W[2]-hardness of Constrained Profile Scheduling

Jiehua Chen[1] Danny Hermelin[2] Matthias Mnich[3]
Manuel Sorge[1]

[1]University of Warsaw, Warsaw, Poland
[1]{jiehua.chen2,manuel.sorge}@gmail.com
[2]Ben-Gurion University of the Negev, Beer Sheva, Israel
[2]hermelin@bgu.ac.il
[3]University of Bonn, Bonn, Germany
[3]mmnich@uni-bonn.de

March 4, 2019

In this paper, we correct some index typos of the proof for the following W[2]-hardness problem [1].

Constrained Profile Scheduling
Input: A set T of unit length tasks, a partial order on $\prec$ on T, a deadline $\Delta \in \mathbb{N}^+$, and a number $m \in \mathbb{N}$ of machines.
Question: Is there a mapping $f \colon T \to \{1, 2, \ldots, \Delta\}$ such that

1. for each two distinct tasks $t, t' \in T$ with $t \prec t'$ it holds that $f(t) < f(t')$, and
2. for each time $i \in \{1, 2, \ldots, \Delta\}$ it holds that $|f^{-1}(i)| \leq m$?

A mapping $f \colon T \to \{1, 2, \ldots, \Delta\}$ for a set T of task is called a *schedule*, and it is called a *feasible schedule* if it satisfies the two constraints above. Accordingly, we say that a task $i \in T$ is *scheduled* at time slot $f(i)$.

Theorem 1. *Parameterized by the number m of machines,* Constrained Profile Scheduling *is W[2]-hard.*

Proof. We reduce from the W[2]-hard Dominating Set problem, parameterized by the size of the dominating set. Let $(G = (V, E), k)$ be an instance of Dominating Set with $V = \{v_0, v_1, \ldots, v_{n-1}\}$ being the set of vertices, E being the set of edges, and k being the size of the desired dominating set. The vertices are indexed beginning with 0 to ease the presentation later on. Without loss of generality, we assume that $n \geq k \geq 1$. We will construct an instance of Constrained Profile Scheduling with $m = 2 \cdot k + 1$ machines and deadline $\Delta = k \cdot n \cdot (n^2 + 1) + 2 \cdot n$. We introduce a set T of tasks and define a directed acyclic graph on T, denoted as $H = (T, F)$. Our precedence constraints are the transitive closure of H. The tasks are divided into the following four parts.

The floor A. We introduce a set A of exactly Δ *floor tasks*, denoted as $a_1, a_2, \ldots, a_\Delta$. We add to F a path of length $\Delta - 1$ on these tasks. Formally, for each $i \in \{1, \ldots, \Delta - 1\}$ let $(a_i, a_{i+1}) \in F$.

Note that each feasible schedule must assign the floor tasks at distinct time slots, one after another. The floor tasks will be used to fix the vertex tasks (defined below) at certain time slots.

The vertex tasks B. We introduce a set B of $k \cdot n^2$ *vertex tasks*, that are "parallel" to the floor tasks of some specific form. We will have $k \cdot n$ vertex tasks for each vertex in the input vertex set. Formally, $B = \{b_{n+1+\alpha \cdot (n^2+1)+i \cdot n} \mid 0 \leq \alpha \leq k \cdot n - 1, 0 \leq i \leq n-1\}$. For each vertex task $b_x \in B$, we add to F the arcs (a_{x-1}, b_x) and (b_x, a_{x+1}).

Note that each vertex task b_x is scheduled at time step x. This creates bottlenecks in the processing capabilities of the machines at these time slots. These bottlenecks will be used to ensure that one of the vertices selected by the selector tasks (defined below) will be adjacent to the vertex corresponding to the vertex task.

The selector tasks C. For each $r \in \{1, \dots, k\}$, we introduce a set C_r of $\Delta - n + 1$ *selector tasks*, denoted as $c_{r,1}, c_{r,2}, \dots, c_{r,\Delta-n+1}$. They form the r^{th} *selector*. Then, we add to F another path of length $\Delta - n$ consisting of the vertices in C_r. Formally, for each r and i with $1 \leq r \leq k$ and $1 \leq i \leq \Delta - n$, we add to F the arc $(c_{r,i}, c_{r,i+1})$. We let $C = \bigcup_{1 \leq r \leq k} C_r$.

