MINIMIZING MAKESPAN IN A FLOW SHOP SCHEDULING PROBLEM

Daniel BĂNCILĂ, Constantin BUZATU
Transilvania University of Brasov, Romania

Abstract. This paper studies the problem of flow shop scheduling in order to minimize the maximum time of completion of all jobs. Taking into consideration all the aspects involved in scheduling this specific case of problem is one of the most important because of its applicability on the engineering field, many solutions being proposed for its solving. By combining priority rules with genetic algorithms, this paper offers a hybrid solution for solving this problem.

Keywords: flow shop scheduling, genetic, algorithm, optimization, priority

1. Introduction

The flow shop scheduling problem occupies an important place in the planning of operations, having an important role in engineering. Problem description is as follows: starting from a set of m machines and n jobs that consists of a set of operations executed in the same order on a specific machine in a known time, it must be obtained the optimal planning scheme for which the completion time of all operations is minimized.

Due to increasing complexity in the case of large problems the flow shop scheduling problem is one of the most difficult combinatorial optimization problems, proving to be NP-hard.

One of the first methods used to solve the problem were the simple rules of priority and branch&bound techniques. Lately there is an increasing use of search methods in the solutions space (genetic algorithms, neural networks, tabu search, optimization methods with swarms of participles) capable of satisfactory results.

In [1] a hybrid flow shop scheduling problem with assembly operations is studied. A number of products of the same kind are produced by assembling a set of several parts. Prior to the assembling all the parts are produced in a flow shop scheduling scheme. The considered objective is to minimize the completion time of all products. In order to resolve the problem an hierarchical branch and bound technique is engaged.

In [2] the permutation flow shop scheduling problem is solved by using a discrete artificial bee colony algorithm, while in [3] a hybrid particle swarm optimization method is proposed for solving the problem of group scheduling in a flow shop scheme. Here, the problem is finding both the sequence for the jobs in each group and the sequence of the groups.

In [4] the flow shop scheduling problem was solved by the means of two methods: integer linear programming, being able to solve problems with up to 10 jobs and an hybrid artificial immune algorithm.

In [5] a no-wait flow shop problem where setup times depend on sequence of operations. The proposed problem considers sequence-independent removal times, release date with an additional assumption that there are some preliminary setup times. There are two objectives of weighted mean tardiness and makespan associated with the proposed model and a two-phase fuzzy programming is implemented.

In [6] in solving the flow-shop scheduling problem with break-down times a genetic algorithm was used and in [7] the flow shop scheduling with sequence-dependent setup times and periodic maintenance activity problem is also solved by using a genetic algorithm.

2. Hybrid genetic algorithm for flow shop scheduling

A genetic algorithm (Figure 1) is a search method in the space of solutions mimicking the evolution of organisms. It is characterized by the existence of a population, individuals being represented by chromosomes consisting of a sequence of genes. In coding the problem a gene represents an operation and a chromosome corresponds to a solution. The quality of each solution is evaluated by the value of the objective function, this value dictating whether that individual will stand in the next population. Schematization of the problem can by either binary or with real values, being one of the most important aspects regarding the algorithm accuracy. In solving this specific problem an integer string is engaged embedded in a matrix. Operations that run on genetic algorithms are chromosomes reproduction, crossover and mutation. The crossover operator plays the most important part in the formation of new generations being responsible for the exchange of sub-pairs between chromosomes.

