Approaches to Solving RCPSP Using Relaxed Problem with Consumable Resources

Ivan A. Rykov

Novosibirsk State Univercity, Russia
rykov@ngs.ru

Summary. In the work a resource-constrained project scheduling problem (RCPSP) is considered. This classical NP-hard problem evokes interest from both theoretical and applied points of view. Thus we can divide effective (i.e. polynomial) approximation algorithms in two groups: fast heuristic algorithms for the whole problem and generalizations and algorithms with performance guarantee for particular cases. In first section we consider the statement of the problem and some generalizations. Along with renewable resources we consider consumable resources which can be stored to be used at any moment after the moment of allocation. A polynomial optimal algorithm for solving the problem with consumable resources only was suggested by Gimadi, Sevastianov and Zalyubovsky [2]. So we can consider polynomially solved relaxation of RCPSP. In this relaxation instead of each renewable resource we have consumable resource which is allocated at each moment of time in one and the same amount. Then we can use information about the solution of this relaxation for approximate solving the original problem in polynomial time (for example, the order of starting times can be used as a heuristic for serial scheduling scheme). Furthermore, the optimal value of relaxation gives the lower bound for the optimal value of the original problem.

Speaking about performance guarantee algorithms we assume the case of void precedence graph, and one type of renewable resource. This is a generalization of binary packing problem which looks similar to a strip packing, but not equal to it. We compare this two problems and find bounds for maximal difference of optimal values for them (optimal value of this difference is unknown yet).

1 Problem Definition

1.1 Resource Constrained Project Scheduling Problem (RCPSP)

Classical NP-hard optimization problem, RCPSP can be stated as the following mathematical model:

$$\begin{cases} \max_j(s_j + p_j) \to \min_{s_j \in Z^+} \\ s_i + p_i + w_{ij} \leq s_j, \qquad \forall(i,j) \in E(G) \\ \sum_{j \in A(t)} r_{kj} \leq R_k, \qquad k = 1 \ldots K,\, t = 1,2 \ldots \end{cases}$$

where $A(t) = \{j | s_j < t \leq s_j + p_j\}$.

The goal is to define integer-valued starting times (s_j) for a set of N jobs of fixed integer duration p_j tied with constraints of two types — precedence constraints and resource constraints. Chosen schedule (i.e. the set of s_j) should minimize project finishing time (so-called makespan). Constraints of the first type are defined by the weighted oriented graph $G = (J,E)$ with jobs-nodes. The weight of each node is equal to the duration of the job and the weight of arc represents time lag between the end of one job and beginning of another. It is known that *the schedule that satisfies all precedence constraints exists if and only if G doesn't contain contours of positive weight* (weight of contour is the sum of weights of all nodes and arcs in it). We will consider only contourless graphs that surely satisfies this criteria. Constraints of second type are resource constraints. In the problem it is considered a set of (K) renewable resources. It means that each resource is allocated in each moment of time in one and the same amount and unused units cannot be stored (the examples of such resources are humans, machines, rooms and etc.) For each job there is set amounts of each resource needed at one moment of executing (r_{kj}). Constraints say that at each moment of time for each resource the amount of this resource needed by all executed jobs $(A(t))$ should not exceed allocating amount R_k.

Though it has such clear definition problem is known to be NP-hard. However, feasible schedule, if exists (see criteria above) can be obtained in polynomial time. Fast heuristic algorithms like well-known serial and parallel scheduling schemes use this property.

Generalizations

Besides its internal mathematical beauty, RCPSP is very interesting for the applied science. ERP systems designed for project management and resource planning are very popular in large-scale industry. Often additional constraints generalizing the problem are considered:

release dates: for some jobs there can be defined moments r_j of their appearance in project. Thus, additional constraints look like $s_j \geq r_j, \forall j$. They simply can be modeled in terms of base problem with the help of fictive job of zero duration precede other jobs with lags equal to the release dates

deadlines (due dates): similarly, for some jobs there can be defined moments d_j of their obligatory ending. Corresponding constraints are the following: $s_j + p_j \leq d_j, \forall j$. Unlike release date, deadlines severely complicate the problem: the search of feasible schedule in such problem is as hard as search of optimal solution in base problem, i.e. NP-hard.