dr hab. inż. Joanna Józefowska
Instytut Informatyki, Politechnika Poznańska

DISCRETE-CONTINUOUS PROJECT SCHEDULING STATE OF THE ART

Abstract. In this paper a survey of models and algorithms for the discrete-continuous project scheduling problems is presented. The problem is formulated and a general solution methodology is described Heuristic approaches are characterised. The problem of correcting the schedule in case of unexpected disturbances during its execution is discussed.

1. INTRODUCTION

A project is a unique undertaking to be accomplished that consists of individual activities each of which requires time and scarce resources for its completion. Moreover there are precedence constraints between activities describing the order in which activities must be performed. Project scheduling is defined as finding start times for all activities such that the precedence and resource constraints are satisfied and a given objective is optimised.

Two types of resources are considered: renewable and non-renewable ones. A resource is renewable, if only its temporal usage, i.e. temporary availability at every moment, is limited. The same resource can be used later when released by activity using it at the moment. A resource is non-renewable, if only its total consumption is limited. If both, the temporal usage and total consumption are limited, the resource is called a doubly-constrained one.

Project scheduling problems occur in many practical situations like construction work, the development and introduction of new products, service systems, strategic long-term planning, software development and many others.

First approaches to solving project scheduling problems were developed in 1950's. It was assumed that resources are not limited and processing times are known a priori. The solution method proposed is, the well known, Critical Path Method (CPM). Practical experience with the Polaris missile project showed, however, that often it is very difficult to determine the duration of activities a priori with satisfactory accuracy. Thus another approach called PERT (Program Evaluation and Review Technique) was developed taking into account the uncertainty of activity times.

Further it was observed that in some situations additional supply of resources may reduce the activity duration. Obviously, added resources also cause increased cost of the project and in consequence a time-cost trade-off problem was considered.

In the Resource Constrained Project Scheduling Problem (RCPSP), the limited availability of resources required by activities is taken into account and the optimisation criterion is the project duration. Unfortunately, it was proved that the RCPSP is strongly NP-hard, so there is unlikely that effective solution procedures can be developed.

Another generalisation of the project scheduling problem is to assume that the activity duration depends on the amount of the resource assigned to this job. So-called processing modes of activities are defined in the problem formulation and the objective is to find the shortest feasible schedule. This problem is called the Multi-Mode Resource-Constrained Project Scheduling Problem. The MRCPSP has been broadly studied in recent years. Several exact and heuristic approaches have been proposed. Talbot [17] was the first one who proposed an exact enumeration scheme, followed by Patterson et al. [14]. These early methods were able to solve instances up to 15 activities. Sprecher and Drexl [15] proposed new dominance criteria making their branch and bound algorithm be able to solve problems up to 20 activities. According to the results presented by Hartmann and Drexl [5] this algorithm is recently the most effective one for exact solving the MRCPSP. Sprecher and Drexl [15] also showed that even the currently most powerful optimisation procedures are unable to solve optimally highly resource-constrained problems with more than 20 activities and more than two modes per activity in reasonable computational time. In consequence, heuristic algorithms to find near-optimal solutions must be applied for larger projects.

Several heuristic approaches for solving the MRCPSP have been already proposed in the literature. Talbot [17] and Sprecher and Drexl [15] suggest to use their branch and bound algorithms as heuristic procedures by imposing a time limit. Hartmann [4] proposed a genetic algorithm with encoding based on a precedence feasible list of activities and a mode assignment. Bouleimen and Lecocq [1] describe a new simulated annealing algorithm. Maniezzo and Mingozzi [13] propose a new mathematical formulation for the MRCPSP and use it to derive two new lower bounds and a new heuristic algorithm based on Benders' decomposition.

Another criterion considered in the field of project scheduling problems is maximisation of the net present value of the cash flow in the project. This criterion was introduced by Russel [] in 1970 and examined by many other researchers. A recent survey of results for this model can be found in [10].

A comprehensive survey of project scheduling models and solution methods was recently developed by Demeulemeester and Herroelen [3].

All the problems mentioned up to this point assumed that the resources can be assigned to activities only in amounts from a given set (discrete numbers). In [7] another model was proposed where at least one resource can be allotted to activities in arbitrary amounts from a given interval (real numbers). Moreover, the activity duration is a decreasing function of the amount of the continuous resource allotted to the job at a time. This model of activity was proposed by Burkov [2] and further examined by Węglarz [18] for the preemptive schedules. Review of more recent results is the subject of this paper.

