Two Deadline Reduction Algorithms for Scheduling Dependent Typed-tasks Systems

Claire Hanen[1,2], Alix Munier-Kordon[1]

[1] Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005 Paris, France
[2] UPL , Université Paris Nanterre , 92000, Nanterre, France
{Claire.Hanen,Alix.Munier}@lip6.fr

Mots-clés : *Scheduling, typed-tasks system, makespan, maximum lateness*

1 Introduction and problem definition

Let us consider the scheduling problem $P\Sigma^k|prec, r_i, d_i|\star$ defined by a set of task T of fixed duration and a precedence graph G . Each task $i \in T$ has a deadline d_i, a release date r_i and a duration p_i. Machines are partitioned into k classes, and each task can be executed by one machine of a fixed class. These constraints include both identical machines (only one class of machines) and dedicated machines (one machine per class). The problem is the existence of a feasible schedule, which is the decision problem associated to the minimization of the makespan (C_{max}) or the maximum lateness (L_{max}).

This problem is $\mathcal{NP}$-hard even if $p_i = 1$, for any task $i \in T$ and $k = 1$[9]. The aim of this talk is to study polynomial methods reducing the deadlines d_i using necessary conditions. These reductions are crucial to reduce the set of feasible schedules, and thus to improve the efficiency of branch and bound algorithms [3]. They also led to polynomial special cases and approximation algorithms (see for example [6]).

Two main classes of algorithms have been developed for unitary duration case (*i.e.* $p_i = 1, \forall i \in T$) to improve the initial evaluation of the deadlines. Their main idea is to reduce the deadlines until a fixed point is reached. The Garey-Johnson algorithm [5](GJ), which was initially considered for scheduling optimally unit duration tasks on 2 identical processors, expressed necessary conditions on the volume of the tasks allowed to particular intervals. The Leung-Palem-Pnueli algorithm [7] (LPP) expresses necessary conditions on deadlines according to precedence constraints and uses Jackson's algorithm to determine the existence of a feasible algorithm meeting resource limitations. Hanen and Munier [2] showed that these two algorithms lead to the same deadlines and an experimental study confirmed that the algorithm LPP is faster than GJ. The purpose of this talk is to develop extensions of these two algorithms for any duration with or without preemption.

2 Extension of Garey-Johnson algorithm

We generalize the Garey-Johnson algorithm by using the energetic reasoning studied for cumulative scheduling problems without precedence constraints [1, 4]. For any task $i \in T$ and any interval $[s, d]$ with $d_i \in [s, d]$ and $r_i \leq s$, we define by $W(i, s, d)$ the sum of the minimum duration that each task must spend in this interval assuming that i ends at time d_i and that the other deadlines are fulfilled. For parallel processors the consistency condition can be stated as $|W(i, s, d)| \leq m(d - s)$, allowing a reduction of d_i if it is not fulfilled.

We then prove that s and d can be reduced to vary in a finite set of values $\{r_j, d_j, d_j - p_j, r_j + p_j\}_{j \in \mathcal{T}}$.This leads to a polynomial deadline modification algorithm based on the iterative computation of values $|W(i, s, d)|$ is designed and proved to lead to a consistent deadline vector $d_i^\star \leq d_i, i \in T$, if such a set exists.

The extension to $P\Sigma^k|prec, r_i, d_i|\star$ is then discussed.

3 Extension of Leung-Palem-Pnueli algorithm

For any task $i \in T$, by assuming that i ends at time d_i, we adjust release times of its successors. Let us consider then the function $BS_i(r, D)$ that returns the maximum value $d \leq D$ such that the (new) release times and deadlines are met assuming preemptive tasks. The decision version of this problem can be polynomially solved using Martel's algorithm [8]. We then prove that $BS_i(r, D)$ is also a polynomial problem using a binary search.

An algorithm is then presented based on the iterative adjustment function $BS_i(r, D)$ that converges to (another) set of deadlines $\tilde{d}_i, i \in T$, such that $BS_i(r, \tilde{d}) = \tilde{d}_i, \forall i \in T$. We then show that $\tilde{d}$ is also consistent with respect to the Garey-Johnson consistency.

4 Conclusion and perspectives

A first perspective of this work is clearly to compare experimentally the two deadline vectors $d^\star$ and $\tilde{d}$ and the complexity of these two methods. Another interesting question is to identify experimentally or theoretically particular structures for which these vectors are equal or different, if any. This study will be then a first step for answering about the existence of two extensions of the LPP and the GJ algorithms converging to same deadline vectors.

References

[1] Ph. Baptiste, C. Le Pape, and W. Nuijten. Satisfiability tests and time-bound adjustments for cumulative scheduling problems. *Annals of Operations Research*, 92(0):305–333, Jan 1999.

[2] Aurélien Carlier, Claire Hanen, and Alix Munier Kordon. The equivalence of two classical list scheduling algorithms for dependent typed tasks with release dates, due dates and precedence delays. *J. Scheduling*, 20(3):303–311, 2017.

[3] Jacques Carlier, Laurent Péridy, Eric Pinson, and David Rivreau. Elimination rules for job-shop scheduling problem. In *Handbook of Scheduling - Algorithms, Models, and Performance Analysis.* 2004.

[4] J. Erschler, P. Lopez, and C. Thuriot. Raisonnement temporel sous contraintes de ressources et problèmes d'ordonnancement. *Revue d'Intelligence Artificielle*, 5(3):7–36, 1991.

[5] M. R. Garey and D. S. Johnson. Two-processor scheduling with start-time and deadlines. *SIAM Journal on Computing*, 6:416–426, 1977.

[6] Claire Hanen and Yakov Zinder. The worst-case analysis of the garey-johnson algorithm. *Journal of Scheduling*, 12(4):389–400, 2009.

[7] Allen Leung, Krishna V. Palem, and Amir Pnueli. Scheduling time-constrained instructions on pipelined processors. *ACM Trans. Program. Lang. Syst.*, 23:73–103, January 2001.

[8] Charles Martel. Preemptive scheduling with release times, deadlines, and due times. *Journal of the Association of Computing Machinery*, 29(3):812–829, July 1982.

[9] J.D. Ullman. NP-complete scheduling problems. *J. Comput. System Sci.*, 10:384–393, 1975.