Fixed task scheduling on single thread processors

Valentina Cacchiani, Alberto Caprara, Paolo Toth
DEIS, University of Bologna, Italy

Introduction

Given:

- ***n* TASKS: start time, end time, weight**
- ***m* SINGLE-THREAD PROCESSORS: capacity, cost**

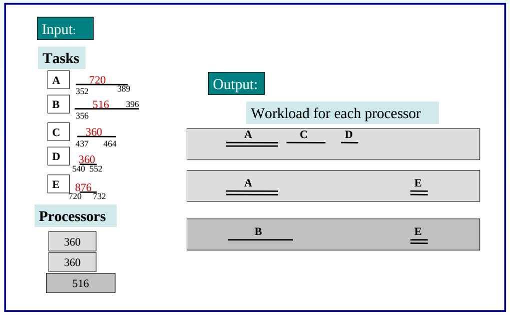
Constraints:

- **Each task must be executed from its start time to its end time by a set of processors with overall capacity at least equal to the task weight**
- **There may be upper bounds (e.g., one or two) on the number of processors on which each task can be executed**
- **There is a setup time from the end of a task to the beginning of the next one on a processor, which in general depends on the two tasks**

Goal:

Determine the minimum cost schedule of the task on the processors such that all the tasks are executed

Example



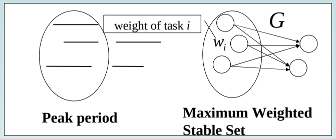
Objectives

Develop a HEURISTIC ALGORITHM that

- **obtains good quality solutions in short computing time**
- **can be applied to different variants of the problem (e.g. TRAIN UNIT ASSIGNMENT PROBLEM arising in Railway Optimization)**

Method

Lower Bound

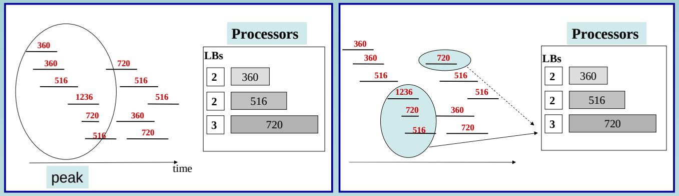


- **Compute the Maximum Weighted Stable Set on this Comparability Graph**
- **Solve an ILP model to determine a lower bound on the number of processors**

Heuristic Algorithm

- **Take the Lower Bound as the number of available processors and the tasks in the stable set as the first ones to be assigned**
- **Assign one task at a time to processor types (based on the task weight and on the capacity of the processors)**
- **Solve an Assignment Problem for each processor type to find the best sequencing of the tasks assigned to the current type up to now**
- **Accept the assignment only if its cost respects the LB. Otherwise, try different processor types. If all the assignments violate the LB, leave the task unassigned**

Example



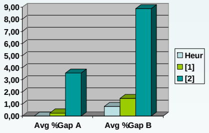
Improvement

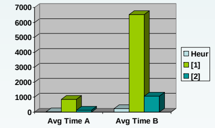
- **Feasibility phase: consider the unassigned tasks and assign them by using all the processors**
- **Iterative execution: after the construction of a solution, take the unassigned tasks as the first one to be assigned (together with the tasks in the stable set)**

Results

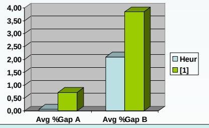
- **A: 13 Real-world instances**
- **B: 23 Realistic instances**

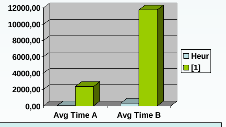
- **Min number of processors and at most 2 processors per task**





- **Min cost of processors and at most 2 processors per task**





References

[1] Cacchiani, V., Caprara, A., and Toth, P., “Solving a Real-World Train Unit Assignment Problem”, *Mathematical Programming* 124, 207–231 (2010).
[2] Cacchiani, V., Caprara, A., and Toth, P., “A Lagrangian Heuristic for a Train-Unit Assignment Problem”, *Discrete Applied Mathematics*, doi: 10.1016/j.dam.2011.10.035.
[3] Cacchiani, V., Caprara A., and Toth, P., “Models and Algorithms for the Train Unit Assignment Problem", in A.R. Mahjoub et al. (Eds.): *ISCO 2012, LNCS 7422*, Springer-Verlag Berlin Heidelberg, 24-35 (2012).