PAPER • OPEN ACCESS

Development of machine schedule at engineering enterprises

To cite this article: I V Kovalev *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **537** 052038

View the article online for updates and enhancements.

Development of machine schedule at engineering enterprises

I V Kovalev[1,2,3], M V Karaseva[1, 2], A A Voroshilova[1,3] and E V Tuev[1]

[1]Reshetnev Siberian State Aerospace University 31, Krasnoyarsky Rabochy Av., Krasnoyarsk, Russian Federation
[2]Siberian State University, 79, Svobodny Av., Krasnoyrsk, Russian Federation
[3]Krasnoyarsk Science and technology city hall, 61, Uritskogo ave., Krasnoyarsk, 660049, Russian Federation

E-mail: karaseva-margarita@rambler.ru

Abstract. The article considers the mathematical formulation of the Resource-Constrained Project Scheduling Problem (RCPSP). The RCPSP structure is represented by a graph. The article regards some special cases of RCPSP for which at present there exist either exact deterministic polynomial algorithms for their solving or approximate heuristic algorithms that find an approximate optimal solution in polynomial time.

1. Introduction
The basic concept for machine scheduling problems (MS) is an operation [1]. As a rule, two "agents" are necessary to specify an operation: an object of an operation (or its "material carrier", i.e. what an operation is performed on) and a subject of the operation, i.e. the one who (or what) performs the operation (a performer). A separate part or a more complex part of the product being created can be used as a material carrier of an operation. A carrier of the operation can be a person served by other people (devices). Subjects of operations are machines (tools, people) needed for these operations performance. In MS models, they are usually called *machines*.

Some subsets of operations are organized (technologically as well) into *jobs*. For example, if an enterprise produces complex products (cars, airplanes, etc.), then all operations related to one product constitute one job [2]. Some jobs may be a subject to a hard directive due date (which corresponds to the term *deadline*, or "death line") and / or the desired ending date (*a due date*), deviation from which is punished with a fine; at the same time, a deadline and a due date of the same work may not coincide. The moments $\{C_j\}$ of completion $\{J_j\}$ ("C" is from English "completion time") play a significant role in evaluation the quality of the developed schedule, as a rule, an objective function $f(C_1, \ldots, C_n)$ of the problem is a function of the completion points $J_1, \ldots, J_n$.

2. Model description
In the classical models of MS [3-5] there are only these three basic concepts (operations, work and machines). At the same time, strict rules are observed:

- each operation refers to only one job;
- each operation is performed by only one machine (possibly selected from a variety of alternative machines);

- no two operations of the same work can be performed simultaneously.

The precedence relations on the set of operations are given by a single directed graph whose vertices are the entire set of operations (it distinguishes this model from the classical models of the job shop or dag shop type, where the precedence is given on the set of operations by each individual work).

3. Mathematical statement of a problem

The Resource-Constrained Project Scheduling Problem (RCPSP) can be formulated as follows. The number of n jobs $i = 1, \ldots, n$ and renewable resources $k = 1, \ldots, r$ are given. A constant amount R_k of the k resource is available at any time. The job$i$ must be done in timep_i. During this time, a constant amount r_{ik} of $k$resource is involved in this job. Moreover, prioritization relationships are defined for the jobs. They are defined as $i \to j$, where $i \to j$ means that the job$j$ cannot release until the job$i$ is completed. The problem of determining the release moments S_i for the jobs $i = 1, \ldots, n$ is so that:

- the total need for a certain resource is less than or equal to the availability of this resource;
- job precedence relationships defined above are observed;
- value $Cmax_{i_{max}}^{\substack{i=1 \\ n}}$ takes its minimum value, where $C_i = S_i + p_i$ is the deadline of the job i.
- The interruption of jobs is possible.

Sometimes it is useful to add an initial pseudo-job 0 and a final pseudo-job $n + 1$, each one is with the zero duration. For all jobs $i = 1, \ldots, n$ conditions $0 \to i$ and $i \to n + 1$ must be observed. Pseudo-jobs do not require resources. S_0 is a start of the project, and S_{n+1} can be interpreted as a deadline.

If interruptions are not allowed, a vector$S = (S_i)$ defines the project scheduling (plan). S is an acceptable project scheduling if all the conditions are observed. Schedules can be graphically represented by Gantt charts.

The RCPSP structure is represented by means of a graph, where the vertices are jobs $G = (V, E)$. Here $V$is a set of all jobs and $E = \{(i, j) | i, j \in V; i \to j\}$ reflects the relationship of precedence. Two sets are assigned to each job i :

$Pred(i) = \{j | (j, i) \in E\}$ and $Succ(i) = \{j | (i, j) \in E\}$.

$Pred(i)$ $(Succ(i))$ is an amount of jobs in the graph just in front of (back from) the job i .

The precedence relationship $i \to j$ can be replaced by "release-release" relationship:

$$S_i + l_{ij} \leq S_j \tag{1}$$

where l_{ij} is any integer. The meaning of relation (1) depends on the sign l_{ij}.

