A new energetic reasoning for the Cumulative Scheduling Problem: a dynamic programming approach

Kristina Kumbria[1], Jacques Carlier[1], Antoine Jouglet[1], Abderrahim Sahli[2]

[1] Université de Technologie de Compiègne, France
{kristina.kumbria, jacques.carlier, antoine.jouglet}@hds.utc.fr
[2] Univ Gustave Eiffel,ESIEE Paris, France
abderrahim.sahli@esiee.fr

Keywords : *cumulative scheduling problem, energetic reasoning, dynamic programming.*

1 Introduction

The Cumulative Scheduling Problem (CuSP) involves scheduling a set $I = \{1, \ldots, n\}$ of n tasks in a resource with a specified capacity of m units. Each task i possesses distinctive attributes such as a release date r_i, a duration p_i, a deadline d_i, and a resource requirement of c_i units. Energetic Reasoning (ER), pioneered by Erschler and Lopez [1], has appeared as a potent approach for addressing the challenges posed by CuSP. ER primarily concentrates on devising feasibility tests, termed ER checkers, along with adjustments related to time constraints. A range of ER checkers has been suggested in existing literature. Baptiste, Le Pape, and Nuijten [2] proposed an $O(n^2)$ checker, which evaluates the energy balance across $O(n^2)$ intervals. Ouellet and Quimper [4] introduced an $O(n \log^2 n)$ checker based on the Monge Matrix and Range trees. We presented an $O(\alpha(n) n \log n)$ checker [3], reducing the number of necessary intervals and following the methodology of Ouellet and Quimper, where $\alpha(n)$ is the inverse Ackermann function. This presentation aims to introduce a new definition of the energetic reasoning method for checkers.

2 Tripartition Problem: Dynamic programming approach

Here, we present a problem formulation involving a tripartition scenario, focusing on a subset of tasks denoted as $\mathcal{J} \subseteq I$. Each task within this subset is characterized by three integer values: a_i, b_i, and c_i. Additionally, there are given two values, $m_{\mathcal{A}}$ and $m_{\mathcal{B}}$, both smaller than the overall capacity m. The objective is to solve the following mathematical program:

$$P(\mathcal{J}, m_{\mathcal{A}}, m_{\mathcal{B}}) \qquad = \max \sum_{i \in \mathcal{J}} (a_i c_i x_i + b_i c_i y_i)$$

$$\textbf{subject to:} \qquad x_i + y_i \leq 1, \forall i \in \mathcal{J} \quad \textbf{and} \quad \sum_{i \in \mathcal{J}} c_i x_i \leq m_{\mathcal{A}} \quad \textbf{and} \quad \sum_{i \in \mathcal{J}} c_i y_i \leq m_{\mathcal{B}}$$

where $\mathcal{A}$ and $\mathcal{B}$ are disjoint subsets of $\mathcal{J}$ and $\mathcal{M} = \mathcal{J} / \mathcal{A} \cup \mathcal{B}$. Binary variables $x_i \in \{0, 1\}$ (resp. $y_i \in \{0, 1\}$) indicate whether task i belongs to $\mathcal{A}$ (or $\mathcal{B}$). We solve this Integer Linear Programming model by using a linear dynamic programming approach.

3 The Tripartition Problem and Energy Evaluation

We establish a lower bound on energy requirement within a given interval $[\alpha, \delta]$ using a dynamic programming model. Minimizing this energy requirement is equivalent to maximizing the parts of tasks that do not fall within the interval. Let $\mathcal{J}(\alpha, \delta)$ be the set of tasks that consistently

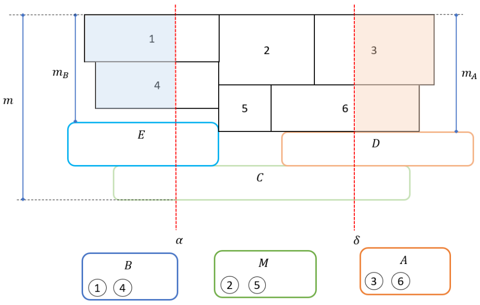

FIG. 1: Tripartition figure

interact with interval $[\alpha, \delta]$, formally defined as $\mathcal{J}(\alpha, \delta) = \{i \in I \mid r_i + p_i \geq \alpha \text{ and } d_i - p_i \leq \delta\}$. In this context, b_i (resp. a_i) denotes the maximum part of task i that can be processed strictly before α (reps. after δ).

