A more powerful energetic reasoning for the Cumulative Scheduling Problem

Jacques CARLIER[1], Antoine JOUGLET[1], Abderrahim SAHLI[2]

[1] Sorbonne universités, Université de Technologie de Compiègne, laboratoire Heudiasyc UMR 7253,
57 avenue de Landshut 60 203 Compiègne cedex, France
{jacques.carlier, antoine.jouglet}@utc.fr

[2] COSYS-GRETTIA, Université Gustave Eiffel, ESIEE Paris, 93162 Noisy-le-Grand, France
{abderrahim.sahli}@esiee.fr

Keywords: *Scheduling problems, algorithms, complexity, energetic reasoning.*

Abstract

In the Cumulative Scheduling Problem (CuSP), a set I of n tasks $\{J_1, \dots, J_n\}$ has to be scheduled on a resource of a given capacity m. Each task J_i has a release date r_i, a duration p_i, a deadline d_i and it requires c_i units of the resource during its processing. Erschler et al. [6] proposed the concept of energetic reasoning which gives a necessary condition of solution existence. This necessary condition, which has to be checked, is based on the energy balance over all intervals relying on the fact that tasks cannot be interrupted. Different checkers have been proposed in the literature. Baptiste, Le Pape and Nuijten proposed an $O(n^2)$ checker that evaluates the energy balance of $15 \times n^2$ intervals [2]. Derrien and Petit [5] showed that it is sufficient to consider only $2 \times n^2$ intervals. Ouellet and Quimper proposed an $O(n \log^2 n)$ checker which uses Monge Matrix and Range trees [9]. In [3], we have proposed an $O(\alpha(n) n \log n)$ checker by limiting the number of useful intervals and applying the methodology of Ouellet and Quimper. The drawback of these checkers is that they do not take into account the integrity of c_i even if they are strong filtering techniques for the Cumulative Scheduling Problem (CuSP). Tesch [11] and Ouellet and Quimper [8] presented algorithms for computing time-bound adjustments for the CuSP that involve energetic reasoning in $O(n^2 \, log \, n)$ time. In [4], a quadratic algorithm for the same purpose was proposed.

The aim of this talk is to propose a new definition of the energy balance of intervals. It is based on a new mathematical model. Let $[\alpha, \delta]$ be an interval and let $J(\alpha, \delta)$ be the set of tasks always in interaction with this interval. Let a_i (resp. b_i) be the maximal part of task i which can be processed strictly before (resp. after) the interval. We have to find a tripartition $\{A, B, C\}$ of $J(\alpha, \delta)$ such that:

$$\sum_{i \in A} c_i \le m$$
$$\sum_{i \in B} c_i \le m$$

and maximizing $\sum_{i \in A} a_i + \sum_{i \in B} b_i$

The evaluation of this model is more accurate than classical ones because of both constraints. We maximize the part of $J(\alpha, \delta)$ which is executed outside of the interval, which permits to minimize the part executed in the interval for getting a lower bounding evaluation. This model can be solved by a dynamic programming method in the general case.

This paper mainly considers the particular case of the CuSP where any c_i is equal to 1. It is the m-machine problem and corresponds to classical checkers as the one of Baptiste et al. [2]. We propose to evaluate the energy balance with the solution of the new formulation. We get nice properties as Monge one. We show how to solve the mathematical model by using Roy flow algorithm which permits to build a quadratic checker. The results are immediately generalized to the CuSP. Finally, we show that we get a more powerful checker but the global difference is not so large even it is strict as we explain

thanks to examples. In [7, 11], the authors introduced two new visions of Energetic Reasoning that explore new perspectives which turn out to be very similar to the one presented in this paper.

The perspective of our work is to apply this new evaluation to the CuSP taking into account integrity of c_i, which is not the case for classical checkers. Of course, we will lose the previous nice properties, and the global complexity resolution will increase. But it will become much more powerful.

References

[1] Artigues, C., & Lopez, P. (2015). Energetic reasoning for energy-constrained scheduling with a continuous resource. Journal of Scheduling. Vol.18, N°3, pp.225-241.

[2] Baptiste, P., Le Pape, C., & Nuijten, W. (1999). Satisfiability tests and time-bound adjustments for cumulative scheduling problems. Annals of Operations Research, 92, 305-333.

[3] Carlier, J., Sahli, A Jouglet, A., Pinson, E., & Sahli, A. (2021), A faster checker of the energetic reasoning for the cumulative scheduling problem. International Journal of Production Research, 1-16.

[4] Carlier, J., Pinson, E., Sahli, A. & Jouglet, A., (2020). An $O\ (n^2)$ algorithm for time-bound adjustments for the cumulative scheduling problem. European Journal of Operational Research, 286(2), pp.468-476.

[5] Derrien, A., & Petit, T. (2014). A new characterization of relevant intervals for energetic reasoning. In International conference on principles and practice of constraint programming, 289-297.

[6] Erschler, J., Lopez, P., & Thuriot, C. (1991). Raisonnement temporal sous contraintes de ressource et problèmes d'ordonnancement. Revue d'intelligence artificielle, 5, 7-32.

[7] Hidri, L., Gharbi, A. and Haouari, M. (2008), Energetic Reasoning Revisited: Application to Parallel Machine Scheduling, Journal of Scheduling, Volume 11, pp. 239-252.

[8] Lahrichi, A. (1982). Ordonnancements : La notion de parties obligatoires et son application aux problèmes cumulatifs. RAIRO no. 3, p. 241-262

[9] Ouellet, Y., & Quimper, C-G. (2018). A $O(n \log^2 n)$ Checker and $O(n^2 \log n)$ Filtering Algorithm for the Energetic Reasoning. In Proceedings of the 15th International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR 2018), pages 477-494.

[10] Tercinet, F., Néron, E., and Lenté, C. (2006). Energetic reasoning and bin-packing problem for bounding a parallel machine scheduling problem. 4OR, 4, 297-318.

[11] Tesch, A. (2018). Improving energetic propagations for cumulative scheduling. In International Conference on Principles and Practice of Constraint Programming, 629-645.