An on/off event-based formulation for RCPSP with production and consumption of resources

Oumar KONE[1], Christian ARTIGUES[1], Pierre LOPEZ[1] and Marcel MONGEAU[2]

[1] CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France
Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France
`{okone,artigues,lopez}@laas.fr`
[2] Université de Toulouse ; UPS, INSA, UT1, UTM ; Institut de Mathématiques de Toulouse
CNRS ; Institut de Mathématiques de Toulouse UMR 5219 ; F-31062 Toulouse, France
`mongeau@math.univ-toulouse.fr`

Keywords: Resource-constrained project scheduling, mixed integer linear programming, production and consumption of resources, on/off formulation.

Introduction

The resource-constrained project scheduling problem (RCPSP) is one of the best-known cumulative scheduling problems due to the interest from the operational research community, and to its many industrial applications. In this article we are concerned with the extension of the RCPSP that, beyond renewable resources considered in its basic version, also allows resources that can be produced and consumed during the execution of activities. This extension is called *RCPSP with production and consumption of resources*. For more detail about the state of the art, see for instance (Laborie 2003, Neumann *et al.* 2003, Bouly *et al.* 2005, Carlier *et al.* 2009).

In our study, after a detailed description of the RCPSP and the considered extension, we present a mixed integer linear programming (MILP) model of the problem which uses variables indexed by *events*. Finally, we present some computational results.

1 Problem description

The resource-constrained project scheduling problem (RCPSP) is defined by a tuple (V, p, E, R, B, b), where V is a set of *activities*, p is a vector of *durations*, E is a set of *precedence relations*, R is a set of renewable *resources*, B is a vector of *resource availabilities*, and b is a matrix of *demands*. A set of n activities, represented by the set $\{0, \ldots, n+1\}$ where activities 0 and $n+1$ are *dummy* activities, must be scheduled on m available renewable resources belonging to the set $R = \{1, \ldots, m\}$. Each activity i has a duration p_i (with $p_0 = p_{n+1} = 0$), and demands b_{ik} for each resource k during its processing. Each resource k is available in quantity B_k. The precedence relations (precedence constraints) are given by a set E of index pairs such that $(i, j) \in E$ means that the execution of activity i must precede that of activity j. Resource constraints dictate that at any time the sum of demands of activities being processed does not exceed the resource availability.

A *schedule* corresponding to a point S (with i^{th} component S_i, the start time of activity i, and $S_0 = 0$ the start of the project), is said *feasible* if it is compatible with both the precedence constraints and the resource constraints. The RCPSP is the problem of finding a *non-preemptive* (with no interrupted activities) schedule S of minimal *makespan* (its end date S_{n+1}) subject to precedence constraints and resource constraints. According to the computational complexity theory, this problem is *NP-hard in the strong sense* (Blazewicz *et al.* 1983, Uetz 2001).

The particularity of the *RCPSP with production and consumption of resources* is that, in addition to using the renewable resources described above, we also deal with specific resources. These resources are consumed (or not) at the start time of an activity in a certain amount and/or then produced in another amount at the completion time of this activity. More specifically, an activity i consumes c_{ip}^{-} units of resource p at the beginning of its processing and produces c_{ip}^{+} units at the end of its processing. Furthermore, the total amount of each resource must remain non negative during all the scheduling horizon.

2 Proposed model

In contrast to the formulations using variables indexed by time (like the formulations proposed by (Pritsker *et al.* 1963) and (Christofides *et. al.* 1987)), we propose here a new formulation that uses variables indexed by events. This is inspired by the work of (Pinto and Grossmann 1995) on batch process problems and on the formulation in (Dauzère-Pérès and Lasserre 1995) for flow-shop problems. Events correspond to start or end times of activities.

Remark that, in any potential optimal solution for the RCPSP (left-shifted schedule for the RCPSP, with finish-to-start precedence relations, with zero time lag), the start time of an activity is either 0 or coincides with the end time of some other activity. Consequently, the number of events to be considered is at most equal to $n + 1$. It can be easily shown that the set of left-shifted (or semi-active) schedules is *dominant* (it suffices to reduce our research to such schedules).

