

The processing times are generated from the discrete uniform distribution [1, 9] and earliness-tardiness penalties are drawn from the discrete uniform distribution [1, 4]. The due dates of each job is drawn from the uniform distribution $[d_{\min} - \lambda, d_{\min} + \lambda]$, where $d_{\min} = P(1 - TEF)$, $P = \sum_{i=1}^{n} p_i$ and $\lambda = P(RDD/2)$. The two parameters TEF and RDD are the tardiness/earliness function and relative range of due dates, respectively. RDD gets the values of 0.2 and 0.5. Also TEF gets the values of 0.2 and 0.35.

B. Experimental results

The comparison between the results of several procedures is shown in Table 1. Numerical examples show that ICA is unable to find optimal solutions in small size problems but if the problem size expands, the ICA obtain good performance in processing JITSP.

Table1
Comparsion of the metaheuristic procedures

n	Heur.	Solution			Solution			Solution			
		Ins1			Ins2			Ins3			
		Best	Avg	Worst	Best	Avg	Worst	Best	Avg	Worst	CPU
30	ICA	4072	4314	4630	3339	4183	3836	3600	3778	3963	12min
	Hybrid	3117	3500	4110	2929	3017	3163	3070	3210	3411	12min
40	ICA	7198	7320	7540	7879	8414	8914	7659	8071	8415	12min
	Hybrid	5711	6301	6945	6995	7505	7983	6513	7111	8027	12min
50	ICA	10214	10787	11353	6735	7068	7297	11688	12181	12709	14min
	Hybrid	8413	8830	9147	5015	5669	6325	9985	10556	10918	14min
60	ICA	16896	17447	17890	17514	17991	18512	13793	14846	15602	14min
	Hybrid	14899	15424	16107	15780	16566	17410	12745	14001	15269	14min
80	ICA	30443	31776	33086	29640	30261	31673	29263	30808	32493	16min
	Hybrid	26945	28296	30673	26856	27817	28524	26084	27268	29151	16min
90	ICA	35421	37160	38896	41002	43861	47772	39315	41816	42802	16min
	Hybrid	34216	35889	37754	35746	37510	39513	34123	36364	38570	16min

The main advantage of the ICA is its high execution speed. Numerical experiments demonstrate that ICA doesn't have the appropriate performance in small size problems, while for medium size and large size problems the performance of ICA is well. Thus, to be concluding that the performance and quality of the hybrid algorithm is better than ICA for small, medium and large size problems.

VI. CONCLUTIONS

In this paper, we developed two meta-heuristic algorithms for the JIT scheduling with preemption and machine idle time: ICA and hybrid algorithm of ICA and GA. The proposed procedures were compared with each other. Indeed, the hybrid algorithm is a better algorithm than the ICA cause of using advantages of the two algorithms. Our computational experiments demonstrated that this approach yields excellent results. Therefore, our future work will consist in using the hybrid algorithm to solve some of more practical optimization problems. Moreover, this work can be extended to more complex settings, such as parallel machine environments. It would also be interesting to study robustness and stability measures in dynamic and stochastic manufacturing settings.

REFERENCES

[1] Atashpaz-Gargari, E., Lucas, C., 2007. Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic. *IEEE Congress on Evolutionary Computation*, 25-28 September 2006 Singapore. 4661–4667.

[2] Ballestín, F., Tautmann,N., 2008. An iterated-local-search heuristic for the resource-constrained weighted earliness-tardiness project scheduling problem. *International Journal of Production Research*, 46 (22), 6231-6249.

[3] Brucker, P., Heitmann,S., Hurink, J., 2003. How useful are preemptive schedules?. *Operations Research Letters,* 31 (2) 129 – 136.

[4] Hepdogan, S., Moraga, R., DePuy, G.W., Whitehouse, G. E., 2009. A Meta-RaPS for the early/tardy single machine scheduling problem. *International Journal of Production Research*, 47 (7), 1717-1732.

[5] Hendel, F., Runge, N., Sourd, F., 2009. The one-machine just-in-time scheduling problem with preemption. *Discrete Optimization,* 6 (1), 10-22.

[6] Lenstra, J. K., Rinnooy Kan, A. H., Brucker, G. P., 1977. Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1, 343-362.

[7] Mahnam, M., Moslehi, Gh., 2009. A branch-and-bound algorithm for minimizing the sum of maximum earliness and tardiness with unequal release times. *Engineering Optimization*, 41 (6), 521-536.