Indian Journal of Science and Technology Vol. 4 No. 10 (Oct 2011) ISSN: 0974- 6846

Solving flexible job-shop scheduling problem using clonal selection algorithm

Mohammad Akhshabi[1]* , Mostafa Akhshabi[2] and Javad Khalatbari[3]

[1*]*Department of Industrial Engineering, Islamic Azad University Ramsar Branch , P.O. Box 46819-66856, Ramsar, Iran*
[2]*Young Researchers Club, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran*
[3]*Department of Managment, Islamic Azad University Ramsar Branch , P.O. Box 46819-66856, Ramsar, Iran*
m.akhshabi@toniau.ac.ir*; mostafa.akhshabi@iau-neyshabur.ac.ir;j.khalatbari@toniau.ac.ir

Abstract
In this paper we present a clonal selection algorithm (CSA) to solve the Flexible Job-shop Scheduling Problem (FJSP). The scheduling objective minimizes the maximal completion time of all the operations, which is denoted by Makespan. The performance of CSA is evaluated in comparison with Lingo 8 software. Experimental results prove that the CSA can be used to solve FJSP effectively.

Keywords: Flexible Job-shop Scheduling Problem, Clonal selection algorithm, Makespan.

Introduction

The job-shop scheduling problem (JSP) has been studied for more than 50 years in both academic and industrial environments. Jain and Meeran (1999) provided a concise overview of JSPs over the last few decades and highlighted the main techniques. The JSP is the most difficult class of combinational optimization. Garey *et al.* (1976) demonstrated that JSPs are non-deterministic polynomial-time hard (NP-hard); hence we cannot find an exact solution in a reasonable computation time. The single objective JSP has attracted wide research attention. Most studies of single-objective JSPs result in a schedule to minimize the time required to complete all jobs, *i.e.*, to minimize the makespan. Many approximate methods have been developed to overcome the limitations of exact enumeration techniques. These approximate approaches include simulated annealing (SA) (Lourenço, 1995), tabu search (Sun *et al.,* 1995; Nowicki & Smutnicki, 1996; Pezzella & Merelli, 2000) and genetic algorithms (GA) (Bean, 1994; Kobayashi *et al.,*1995; Gonçalves *et al.,* 2005; Wang & Zheng, 2001). However, real world production systems require simultaneous achievement of multiple objective requirements. This means that the academic concentration of objectives in the JSP must been extended from single to multiple.

Flexible Job-shop Scheduling Problem (FJSP) is an extension of the classical Job-shop Scheduling Problem (JSP) which allows an operation to be processed by any machine from a given set. Currently the approaches for solving FJSP mainly include GA (Chen *et al.*,1999), Simulated Annealing (SA) (Najid *et al.*, 2002), Tabu Search (TS) (Barnes & Chambers,1996) and some hybridization algorithms (Kacem *et al.,* 2002). Xia and Wu (2005) developed an easily implemented approach for the multi-objective flexible JSP based on the combination of PSO and SA. They demonstrated that their proposed algorithm was a viable and effective approach to the multi-objective flexible JSP, especially for large-scale problems.

Mastrolilli and Gambardella (2000) proposed a tabu search procedure with effective neighborhood functions for the flexible job shop problem. Many authors have proposed a method of assigning operations to machines and then determining sequence of operations on each machine. Pezzella *et al.* (2008) and Gao *et al.* (2008) proposed the hybrid genetic and variable neighborhood descent algorithm for this problem. There are only a few papers considering parallel algorithms for the FJSP. Yazdani *et al.* (2010) propose a parallel variable neighborhood search (VNS) algorithm for the FJSP based on independent VNS runs. Defersha and Chen (2009) describe a coarse-grain version of the parallel genetic algorithm for the considered. FJSP basing on island-model of parallelization focusing on genetic operators used and scalability of the parallel algorithm. Both papers are focused on parallelization side of the programming methodology and they do not use any special properties of the FJSP.

The clonal selection algorithm (CSA) is a population-based stochastic algorithm inspired by the clonal selection principle of acquired immunity. In this paper, the optimization mechanism of the CSA is analyzed and a general optimization model is proposed. A clonal selection algorithm according to the model is presented to tackle the FJSP. Experimental results on benchmark problems show that FJSP can be solved by CSA effectively.