In Section 2 the formulation of discrete-continuous scheduling problems as well as general solution approach are presented. In Section 3 some heuristics for the minimisation of the project duration are described. Section 4 discusses the problem of updating the schedule in case of disturbances observed during its execution. Summary and directions for further research complete the paper.

2. PROBLEM FORMULATION

The problem of discrete-continuous project scheduling is defined as follows [7]. Let us consider a project consisting of n precedence- and resource-constrained, nonpreemptable activities that require renewable resources of two types: discrete and continuous ones. Assume m discrete resources are available and vector $\mathbf{r}_i = [r_{i1}, r_{i2}, \ldots, r_{im}]$, $i = 1,\ldots, n$, determines the (fixed) discrete resource requirements of activity i. The total number of units of the discrete resource j, $j = 1,\ldots, m$, is limited by R_j. The single continuous, renewable resource can be allotted to activities in (arbitrary) amounts from the interval [0, 1]. The amount (unknown in advance) of the continuous resource $u_i(t)$, allotted to activity i at time t determines the processing rate of activity i as it is described by the following equation:

$$\dot{x}_i(t) = \frac{dx_i(t)}{dt} = f_i[u_i(t)],\ x_i(0) = 0,\ x_i(C_i) = \tilde{x}_i \tag{1}$$

where $x_i(t)$ is the state of activity i at time t,

f_i is a continuous, nondecreasing function, where $f_i(0) = 0$,

$u_i(t)$ is the amount of the continuous resource allotted to activity i at time t,

C_i is the completion time (unknown in advance) of activity i,

$\tilde{x}_i$ is the processing demand (final state) of activity i.

State $x_i(t)$ of activity i at time t is an objective measure of work related to the processing of activity i up to time t. It may denote, for example, the number of man-hours already spent on processing of activity i, the number of standard instructions in processing of computer program i, and so on. All activities are available at the start of the process.

The problem is to find a feasible assignment of discrete resources and, simultaneously, a continuous resource allocation which minimise the schedule length, i.e. the project duration, $M = \max\{C_i, i = 1, 2,\ldots, n\}$. The continuous resource allocation is defined by a piece-wise continuous, non-negative vector function $\mathbf{u}^*(t) = [u_1^*(t), u_2^*(t),\ldots,u_n^*(t)]$ whose values $\mathbf{u}^* = [u_1^*, u_2^*,\ldots,u_n^*]$ are (continuous) resource allocations corresponding to M^* — the minimal value of M.

The general methodology developed for the discrete-continuous project scheduling problems uses the idea of a feasible sequence defined first in [12]. Let us divide a feasible schedule (i.e. a solution of a discrete-continuous project scheduling problem) into $p \le n$ intervals of lengths M_k, $k = 1,\ldots, p$, defined by the completion times of the consecutive activities.

Let Z_k denote the combination of activities corresponding to the k-th interval. Thus, a *feasible sequence* S of combinations Z_k, $k = 1,2,\ldots, p$, is associated with each feasible schedule. The feasibility of such a sequence requires that:

1. the number of units of the discrete resource j, $j = 1,\ldots, m$, assigned to all activities in combination Z_k, $k = 1,\ldots, p$, does not exceed R_j, i.e. $\sum_{i\in Z_k} r_{ij} \le R_j$, $j = 1,\ldots, m$, $k = 1,\ldots, p$,
2. each activity appears in at least one combination,
3. precedence constraints between activities are satisfied,
4. nonpreemptability of each activity is guaranteed.

The last condition requires that each activity appeared in exactly one or in consecutive combinations in S.

It has been proved in [7] that for convex processing rate functions f_i, $i = 1, 2,\ldots, n$, a makespan optimal schedule is obtained by allotting the total amount of the continuous resource to one activity at a time only. Concluding, in an optimal schedule activities are ordered in a sequence fulfilling the precedence constraints and the total amount of the continuous resource is allotted to each activity. Of course, if we assume $r_{ij} \le R_j$, $i = 1, 2,\ldots, n$, $j = 1, 2,\ldots, m$, the discrete resource constraints are fulfilled, because only one activity is performed at a time. For concave processing rate functions in a makespan optimal schedule as many activities as possible are performed in parallel (see [7]). In this case for a given feasible sequence S one can find an optimal division of processing demands of activities, $\tilde{x}_i$, $i = 1,2,\ldots, n$, among combinations in S, i.e. a division which leads to a minimum length schedule from among all feasible schedules generated by S. To this end a nonlinear programming problem can be formulated in which the sum of the minimum-length intervals (i.e. parts of a feasible schedule) generated by consecutive combinations in S, as functions of the $\{\tilde{x}_{ik}\}_{k\in Z_k}$, where $\tilde{x}_{ik}$ is a part of activity i processed in combination Z_k, is minimized subject to the constraints that each activity has to be completed. For concave processing rates f_i, $i = 1,\ldots, n$, it is sufficient to consider feasible schedules in which the resource allocations among activities remain constant in each interval k, $k = 1, 2,\ldots, p$. In the sequel we will assume that f_i, $i = 1, 2,\ldots, n$, are concave.