Intuitively, if all selector tasks in C_r would be scheduled at consecutive time slots without gaps, then, because there are exactly $\Delta - n + 1$ such vertex tasks, there would be n possible ways to schedule them. Each of these possibilities shall correspond to a vertex that is chosen into the dominating set by the r^{th} selector.

The non-edge tasks D. For each r with $1 \leq r \leq k$, we introduce a set D_r of $2 \cdot k \cdot n \cdot (\binom{n}{2} - |E|)$ vertices that represent the non-edges in G and are put "parallel" to the selector tasks in C_r of some specific form. Formally, $D_r = \{d_{r,n+1+\alpha \cdot (n^2+1)+i \cdot n-j} \mid 0 \leq \alpha \leq k \cdot n - 1, 0 \leq i, j \leq n - 1 \text{ with } \{v_i, v_j\} \notin E\}$. For each $d_{r,x} \in D_r$, we add to F two arcs $(c_{r,x-1}, d_{r,x})$ and $(d_{r,x}, c_{r,x+1})$. We let $D = \bigcup_{1 \leq r \leq k} D_r$.

Note that for each set S from the $2k+2$ sets $A, B, C_1, C_2, \dots, C_r, D_1, D_2, \dots, D_r$ and at each time slot z, there is at most one task from S that is scheduled at z. Our constructed instance for Constrained Profile Scheduling is as follows $I = (T, \prec, \Delta = k \cdot n \cdot (n^2+1), 2k+1)$, where T consists of the tasks from $A \cup B \cup C \cup D$, and the precedence constraints $\prec$ is the transitive closure of the acyclic graph (T, F) defined above. The construction can clearly be done in polynomial time. Now we show that the input graph G admits a dominating set of size k if and only if there is a schedule $f \colon T \to \{1, 2, \dots, \Delta\}$ for all the tasks in T, which satisfies the precedence constraints $\prec$ and uses at most $m = 2 \cdot k + 1$ machines at each time slot.

For the "only if" direction, assume that $V' = \{v_{q_1}, v_{q_2}, \dots, v_{q_k}\} \subseteq V$ is a dominating set of size k. We show that the following mapping $f \colon T \to \{1, 2, \dots, \Delta\}$ is a scheduling satisfying the conditions in the statement.

1. For each floor $a_i \in A$, let $f(a_i) = i$.
2. For each vertex task $b_i \in B$, let $f(b_i) = i$.
3. For each selector task $c_{r,x} \in C$, let $f(c_{r,x}) = x + q_r$.
4. For each non-edge task $d_{r,x} \in D$, let $f(d_{r,x}) = x + q_r$.

First, all tasks from $A \cup B$ are scheduled within deadline Δ. Second, for each $c_{r,x} \in C$ we have that $1 \leq x \leq \Delta - n + 1$ and $0 \leq q_r \leq n - 1$. By the definition of $f(c_{r,x})$, we thus have $1 \leq f(c_{r,x}) \leq \Delta$. Next, for each $d_{r,x} \in D$ we have that $0 \leq q_r \leq n - 1$ and, for some $i, j \in \{0, \dots, n-1\}$, $\alpha \in \{0, \dots, k \cdot n - 1\}$ that $x = n + 1 + \alpha \cdot (n^2+1) + i \cdot n - j$ and, thus,

$$2 \leq x \leq n + 1 + (k \cdot n - 1) \cdot (n^2 + 1) + (n-1) \cdot n = \Delta - 2 \cdot n.$$

By the definition of $f(d_{r,x})$ we have $2 \leq f(d_{r,x}) \leq \Delta - n - 1$. Further, it is easy to verify that the mapping f satisfies the precedence constraints.

It remains to show that at each time slot $z \in \{1, 2, \dots, \Delta\}$, we have $|f^{-1}(z)| \leq 2 \cdot k + 1$. If z is not of the form $n+1+\alpha \cdot (n^2+1)+i \cdot n$ for some $\alpha \in \{0, 1, \dots, k \cdot n-1\}$ and some $i \in \{0, 1, \dots, n-1\}$, then $b_z \notin B$ and thus one floor task, no vertex task, at most k selector tasks, and at most k non-edge tasks are scheduled at time z. Thus, for such z, it holds that $|f^{-1}(z)| \leq 2 \cdot k + 1$.

