Buletin Teknik Elektro dan Informatika
(Bulletin of Electrical Engineering and Informatics)
Vol.2, No.1, March 2013, pp. 59~64
ISSN: 2089-3191

An Effective Multi-Population Based Hybrid Genetic Algorithm for Job Shop Scheduling Problem

Somayeh Kalantari[1], Mohamad SanieeAbadeh[2]
[1]Department of Electrical, IT and Computer, Islamic Azad University, Qazvin Branch, Qazvin, Iran
[2]Department of computer, Tarbiat Modares University, Tehran, Iran
e-mail: sk_kalantari@yahoo.com[1], sanie@modares.ac.ir[2]

Abstract

The job shop scheduling problem is a well known practical planning problem in the manufacturing sector. We have considered the JSSP with an objective of minimizing makespan. In this paper, a multi-population based hybrid genetic algorithm is developed for solving the JSSP. The population is divided in several groups at first and the hybrid algorithm is applied to the disjoint groups. Then the migration operator is used. The proposed approach, MP-HGA, have been compared with other algorithms for job-shop scheduling and evaluated with satisfactory results on a set of JSSPs derived from classical job-shop scheduling benchmarks. We have solved 15 benchmark problems and compared results obtained with a number of algorithms established in the literature. The experimental results show that MP-HGA could gain the best known makespan in 13 out of 15 problems.

Keywords*: Hybrid Genetic Algorithm, Job shop scheduling problem, multi-population*

1. Introduction

Scheduling is the process of allocating resources to activities over time [1]. It also is one of the most important issues in the planning and operation of manufacturing systems [2]. The job-shop scheduling problem consists of a set of jobs as job= $\{j_1, j_2 \ldots j_n\}$, and a set of machines as machine= $\{m_1, m_2 \ldots m_n\}$. In the general JSSP, each job comprises a set of tasks which must each be done on a different machine for different specified processing times, in a given job-dependent order. The standard job-shop scheduling problem makes the following constraints and assumptions [3]:

- The processing time for each operation using a particular machine is defined.
- There is a pre-defined sequence of operations that has to be maintained to complete each job.
- Delivery times of the products are undefined.
- There is no setup or tardiness cost.
- A machine can process only one job at a time.
- Each job is performed on each machine only once.
- No machine can deal with more than one type of task.
- The system cannot be interrupted until each operation of each job is finished.
- No machine can halt a job and start another job before finishing the previous one.
- Each and every machine has full efficiency.

Several measures of schedule quality exist, but shortest make span is the simplest and most widely used criterion. So In this paper we have considered the JSSP with an objective of minimizing makespan, the total elapsed time between the beginning of the first task and the completion of the last task. Due to the practical significance of JSP, it has drawn the attention of researchers for the last decades. Bruker and Schile [4] were the first to address this problem in 1990. They developed a polynomial graphical algorithm for a two jobs problem. Haung and Yin used an improved shifting bottleneck procedure for JSSP [5]. Chen and Luh used a new alternative method to Lagrangian relaxation approach [6]. A taboo search algorithm for JSSP was applied by Nowicki and Smutnicki [7]. Yang et al. used a clonal selection based Memetic algorithm for JSSP [8].

Received November 15, 2012; Revised January 19, 2013; Accepted February 3, 2013

The rest of this paper is organized as follows. Section 2 describes the Hybrid Genetic Algorithm. The proposed approach framework is shown in section 3. Section 4 reports experimental results. Concluding remarks are given in section 5.

2. Hybrid Genetic Algorithm

Hybrid Genetic Algorithms have been applied in a number of different areas and problem domains. Many researchers improve the performance of GA by incorporating different search and heuristic techniques [9].Researchers and practitioners have shown that a combination of global and local search is almost always beneficial [10].

2.1. Genetic Algorithm

Genetic algorithms are adaptive methods, which may be used to solve search and optimization problems [11]. They are based on the genetic process of biological organisms. Over many generations, natural populations evolve according to the principles of natural selection, i.e. survival of the fittest, first clearly stated by Charles Darwin in The Origin of Species. By mimicking this process, genetic algorithms are able to evolve solutions to real world problems, if they have been suitably encoded. Before a genetic algorithm can be run, a suitable encoding (or representation) for the problem must be devised. A fitness function is also required, which assigns a figure of merit to each encoded solution. During the run, parents must be selected for reproduction, and recombined to generate offspring (see Figure 1).

