A polynomial algorithm for periodic scheduling

Maël Guiraud[1], Yann Strozecki[2]

[1] LINEACT, CESI, Nanterre
mguiraud@cesi.fr
[2] DAVID, Université Paris Saclay
yann.strozecki@uvsq.fr

Keywords : *Periodic scheduling, Integer linear programming, Totally unimodular constraint matrix.*

1 Introduction

We are interested in problems related to periodic scheduling, driven by applications using networks in which terminals send identical flows periodically. The management of such flows must not only minimize latency, but also jitter [4], and we have shown that classical methods of sizing networks by statistical multiplexing are not suitable for this situation [3].

2 Studied Problem

We begin by focusing on a singular resource, which is the output port of a node in the case of networks. The tasks (or flows for networks) have a release time and a deadline, are all the same size and the same period. Our goal is to determine a schedule, i.e. date of allocation of the resource for each task, which fulfills these imposed time bounds. If we neglect the aspect of periodicity, we are presented with a fundamental scheduling problem. The aim is to decrease the makespan, and this particular problem is solvable in $\mathcal{O}(n \log n)$ [2].

We address here a decision problem rather than a minimization problem. Adding periodicity to our model significantly modifies our objective, as we are no longer attempting to minimize the makespan, but searching for a valid periodic schedule. In a **valid** periodic schedule, we want to associate to every task a date in the period such that the resource is not used by another task. The schedule must not only avoid collisions with tasks from the same period, but also from preceding or succeeding periods. Figure 1 shows a collision caused by periodicity. Therefore, the periodic scheduling problem we are studying is as follows: Given a period (identical for each task), and a set of tasks with each its own release time and deadline, is there a valid periodic schedule?

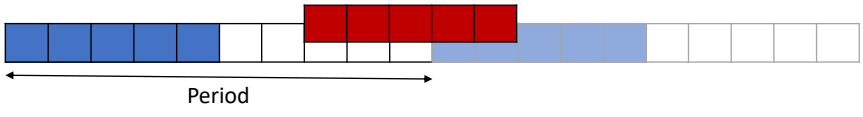

FIG. 1: A collision due to periodicity.

3 Achievements

For the non-periodic problem, a mixed-integer linear program is proposed by [1] in which the constraint matrix is totally unimodular, enabling it to be solved in polynomial-time. Building on this, we propose an integer linear program that takes periodicity into account. The solution to this linear program provides a set of dates at which tasks must be scheduled, rather than the periodic scheduling itself. Applying HALL's theorem, which guarantees a perfect matching between the set of release times and the set of dates provided by the solution, we are able to calculate the periodic solution.

4 Perspectives

It is possible to transform the linear program into an instance of shortest path problem, an idea mentioned in [1]. We aim to adapt this approach to solve our periodic problem with the expectation of achieving better complexity in both theoretical and practical cases. We intend to avoid generating the linear program and relying on a solver by generating the graph on which we compute the shortest path directly, preferably implicitly. This algorithm will help the general problem of periodic scheduling in general topologies: it serves as a relaxation in a branch and bound method and at the terminal nodes within the network [3].

References

[1] Christoph Dürr and Mathilde Hurand. Finding total unimodularity in optimization problems solved by linear programs. *Algorithmica*, 59(2):256–268, 2011.

[2] Michael R Garey, David S. Johnson, Barbara B. Simons, and Robert Endre Tarjan. Scheduling unit–time tasks with arbitrary release times and deadlines. *SIAM Journal on Computing*, 10(2):256–269, 1981.

[3] Maël Guiraud. *Ordonnancement periodiques de messages pour minimiser la latence dans les réseaux dans un contexte 5G et au delà.* PhD thesis, université Paris-Saclay, 2021.

[4] Jinoo Joung and Juhyeok Kwon. Zero jitter for deterministic networks without time-synchronization. *IEEE Access*, 9:49398–49414, 2021.