MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
SUMY STATE UNIVERSITY
UKRAINIAN FEDERATION OF INFORMATICS

PROCEEDINGS

OF THE V INTERNATIONAL SCIENTIFIC CONFERENCE

ADVANCED INFORMATION SYSTEMS AND TECHNOLOGIES

AIST-2017

(Sumy, May 17–19, 2017)

SUMY
SUMY STATE UNIVERSITY
2017

The task of schedule optimizing for partially ordered jobs on machines with different productivity in the presence of idle time

Raymond Keller, Olesia Ilchuk
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", raybed@ukr.net

***Abstract* – The subject of this article is the job shop scheduling problem and methods for solving this problem. It contains model for the task of schedule optimizing for partially ordered jobs on machines with different productivity in the presence of idle time and algorithm for getting initial permissible solution.**

***Keywords* – job shop scheduling; optimizing; initial acceptable solution; algorithm; theory of schedule.**

I. INTRODUCTION

Scheduling theory deals with "ordering" and "scheduling". Ordering - it queuing operations performed by a machine, and scheduling - a task action sequences for several cars. In this scientific paper the "scheduling", namely the class of problems in which we have a service system that consists of machines. Each machine performs (some basic tasks). These operations are partially ordered, that perform one operation may depend on a previous execution did not depend. However, the duration of the operation is not a constant, it depends on the machine on which it runs. At the time axis may be "gaps" that forced the machine idle periods, when it cannot perform any operations. The set of machines, transactions and operations disciplines relevant purpose machines called maintenance. Scheduling this process means that for every transaction on the time axis corresponds to the subset in which the operation is performed in accordance machine. The main objective of this class of problems is to reduce maintenance time.

II. EXISTING APPROACHES TO SOLVING PROBLEMS SCHEDULING

A. Local Search

Local Search - search carried out by local search algorithms, a group of algorithms which search is conducted only on the basis of the current state and earlier passed states are not counted and not memorable.

The main purpose of the search is not an optimal path to the target point, and some optimization objective function, so the problem solved similar algorithms, called optimization problems. To describe the state space such problems using state space landscape, this presentation of the problem is to find the state of global maximum (or minimum) at this landscape.

B. Branch and bound method

Algorithmic method for finding the best solutions of various optimization problems, in particular, NP-complete problems, such as the problem of traveling salesman. Method is a combinatorial (sorting algorithm) with the rejection of subsets of feasible solutions that do not contain optimal solutions. The idea of the method can be described by the example of finding a minimum or maximum of a function f (x) on the set of admissible values of x. The method comprises two procedures: branching and evaluation (search boundaries).The first is to partition the set of feasible solutions to smaller subsets (technology "divide and conquer"). The procedure can be repeated recursively for all subsets. The result will be obtained tree is called a tree branch and bound tree or search nodes which are received subset. The assessment procedure is to find upper and lower bounds for the optimal value on a subset of feasible solutions.Screenings decision is made in the following way. To minimize problems, if the lower limit for the subset A search tree than the upper limit of any of the previously discussed subsets of B, then A can be eliminated from further consideration. The smallest of the received top ratings memorize in the variable m. Any node tree search, the lower limit is greater than the value m, can be excluded from further consideration. If the lower boundary coincides with the node tree top, this value is the minimum of the function at the appropriate subset.

C. Genetic algorithms

Genetic Algorithms - adaptive search, which recently often used for solving functional optimization. They are based on the genetic processes of biological organisms, biological populations develop over several generations, obeying the laws of natural selection and the principle of "survival of the fittest".