It is assumed that a potential solution to a problem may be represented as a set of parameters. These parameters (known as genes) are joined together to form a string of values (chromosome). In genetic terminology, the set of parameters represented by a particular chromosome is referred to as an individual. The fitness of an individual depends on its chromosome and is evaluated by the fitness function. The individuals, during the reproductive phase, are selected from the population and recombined, producing offspring, which comprise the next generation. Parents are randomly selected from the population using a scheme, which favours fitter individuals. Having selected two parents, their chromosomes are recombined, typically using mechanisms of crossover and mutation. Mutation is usually applied to some individuals, to guarantee population diversity [12].

Selection: The selection phase is in charge to choose the better individuals to prepare for the crossover. In this paper the roulette wheel selection is adopted.

Crossover: Crossover can be regarded as the backbone of genetic search. It intends to inherit nearly half of the information of two parent solutions to one or more offspring solutions. Provided that the parents keep different aspects of high quality solutions, crossover induces a good chance to find still better offspring [12].

Mutation: Mutation usually works on a single chromosome and creates another chromosome through alternation of the value of a string position or exchange of the values of two string positions in order to maintain the diversity of population. In this paper, an exchange order mutation operator is used. That is, two operations are chosen randomly and then their positions are exchanged [13].

Stopping criteria: Fixed number of generations is considered as stopping criteria. If the stop criterion is satisfied, the best chromosome is given as output.

```
Genetic Algorithm
{
Generate initial population Pt
Evaluate population Pt
While stopping criteria not satisfied Repeat
 {
    Select elements from Pt to copy into Pt+l
    Crossover elements of Pt and put into Pt+l
    Mutation elements of Pt and put into Pt+l
    Evaluate new population Pt+l
    Pt = Pt+l
 }
}
```

Figure 1. A standard genetic algorithm

2.2. Local Search Method

The local search combined to the genetic part keeps the balance of the exploitation and exploration. It has been observed that applying the local search for a limited number of iterations enables better results in the long run as reported in [14] for the maintenance scheduling problem. The local search for all chromosomes may cost much computation time. An approach that has been used in literatures is to apply local search to only good offspring to improve the search ability of their genetic local search approach [9, 15]. Figure 2 shows pair wise interchange method [16]. In MP-HGA, local search is performed through a pair wise interchange heuristic. Merz and Freisleben in [17] used that for the quadratic assignment problem. In this method, the local-search neighbourhood is defined as the set of all solutions that can be reached from the current solution by swapping two elements in the chromosome. For a chromosome of length n, the neighbourhood size for the local search is:

$$N = \frac{1}{2} \times n \times (n-1) \quad (1)$$

The number of swaps and consequently the size of the neighbourhood grow quadratically with the chromosome length (problem variables). In order to reduce processing time, Merz and Freisleben [17] suggested stopping the pair wise interchange after performing the first swap that enhances the objective function of the current chromosome.

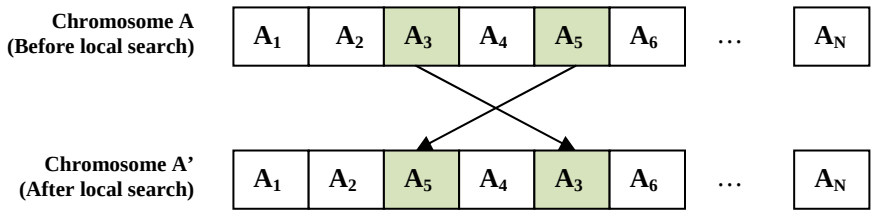

Figure 2. Applying local search using pair wise interchange

3. The MP-HGA Framework

The diagram of the proposed approach is shown in figure3. In this approach, using a random way the whole population is generated at first. A population consists of a set of solutions that will be partitioned into several groups. Here, the solutions indicate a schedule for jobs. Then the entire population is divided into G groups, each containing S solutions. Group formation is based on Shuffled Frog Leaping Algorithm. That is, the solutions are sorted in a descending order according to their fitness. Then the entire population is divided into m sub populations, each containing n solutions. Finally, a process similar to the SFL is applied. That is, the first solution goes to the first sub population, the second solution goes to the second sub population, population m goes to the m^{th} sub population, and solution m+1 goes back to the first sub population, etc. The Social topology is considered as population topology and the neighbor groups are connected by communication channels [18]. Each group is connected to all groups in the population. Groups are separated from one another (geographic isolation) and individuals compete only within a group [19]. After creation of groups, genetic and local search parts are done respectively. Then Re-Grouping operator is applied and based on a rule used in the Shuffled Frog Leaping Algorithm; individuals migrate to any other group. Details of genetic and local search part are explained in the previous section.