Otherwise, $z = n+1+\alpha \cdot (n^2+1)+i \cdot n$ for some $\alpha \in \{0, 1, \dots, k \cdot n-1\}$ and $i \in \{0, 1, \dots, n-1\}$. We distinguish between two cases, depending on whether v_i is in the dominating set V'.

Case 1: $v_i \in V'$**.** This means that there is an $r \in \{1, 2, \dots, k\}$ such that $i = q_r$. Observe that $d_{r,z-i} = d_{n+1+\alpha \cdot (n^2+1)+i \cdot n-i}$ does not exist in D. By the definition of f, at most $k-1$ non-edge tasks are scheduled at time z. Consequently, at most $2 \cdot k + 1$ tasks are scheduled at z, including one floor task, one vertex task, at most k selector tasks, and at most $k-1$ non-edge tasks.

Case 2: $v_i \notin V'$**.** This means that there is an $r \in \{1, 2, \dots, k\}$ such that $\{v_i, v_{q_r}\} \in E$, meaning that the task $d_{r,z-q_r}$ does not exist. Again, by the definition of f, at most $k-1$ non-edge tasks are scheduled at z.

Before we proceed with the back direction, we first present some useful observations.

Claim 1. *For each mapping $f \colon T \to \{1, 2, \dots, \Delta\}$ that satisfies the precedence constraints $\prec$, the following holds.*

1. *For each $i \in \{1, \dots, D\}$ and each floor task $a_i \in A$ we have $f(a_i) = i$, and if there is a vertex task $b_i \in B$, then $f(a_i) = f(b_i) = i$.*
2. *For each $r \in \{1, \dots, k\}$ and for each selector task $d_{r,x} \in D_r$, we have that $f(c_{r,x}) = f(d_{r,x})$.*
3. *For each $r \in \{1, \dots, k\}$ and for each selector task $c_{r,x} \in C_r$, we have that $x \leq f(c_{r,x}) \leq x + n - 1$.*

Proof. Since there are Δ floor tasks which form a path in the precedence constraint and since the deadline is Δ, each floor task must obtain a unique time slot. Thus, $f(a_i) = i$. By the precedence constraints of b_x, we can deduce that $f(b_i) = f(a_i) = i$. The second statement follows from the precedence constraints for the non-edge tasks. The first inequality of last statement follows from the fact that there are exactly $x - 1$ tasks in C_r that have to be scheduled before $c_{r,x}$ such that each of them obtain a distinct time slot. The last inequality of the last statement follows from the fact that there are $\Delta - n + 1 - x$ tasks that have to be scheduled after $c_{r,x}$, sequentially and within deadline Δ. (of Claim 1) $\diamond$

From the construction of the tasks' precedence constraints, we can deduce that there is a range of time slots where all selector tasks are executed consecutively:

Claim 2. *Let $f \colon T \to \{1, 2, \dots, \Delta\}$ be a scheduling for the tasks in T that satisfies the precedence constraints $\prec$. Then, there is a time slot $s = n + 1 + \delta \cdot (n^2 + 1)$ for some δ with $0 \leq \delta \leq k \cdot n - 1$ such that for each time slot z with $z \in \{s, s + 1, \dots, s + n^2\}$ and for each selector $r \in \{1, \dots, k\}$ it holds that $f^{-1}(z) \cap C_r \neq \emptyset$.*

Proof. For each selector $r \in \{1, 2, \dots, k\}$, let S_r denote the set of integers σ where at most n^2 tasks from C_r are assigned time slots between $n + 1 + \sigma \cdot (n^2 + 1)$ and $n + 1 + \sigma \cdot (n^2 + 1) + n^2$, that is, $S_r = \{\sigma \in \{0, 1, \dots, k \cdot n - 1\} \mid \exists z \in \{n + 1 + \sigma \cdot (n^2 + 1), n + 1 + \sigma \cdot (n^2 + 1) + 1, \dots, n + 1 + \sigma \cdot (n^2 + 1) + n^2\} \colon f^{-1}(z) \cap C_r = \emptyset\}$. Note that the sets of jobs scheduled for $\sigma, \tau \in S_r$, $\sigma \neq \tau$, are disjoint. Since all tasks in C_r occupy exactly $\Delta - n + 1$ time slots and the total number of time slots are Δ, it follows that $|S_r| \leq n - 1$. In total, $|\bigcup_{1 \leq r \leq k} S_r| \leq k \cdot (n - 1) < k \cdot n$. Now, consider an arbitrary index $\delta \in \{0, 1, \dots, k \cdot n - 1\} \setminus \left(\bigcup_{1 \leq r \leq k} S_r\right)$. By the definition of S_r, for each time slot in $\{n + 1 + \delta \cdot (n^2 + 1), n + 1 + \delta \cdot (n^2 + 1) + 1, \dots, n + 1 + \delta \cdot (n^2 + 1) + n^2\}$ it holds that $|f^{-1}(z) \cap C| = k$. (of Claim 2) $\diamond$