If $l_{ij} > 0$, then the job j cannot start until the time l_{ij} has passed after the release of the job i, i.e. the job$j$ cannot start until the job i releases and l_{ij} is a minimum difference between the moments two jobs release. If $l_{ij} < 0$, then the job i should release no later than in $-l_{ij}$ after the job releases.

l_{ij} is called a positive (negative) time delay if (1) is executed and $l_{ij} > 0$ $(l_{ij} < 0)$.

The relations (1) are generalized. For example, if in (1) $l_{ij} = p_i$, then this case is nothing else but $i \to j$. If between the deadline of the job $i$and the release of the job$j$is no less d_{ij} time should pass, and then it can be written as$S_i + p_i + d_{ij} \leq S_j$. If the conditions $S_i + p_i + d_{ij} \leq S_j$ and $S_j - u_{ij} - p_i \leq S_i$ are met where$0 \leq d_{ij} \leq u_{ij}$, then the time between the end of the job i and the start of the job j should be no less than d_{ij}, but no more than u_{ij}. The latter includes a special $0 \leq d_{ij} = u_{ij}$ case where the job$j$ should release exactly in l_{ij} time units after the end of the job i.

For the job i, early release terms r_i (r - release) and late deadlines d_i (d - deadlines) can be modeled by the relations (1):$S_0 + r_i \leq S_i$, $S_i - (d_i - p_i) \leq S_0$. The time interval $[r_i, d_i]$ is called the job time window$i$. The job i should start and end between the boundaries of the given time window.

The renewable resource $k$with the capacityr_{jkm} is for job $j$with the duration p_{jm}.

Consider some special cases of RCPSP.

Scheduling for identical parallel machines.

We have m machines $M_1, \ldots, M_m$. The duration p_j of the job j does not depend on which machine will fulfill the job j. This problem is interpreted as RCPSP with $r = 1$, $R_1 = m$ and $r_{j1} = 1$ for $j = 1, \ldots n$.

The general formulation of the problem of the workshop scheduling.

There exist jobs $j = 1, \ldots, n$ and machines $M_1, \ldots, M_m$. The job j consists of n_j operations. Two operations of the same job cannot be done simultaneously. The operation has a duration p_{ij} if it is performed by the machine $\mu \in \{M_1, \ldots, M_m\}$. There exists the precedence relationship between any operations. The general problem of making a workshop schedule can be interpreted as RCPSP with $r = m + n$ renewable resources, where $R_k = 1$ for $k = 1, \ldots, m + n$, and with a number of operations O_{ij} equal to $\sum_{j=1}^{n} n_j$. Moreover, the operation O_{ij} uses a resource volume k equal to r_{ijk} where

$$r_{ijk} = \begin{cases} 0 & if \quad \mu_{ij} = M_k \quad or \quad k = m + j \\ & in \ other \ case \end{cases}$$

Resources $k = 1, \ldots, m$ correspond to the equipment, while resources $m + j$ $(j = 1, \ldots, n)$ are needed to simulate situations where different operations of the same job j cannot be performed simultaneously.

The "Job shop problem", "flow shop problem and "open shop problem" are important special cases of the generalized problem of workshop scheduling.

«Job shop problem».

In this problem, the sequence of operations is as follows:

$$O_{1j} \to O_{2j} \to \cdots \to O_{n_j, j}$$

for $j = 1, \ldots, n$, i.e. there is no temporary connection between the operations of different works, and the sequence of operations within each job is a strictly defined chain.

"Flow shop problem".

It is a special case of the "job shop problem" with $n_j = m$ for all $j = 1, \ldots, n$ and $\mu_{ij} = M_i$ and for all $i = 1, \ldots, m$, $j = 1, \ldots, n$, i.e. the operation O_{ij} must be done by machine M_i.

4. Conclusion

Currently there exist either exact deterministic polynomial algorithms for their solving, or approximate heuristic algorithms that find an approximate optimal solution in polynomial time for these particular cases of solving a general problem of the workshop scheduling. There are algorithms using the branch and bound method, "Tabu Search Method", heuristic methods based on different rules for choosing the sequence of operations (first, the shortest operation or the longest is selected, etc.). As it was mentioned above, the early release r_i *(r* - release) and late deadlines d_i (*d* - deadlines) can be added to the restrictions that already exist in these settings.

References

[1]   Thoben K-D, Eschenbächer J and Jagdev H S 2003 *Emerging concepts in E-business and extended products* (N.Y.: Springer-Verlag) pp 17–38

[2]   Fischer K, Muller J P, Heimig I and Scheer A-W 1996 Intelligent agents in virtual enterprises *Proc. of the 1st Intern. Conf. and Exhibition on the Practical Applications of Intelligent Agents and Multi-Agent Technology, London, UK* pp 205–23

[3]   Kolisch R and Drexl A 1997 Local search for nonpreemptive multi-mode resource-constrained project scheduling *IIE Transact* **29(11)** 987–99

[4]   Smith R G 1980 The contract net protocol: High-level communication and control in a distributed problem solver *IEEE Transact. on Computers* **29(12)** 1104–13

[5]   Kolisch R and Sprecher A 1997 PSPLib – A project scheduling problem library *European J. of Operational Research* **96(1)** 205–16