

Flow-shop Sequencing Model using Genetic Algorithm

Smita Verma¹ and Paheli Singhal²

¹*Department of Applied Mathematics & Computational Science
S.G.S. Institute of Technology & Sciences, 23-Park Road, Indore (M.P.)*

²*Department of Mathematics, Faculty of Applied Sciences,
TRUBA College of Engineering, Indore*

¹*Email: yvsmita@rediffmail.com*

²*Email: paheli_bansal@rediffmail.com*

Abstract

This paper considers the problem of scheduling in flow-shop by classical method and genetic algorithm method for finding an optimal sequence for n-jobs m-machine problem based on minimum elapsed time. The results of the tests showed that, for processing n-jobs through m-machine, genetic algorithm provide better result. Once the sequence is obtained, total elapsed time as well as the idle time of m-machines can be obtained.

Keywords: Flow-shop, scheduling, genetic algorithm.

1. Introduction

Sequencing problems have been most commonly encountered in production shops where different products are to be processed over various combinations of machines. The selection of appropriate order in which jobs are to be performed is called job sequencing. The objective is to determine an appropriate sequence or order for jobs to be done on a finite number of service facilities in some pre-assigned order, so as to minimize the total involved resources. There are total $(n!)^m$ possible ways by which n-jobs can be processed on m-machines. Here, the aim is to find out one sequence out of $(n!)^m$ that minimizes the total elapsed time. Practically it is not possible so we use some easier method to deal with such problems. Sequencing problems fall into two categories namely; Flow-shop sequencing problem and Job-shop sequencing problem. In this paper we only deal with Flow-shop sequencing problem.

2. Methodology

2.1 Flow-shop problem

The regular flow shop problem consists of two main elements; a group of m -machines and a set of n -jobs to be processed on this group of machine. Each of the n -jobs has the same ordering of machines for its process sequence. Each job in the sequence has to pass through each machine. In fact, the machine sequence is fixed and the technological order of all jobs is same, but the job sequence is to be determined. Each job is processed only once on each machine. Operations are not preempted and set-up times of operations are independent of the sequences and therefore can be included in the processing time.

Procedure

The following assumptions are made while solving a sequencing problem:

- (a) no machine can process more than one job at a time
- (b) each operation, once started, must be performed till completion
- (c) each operation must be completed before any other operation, which it must precede, can begin
- (d) all machines are of different types
- (e) a job is processed as soon as possible subject to ordering requirements
- (f) processing times are independent of order of performing the operations
- (g) the time involved in moving a job from one machine to other is negligibly small
- (h) all jobs are completely known and are ready for processing before the period under consideration begins

2.2 Genetic Algorithm

The Genetic Algorithm (GA) is an optimization and search technique based on the principles of genetic and natural selection. A GA allows a population composed of many individuals to evolve under specific selection rules to a state that maximizes the fitness (i.e., minimizes the cost function). This method was developed by John Holland (1975). Algorithm is started with a set of solutions (represented by chromosomes) called population. In n -jobs m -machine problem, the job sequencing is represented as a string. Here, two point crossover and mutation operators to produce an offspring are used. The crossover probability is calculated as the minimum elapsed time taken over maximum elapsed time of the optimal sequence. The purpose of using the crossover probability is to select the better individuals for crossover and mutation. In this paper, by using GA a sequence of jobs is formed from the given set of jobs. (Method is explained by the algorithm in Section 3.2). This is motivated by the hope that the new sequence so obtained will be either equivalent or better than the optimal sequence obtained by the S M Johnson's method (Section 3.1).

3. Algorithms

3.1 Algorithm for processing n -jobs through 2-machines (S M Johnson's method)

- *start*

- *read the number of jobs and processing times for each jobs on machine m_1 and machine m_2*
- *from the current job list find the job with the smallest processing time, say j*
- *if the smallest processing time falls into machine m_1 , put job j as close to the beginning of the sequence as possible otherwise to the end of the sequence*
- *delete job j from further consideration*
- *repeat the above procedure and check that all the jobs been assigned a position in the optimal sequence*
- *calculate the overall elapsed time*
- *calculate the idle time for machine m_1 and machine m_2*
- *print optimal sequence, idle time of machine m_1 and machine m_2*
- *end*

3.2 Genetic Algorithm for processing n-jobs through 2-machines:

- *start*
- *read the number of jobs and processing times for each jobs on machine m_1 and machine m_2*
- *add the processing times of machine m_1 and machine m_2*
- *select the job with maximum processing time and put it into the centre of the sequence*
- *delete the particular job from further consideration*
- *select the job with maximum processing time and put it into the left of the central job (which has already be assigned)*
- *delete the particular job from further consideration*
- *select the job with maximum processing time and put it into the right of the central job*
- *repeat the process till the last job is assigned in the sequence*
- *calculate the overall elapsed time*
- *calculate the idle time for machine m_1 and machine m_2*
- *print optimal sequence, idle time of machine m_1 and machine m_2*
- *end*

4. Discussion and Conclusion

For processing n-jobs through three-machines, S M Johnson extended the method by deriving two special conditions; (i) the smallest processing time for the first machine is at least as great as the largest processing time for the second machine; and (ii) the smallest processing time for the third machine is at least as great as the largest processing time for the second machine. And then by using these conditions, the problem of n-jobs three-machines is converted into n-jobs two-machines. But if neither of the two conditions holds good, the method fails.

The GA procedure mentioned above can also be applied to the n-jobs three-machine problem or in general to the n-jobs m-machine problems. Many examples have been solved by both the methods which support the assumption.

References

- [1] Gillett, Billey E.: Introduction to Operations Research – A Computer Oriented Algorithmic Approach, Tata McGraw Hill Publication.
- [2] Johnson, S. M.: Two and three stage production schedules with setup times included, Nav. Res. Logistic Q., Vol.1, No.1, 1954.
- [3] Chilte A.K. and J. Negi: Text Book of Operations Research, JB Publication.
- [4] S. Reza Hajazi and S. Saghafian, Flow-shop-scheduling Problems with Make Span Criterion: A Review, International Journal of Production Research, Vol. 43, No. 14, 15 July 2005, 2895–2929.