# A Note on the Complexity of Scheduling Coupled Tasks on a Single Processor*

Jacek Błażewicz[1], Klaus Ecker[2], Tamás Kis[3], Michał Tanaś[1]

[1] Institute of Computing Science, Poznań University of Technology, Poznań, Poland

[2] Institute of Computing Science, Technical University Clausthal, Germany

[3] University of Siegen, Germany

## Abstract

This paper considers a problem of coupled task scheduling on one processor, where all processing times are equal to 1, the gap has exact length $h$, precedence constraints are strict and the criterion is to minimise the schedule length. This problem is introduced e.g. in systems controlling radar operations. We show that the general problem is NP-hard.

**Keywords:** Scheduling, coupled tasks, NP-hardness.

## 1 Introduction

A scheduling problem is, in general, a problem answering a question of how to allocate some resources over time in order to perform a given set of tasks [1]. In practical applications resources are processors, money, manpower, tools, etc. Tasks can be described by a wide range of parameters, like ready times, due dates, relative urgency factors, precedence constraints and many more. Different criteria can be applied to measure the quality of a schedule. The general formulation of scheduling problems and the commonly used notation can be found in books such as [5], [16], [2], [15], [3] and [4]. A survey of the most important results is given in [11].

One branch of scheduling theory is concerned with scheduling of coupled tasks. A task is called *coupled* if it contains two operations where the second has to be processed some time after a completion of the first one. This problem, described in [17] and [18], often appears in radar-like devices, where the first operation is the transmission of an electromagnetic pulse and the second is the reception of its echo. Several algorithms designed to solve the problem of radar pulse scheduling can be found in [9] and [12].

The complexity of various scheduling problems with coupled tasks has been studied in [13]. Although most of the cases are NP-hard [13], some polynomial algorithms were found in [14].

A coupled task scheduling problem with variable length gap is surveyed in [10] and [7]. NP-hardness of this case is proved in [19], where some interesting connections between coupled tasks and flow shops are also given.

In this note, we complement the above results by presenting the NP-completeness proof for the problem of scheduling coupled tasks on a single processor, with all processing times equal to 1, exact, integer gap length, general strict precedence constraints and the optimisation criteria of minimising the schedule length.

An organisation of the paper is as follows. The problem is formulated in Section 2. The NP-hardness proof is presented in Section 3. We conclude in Section 4.

## 2 Problem formulation

We consider the problem of scheduling $n$ coupled tasks on a single machine, where each coupled task $T_i$ consists of two operations $T_{i1}$ and $T_{i2}$ and a *gap* between them. During the gap, another task can be processed. Let $p_{i1}$ and $p_{i2}$ denote the processing times of operations $T_{i1}$ and $T_{i2}$, respectively.

The gap is exact when operation $T_{i2}$ has to start exactly $h_i$ units of time after the end of operation $T_{i1}$, where $h_i$ denotes a length of the gap. In this paper, the only cases considered are those where all $h_i$ are equal, i.e. $h_i = h$, $i = 1, 2, \ldots, n$.

*Precedence constraints* of coupled tasks can be strict or weak. $T_i \prec T_j$ means that $T_{i2} \prec T_{j1}$ if precedence constraints are *strict*, and $T_{i1} \prec T_{j1} \wedge T_{i2} \prec T_{j2}$ if they are *weak*.

The special case of a coupled task problem involves identical tasks. Commonly, such tasks are

*Partially supported by the KBN grant.

denoted by $(p_1, h, p_2)$, where $p_1 = p_{i1}$, $p_2 = p_{i2}$, $h = h_i$ for all $1 \le i \le n$.

Adapting the commonly accepted notation for scheduling problems [8], the scheduling problem considered in this paper can be denoted by $1|(1,h,1) - coupled,\ strict\ prec,\ exact\ gap|C_{max}$, which means:

- There is one processor in a system.
- Tasks are coupled and identical, with processing times $p_{i1} = p_{i2} = 1$, $\forall_{1 \le i \le n}$
- Gaps are exact and have uniform length $h$.
- Precedence constrains are strict.
- The optimisation criterion is to minimise the schedule length $C_{max} = \max\{t_{j2}\}$, where $t_{j2}$ is a completion time of $T_j$ (its second operation).