Extension of previously-proposed event-based models to the RCPSP is not straightforward. Indeed, such approaches were based on sequential variables on machines i.e. *where resource availability is equal to 1*, where all events are totally ranked on each machine, and such activities assigned to distinct events cannot overlap in time. For resource of availability greater than 1, as in the RCPSP, there is only a *partial* order between activities.

Zapata *et al.* 2008 propose such an event-based formulation for a multimode RCPSP. Their formulation considers that an event occurs when an activity starts or ends. Transposed to the RCPSP, this model involves three types of binary variables. Our proposition uses only *one* type of binary variable per event. We define a decision variable z_{ie} which remains equal to 1 for the duration of the process of activity i. That is why we call this model, the on/off event-based formulation (noted **OOE**). In this model, the number of events can be restricted to the number of activities n. A continuous variable t_e represents the date of event e. We also use one single extra continuous variable: $C_{\max}$ (the makespan). With formulation OOE, the resource constraints are modeled in a straightforward manner.

The OOE model does *not* involve the use of dummy activities and, as the flow-based continuous-time formulation (FCT) proposed in (Artigues *et al.* 2003), has also the advantage of being able to deal with instances containing *non-integer* activity processing times. Above all, it involves *fewer variables* compared to the models indexed by time. Remark also that the event-based formulations we are introducing in this paper do *not* involve any big-M constant.

We also define **OOE_Prec**, a preprocessed variant of OOE. Roughly speaking, it is obtained from OOE by removing, from the set of possible events for an activity, all the first events during which the activity cannot or does not need to be in processed because of its predecessors. Symmetrically, we remove the last events during which the activity cannot, or does not need to, be in process because of its successors.

3 Computational results

We perform a series of tests and we compare the results obtained by OOE et OOE_Prec with those obtained by time-indexed formulations proposed in (Pritsker *et al.* 1963) (DT), Christofides (DDT) (Christofides *et al.* 1987), and flow-based continuous-time formulation (FCT) (Artigues *et al.* 2003). We use different types of modified RCPSP instances where we randomly generate resource productions and consumptions: KSD30 (Kolisch and Sprecher 1997), BL (Baptiste and Le Pape 2000), PACK (Carlier and Néron 2001). We use instances PACK_d and KSD15_d obtained from the PACK and KSD30 instances by increasing the range of processing times as this feature is common in the process industry (Pinto and Grossmann 1995). We obtained the following results with ILOG-CPLEX (version 11) on Xeon 5110 biprocessor Dell PC clocked at 1.6GHz with 4GB RAM, running Linux Fedora as operating system. In Table 1, *%integer* gives the percentage of instances for which a (non-necessarily optimal) integer solution was found within 500 seconds of CPU time, *%Optimal* is the percentage for which an optimal solution was found, *BestSol gap* represents the average deviation percentage for solved instances from the value of the best solution known, *ΔCPM gap* provides the average deviation percentage for solved instances from the critical-path method lower bound, and *Time* is the average time (in seconds) to find an optimal solution.

Table 1. RCPSP with resource consumption and production

Inst.	Model	%integer	%Optimal	BestSol gap	ΔCPM gap	Time
KSD30	DT	84	63	10,21	25,20	52,60
	DDT	82	71	0.13	7.65	83.87
	OOE_Prec	78	1	52.63	76.13	415.70
	FCT	69	20	46.65	70.35	289.39
	OOE	1	0	65.96	65.96	
PACK	OOE	94.55	1.82	20.44	277.83	110.85
	OOE_Prec	92.73	3.64	13.13	258.18	449.26
	DT	90.91	18.18	48.04	365.49	126.63
	DDT	47.27	32.73	1.33	245.66	168.04
	FCT	9.09	0	5.90	96.41	
BL	OOE_Prec	100	0	17.61	72.57	
	DDT	94.87	48.72	1.32	47.62	125.55
	DT	87.18	38.46	49.82	119.82	108.60
	OOE	74.36	0	27.64	87.56	
	FCT	20.51	0	26.79	72.51	
KSD15_d	FCT	100	93.94	0.12	10.15	18.26
	OOE_Prec	100	80.81	0.05	10.08	30.69
	OOE	100	79.80	0.10	10.14	62.33
	DT	0	0			
	DDT	0	0			
PACK_d	OOE_Prec	96.36	5.45	1.62	249.77	252.09
	OOE	96.36	5.45	5.80	264.41	320.62
	FCT	5.45	1.82	0	44.02	100.01
	DT	0	0			
	DDT	0 %	0 %			