The task is coded so that its solution could be represented in the form similar to the array data structure of the chromosome. This array often called just that: "chromosome". Random array creates a certain amount of initial elements "Persons" or initial population. Individuals are valued using the tool accessories, which resulted in each person is assigned a certain value fitness, which determines the possibility of the survival of the individual. Then using obtained values adaptability selected persons admitted to crossing (selection). Persons used "genetic operators" (in most cases an operator intersection (Crossover) and mutation operator (mutation)), thus creating the next generation of people. Those next generations also evaluated using genetic operators and performed selection and mutation. Yes, simulated evolutionary process continues several life cycles (generations) until stopping criterion satisfied algorithm

Genetic algorithm consists of several stages:

1. creation of initial population (chromosome create for each individual)
2. the calculation of each individual fitness function (that it shows who fit best in this population)
3. Selection of the best representatives for further education of offspring
4. crossover
5. mutation
6. When (4) get the children, some of which passes through (5). The output obtain offspring
7. Selection of parents and children in the next generation
8. return to step (2) if the values that give our children is not satisfied

D. Simulated annealing

Simulated annealing - general algorithmic method of solving the problem of global optimization, especially in discrete and combinatorial optimization, in which the procedure for finding global solution simulates the physical process of annealing.

Exotic names associated with the algorithm of simulation methods in statistical physics, based on the technique of Monte Carlo. Investigation of the crystal lattice and behavior of atoms slow cooling of the body led to the appearance of the light probabilistic algorithms that have proven extremely effective in combinatorial optimization. It was first observed in 1983. Today this algorithm is popular among practitioners because of its simplicity, flexibility and efficiency, as well as among theorists, since the algorithm cannot analytically investigate its properties and prove asymptotic convergence.

Simulated annealing refers to a class of local search algorithms threshold. At each step of the algorithm for this decision in its neighborhood chosen a solution and if the difference in the objective function between the new and the current decision does not exceed the predetermined threshold, the new solution replaces the current one. Otherwise, select the new adjacent solutions. The general scheme of threshold algorithm can be represented as follows.

E. Taboo search

Taboo search is a meta-heuristic algorithm, local search is to prevent it from falling into the trap of premature local optimum by prohibiting the movement of those who are forced to return to previous decisions and cyclic operation. Taboo search begins with the initial decision. At each iteration margin generated solutions and best for this neighborhood gets a new solution. Certain attributes stored in previous decisions taboo-list, which is updated at the end of each iteration. Choosing the best solution going around so that it does not take any of the prohibited attributes. Most acceptable solution is now being updated when new and better solution to the current limit. The procedure continues until the fail any of the two criteria breakpoint, which is the maximum number of iterations performed and the maximum number of iterations during which the current solution is not improving.

III. THE FEASIBILITY OF THE RESEARCH

An interesting example is the guild task scheduling theory routing generalizing guild tasks commonly known metric traveling salesman problem. Performances such problems appeared independently in considering problems arising both in manufacturing and in the service industry. Most tasks scheduling theory is NP-difficult.

Sorting algorithms exponential type require considerable computational cost even when solving examples of average dimensions. Therefore, one of the important areas of research is accelerating algorithms for NP-hardness. Currently, this area has gained a huge number of supporters among researchers engaged in computer mathematics. Here are the main reasons for the popularity:

- there are many optimization problems that require solutions, and most NP-complex. For many of them not necessarily find the exact solution, but rather to build about, but in a short time, the more successfully cope with approximate algorithms;
- in practice, most optimization problems are very complex, because it includes many additional restrictions that do not allow to find a satisfactory approximate solution. But usually approximate algorithms for simpler versions of the same tasks suggest new ideas for heuristics, which then successfully used for practical tasks;
- in terms of exact solutions of NP-almost all challenges are equivalent to each other. Construction of approximate algorithms or proved the impossibility of constructing differences in the classes P and NP makes it possible to compare th NP-complex tasks by how they can be approximated.

Thus, we see that the current algorithms can be improved at the expense of speeding.