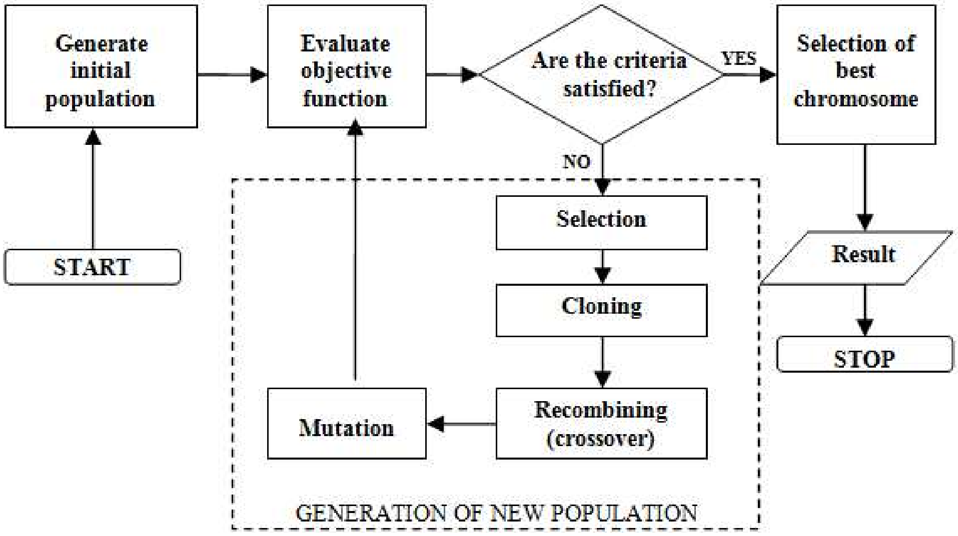

Figure 1. The fundamental structure of a genetic algorithm [8]

In the present problem, the reproduction operator of the genetic algorithm is an elitist one in which individuals is arranged in the ascending order of their value of the objective function. The next generation is obtained by using the roulette wheel reproduction operator (each individual is assigned a portion of the wheel proportional to its value of the objective function). Crossover operator is two-point ring-like whilst the mutation operator is position based with random choice of the mutation position, the rates being 50 and 10 %. In the case of genetic algorithms one of the most common problems is the convergence to a local optimum point thus the tuning of the algorithm (Figure 2) using a set of restrictions is necessary.

In the first phase generating initial population is no longer random, but by using simple rules of priority (SRPT, LRPT, MS, LPT, SPT) and only the remaining individuals are randomly selected to complete the population given as a parameter of the algorithm based on the problem's complexity.

In the following step, using the genetic operators (selection, crossover and mutation) the global search is initiated. To ensure a degree of diversity within generations the inverse matrix (Figure 3) is introduced by mirroring each chromosomes genes. Both obtained generations are then evaluated, assigning to each individual the value of its objective function. The next generation is formed by choosing from these two generations the most well-of individuals. As a stopping criterion a limited number of generations are established.

One of the advantages of using this hybrid algorithm is the inverse matrix, thus insuring a greater degree of avoiding premature convergence to a local optimum, also reducing the number of generations needed to find the optimum. Also, by generating the initial population under some restrictions, unfeasible solutions are eliminated from the start.

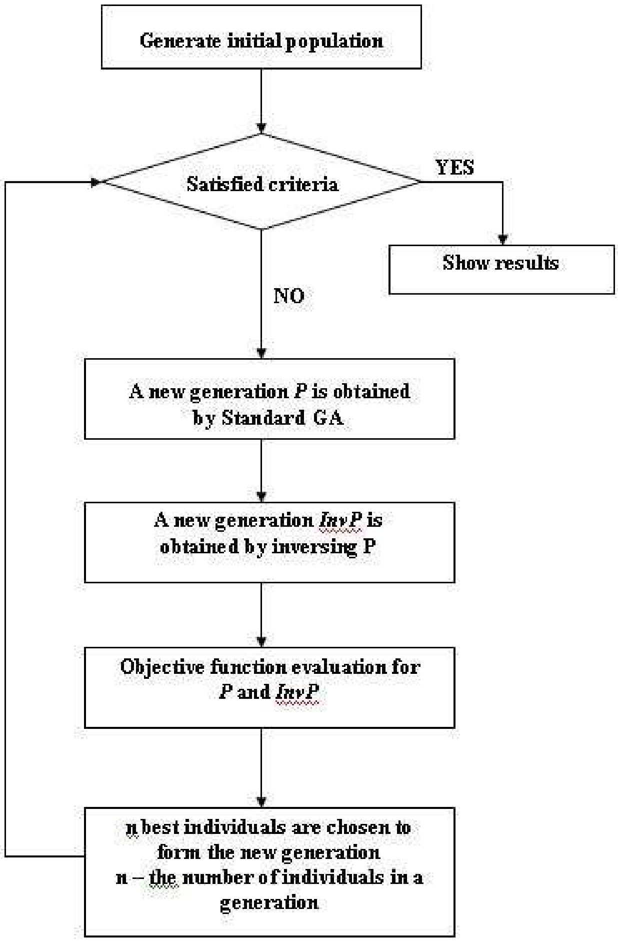

Figure 2. The tuning of the genetic algorithm procedure [9]