Let $M_k^*(\tilde{\mathbf{x}}_k)$ be the minimum length of the part of the schedule generated by $Z_k \in S$, as a function of $\tilde{\mathbf{x}}_k = \{\tilde{x}_{ik}\}_{i\in Z_k}$. Let K_i be the set of all indices of Z_k's such that $i \in Z_k$. The following mathematical programming problem is obtained to find an optimal demand division of activities for a given feasible sequence S.

Problem P

Minimize $$M\left(\{\tilde{\mathbf{x}}_k\}_{k=1}^{p}\right) = \sum_{k=1}^{p} M_k^*(\tilde{\mathbf{x}}_k) \qquad (2)$$

subject to $$\sum_{k\in K_i} \tilde{x}_{ik} = \tilde{x}_i, \qquad i = 1,2,\ldots, n \qquad (3)$$

$$\tilde{x}_{ik} \ge 0, \qquad i = 1,2,\ldots, n;\ k \in K_i \qquad (4)$$

where $M_k^*(\widetilde{\mathbf{x}}_k)$ is calculated as a unique positive root of the equation:

$$\sum_{i \in Z_k} f_i^{-1}(\widetilde{x}_{ik} / M_k) = 1 \tag{5}$$

which can be solved analytically for some important cases (see [18]).
Of course, constraints (3) correspond to the condition of fulfilling processing demands of all activities. It was proved by Węglarz in [18] that the objective function (2) is always a convex function. In consequence, our problem is to minimize a convex function subject to linear constraints.
After finding an optimal division $\widetilde{x}_{ik}$, $i = 1, 2, \ldots, n$, $k \in K_i$ of $\widetilde{x}_i$'s the corresponding continuous resource allocation for combination Z_k is given as

$$u_{ik}^* = f_i^{-1}\left(\widetilde{x}_{ik}^* / M_k^*\right), \; i \in Z_k \tag{6}$$

Thus, an optimal continuous resource allocation for a given feasible sequence can be calculated by solving Problem P. In consequence, a globally optimal schedule can be found by solving the continuous resource allocation problem optimally for all feasible sequences and choosing a schedule with the minimum length. In this sense the process of finding an optimal schedule can be viewed as a search process over the space of all feasible sequences for a given problem instance. Unfortunately, in general, the number of all feasible sequences grows exponentially with the number of activities.

3. HEURISTIC ALGORITHMS

3.1. Metaheuristics

We have stated above that a way to find an optimal schedule is to calculate optimal resource allocations for all the feasible sequences and choose the best one of all the schedules generated. Therefore it is possible to apply local search metaheuristics such as Simulated Annealing (SA), Tabu Search (TS) or Genetic Algorithms (GA) operating on a space of all feasible sequences [7].
However, since the number of combinations in a feasible sequence p is less than or equal to n, it is easy to observe that in order to find an optimal schedule, it is sufficient to look through n-element feasible sequences only.
Thus we define a *feasible solution* as an n-element feasible sequence. The set of all such feasible solutions constitutes the search space for all the three heuristics.
We may assume without loss of generality that activities are numbered in such a way that if $i < j$, then i precedes j. This can be always accomplished using the level numbering of nodes in a directed acyclic graph. For each feasible solution the objective function is calculated by solving Problem P for the corresponding feasible sequence. The stop criterion has been defined as a number of feasible solutions visited.
The results presented in [7] show that SA has found the best solutions of all the three heuristics for all instances generated. The average relative deviation of the solutions found by both the remaining heuristics from the solution found by SA is less than 2%,

while the maximum deviation is over 10% for TS. GA has produced slightly better results than TS, especially the number of instances for which the best solution has been found is larger for GA.