## 3 NP-hardness of the general case

In case where precedence constraints are general the problem of scheduling coupled tasks on a single processor is NP-hard even for unit processing times. We will prove this by a series of lemmae showing NP-hardness of some intermediate problems.

**Lemma 1** Problem of *Balanced Colouring of Graphs with Partially Ordered Vertices* is NP-hard.

**Proof:** The problem of *Balanced Colouring of Graphs with Partially Ordered Vertices* (*BCGPOV* for short) can be stated as follows:
*Instance:* A directed, acyclic graph $G = (V, E)$ where $|V| = q$. (It is clear that the arcs define a partial order in set $V$.)
*Question:* Can the vertices of $G$ be coloured with $l$ colours such that no pair of adjacent vertices shares the same colour and exactly $q/l$ vertices are coloured with the same colour. (We will call such a colouring the *balanced colouring*.) Without loss of generality we can assume that $q/l \in \mathbb{Z}^+$.

Firstly, we prove that *BCGPOV* belongs to *class NP*. To verify a solution of *BCGPOV* is enough to verify that

No pair of adjacent vertices is monochromatic. Because each vertex has no more than $(q-1)$ adjacent vertices the complexity of this step is $O(q^2)$.

Exactly $q/l$ vertices are coloured with each colour. Complexity of this step is $O(q)$.

A solution of *BCGPOV* can be verified in polynomial time, which means that the problem *BCGPOV* belongs to *class NP*.

In order to prove NP-completeness of the problem *BCGPOV* we will use the *3-Partition* problem, which is NP-complete in the strong sense according to [6]. The *3-Partition* problem is defined as follows:
*Instance:* A collection $A$ of $3r$ items, bound $B \in \mathbb{Z}^+$, and size $s(a_j) \in \mathbb{Z}^+$, $\forall_{a_j \in A}$ such that $B/4 < s(a_j) < B/2$ and such that $\sum_{a_j \in A} s(a_j) = rB$.
*Question:* Can $A$ be partitioned into $r$ disjoint sets $A_1, \dots, A_r$ such that for $1 \le i \le r$, $\sum_{a_j \in A_i} s(a_j) = B$ (note that each $A_i$ must contain exactly three elements from $A$)?
*The transformation:* For any instance of the *3-Partition* problem let us define the corresponding instance of the *BCGPOV* problem as follows:

- $q = 3r^2 + rB$
- $l = r$
- For each item $a_j \in A$, $1 \le j \le 3r$ let us construct graph $G_j$ in the following way:
  1. Construct complete graph $K_r^j$ on $r$ vertices. Denote one vertex of $K_r^j$ by $v_j$.
  2. Construct set $D^j_{s(a_j)}$ of $s(a_j)$ independent vertices.
  3. Create all possible edges $(v_x, v_y)$ such that $v_x \in K_r^j$, $v_x \ne v_j$ and $v_y \in D^j_{s(a_j)}$.

  It is clear that edges of graph $G_j$ can be directed to define a partial order in the set of vertices of $G_j$.

  An example of such a graph is shown in Fig 1.
- $G = \bigcup_{i=1}^{3r} G_j$

It is clear that $G_j$ can be coloured with $r$ colours only in the following way: each vertex from $K_r^j$ has a different colour and all vertices from $D^j_{s(a_j)}$ are coloured with the same colour as vertex $v_j$.

No two vertices of any subgraph $K_r^j$, $1 \le j \le 3r$ can share the same colour, so after colouring of all $K_r^j$, exactly $3r$ vertices are coloured with the same colour. All vertices of $D^j_{s(a_j)}$ are connected with all but $v_j$ vertices of $K_r^j$, so all vertices of $D^j_{s(a_j)}$ have to be coloured with the same colour as vertex $v_j$. So, the set of vertices of any $D^j_{s(a_j)}$ has to be monochromatic.

