THE SOLUTION OF OPERATIONS SCHEDULING BY USAGE OF GENETIC ALGORITHMS

Miriam IRINGOVÁ, Pavol BEZÁK

Authors: ***Miriam Iringová, MSc. Eng., Pavol Bezák, MSc. Eng.***
Workplace: ***Institute of Applied Informatics, Automation and Mathematics, Faculty of Materials Science and Technology, Slovak University of Technology***
Address: ***Hajdóczyho Street 1, 917 24 Trnava***
Email: ***miriam.iringova@stuba.sk, pavol.bezak@stuba.sk***

Abstract

This contribution contains description of optimization problem solving using evolution computational techniques. It contains design of solution using genetic algorithms.

Key words

scheduling, job shop, production machine, genetic algorithm

Introduction

The creation of production schedules is important task in production on operative level. The scheduling problem is generally set by ending group of products that has to be produced and by limited number of production machines that are available. There is given job description that designates the kind and order of operations, their times and also assigns certain type of machine to every operation.

In case there is various order of machines for every product, we call it *job shop scheduling*. If there is the same order for every products, we call it *flow shop scheduling*. The task of production scheduling is to find the best schedule it means optimal order of operations executed by the machines. As the criteria of schedule it is possible to use the overall time of the flow time, time of machine downtime, losses due to the incompletion of tasks in required dates, the state of work-out etc.

Practically we can meet with various modifications of scheduling problem (including set up times, traffic and technological times, respect for production shops, mounting operations scheduling, production machines choice, limited buffer stocks capacity, uncertainty of input data etc.

The choice of production machines

The scheduling problem of job shop production in real conditions concerns also the machine choice problem [1], when there must not be firmly set specific machine for one of the operations but the list of machines, from which we choose one that will do the operation. The operation duration time as well as the operation execution price can be different on various machines. Besides the list of node evaluation that belong to operations and the list of evaluation of conjunctive edges corresponding to the technological order of operations, the problem formulation contains the list of alternative machines with duration times.

On Tab.1 there is an example of problem formulation with operation axes. For example operation op_4 can be executed on machine S_1 or S_2. In both cases duration time is equal to 3. For operation op_8 we can use machine S_2 with duration time equal to 4 or machine S_3 with duration time equal to 6.

EXAMPLE OF PROBLEM FORMULATION WITH OPTIONAL MACHINES

Table 1

task	operations								
	n.	machine	time	n.	machine	time	n.	machine	time
U_1	op_1	S_1	6	op_2	S_2	6	op_3	S_3	3
U_2	op_4	S_1	3	op_5	S_3	7			
		S_2	3						
U_3	op_6	S_2	7	op_7	S_1	4	op_8	S_2	4
								S_3	6

The setting of production operations on machines and the choice of machines from the list of production equipment is the task of heuristic method.

The schedule π is represented by two strings of parameters $\pi = \{X, Y\}$. The first string has the count of elements equal to the count of operations, for which the machine is determined from the group of machines. Every element in string X is expressed by index of machine utilized. In upper example this string can be for example $X = (1, 3)$, which means, that operation o_4 has to be executed on machine S_1 and operation op_8 on machine S_3. The second string is describing the order of operation execution on individual machines. If we choose representation based on operations, the example is the list $Y = (2, 3, 1, 2, 1, 3, 1, 3)$. By the same mean we can decode schedule {(4,1,7),(6,2),(5,3,8)} from the list and assign an evaluation to it according to the fitness function.

For the given example it is possible to design special genetic algorithms described in additional parts of this contribution. Think over an individual that is represented by two parts of solution X and Y. It is possible to assign the value of fitness function to every individual. The task is to find the individual which fitness function represents minimization of continuous time. In every part of solution X, Y it is possible to define operator of crossover and mutation. In string X it is possible to define mutations for example the change of one element of the string (it means change of the used machine to another in one operation). For example if machine of operation op_8 changes to S_2, than the chromosome is: {(1,2),(2,3,1,2,1,3,1,3)} and the schedule corresponding to it: $\{(4,1,7),(6,2,8),(5,3)\}$. In both cases it is possible to execute the crossover operation by one-point or two-point mean [2].

Genetic algorithm for solving problem with various parameters

Initial population is generated randomly. In each iteration a subgroup is randomly selected from the population and two best individuals become parents. Random selection decides whether the crossover in part X or part Y will be executed. Two children arise due to crossover. Every child goes to mutate with small probability. Randomization again decides whether mutation will be executed in part X or in part Y. At the end the newly arose children replace two worst individuals from randomly selected subgroup of population. Algorithm will be ended by execution of the given count of iterations or by expired time limit. During the algorithm we register the individual with the best value of fitness function.

It is possible to improve the algorithm by selecting each individual to crossover maximally 10 times. If one individual already has reproduced itself 10 times, it is overruled in elimination from the population regardless to the value of its fitness value.

Genetic algorithm with migration

Algorithm works with two populations and in one of them one part of solution X is being reproduced and in other one the second part of solution Y is being reproduced. Selected individuals migrate between these populations.

Phases of the algorithm are:

1. random generation of two populations at the beginning,
2. execution of N_1 iterations in the first population by applying crossover operator and mutation in part X of solution; than M_1 selected „strong" individuals migrate to the second population, where they replace M_1 „weak" individuals,
3. execution of N_2 iterations in the second population by applying crossover operator and mutation in part Y of solution; than M_2 selected „strong" individuals migrate to the second population, where they replace M_2 „weak" individuals,
4. in the end of algorithm, if the end condition is satisfied (equal as in previous algorithm), or it continues from the event 2 [3].

Genetic algorithm with migration with forcing part of the individual to the whole population

This method alternates the solution of two problems: optimization of the part X of solution when Y is constant and optimization of the part Y of solution when X is constant. This method works also with two populations but the individuals in the first population are represented only by part X and the individuals of the second population only by part Y.

Phases of the given algorithm are:

1. random generation of two populations at the beginning where value Y is equal for every individual in the first population,
2. execution of N_1 iterations in the first population by applying crossover operator and mutation in part X of solution; the value of Y remains constant for every individual of the first population; part X of the most successful individual from the first population will be forced to every individual of the second population,

3. execution of N_2 iterations in the first population by applying crossover operator and mutation in part Y of solution; part Y of the most successful individual from the first population will be forced to every individual of the second population,
4. in the end of algorithm, if the end condition is satisfied (equal as in previous algorithm), or it continues from the event 2.

Conclusion

This contribution deals with the problem of scheduling in the job shop production, especially the choice of production machines. Generally the problem of scheduling is given by the finite group of products that is needed to produce and by limited number of production machines that are available. Job description defines operations for every product in specified order and certain machine is assigned to every operation. Various methods of assigning production machines to production operations exist. There is one of stochastic heuristic methods described in this contribution and it is the use of genetic algorithms.

References:

[1] DAUZERE-PERES, S., ROUX, W., LASSERRE, J.B. Multi-Resource Shop Scheduling with Resource Flexibility. In *European Journal of Operational Research*, 1998, vol. 107, pp. 289-305.
[2] MAJER, P. *Moderní metody rozvrhování výroby*. Dizertačná práca. Brno: VUT, FSI, 2004, 87 s.
[3] DVOŘÁK, J., MAJER, P. Application of Genetic Algorithms to Scheduling and Lot Sizing in a Flow Shop. In *Proceedings of the 6th International Conference on Soft Computing MENDEL ‘2000*. Brno, 2000, 7 p.