Finally, let us stress that the high computational effort required follows from the fact that for every feasible solution visited by a heuristic an optimal resource allocation has been calculated by solving Problem P. Solving the nonlinear mathematical programming problem was very time consuming. Thus, another approach to solving the discrete-continuous scheduling problems was proposed in [8]

3.2. Dicretisation of the continuous resource

Observe that the mathematical programming problems for finding optimal resource allocation, although theoretically valuable, are in general computationally intractable, as we have already mentioned above. Thus in this section we describe an alternative approach, the continuous resource allotments are dicretised. Denote by $r_i^{l(i)} = u_i^{l(i)} \in [0,1]$, the discretised continuous resource allotment for activity i, $l(i) = 1, 2, \ldots, L$, where L is the maximal number of allotments. Then from (1) we obtain the processing time of activity i for this allotment:

$$\tau_i^{l(i)} = \frac{\bar{x}_i}{f_i\left(r_i^{l(i)}\right)} \tag{7}$$

Treating the discretised continuous resource as an additional discrete resource, i.e. resource $m+1$, we obtain the MRCPSP in which resource requirements of activity i, $i = 1, 2, \ldots, n$, processed in mode $l(i)$, $l(i) = 1, 2, \ldots, L$, are determined by vector $\mathbf{r}_i^{l(i)} = \left[r_{i1}, r_{i2}, \ldots, r_{im}, r_{im+1}^{l(i)}\right], \sum_i r_{im+1}^{l(i)} \leq 1$, and processing times are defined by (7). Of course, resource $m+1$ is renewable and its total available amount is equal to 1.

The discretisation of the continuous resource converts our problem to the Multi-Mode Resource-Constrained Project Scheduling Problem. The latter one is solved using a customised simulated annealing algorithm. Two versions of the simulated annealing algorithm were proposed in [8] and computationally compared with the SA algorithm developed for the original discrete-continuous model. The results showed that the considered approach allows to reduce computational time up to 70 times while loosing not more than 5% of solution quality.

4. UNCERTAINTY IN DISCRETE-CONTINUOUS SCHEDULING

The methodology presented above assumes complete information about the scheduling problem and a static deterministic environment within which the schedule will be executed. This is a general characteristic of the deterministic approaches to scheduling. A schedule constructed basing on such assumptions is called a *predictive* one [11].

However, in practical applications the problem parameters are subject to uncertainty. Quite often it may be difficult to predict the processing demand and processing rate function of the activities. In such situations estimation from analysis of historical data is used to build the predictive schedule. Herroelen and Leus [6] present an extensive survey of approaches to project scheduling under uncertainty. They examine the following methods: reactive scheduling, stochastic project scheduling, stochastic GERT network scheduling, fuzzy project scheduling, robust (proactive) scheduling and sensitivity analysis.

In this section we would like to exploit some properties of discrete-continuous scheduling in order to create good reactive schedules without changing the discrete part of the predictive schedule and thus saving a lot of computational time.

By a disturbance in a schedule we will understand a situation where a deviation of the completion time of an activity is observed (delayed or early completion). There may be several reasons of disturbances. However, the availability of resources is the most common source if disturbances. It is easy to observe that variation in the available number of units of a discrete resource may not influence the schedule, while any decrease of the available amount of the continuous resource must result in increased processing time of at least one activity. Similarly, resource requirements of an activity may not influence the schedule in case of discrete resources. In case of continuous resources this demand is represented by the processing rate function. A change of the processing rate function usually results in different than expected completion time. On the other hand, however, if a critical discrete resource is not available, the delays may be significant.

In [11] a scheduling algorithm was proposed where the continuous resource allocation is modified in response to the late or early completion of activities performed according to the predictive schedule. As it was already mentioned the given feasible sequence of activities remains unchanged. Otherwise the feasibility of the allocation of the discrete resources could be violated. The proposed method is dedicated to the case where functions f_i $i = 1, 2, \ldots, n$, are concave. Moreover, the method bases on the assumption that the present state of an activity is unknown during its execution. Although the proposed model of activity execution (1) allows changing the allocation of the continuous resource to an activity during its execution, we will assume that this allocation may be modified at the completion time of an activity only.

The idea of the algorithm is based on the assumption that any disturbances (late or early completion of an activity) in the execution of the schedule result from an unforeseen change of the processing demand of an activity. The adequate reactive allocation of the continuous resource to the executed activities bases on periodically modified information on the processing demand of the portion of these activities that is not yet completed. To this end the actual processing demands of the executed activities are measured and compared with the processing demands assumed in the predictive schedule.