Let $A_1, \dots, A_j$ be a solution for the given instance of *3-Partition* problem and let $A_i = \{a_{i(1)}, a_{i(2)}, a_{i(3)}\}$. Let us denote $S_i = G_{i(1)} \cup G_{i(2)} \cup G_{i(3)}$. $G$ can be coloured such

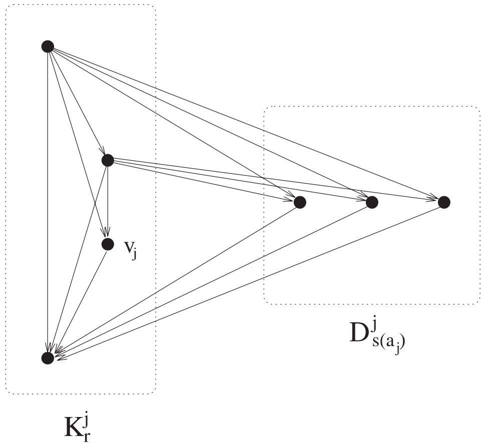


Figure 1: An example of graph $G_j$, where $q = 4$, $s(a_j) = 3$.

that sets $D_i$ and $D_j$ share the same colour if and only if both $D_i$ and $D_j$ are in the same $S_i$. It means that for all $i$ exactly $s(a_{i(1)})+s(a_{i(2)})+s(a_{i(3)}) = B$ vertices of sets $D$, so exactly $3r + B$ vertices of graph $G$, are coloured with each colour.

On the other hand, let $S_1, S_2, \ldots, S_r$ be a disjoint sets of vertices of graph $G$, such that $V(G) = \sum_{i=1}^{r} S_i$, each vertex in $S_i$ is coloured with $i$-th colour and $\forall_i\ |S_i| = 3r + B$. Let $S_i^D$ be a subset of $S_i$ that contains only vertices belonging to $D^j_{s(a_j)}$ subgraphs. $S_i$ has to contain $3r$ vertices belonging to $K_r^j$, so $|S_i^D| = B$ for each $i$. For each feasible colouring each $D^j_{s(a_j)}$ is monochromatic, so $\forall_{1\le j\le 3r}\ \exists_{1\le i\le r}\ D^j_{s(a_j)} \in S_i$. Because $\forall_{1\le j\le 3r}\ B/4 < s(a_j) < B/2$ each $S_i$ has to contain exactly 3 subgraphs $D^{i(j)}_{s(a_{i(j)})}$, $j = 1, 2, 3$. So, we can denote $A_i = \{a_{i(j)} : j = 1, 2, 3\}$, $i = 1, 2, \ldots, r$ and this is the solution of the *3-Partition* problem.

The complexity of this transformation is $O(r^2 + r\,B)$. Let us note that this is a pseudopolynomial transformation. On the other hand, *3-Partition* is NP-complete, thus, we have the desired result.

□

Now, we will show that problem *BCGPOV* polynomially transforms to our scheduling problem.

**Lemma 2** The problem of *Balanced Colouring of Graphs with Partially Ordered Vertices* polynomially transforms to $1|(1,h,1) - coupled,\ strict\ prec,\ exact\ gap|C_{max}$.

**Proof:** Let $G = (V, E)$ be an instance of problem *BCGPOV*. Let it contain all transitive arcs. Let us define a corresponding instance of $1|(1,h,1) - coupled,\ strict\ prec,\ exact\ gap|C_{max}$ problem in the following way:

- $n = q$
- $h = q/l - 1$
- For each vertex $v_i$ of graph $G$ define the coupled task $T_i$.
- For each arc $(v_i, v_j)$ of graph $G$ define an arc $T_i \prec T_j$ of precedence constraints in the scheduling problem.
- $y = C_{max} = 2n$.

Let us assume that a balanced colouring of $G$ exists. Let $S_1, S_2, \ldots, S_l$ be subsets of $V(G)$ such that $\bigcup_{i=1}^{l} S_i = V(G)$, and all vertices that belong to $S_i$ are coloured with the $i$-th colour and $\forall_{1\le i\le l}\ |S_i| = q/l$. Sets $S_i$ are partially ordered because vertices of $G$ are partially ordered and $G$ contains all transitive arcs. Schedule sets $S_i$ in accordance to the partial order using the following algorithm:
***Algorithm 1***

**begin**
  $s := 0$
  **repeat**
    `Get a task` $T_j \in S_i$`.`
    `Let` $s$ `be the starting time of the task` $T_j$`.`
    $S_i := S_i \backslash T_j$
    $s := s + 1$
  **until** $S_i \neq \emptyset$
**end;**

The schedule generated in such way is shown in Fig. 2.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $T_{a1}$ | $T_{b1}$ | $T_{c1}$ | $T_{d1}$ | $T_{e1}$ | $T_{a2}$ | $T_{b2}$ | $T_{c2}$ | $T_{d2}$ | $T_{e2}$ |

Figure 2: Schedule of set $S_i = \{T_a, T_b, T_c, T_d, T_e\}$ where $h = 4$.