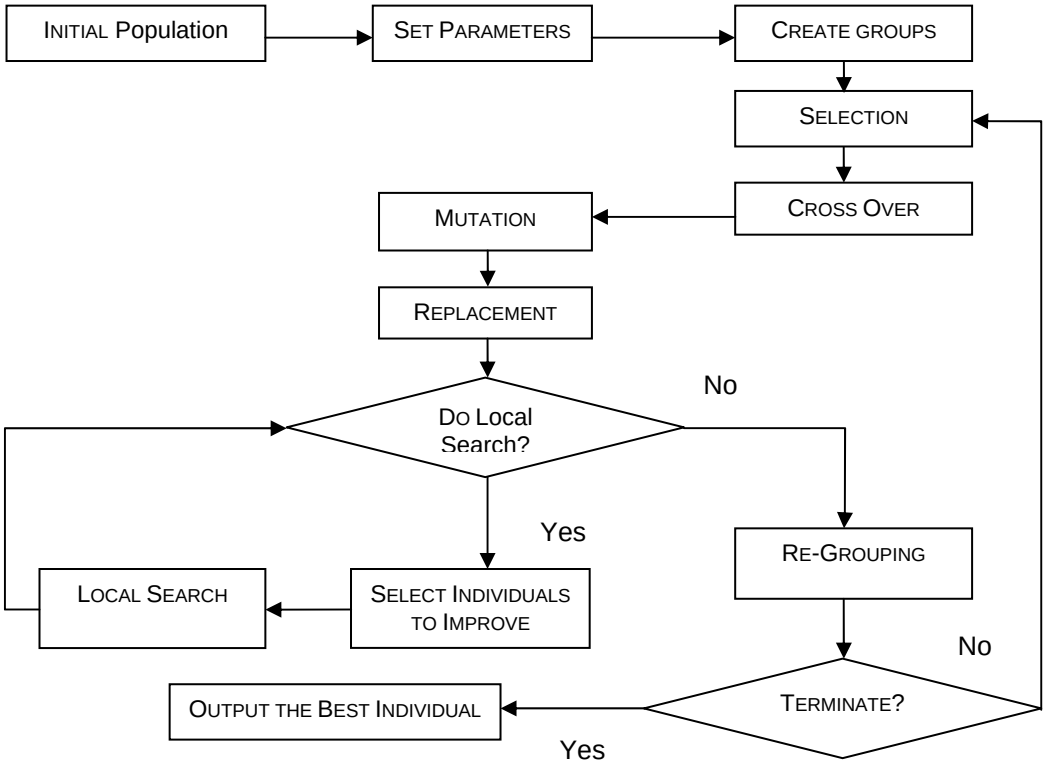

Figure 3. Diagram for the proposed approach

4. Computational Results

The proposed approach was implemented in Matlab and the tests were run on a computer with a 2.8 GHz Pentium five CPU, and 2GB RAM. The parameters used in this approach are chosen experimentally to get a satisfactory solution in an acceptable time span. Through experimentation, the parameter values were chosen as shown in Table1.

Table1. Parameter setting

Parameter Name	value
Number of generations	150
Number of groups	6
Size of each group	100
Size of the Population	600
Probability of Crossover	0.8
Probability of Mutation	0.01

In order to give a rough idea about the quality achieved, we confine to the 15 famous problems of Lawrence [20]. MP-HGA is tested on Lawrence's data set (LA01-LA15). Applied instances of this data set consist of 15 problems with 10, 15 or 20 jobs, 5 machines, and 5 operations. We ran the proposed approach five times on the same instance to obtain meaningful results. The results appear in Table2. It respectively lists problem name, problem size (number of jobs × number of operations), the best known solution (BKS), and the best solution obtained in this paper (MP-HGA). The solutions obtained by the following literatures: Park et al. [21], Gao et al. [22], and Yang et al. [23] along with the relative deviations, Dev (%), are shown in the next columns of the table. Dev (%) columns refer to the relative deviation of those algorithms with respect to MP-HGA. The relative deviation is defined in Eq. (2).

$$dev = \left[\frac{Mk_{comp} - Mk_{MP-HGA}}{Mk_{comp}}\right] \times 100 \quad (2)$$

Where, MK_{MP-HGA} is the makespan obtained by our approach, and MK_{comp} is the makespan of the algorithm we compared to. Results show that for 86.7% of the cases, the numbers typed bold in column MP-HGA of Table2, our approach could gain the Best Known Solution. The last row of the table shows the average improvement of the other literatures with respect to the MP-HGA. In Figure 4, the makespan of MP-HGA and other three approaches is drawn.