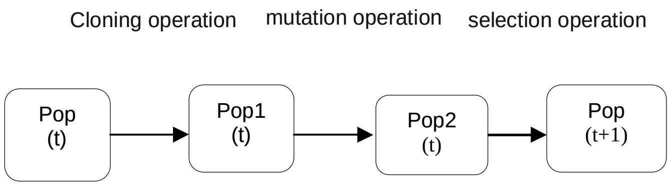
Flexible Job-shop Scheduling Problem

The n×m Flexible Job-shop Scheduling Problem (FJSP) can be formulated as follows:
There is a set of n jobs j= $\{j_{1,}j_{2,...,,}j_{n,}\}$ and a set of m machines M=$\{m_{1,}mj_{2,...,,}m_{m,}\}$ Each job J consists of a sequence of operations $o_{i1}, o_{i2}, \ldots, o_{in_i}$ where n_i is the number of operations that j_i comprises. Each operation o_{ij} has to be processed by one machine out of a set of given

machines $M_{ij} \in M$. the problem is thus to both determine an assignment and a sequence of the operations on the machines so that some criteria are attained. In this paper, the scheduling objective is to minimize the maximal completion time of all the operations, which is denoted by Makespan.

Clonal selection algorithm

Fig. 1. Basic state transfer flowchart of CSA0



CSA contains three operations, which are cloning, mutation, and selection. A set of antibodies are selected, which have the best highest affinity with antigen from the antibodies. The selected antibodies are cloned in proportion to their affinity with the antigen (rank based). All the copies are mutated. The mutation degree is inversely rated to their parent's affinity. All the copies are added to the antibodies. A set of antibodies from all antibodies that have the best highest affinity with the antigen are re-selected. Fig.1 presents the basic state transfer flowchart of CSA. Since CSA has the advantage of being simple to implement and easy to understand, it has shown considerable success in solving a variety of optimization problem. CSA also has been reported in scheduling area. For example, Yang proposed a clonal selection-based memetic algorithm for solving job shop scheduling problems. Kumar *et al*. (2008) extended the artificial immune system approach by proposing a new methodology termed as psycho-clonal algorithm to solve m-machine no-wait flow shop problem.

Encoding

Clonal are corresponding to the solutions of FJSP. In our coding each Clonal consists of two parts of strings, A-string and B-string. A-string defines the routing policy of the problem, and B-string defines the sequence of the operations on each machine. The elements of the strings respectively describe a concrete allocation of operations to each machine and the sequence of operations on each machine.

Assuming the total number of operations is l, and the i^{th} operation can be processed by a machine set $S_i=\{m_{i1,}mi_{2,...,}m_{in_i}\}$. The length of A-string is l and it can be denoted i^{th} as $g_1, g_{2,} \dots, g_i$. The i^{th} element g_i is an integer between 1 an in and n_i it means that the i^{th} operation is assigned to $g_i{}^{th}$ the machine m_{ig_i} in S_i.

B-string ha a length of l too. It is consisted of a sequence of job numbers in which job number j occurs o_j Times. o_j is the number of total operations that the j^{th} job comprises. When decoding a particle, B-string is converted to a sequence of operations at first. Then each operation is assigned to a processing machine according to A-string. At last each operation in the sequence will be scheduled in its earliest permitted time. In this way, each particle is corresponding to a feasible FJSP schedule.

Initial population

Each particle has its own position and velocity. The algorithm first starts by generating a population P, which is composed of N_p particles. Each individual $x_i$also has D-dimensional parameter vectors. The index t denotes the iteration number of the algorithm. The population P is used to represent the positions of particle. The initial position population can be generated randomly by using Eq. 5. m represents the number of identical parallel machines, and rand is a random value between 0 and 1.

$p(t) = \{x_1(t), x_2(t), \dots . x_i(t), \dots \dots, x_{N_p}(t)\}$ (3)

$x_i(t) = \{x_{1,i}(t), x_{2,i}(t), \dots \dots \dots, x_{j,i}(t), \dots \dots \dots, x_{D,i}(t)\}$ (4)

$x_{ij}(0) = rand() \times m + 1$

$i=1,2,\dots,N_p; \quad j=1,2,\dots,d; \quad t=1,2,\dots,T$ (5)

A population V, which is composed of N_p individuals, denotes the velocity of each particle. It also has D-dimensional parameter vectors. The initial velocity population can be generated using Eq. 8. The velocity of each particle is equal to zeros. It means that all particles have no velocity in the initial generation.