This procedure guarantees that no precedence constraint will be violated. The schedule contains no idle intervals, so its length is $y$.

On the other hand, let us assume that a schedule of length $y$ for the given instance of the coupled tasks problem exists. The schedule does not contain idle intervals, so it has to be a sequence of segments as shown in Fig. 2. Each segment contains

$h+1$ independent tasks, which means that the corresponding vertices in $G$ are also independent. So, the vertices from one segment can be coloured with the same colour, which means $G$ can be coloured with $n/(h+1)$ colours such that exactly $h+1$ vertices shares the same colour.

□

Now we can conclude.

**Theorem 1** Problem $1|(1,h,1) - coupled, strict\ prec,\ exact\ gap|C_{max}$ is NP-hard.

**Proof:** Follows immediately Lemmae 1 and 2.

□

## 4 Conclusions

In the paper, scheduling of coupled tasks has been considered. General precedence constraints resulted in a strong NP-hardness of the problem, even for unit processing times and equal gap lengths for all the tasks. The other cases, especially where precedence constraints are chain-like or tree-shaped are still open.

## References

[1] K. Baker, *Introduction to Sequencing and Scheduling*, J. Wiley, New York, 1974

[2] J. Błażewicz, W. Cellary, R. Słowiński and J. Węglarz, *Scheduling under Resource Constraints: Deterministic Models*, J. C. Baltzer, Basel, 1986.

[3] J. Błażewicz, K. H. Ecker, E. Pesch, G. Schmidt and J. Węglarz, *Scheduling Computer and Manufacturing Processes* (2nd edition), Springer, Berlin, New York, 2001.

[4] J. Błażewicz, K. H. Ecker, B. Plateu, D. Trystam, *Handbook of Parallel and Distributed Processing*, Springer, Berlin, New York, 2000.

[5] R. W. Conway, W. L. Maxwell and L. W. Miller, *Theory of Scheduling*, Addison-Wesley, Reading, Mass. 1967.

[6] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco, 1979.

[7] J. N. D. Gupta *Single facility scheduling with two operations per job and time-lags*, preprint, 1994.

[8] L. R. Graham, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling theory: a survey, *Ann. Discrete Math. 5*, 1979, 287-326.

[9] A. Farina, P. Neri, Multitarget interleaved tracking for phased radar array, *IEE Proceedings F* **27**, 1980, 312-318.

[10] W. Kern, W. M. Nawijn, *Scheduling multi-operations jobs with time lags on a single machine*, preprint, 1991.

[11] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnoy Kan, D. B. Shmoys, Sequencing and scheduling: algorithms and complexity, in S. C. Graves et al, *Handbooks in Operations Research and Management Science* **4**, North-Holland, Amsterdam, 1993

[12] D. J. Milojevic, B. M. Popovic, Improved algorithm for the interleaving of radar pulses, *IEE Proceedings F* **139**, 1992, 98-104..

[13] A. J. Orman, C. N. Potts, On the complexity of coupled tasks scheduling, *Discrete Applied Mathematics* **72**, 1997, 141-154.

[14] A. J. Orman, C. N. Potts, A. K. Shahani, A. R. Moore, Scheduling for the control of a multifunctional radar system, *European Journal of Operational Research* **90**, 1996, 13-25.

[15] M. Pinedo, *Scheduling: Theory, Algorithms and Systems*, Prentice Hall, Englewood Cliffs, N. J., 1995.

[16] A. H. G. Rinnooy Kan, *Machine Scheduling Problems: Classification, Complexity and Computations*, Martinus Nijhoff, The Hague, 1976.

[17] R. D. Shapiro, Scheduling coupled tasks, *Naval. Res. Logist. Quart.* **27**, 1980, 477-481.

[18] A. J. Orman, A. K. Shahani and A. R. Moore, Modelling for the control of a complex radar system, *Computers Ops Res.* **25**, 1998, 239-249.

[19] W. Yu, *The Two-machine Flow Shop Problem with Delays and the One-machine Total Tardiness Problem*, Ph. D. Thesis, Technische Universiteit Eindhoven, 1996.