Table2. Experimental results

Problem	Size	BKS	MP-HGA	Gao	Dev (%)	Yang	Dev (%)	Park	Dev (%)
LA01	10×5	666	**666**	666	0	666	0	666	0
LA02	10×5	655	**655**	655	0	655	0	666	0
LA03	10×5	597	621	597	-4.02	597	-4.02	597	-4.02
LA04	10×5	590	602	590	-2.03	590	-2.03	590	-2.03
LA05	10×5	593	**593**	593	0	593	0	593	0
LA06	15×5	926	**926**	926	0	926	0	926	0
LA07	15×5	890	**890**	890	0	890	0	890	0
LA08	15×5	863	**863**	863	0	863	0	863	0
LA09	15×5	951	**951**	951	0	951	0	951	0
LA10	15×5	958	**958**	958	0	958	0	958	0
LA11	20×5	1222	**1222**	1222	0	1222	0	1222	0
LA12	20×5	1039	**1039**	1039	0	1039	0	1039	0
LA13	20×5	1150	**1150**	1150	0	1150	0	1150	0
LA14	20×5	1292	**1292**	1292	0	1292	0	1292	0
LA15	20×5	1207	**1207**	1207	0	1207	0	1207	0
Average improvement					-6.05		-6.05		-6.05

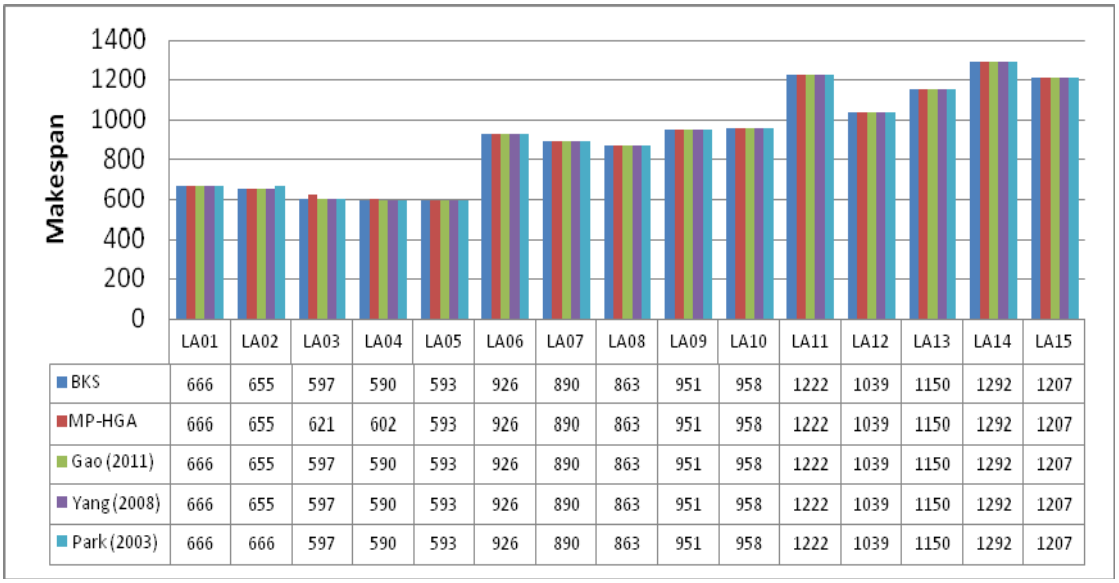

Figure 4. The makespan of the MP-HGA and other approaches

5. Conclusion

In this paper, a multi population based Hybrid Genetic Algorithm is proposed. It uses the Genetic algorithm and the pair wise interchange method for solving job-shop scheduling problem. The approach is compared with some other algorithms in the literature. It is tested on a set of 15 standard instances and compared with the three other approaches. The computational results show that the approach has produced good solutions on all instances tested. It could gain the same results on 13 problems out of the 15 problems.