IV. FORMULATION OF THE PROBLEM

The problem purpose partially ordered work to handle their various capacities on machines that contain certain periods of downtime if downtime to minimize the time of execution of works. Suppose we have a set of works

$J = \{J_i\}, i = \overline{1,n}$ i and set of cars $M = \{M_j\}, j = \overline{1,m}$. Time for each job on each machine defined matrix $(x_{ij})_{n\times m}$. Also considered a given set period of inactivity $H = \{H_l\}, l = \overline{1,k}$, along with three vectors $(\beta_1,\ldots,\beta_k)$, $(r_1,\ldots,r_k)$, $(\tau_1,\ldots,\tau_k)$, which determine the length of downtime, the number of machines to which he belongs, and the time it starts, respectively. The problem can be represented by two vectors — $(s_1,\ldots,s_n)$, $(z_1,\ldots,z_n)$, which specifies the number of machines on which work is scheduled and the start of its implementation. Based on the above notation, a formal model of the problem can be represented as follows:

$$(x_{ij}) > 0, i = \overline{1,n}, j = \overline{1,m},$$

$$\beta_l > 0, l = \overline{1,k},$$

$$s_i \in \{1,\ldots,m\}, i = \overline{1,n},$$

$$r_l \in \{1,\ldots,m\}, l = \overline{1,k},$$

$$z_i > 0, i = \overline{1,n},$$

$$\tau_l > 0, l = \overline{1,k},$$

For all $i \in \{1,\ldots,n\}, \nu = \{1,\ldots,n\}$ such that $s_i = s_\nu, z_i < z_\nu$:

$$z_i + x_{is_i} \le z_\nu \qquad (1)$$

For all $i \in \{1,\ldots,n\}, \nu = \{1,\ldots,k\}$ such that $s_i = r_\nu, z_i < \tau_\nu$:

$$z_i + x_{is_i} \le \tau_\nu \qquad (2)$$

For all $i \in \{1,\ldots,n\}, \nu = \{1,\ldots,k\}$ such that $s_i = r_\nu, \tau_\nu < z_i$:

$$\tau_\nu + \beta_\nu \le z_i \qquad (3)$$

$$\max_{i=\overline{1,n}} (z_i + x_{is_i}) \to \min \qquad (4)$$

Limit (1) sets no crossing of the planned works on the same machine, constraints (2) and (3) - not crossing works and downtime. Limitation (4) - objective function that provides minimizing completion time of the most recent (v. BC. Guillotine cutting).

V. ALGORITHM TO OBTAIN AN INITIAL ACCEPTABLE SOLUTION

Step 1. Choose a car with the most power.
Step 2. Go to Step 4.
Step 3. CHOOSE car with less power.
Step 4. CHOOSE task with the greatest of less than or equal intervals to stop the machine.
Step 5. delete the selected task from the list available.
Step 6. Set the selected task in the processing machine.
Step 7. If the current machine is not the least power, then go to Step 3.
Step 8 select the first car
Step 9. Go to Step 11.
Step 10. Step the Next machine
Step 11. CHOOSE task with the greatest of less than or equal intervals to stop the machine.
Step 12. If the problem is not selected TO CHOOSE next interval after stopping the car OTHERWISE go to step 14.
Step 13. Go to Step 11.
Step 14. Set the time the machine flat end time of the end processing tasks.
Step 15. If the current machine is not the last, then go to Step 10.
Step 16. CHOOSE car with the lowest end of time.
Step 17. delete the selected task for this machine from a list.
Step 18. Set the selected task in handling this machine.
Step 19. If the list of available tasks does not blank, then go to step 8.

CONCLUSIONS

As the previous review, all known methods for scheduling the task to be performed with NP-complexity. Given the current trends in computing, it should focus on reducing the complexity of existing algorithms, which will accelerate their work and productivity software that will use them. Were considered well-known methods for solving the above problem and algorithm to obtain an initial acceptable solution.

REFERENCES:

[1] Hulyanytskyy L, Turin V. “Mathematical model of a class of independent tasks scheduling machines”, *Computer mathematics. Kyiv Institute of Cybernetics, vol. 2*, 2012, pp. 1-6.
[2] S. Kolliopoulos, G. Steiner. “Approximation algorithms for minimitation the total weighted tardiness on single machine”, 2006, pp.1-10
[3] A. Lazarev, E Rashidovich “Theory of schedules”, Moscow, 2011, pp. 222.