



HAL
open science

Proof of NP-completeness for a scheduling problem with coupled-tasks and compatibility graph

Gilles Simonin

► **To cite this version:**

Gilles Simonin. Proof of NP-completeness for a scheduling problem with coupled-tasks and compatibility graph. [Research Report] RR-08007, LIRMM. 2008. lirmm-00262286

HAL Id: lirmm-00262286

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00262286>

Submitted on 11 Mar 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

In this technical rapport, we investigate a scheduling problem with coupled-tasks (denoted acquisition tasks) and a compatibility graph. We will show that the problem $1|prec, coupled-task, (p_{a_i} = p_{b_i} = 1, L_i = \alpha) \cup (p_{T_i}, p_{mtn}), G_c|C_{max}$ is \mathcal{NP} -complete, with $\alpha \geq 3$. This problem is denoted by Π .

Theorem 1 *Let n be the acquisition tasks number, the problem to decide if an instance of the problem Π has a scheduling length $C_{max} = 2n + \sum_{T_i \in \mathcal{T}} p_{T_i}$, is \mathcal{NP} -complete.*

Proof

Our approach is similar to the proof of Lenstra and Rinnoy Kan [1] for the problem $P|prec; p_j = 1|C_{max}$. This demonstration is based on the *Clique* decision problem (see Garey and Johnson GT19 [2]):

INSTANCE: A graph $G = (V, E)$ where $|V| = n$, and an integer K .

QUESTION: Can we find a clique of size K in G ?

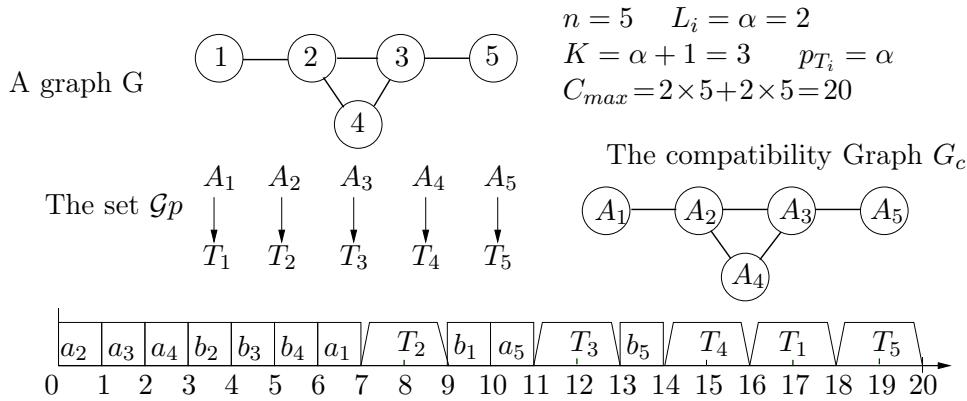


Figure 1: Illustration of polynomial-time transformation $\text{Clique} \propto \Pi$

Our proof is based on the polynomial-time transformation $\text{Clique} \propto \Pi$. It is easy to see that the problem Π is in \mathcal{NP} .

Let I^* an instance of Clique , we will construct an instance I of Π with $C_{max} = 2n + \sum_{T_i \in \mathcal{T}} p_{T_i}$ in the following way:

Let $G = (V, E)$ a graph in the instance I , with $|V| = n$:

- $\forall v \in V$, an acquisition task A_v is introduced, composed of two sub-tasks a_v and a'_v with processing time $p_{a_v} = p_{a'_v} = 1$ and with a latency time, between these two sub-tasks, of length $\alpha = (K - 1)$, called *slot*.
- For each edge $e = (v, w) \in E$, there is a compatibility relation between the two acquisition tasks A_v and A_w .
- For each task A_v , we introduce a treatment task T_v which is its successor.

- Each T_v has a processing time noted $p_{T_v} = \alpha$. Thus, the treatment tasks will replace all the inactivity slot of all the A_v after the clique.
- We suppose that there is a clique of length $K = (\alpha + 1)$ in the graph G . Let us show that there is a scheduling in $C_{max} = (2n + \sum_{T_i \in \mathcal{T}} p_{T_i})$ units of time. For that, consider the following scheduling:
 - From $t = 0$ to $t = \alpha$, we schedule the $K = (\alpha + 1)$ tasks which represent the vertices of the clique of size K .
 - From $t = (2\alpha + 2)$, we schedule the $(n - K)$ remaining tasks A_v .
 - In each slot from these $(n - K)$ tasks A_v , we schedule the tasks T_v . Since each T_v has as a value $p_{T_v} = \alpha$, by scheduling $(n - K)$ tasks T_v , we will fill each slot of length α of the $(n - K)$ tasks A_v .
 - Remaining treatment tasks are scheduled at the end of the schedule.

With this allocation, we fill all the slots and we give a valid scheduling in $(2n + \sum_{T_i \in \mathcal{T}} p_{T_i})$ units of time.

- Reciprocally, let us suppose that there is a scheduling in $(2n + \sum_{T_i \in \mathcal{T}} p_{T_i})$ units of time without inactivity time, then let us show that the graph G contains a clique of size $K = (\alpha + 1)$.

From these suppositions, we make essential comments:

- With the precedence constraints between the tasks A_v and T_v , it is easy to see that we can schedule only tasks A_v at $t = 0, \forall v \in V$. Thus, the first treatment task could be scheduled only starting from $t = (\alpha + 2)$.
- Let a_{p_1} be the first sub-task of acquisition scheduled at $t = 0$, with a slot of length α . We need a clique of size $(\alpha + 1)$ to obtain a scheduling without inactivity slot.

Thus we have $(\alpha + 1)$ acquisition tasks which are compatibles. And in the compatibility graph G_c , we will have an edge between each couple of these tasks A_v . Consequently, the tasks $A_{p_1}, A_{p_2}, \dots, A_{p_\alpha}$, associated to the vertices of the graph G , form a clique of size $K = (\alpha + 1)$.

This concludes our proof of Theorem 1.

□

Bibliographie

1. R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5:287–326, 1979.
2. M.R. Garey and D.S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.