References

[1] Aggoun, A. & Beldiceanu, N., (1993)" Extending CHIP in order to solve comple scheduling and placement problems", Mathematical and Computer Modeling, 17(7), pp. 57-73

[2] Zhang G. & Gao L. & Shi Y., (2008) "A genetic algorithm and tabu search for solving flexible job shop schedules", In proc. computational intelligence and design International symposium, pp369-372

[3] Giovanni L. D. & Pezzella F., (2010)"An improved genetic algorithm for the distributed and flexible job shop scheduling problem" European journal of operational research, 200, pp395-408

[4] Brucker P. & Schile R., (1990)," Job-shop scheduling with multi-purpose machines", Computing, 45(4), pp369-375

[5] Huang W. Q. & Yin A. H., (2004)"An improved shifting bottleneck procedure for the job shop scheduling problem", Computers & Operations Research, 31, pp2093–2110

[6] Chen H. X. & Luh P. B., (2003)" An alternative framework to Lagrangian relaxation approach for job shop scheduling", European Journal of Operational Research, 149, pp499–512

[7] Nowicki E. & Smutnicki C., (1996)" A fast taboo search algorithm for the job shop problem", Management Science, 42, pp797–813

[8] Yang J. & Sun L. & Lee H. & Qian Y. & Liang Y., (2008)" Clonal Selection Based Memetic Algorithm for Job Shop Scheduling Problems" Journal of Bionic Engineering, 5, pp111-119

[9] Gao L. & Zhang G. & Zhang L. & Li X., (2011)" an effective Memetic algorithm for solving the job-shop scheduling problem", journal of Computers and Industrial engineering, 60, pp699-705

[10] Krasnogor N. & smith J., (2000)"a memetic algorithm with self adaptive local search: TSP as a case study", In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000), Morgan Kaufmann, San Francisco, USA, pp987-994

[11] Beasley D. & Bull D. & Mratin R., (1993)"an overview of genetic algorithms: part 1- fundamentals", University computing, 15(2), pp58-69

[12] Bierwirth C., (1995)"A generalized permutation approach to job shop scheduling with genetic algorithms", OR Spectrum, pp1787-1792

[13] Xing Y. J. & Wang Z. Q. & Sun J. & Meng J., (2006)"A multi objective fuzzy genetic algorithm for job shop scheduling problems", journal of achievements in materials and manufacturing engineering, 17, pp297-300

[14] Bruke E. & Smith A., (1999)" a memetic algorithm to schedule planned maintenance for the national grid", ACM Journal of experimental algorithmics, 4(1), pp1084-1096

[15] Ishibuchi H. & Yoshida T. & Murata T., (2002)" selection of initial solutions for local search in multi objective genetic local search", proceeding of the 2002 congress on evolutionary computation (CEC 2002), pp950-955

[16] Elbeltagi E. & Hegazy T. & Grierson D., (2005)" Comparison among evolutionary-based optimization algorithms", journal of Advanced Engineering Informatics, 19, pp43-53

[17] Merz P. & Freisleben B., (1997)"A genetic local search approach to the quadratic assignment problem", In Proceedings of the 7th international conference on genetic algorithms, San Diego, CA: Morgan Kaufmann, pp465–72

[18] Akbari R. & Ziarati K., (2011)"a multi level evolutionary algorithm for optimizing numerical functions", International journal of Industrial Engineering Computation, 2, pp419-430

[19] Nowostawski M. & Poli R., (1999)"Parallel Genetic Algorithm Taxonomy"

[20] Lawrence S., (1984)"Supplement to resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques", Pittsburgh, PA: GSIA, Carnegie Mellon University

[21] Park B. J. & Choi H. R. & Kim H. S., (2003)" A hybrid genetic algorithm for the job shop scheduling problems", Journal of Computers & Industrial Engineering, 45(4), pp597–613

[22] Gao L., Zhang G., Zhang L., Li X., (2011)"an effective memetic algorithm for solving the job- shop scheduling problem", journal of Computers and Industrial engineering, 60, pp699-705

[23] Yang J., Sun L., Lee H., Qian Y., Liang Y., (2008)"Clonal Selection Based Memetic Algorithm for Job Shop Scheduling Problems", Journal of Bionic Engineering, 5, pp111-119