The proposed heuristic generates results satisfactory from the practical point of view. The maximum relative deviation from the schedule with optimum resource allocation may exceed 30%, but simple realisation of the predictive schedule may result in much worse schedules (125% deviation from optimum).

5. SUMMARY

Discrete-continuous scheduling problems have been defined in the last decade. This paper presents the state of the art in this interesting area. Two general solution approaches are presented based on search techniques. One of the approaches assumes that the continuous resource allocation is calculated exactly by solving a mathematical programming problem, while the other approach converts the problem to the multi-mode resource constrained scheduling problem via discretisation of the continuous resource. The first one is more time consuming, but provides slightly better solutions. Another issue discussed in this paper is adjusting the schedule in case of unexpected disturbances during its execution. This is also one of possible and important directions for further research. Another topic for further investigation may be consideration of other optimisation criteria, like for example maximisation of the net present value.

LITERATURE

1. Bouleimen K., Lecocq H., *A new efficient simulated annealing algorithm for the RCPSP and its multiple mode version.* Proceedings of the 6th International Workshop on Project Management and Scheduling, Istanbul, July 7-9, 1998.
2. Burkov V. N., Optimal Project Control, *Prace IV Kongresu IFAC*, t. 35, Warszawa 1969, 46-57.
3. Demeulemeester E. L., Herroelen W. S., *Project Scheduling - A Research Handbook*, Kluwer Academic Publishers, Boston 2002.
4. Hartmann S., *A competitive genetic algorithm for resource-constrained project scheduling*, Naval Research Logistics, 45, 1998, 733-750.
5. Hartmann S., Drexl A., *Project scheduling with multiple modes: A comparison of exact algorithms*, Networks, 32, 1998, 283-297.
6. Herroelen W., R. Leus, *Project scheduling under uncertainty*, invited paper to be published in the special issue of EJOR containing selected papers from PMS 2002.
7. Józefowska J., M. Mika, R. Różycki, G. Waligóra, J. Węglarz, *Project scheduling under discrete and continuous resources*, w: J. Węglarz (Ed.), Project Scheduling: Recent Models, Algorithms and Applications, Kluwer Academic Publishers, Boston, 1999, 289-308.
8. Józefowska J., M. Mika, R. Różycki, G. Waligóra, J. Węglarz, *Solving discrete-continuous project scheduling problem via its discretization*, Mathematical Methods of Operations Research, 52/3, 2000, 489-499.
9. Józefowska J., M. Mika, R. Różycki, G. Waligóra, J. Węglarz, *Simulated annealing for multi-mode resource-constrained project scheduling*, Annals of Operations Research 102, 2001, 137-155.
10. Józefowska J., M. Mika, G. Waligóra, J. Węglarz, *Multi-mode resource-constrained project scheduling problem with discounted positive cash-flows - simulated annealing vs. tabu search*, Foundations of Computing and Decision Sciences, 27, 2002, 97-114.
11. Józefowska J., Różycki R., Węglarz J., *Uncertainty in discrete-continuous project scheduling,* Proc. of the Conf. AUTOMATION'03, Warszawa 2003. (to appear)

12. Józefowska J., Węglarz J., *On a methodology for discrete-continuous scheduling*, European Journal of Operational Research, 107/2, 1998, 338-353.
13. Maniezzo V., Mingozzi A., *A heuristic procedure for the multi-mode project scheduling problem based on Bender's decomposition*. In Węglarz J. (ed.) *Project Scheduling: Recent Models, Algorithms and Applications*. Kluwer, Dordrecht, 1999, 179-196.
14. Patterson J.H., Słowiński R., Talbot F.B., Węglarz J., *An algorithm for a general class of precedence and resource-constrained scheduling problems*. In Słowiński R., Węglarz J. (eds.) *Advances in project scheduling*. Elsevier, Amsterdam, 1989, 3-28.
15. Russel A. H., *Cash flows in networks*, Management Science, 16, 1970, 357-373.
16. Sprecher A., Drexl A., *Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm*, European Journal of Operational Research, 107, 1998, 431-450.
17. Talbot F.B., *Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case*, Management Science 28, 1982, 1197-1210.
18. Węglarz J., Modelling and control of dynamic resource allocation project scheduling systems. In: S. G. Tzafestas (ed.), Optimization and Control of Dynamic Operational Research Models. North-Holland, Amsterdam 1982, 105-140.