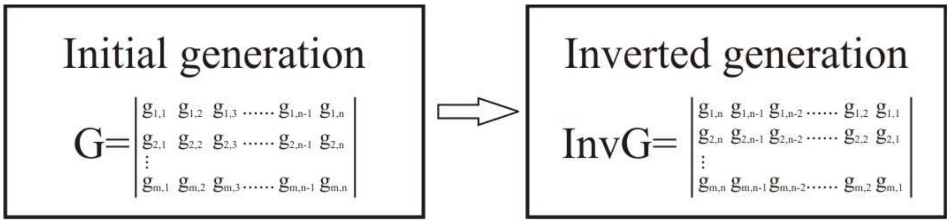

Figure 3. Inverting procedure [9]

3. Computational results and comparison

In this section the computational results of the hybrid algorithm developed for the flow shop scheduling problem are discussed. In order to balance performance and efficiency, the hybrid algorithm employed the generation limit as the stop criterion, terminating when it iterates the pre-set number of searching generations.

In order to test the efficiency of the hybrid algorithm the results obtained from its use were compared with those achieved from the employment of a standard genetic algorithm.

A problem consisting of six jobs which have to be flow shop scheduled on six machines is considered. The time cost matrix is given in Table 1.

Table 1. Time costs for all jobs on all machines

Jobs Machines	M1	M2	M3	M4	M5	M6
J1	2	1	2	2	1	1
J2	1	3	2	3	1	3
J3	3	2	1	4	1	1
J4	1	1	1	4	5	1
J5	1	1	1	1	2	1
J6	2	2	1	1	2	1

Both hybrid and standard algorithm's population consists of ten individuals, the crossover, mutation and reproduction operators are the same and the number of generations is set to 200. In the second case the population consists of 30 individuals and the number of generations is 1000. In both these cases as well as in others the results obtained by using the hybrid algorithm were better than those obtained by using the standard algorithm. In figure 4 and figure 5 it can be observed that the hybrid algorithm improves much faster without the need for additional time execution.

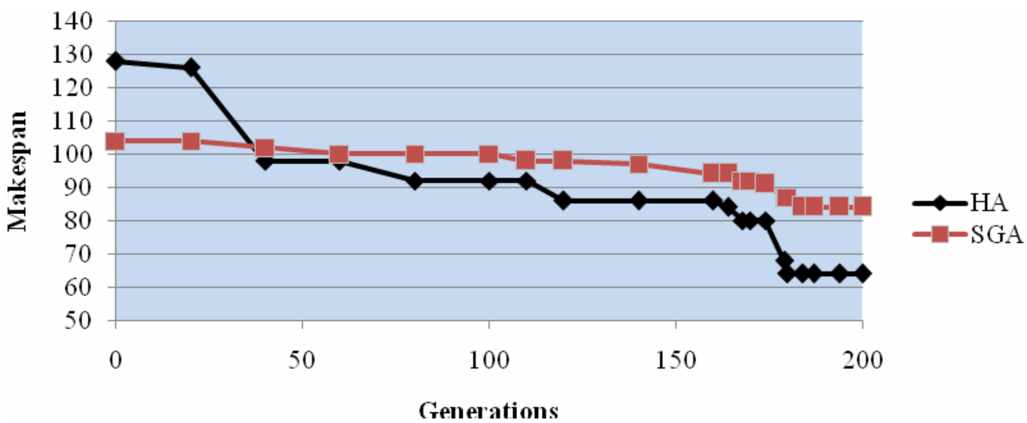

Figure 4. Comparison between HA and SGA – 6 jobs on 6 machines

4. Conclusions

The flow shop scheduling problem is presented in this paper, a hybrid algorithm based on the integration of a standard genetic algorithm and simple priority rules being proposed to solve it. The main purpose of the research is to find new effective methods for solving flow shop scheduling.

The algorithm presented in this paper, based on its construction, has two great advantages compared to a standard genetic algorithm: by generating a controlled initial population the unfeasible solutions are eliminated from the start and the usage of the inverse matrix assures the diversity of the new formed generations and the avoidance of premature convergence to a local optimum.