$V(t) = \{v_1(t), v_2(t), \dots . v_i(t), \dots \dots, v_{N_p}(t)\}$ (6)

$v_i(t) = \{v_{1,i}(t), v_{2,i}(t), \dots \dots \dots, v_{j,i}(t), \dots \dots \dots, v_{D,i}(t)\}$ (7)

$V(0) = zeros(N_p, D)$ (8)

Clonal selection operations

In the following, clonal selection operation involving cloning operation, mutation operation, and selection operation are described in detail.

Cloning operation: The cloning operation can expand the number of individuals that have higher affinities. In this paper, we adopt the total tardiness to calculate the affinities, i.e., the smaller total tardiness (higher affinity), the higher the number of copies and vice versa. A set of L antibodies are selected that have the best highest affinity with antigen from the K antibodies. The number of clones generated from each of a set of L-selected antibodies is rated to their affinity using a rank measure. The L-selected antibodies are sorted in ascending order by their affinity to the antigen. The number of clones for each antibody is based on the current position and can be described as follows:

$LN(i) = round(\beta \times L/i)$ (9)

Where i is the antibody current rank or position, i {1,2,..., L}, β the clonal factor, LN(i) the number of clones for antibody i, and L the size of selected antibodies to be cloned.

Mutation: Mutation is an important operation in CSA, which is carried out to maturate the freshly generated clones. It can maintain the diversity in the population and allow the algorithm to avoid local minima by preventing the individuals in a population from becoming too similar.

Table 1. Comparing lingo8 and CSA

Solution time (second)	objective value	Algorithm	Machine number	Job number	Number
0.6	0.349	*CSA*	3	3	1
1	0.358	*LINGO8*			
0.6	0.426	*CSA*	3	4	2
1	0/421	*LINGO8*			
0.6	0.465	*CSA*	4	4	3
10	0.476	*LINGO8*			
0.8	0/474	*CSA*	6	4	4
60	0.536	*LINGO8*			
0.8	0.522	*CSA*	4	7	5
1625	0.601	*LINGO8*			
0.8	0.662	*CSA*	6	5	6
8645	0.631	*LINGO8*			
1	0.741	*CSA*	14	18	7
>36000	………	*LINGO8*			
2.48	0.784	*CSA*	16	20	8
………	………	*LINGO8*			
3.35	0.831	CSA	20	25	9
………	………	LINGO8			
5.78	0.874	CSA	24	30	10
………	………	LINGO8			
7/62	0/912	CSA	30	40	11
………	………	LINGO8			
9/16	0/954	CSA	45	50	12

The clones (set of duplicate antigens) are then subjected to an affinity maturation process to better match the antigen in question. The degree of maturation is inversely proportional to their parent's affinity, meaning that the greater the affinity, the lower the mutation. In this research, we apply swap mutation operation to the problem, and the degree of maturation P_{hm} is equal to 1. This mutation operation just changes the two genes. Randomly generate two points and the values in two positions are exchanged.

Selection: In GA, we use the roulette wheel selection technique. However, this operator does not guarantee that each antibody that is selected to enter next the generation is better than the last generation. In order to overcome this shortcoming, we adopt another selection strategy here. For each antibody, we select the best antibody among the clones of each antibody to enter next generation. In this paper, we have fixed the parameter values for clonal selection as follows, and these parameter values are often used in CSA.

$L = 50\% \times N_p$; $\beta=0.9$; $P_{hm} = 1$

Experimental results

To illustrate the effectiveness and performance of algorithm proposed in this paper, it is implemented in MATLAB 7 on a laptop with Pentium IV Core 2 Duo 2.53 GHz CPU. The outputs of the CSA are compared with that achieved by Lingo 8 software. To study the function of Clonal selection Algorithm, some example questions are randomly created and then the results obtained from the calculation of presented mathematical model by Lingo 8 software are compared with the calculation of the question by CSA and are analyzed.

Table 1 shows the results after Lingo8 software calculation; CSA calculation for 9 to 2250 dimensions have been compared. It seem's that the mean of LINGO objective value equals the mean of CSA objective value. For exact statistical analysis, SPSS16 software was used and was calculated Independent Sample T Test, shown in Table 2. The significance value (0.034) indicates the equal variances not assumed. The significance value in second row *i.e.* 0.067 shows that the two means are equal.