Definition 1 *A task i is called a crossing task of the interval if $(d_i - p_i) < \alpha$ and $(r_i + p_i) > \delta$. A task i is a plus-semi-crossing task if $(r_i + p_i) > \delta$ and $\alpha \leq (d_i - pi) \leq \delta$. It is a minus-semi-crossing task if $(d_i - p_i) < \alpha$ and $\alpha \leq (r_i + p_i) \leq \delta$.*

There are a total of m' crossing tasks, leaving (m - m') resources available. However, some of these resources are still required by semi-crossing tasks. This leads to $m_{\mathcal{B}}$ available resources before α and $m_{\mathcal{A}}$ available resources after δ. Define $\mathcal{C}(\alpha, \delta)$ (resp. $\mathcal{D}(\alpha, \delta); \mathcal{E}(\alpha, \delta)$) to be the set of crossing (resp. semi-crossing) tasks, and w_i the minimum energy required by crossing and semi-crossing tasks. Then, let $\mathcal{J}'(\alpha, \delta)$ be the set of tasks in $\mathcal{J}(\alpha, \delta)$ that do not belong to $\mathcal{C}(\alpha, \delta) \cup \mathcal{D}(\alpha, \delta) \cup \mathcal{E}(\alpha, \delta)$. The total energy over the time interval $[\alpha, \delta]$, denoted as $W(\alpha, \delta)$, can be computed by determining a tripartition $(\mathcal{A}, \mathcal{B}, \mathcal{M})$ of $\mathcal{J}'(\alpha, \delta)$.

$$W(\alpha, \delta) = \sum_{i \in \mathcal{C}(\alpha,\delta) \cup \mathcal{D}(\alpha,\delta) \cup \mathcal{E}(\alpha,\delta)} w_i + \sum_{i \in \mathcal{J}'(\alpha,\delta)} p_i c_i - P(\mathcal{J}'(\alpha, \delta), m_{\mathcal{A}}, m_{\mathcal{B}})$$

4 Checker

We have proposed a new ER checker for the CuSP based on dynamic programming. Recognizing the drawbacks of evaluating all possible intervals, we pinpointed useful relevant intervals denoted by $[\alpha, \delta]$, where $\alpha \in \{r_i, r_i + p_i, d_i - p_i \mid i \in I\}$ and $\delta \in \{r_i + p_i, d_i, d_i - p_i \mid i \in I\}$.

References

[1] J. Erschler and P. Lopez, "Energy-based approach for task scheduling under time and resources constraints", In Proceedings of the 2^{nd} international workshop on project management and scheduling, pages 115–121, 1990.

[2] P. Baptiste, C. Le Pape, and W. Nuijten, 1999, "Satisfiability tests and time-bound adjustments for cumulative scheduling problems", *AOR*, Vol. 92, pp. 305-333.

[3] J. Carlier, A. Sahli, A. Jouglet and E. Pinson, 2022, "A faster checker of the energetic reasoning for the cumulative scheduling problem", *International Journal of Production Research*, 60:11, 3419-3434, DOI: 10.1080/00207543.2021.1923853

[4] Y. Ouellet, and C-G. Quimper, 2018, "An $O(nlog^2n)$ Checker and $O(n^2logn)$ Filtering Algorithm for the Energetic Reasoning", *CPAIOR*, pp. 477-494.