SOLVING DYNAMIC RCPSP USING EXPLANATION-BASED CONSTRAINT PROGRAMMING

Abdallah Elkhyari, Christelle Guéret and Narendra Jussien

École des Mines de Nantes and IRCCyN, France
{aelkhyar,gueret,jussien}@emn.fr

The Resource Constrained Project Scheduling Problem (RCPSP) is a general scheduling problem. It consists of a set of activities and a set of renewable resources. Each resource is available in a given constant amount. Each activity has a duration and requires a constant amount of resource to be processed. Preemption is not allowed. Activities are related by two sets of constraints: temporal constraints modelled through precedence constraints, and resource constraints that state that for each time period and for each resource, the total demand cannot exceed the resource capacity. The objective considered here is the minimization of the makespan (total duration) of the project. This problem is *NP-hard* [3].

Most work about RCPSP consider static problems in which activities are known in advance and constraints are fixed. However, every schedule is subject to unexpected events (consider for example a new activity to schedule, or a resource failure – *eg.* machine breakdown). When such a situation arises, a new solution, taking these events into account, is needed in a preferably short time. Two classical methods used to solve such problems are: recomputing a new schedule from scratch each time an event occurs (a quite time consuming technique) and constructing a partial schedule and completing it progressively as time goes by (like in on-line scheduling problems – this is not compatible with planning purposes). Recently, *Artigues et al.* [1] developed a polynomial algorithm based on a flow network model to update an initial static schedule when considering the insertion of an unexpected activity.

Constraint Satisfaction Problems (CSP) are increasingly used for solving scheduling problems. However, dynamic CSP (an extension of the CSP framework where the set of variables or/and constraints evolves throughout computation [7]) are not as spread. We introduce here such a new method applied to dynamic RCPSP.

Efficiently solving dynamic problems requires incremental addition and retraction of constraints. Even though incremental constraint addition is naturally handled by modern constraint solvers, incremental retraction of constraints is often performed with recording trace/undo information [2]. Such an information is used to determine past effects of removed constraint that need to be undone.

Explanations are a generalization of that information. An explanation is a set of constraints that justifies an action of the solver (classically value removals) *i.e.* as long as each constraint that appears in the explanation remains active (not removed) the value removal is valid; thus, no valid solution can be build from this partial assignment [10].

Providing explanations for temporal binary constraints is straightforward (as they are binary mathematical relations – see [10]). Explanations for resource management constraints are not that easy. It is necessary to study the algorithms used for propagation (resource limitations maintenance): *core-times* [11], *task-interval* [6, 5] and *resource-histogram* [6, 5].

We added explanation capabilities to these techniques [8]. Moreover, we developed a branch and bound search using explanation-based techniques (as in [9]) and inspired from a branch and bound algorithm from [4]. Our interactive system accepts several types of modification on the scheduling problem: temporal events (adding/removing precedence/overlapping/ disjunctive relations, modifying time-windows), activity related events (adding/removing),

Table 1: Some Kolish, Sprecher and Drexl instances (4 consecutive dynamic events). (`www.wior.uni-karlsruhe.de/RCPSP/ProGen.html`). Relative speed-up (in %)

12 act./4 res.	Modif. 1	Modif. 2	Modif. 3	Modif. 4
# 1	46.24	54.92	62.68	63.13
# 2	3.81	26.38	46.62	55.92
# 3	22.13	10.64	21.08	22.21
# 4	-29.21	-0.35	15.74	30.27
# 5	32.01	67.90	75.15	75.65
# 6	46.72	52.58	47.75	49.27
# 7	12.76	26.09	35.84	35.58
# 8	35.88	44.12	45.82	45.87

resource related events (adding/removing/modifying).

Table 1 presents some first results on dynamic RCPSP. In this table, we report the relative time speed-up obtained using explanation-based dynamic constraint solving compared to solving each problem from scratch. The table reports results considering 4 consecutive modifications from an original problem. As we can see, those results are quite promising. Even bad results (instance 4) get better in the long run. However, notice that some results (not reported here) show that dynamic handling is not always the panacea and rescheduling from scratch can be very quick.

We are currently improving our system with user-interaction capabilities still using explanations. Moreover, we are conducting high scale experiments (larger problems, easy/medium/ hard problems, ...) to validate our approach.

References

[1] C. Artigues and F. Roubellat. A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes. *EJOR*, 127(2):297–316, 2000.

[2] Christian Bessière. Arc consistency in dynamic constraint satisfaction problems. In *Proceedings AAAI'91*, 1991.

[3] J. Blazewicz, J.K. Lenstra, and A.H.G. Rinnoy Kan. Scheduling projects subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5:11–24, 1983.

[4] P. Brucker, S. Knust, A. Schoo, and O. Thiele. A branch and bound algorithm for the Resource-Constrained Project Scheduling Problem. *European Journal of Operational Research*, 107:272–288, 1998.

[5] Y. Caseau and F. Laburthe. Cumulative scheduling with task-intervals. In *Joint International Conference and Symposium on Logic Programming (JICSLP)*, 1996.

[6] Yves Caseau and François Laburthe. Improving clp scheduling with task intervals. In P. Van Hentenryck, editor, *Proc. of the 11th International Conference on Logic Programming, ICLP'94*, pages 369–383, 1994.

[7] R. Dechter and A. Dechter. Belief maintenance in dynamic constraint networks. In *AAAI'88*, pages 37–42, St Paul, MN, 1988.

[8] Abdallah Elkhyari, Christelle Guéret, and Narendra Jussien. Conflict-based repair techniques for solving dynamic scheduling problems. In *Proceedings CP'02*, LNCS 2470, pages 702–707, 2002.

[9] Christelle Guéret, Narendra Jussien, and Christian Prins. Using intelligent backtracking to improve branch and bound methods: an application to open-shop problems. *European Journal of Operational Research*, 127(2):344–354, 2000.

[10] Narendra Jussien. e-constraints: explanation-based constraint programming. In *CP01 Workshop on User-Interaction in Constraint Satisfaction*, Paphos, Cyprus, 1 December 2001.

[11] R. Klein and A. Scholl. Computing lower bounds by destructive improvement: an application to Resource-Constrained Project Scheduling Problem. *European Journal of Operational Research*, 112:322–345, 1999.