Conclusions

In this paper we present a clonal selection algorithm to solve the Flexible Job-shop Scheduling Problem with regard to being NP-hard using MATLAB 7.0 software. The quality of the results with their time of calculation is compared with the results obtained from Lingo8. The experiments prove that the CSA can be used to solve FJSP effectively and the efficiency of CSA has been perfectly shown by the expansion of the said model for other state of production such as parallel machine series. Other Meta heuristic methods like Memetic algorithm, GA algorithm and Forbidden search algorithm could be used for the presented models as well.

Acknowledgements

The author gratefully acknowledges the support provided by the Islamic Azad University Ramsar Branch for their kind support and cooperation during researched plan that this paper resulted from it.

References

1. Barnes JW and Chambers JB (1996) Flexible job shop scheduling by tabu search. In: Graduate program in operations research and industrial engineering, The University of Texas at Austin, *Technical report series.* ORP. pp: 96-09.
2. Bean J (1994) Genetic algorithms and random keys for sequencing and optimization. *Operations Res. Soc. Am. J.Computing (ORSA).* 6, 154-160.
3. Chen H, Ihlow J and Lehmann C (1999) A genetic algorithm for flexible job-shop scheduling. In: Proc. of the *Intl. Conf. Robotics & Automation.* NY. IEEE. pp. 1120-1125.
4. Defersha FM and Chen M (2009) A coarse-grain parallel genetic algorithm for flexible job-shop scheduling with lot streaming. *Proc. Intl. Conf. Compu. Sci. & Engg.* 1, 201-208.
5. Gao J, Sun L and Gen M (2008) A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. *Compu. & Operations Res.* 35, 2892-2907.
6. Garey MR, Johnson DS and Sethi R (1976) The complexity of flowshop and jobshop scheduling. *Maths. Operations Res.* 1, 117-129.
7. Gonçalves JF, Mendes JJM and Resende MGC (2005) A hybrid genetic algorithm for the job shop scheduling problem. *Europ. J. Operational Res.* 167(1), 77-95.
8. Jain AS and Meeran S (1999) Deterministic job-shop scheduling: Past, present and future. *Europ. J. Operational Res.* 113, 390-434.
9. Kacem I, Hammadi S and Borne P (2002) Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. IEEE Transactions on Systems, Man and Cybernetics Part C. *Appl. & Rev.* 32(1), 1-13.
10. Kobayashi S, Ono I and Yamamura M (1995) An efficient genetic algorithm for job shop scheduling problems. In LJ. Eshelman (Ed.). *Proce. 6th Intl. conf. on genetic algorithms.* San Francisco, CA:*Morgan Kaufman Publ.* pp: 506-511.
11. Kumar V, Prakash MK Tiwari and Felix T S Chan (2008) Stochastic make-to-stock inventory deployment problem: an endosymbiotic psychoclonal algorithm based approach. *Intl. J. Production Res*. 44(11), 2245-2263.
12. Lourenço HR (1995) Local optimization and the job-shop scheduling problem. *Eur. J. Operational Res.* 83, 347-364.
13. Mastrolilli M and Gambardella LM (2000) Effective neighborhood functions for the flexible job shop problem. *J. Scheduling.* 3(1), 3-20.
14. Najid M, Stephane DP and Zaidat A (2002) A modified simulated annealing method for flexible job shop scheduling problem. *Proce. 2002 Intl. Conf. Sys. Man & Cybernetics*. NY. IEEE. pp: 89-94.
15. Nowicki E and Smutnicki C (1996) A fast taboo search algorithm for the job shop problem. *Managt. Sci.* 42(6), 797-813.
16. Pezzella F and Merelli E (2000) A tabu search method guided by shifting bottleneck for the job shop scheduling problem. *Eur. J. Operational Res.* 120(2), 297-310.
17. Pezzella F, Morganti G and Ciaschetti G (2008) A genetic algorithm for the flexible job-shop scheduling problem. *Compu. & Operations Res.* 35, 3202-3212.
18. Sun D, Batta R and Lin L (1995) Effective job shop scheduling through active chain manipulation. *Compu. & Operations Res.* 22(2), 159-172.
19. Wang L and Zheng DZ (2001) An effective hybrid optimization strategy for jobshop scheduling problems. *Compu. & Operations Res.* 28, 585-596.
20. Xia W and Wu Z (2005) An effective hybrid optimization approach for multiobjective flexible job-shop scheduling problems. *Compu. & Indust. Engg.* 48, 409-425.
21. Yazdani M, Amiri M and Zandieh M (2010) Flexible job-shop scheduling with parallel variable neighborhood search algorithm. *Expert Sys. with Appli. Intl. J.* 37(1), 678-687.