We note that, in terms of number of (not necessarily optimal) solutions found, the model OOE and its variant OOE_Prec have almost the best performance on three types of instances (PACK, BL and PACK_d), the second best performance on KSD15_d, and the third best performance on the last one type (KSD30). Overall, these results allow us to conclude that OOE and its variant OOE_Prec are the best overall at finding integer solutions. Moreover, thanks to the preprocessing, OOE_Prec obtains better results than OOE. In terms of optimal solutions found, OOE models are outperformed by DT and DDT for the instances with small duration ranges, which is due to their weaker relaxations. However, OOE models are uncomparably better than DT and DDT for the instances with high duration ranges.

Conclusion

In this paper, we proposed a new MILP formulation for the RCPSP with consumption and production of resources using variables indexed by events. Compared to other classical formulations issued from the literature, our formulation provides encouraging results. Future work should consider designing MILP approaches that combines the advantages of time-indexed and event-based formulations.

References

C. Artigues, P. Michelon, and S. Reusser, "Insertion techniques for static and and dynamic resource-constrained project scheduling", *European Journal of Operational Research*, vol. 149(2), pp. 249-267, 2003.

P. Baptiste and C. Le Pape, "Constraint propagation and decomposition techniques for highly disjunctive and highly cumulative project scheduling problems", *Constraints*, vol. 5(1-2), pp. 119-139, 2000.

J. Blazewicz, J. Lenstra, and A.H.G. Rinnooy Kan, "Scheduling subject to resource constraints: Classification and complexity", *Discrete Applied Mathematics*, vol. 5(1), pp. 11-24, 1983.

H. Bouly, J. Carlier, A. Moukrim and M. Russo, "Solving RCPSP with resources production possibility by tasks", In : *MHOSI'2005, 24–26 April*, 2005.

J. Carlier and E. Néron, "On linear lower bounds for resource constrained project scheduling problem", *European Journal of Operational Research*, vol. 149, pp. 314-324, 2001.

J. Carlier, A. Moukrim and H. Xu, "The project scheduling problem with production and consumption of resources: A list-scheduling based algorithm", *Discrete Applied Mathematics*, 2009.

N. Christofides, R. Alvarez-Valdès, and J.M. Tamarit, "Project scheduling with resource constraints: A branch and bound approach", *European Journal of Operational Research*, vol. 29(3), pp. 262-273, 1987.

S. Dauzère-Pérès and J.B. Lasserre, "A new mixed-integer formulation of the flow-shop sequencing Problem", *2nd Workshop on Models and Algorithms for Planning and Scheduling Problems*, Wernigerode, Germany, May 1995.

R. Kolisch and A. Sprecher, "PSPLIB - A project scheduling library", *European Journal of Operational Research*, vol. 96(1), pp. 205-216, 1997.

P. Laborie, "Algorithms for propagating resource constraints in AI planning and scheduling: Existing approaches and new results", *Artificial Intelligence*, vol. 143, pp. 151-188, 2003.

K. Neumann, C. Schwindt, and J. Zimmermann, "Project Scheduling with Time Windows and Scarce Resources", *Springer*, 2003.

J. M. Pinto and I. E. Grossmann, "A continuous time MILP model for short term scheduling of batch plants with pre-ordering constraints", *Industrial & Engineering Chemistry Research*, vol. 34(9), pp. 3037–3051, 1995.

A. Pritsker, L. Watters, and P. Wolfe , "Multi-project scheduling with limited resources: A zero-one programming approach", *Management Science*, vol. 16, pp. 93-108, 1969.

M. Uetz, Algorithms for Deterministic and Stochastic Scheduling, PhD thesis, Technische Universität Berlin, 2001.

J. C. Zapata, B. M. Hodge, and G. V. Reklaitis, "The multimode resource constrained multiproject scheduling problem: Alternative formulations", *AIChE Journal*, vol. 54(8), pp. 2101-2119, 2008.