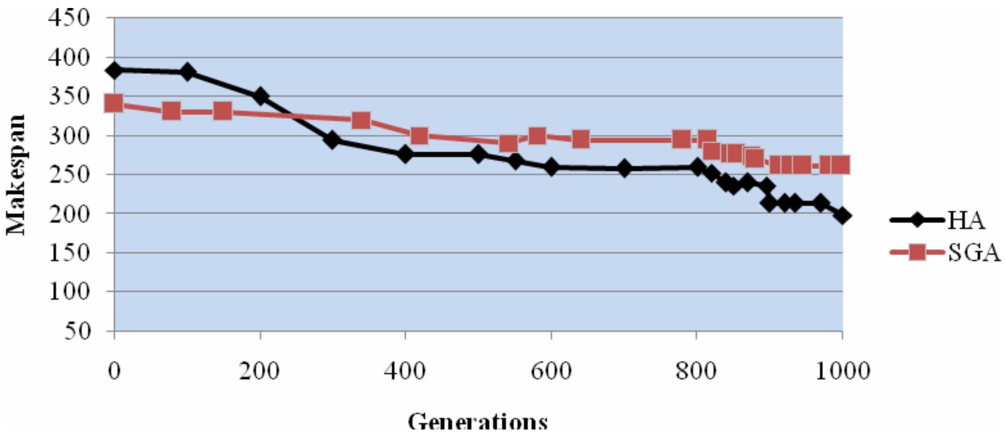

Figure 5. Comparison between HA and SGA – 10 jobs on 10 machines

Two cases are presented and the computational results show that the hybrid algorithm improves much faster and has better results than a standard genetic algorithm.

Acknowledgment

This work was partially supported by the strategic grant POSDRU/159/1.5/S/137070 (2014) of the Ministry of National Education, Romania, co-financed by the European Social Fund – Investing in People, within the Sectoral Operational Programme Human Resources Development 2007-2013.

References

1. Fattahi, P., Hosseini, S.M.H., Jolai, F., Tavakkoli-Moghaddam, R. (2014) *A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations*. Applied Mathematical Modelling, ISSN 0307-904X, Vol. 38, no.1, p. 119-134
2. Li, X., Yin, M. (2012) *A discrete artificial bee colony algorithm with composite mutation strategies for permutation flow shop scheduling problem*. Scientia Iranica, ISSN 1026-3098, Vol. 19, no. 6, p. 1921-1935
3. Hajinejad, D., Salmasi, N., Mokhtari, R. (2011) *A fast hybrid particle swarm optimization algorithm for flow shop sequence dependent group scheduling problem*. Scientia Iranica, ISSN 1026-3098, Vol. 18, no. 3, p. 759-764
4. Najafi, E., Naderi, B., Sadeghi, H., Yazdani, M. (2012) *A Mathematical Model and a Solution Method for Hybrid Flow Shop Scheduling*. Journal of Optimization in Industrial Engineering, ISSN 2251-9904, Vol. 5, no. 10, p. 65-72
5. Naderi-Beni, M., Tavakkoli-Moghaddam, R., Naderi, B., Ghobadian, E., Pourrousta, A. (2012) *A two-phase fuzzy programming model for a complex bi-objective no-wait flow shop scheduling*. International Journal of Industrial Engineering Computations, ISSN 1923-2926, Vol. 3, no. 4, p. 617-626
6. Rosas-Gonzalez, A., Clemente-Guerrero, D.M., Caballero-Morales, S.O., Flores-Juan, J.C. (2012) *An Evolutionary Approach for Solving the N-Jobs M-Machines Permutation Flow-Shop Scheduling Problem with Break-Down Times*. International Journal of Computer Applications, ISSN 0975-8887, Vol. 83, no. 1, p. 1-6
7. Gholizadeh, H., Tavakkoli-Moghaddam, R., Tootooni, B. (2012) *Minimizing the Makespan in a Flow Shop Scheduling Problem with Sequence-Dependent Setup Times and Periodic Maintenance by a Hybrid Algorithm*. Proceedings of the 3rd International Conference on Industrial Engineering and Operations Management, ISBN 978-0-9855497-0-1, p. 806-814, Istanbul, Turkey
8. Blaga, F.S. (2003) *Funcția de ordonanțare în fabricația flexibilă a familiilor de piese* (*The authorizing function in flexible manufacturing families of parts*). Ph.D. thesis. University of Oradea, Romania, p.50 (in Romanian)
9. Buzatu, C., Băncilă, D. (2008) *A Hybrid Algorithm for Job Shop Scheduling*. Proceedings of 6th International Conference of DAAAM Baltic Industrial Engineering, ISBN 9985894928, p. 309-314, Tallinn, Estonia

Received in October 2014