We continue to show the "if" direction: Let there be a feasible schedule $f\colon T \to \{1,2,\dots,\Delta\}$. By Claim 2, let $\delta \in \{0,1,\dots,n-1\} \setminus \left(\bigcup_{1\le r\le k} S_r\right)$ and $s = n+1+\delta\cdot(n^2+1)$ such that for each time slot $z \in \{s, s+1, \dots, s+n^2\}$ it holds that $|f^{-1}(z)\cap C| = k$. This means that we can find k tasks, denoted as $c_{1,s-q_1}, c_{2,s-q_2}, \dots, c_{k,s-q_k}$, such that

$$\forall t \in \{0,1,\dots,n^2\}\colon s+t = f(c_{1,s-q_1+t}) = f(c_{2,s-q_2+t}) = \dots = f(c_{k,s-q_k+t}). \tag{1}$$

By the third statement in Claim 1, we have that for each $r \in \{1,2,\dots,k\}$, it holds that $c_{r,s-q_r}$ has a time slot $s = f(c_{r,s-q_r})$ between $s-q_r$ and $s-q_r+n-1$, implying that $0 \le q_r \le n-1$. We claim that $V' = \{v_{q_1}, v_{q_2}, \dots, v_{q_k}\}$ is a dominating set for G. As already reasoned, for each $r \in \{1,2,\dots,k\}$, it holds that $q_r \in \{0,1,\dots,n-1\}$.

Now, for each vertex $v_i \in V$, we show that $v_i \in V'$ or v_i is adjacent to a dominating vertex from V'. To show this, consider the specific time slot $\hat{z} = s + i\cdot n = n+1+\delta\cdot(n^2+1)+i\cdot n$. Note that the vertex task $b_{\hat{z}}$ exists. By the first statement in Claim 1, we have that $f(a_{\hat{z}}) = f(b_{\hat{z}}) = \hat{z}$. We claim that for each $r \in \{1,\dots,k\}$ we have $f(c_{r,\hat{z}-q_r}) = \hat{z}$. To see this, note that $\hat{z} - q_r = s - q_r + i\cdot n$. By Equation 1, we deduce $f(c_{r,\hat{z}-q_r}) = f(c_{r,s-q_r+i\cdot n}) = s + i\cdot n$ which equals $\hat{z}$ by definition. Hence, there are exactly k selector tasks from C that are scheduled at time slot $\hat{z}$, namely $c_{1,\hat{z}-q_1}, c_{2,\hat{z}-q_2}, \dots, c_{r,\hat{z}-q_r}$. Consequently, there can be at most $k-1$ non-edge tasks that are scheduled at $\hat{z}$. However, by the third statement in Claim 1, each non-edge task of the form $d_{r,\hat{z}-q_r}$ would be scheduled at $f(c_{r,\hat{z}-q_r}) = \hat{z}$. This means that there exists a selector ℓ such that $d_{\ell,\hat{z}-q_\ell}$ does not exists. By the definition of $\hat{z}$ and the definition of the non-edge tasks, $d_{\ell,\hat{z}-q_\ell}$ does not exists if only if $i = q_\ell$ or $\{v_i, v_\ell\}$. In the first case v_i belongs to V', and in the second case v_i is adjacent to some vertex in V'. Hence, V' is a dominating set of G. ☐

References

[1] H. L. Bodlaender and M. R. Fellows. W[2]-hardness of precedence constrained k-processor scheduling. *Operations Research Letters*, 18(2):93–97, 1995. → p. 1.