W[2]-hardness of Precedence Constrained K-processor Scheduling

Hans L. Bodlaender * Michael R. Fellows †

Abstract

It is shown that the PRECEDENCE CONSTRAINED K-PROCESSOR SCHEDULING problem is $W[2]$-hard. This means that there does not exist a constant c, such that for all fixed K, the PRECEDENCE CONSTRAINED K-PROCESSOR SCHEDULING problem can be solved in $O(n^c)$ time, unless an unlikely collapse occurs in the parameterized complexity hierarchy introduced by Downey and Fellows (see [5]). That is, if the problem can be solved in polynomial time for each fixed K, then it is likely that the degree of the running time polynomial must increase as the number of processors K increases.

1 Introduction

The PRECEDENCE CONSTRAINED K-PROCESSOR SCHEDULING problem is a well studied problem. In this problem, we look for a schedule of a set of unit length tasks T on a set of K processors, that meets a given deadline D, and satisfies a given partial order on the set of tasks T. In practical situations the set of tasks will normally be much larger than the set of processors. Thus it

*Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB The Netherlands. The research of this author was partially supported by the ESPRIT II Basic Research Actions of the EC under Contract No. 3075 (project ALCOM). hansb@cs.ruu.nl

†Computer Science Department, University of Victoria, Victoria, British Columbia V8W 3P6, Canada. Research supported in part by the National Science and Engineering Council of Canada and by the United States National Science Foundation under grant MIP-8919312. mfellows@csr.uvic.ca

is an interesting problem is to look for efficient scheduling algorithms when the number of processors K is a fixed integer. So far, a polynomial time algorithm is known only for the case of $K = 2$ [7], and the question whether there exists a polynomial time algorithm for this problem for each fixed K is a famous open problem. (See e.g. [8], [OPEN 8].) If K is variable, then the problem is NP-hard. Many special cases have been investigated; see e.g. [9] for an overview.

Although it is often believed that 'polynomial time' is a synonym for 'practical', this is not always the case. Polynomial time algorithms with a running time of $\Theta(n^K)$ will be slow, even for very small values of K. This observation motivates the study of the structural complexity of parameterized problems: problems that have as part of their input a parameter, usually an integer in $\mathbf{N}^+$. For some parameterized problems that are solvable in polynomial time for a fixed parameter K, there exists a constant c, such that for all fixed K, there exists an algorithm for the problem with fixed parameter K, that runs in time $O(n^c)$. This (desirable) complexity behaviour is termed *fixed-parameter tractability*. For other parameterized problems, it seems that the degrees of the polynomials bounding the running times must depend on K. Well-known examples of the former include K-VERTEX COVER and K-MIN CUT LINEAR ARRANGEMENT (see [8] for the definitions). Each of these is solvable in linear time for each fixed K. Examples of the latter include K-DOMINATING SET and K-BANDWIDTH, for which the best known algorithms require time $\Omega(n^K)$ and are based on forms of exhaustive search. The difference between these two kinds of complexity behaviour is reminiscent of the contrast we often see between problems in P and problems which are NP-complete, with the latter often solvable (apparently) only by means of (exponential) exhaustive search.

Just as the theory of *NP*-completeness can be used to show that problems are unlikely to be solvable in polynomial time, the theory of fixed parameter complexity, introduced in [5], can be used to demonstrate the unlikelihood of fixed-parameter tractability. Some of the basic notions of this theory are reviewed in Section 2.

The main result of this paper is that the PRECEDENCE CONSTRAINED K-PROCESSOR SCHEDULING problem is hard for the complexity class $W[2]$. This means that it is most likely that if the problem is solvable in polynomial time for fixed K, then the problem exhibits the second type of behavior, i.e., that it is unlikely that there exists a c, such that for each fixed K, there exists an $O(n^c)$ algorithm for the problem. Namely, if such an algorithm would exist for PRECEDENCE CONSTRAINED K-PROCESSOR SCHEDULING, this would imply such algorithms for all problems in the parameterized complexity classes $W[1]$ and $W[2]$, including K-INDEPENDENT SET, K-CLIQUE, K-PERFECT CODE, K-SUBSET SUM, K-SUBSET PRODUCT, K-SQUARE TILING, and K-STEP HALTING PROBLEM FOR NONDETERMINISTIC TURING MACHINES [3, 4, 5, 6]. Although we do not solve the problem [OPEN 8] from [8], our result can be interpreted as bearing on the practical significance of this problem, showing that even if there is no particular K for which the problem is *NP*-complete, it is still likely to be computationally intractable for the fixed parameter values that are important in many applications.

2 Definitions

In this section we give some of the basic definitions from the theory of fixed parameter intractability. We also give the formal definition of the PRECEDENCE CONSTRAINED K-PROCESSOR SCHEDULING problem:

> PRECEDENCE CONSTRAINED K-PROCESSOR SCHEDULING
> **Instance:** Set T of unit length tasks, partial order $\prec$ on T, a deadline $D \in \mathbf{N}^+$, number of processors $K \in \mathbf{N}^+$.
> **Question:** Does there exist a mapping $f : T \to \{1, \ldots, D\}$, such that for all $t, t' \in T$: $t \prec t' \Rightarrow f(t) < f(t')$, and for all i, $1 \le i \le D$: $|f^{-1}(i)| \le K$?
> **Parameter:** K.

A *parameterized problem* is a set $L \subseteq \Sigma^* \times \Sigma^*$ where Σ is a fixed alphabet. For convenience, we consider that a parameterized problem L is a subset of $L \subseteq \Sigma^* \times \mathbf{N}^+$. For a parameterized problem L and $K \in \mathbf{N}^+$ we write L_K to denote the associated fixed-parameter problem $L_K = \{x | (x, K) \in L\}$. We say that a parameterized problem L is (uniformly) *fixed-parameter tractable* if there is a constant c and an algorithm Φ such that Φ decides if $(x, K) \in L$ in time $f(K)|x|^c$ where $f : \mathbf{N}^+ \to \mathbf{N}^+$ is an arbitrary function. Let A, B be parameterized problems. We say that A is (uniformly many:1) *reducible* to B if there is an algorithm Φ which transforms (x, K) into $(x', g(K))$ in time $f(K)|x|^c$, where $f, g : \mathbf{N}^+ \to \mathbf{N}^+$ are arbitrary functions and c is a constant independent of K, so that $(x, K) \in A$ if and only if $(x', g(K)) \in B$.

In [5], Downey and Fellows define complexity classes FPT, $W[1]$, $W[2]$, $\ldots$, $W[P]$, where FPT is the class of fixed-parameter tractable problems. The following containment relations hold:

$$FTP \subseteq W[1] \subseteq W[2] \subseteq \ldots \subseteq W[P]$$

Problems that are hard for $W[1]$ (and hence problems hard for any larger class) are believed not to be fixed-parameter tractable. However, showing that the W hierarchy is proper would be very hard, as this would imply $P \neq NP$. Thus a completeness theory for exploring the issue of fixed-parameter tractability is a reasonable way to proceed. It can be shown that if the W hierarchy collapses, then a strong *quantitative* form of the $P \neq NP$ conjecture fails [1].

A set of vertices $W \subseteq V$ is a *dominating set* of an undirected graph $G = (V, E)$, if for all $v \in V$, either $v \in W$ or v is adjacent to a vertex $w \in W$. The DOMINATING SET problem is the following:

> DOMINATING SET
> **Instance:** Undirected graph $G = (V, E)$, integer $K \in \mathbf{N}^+$.
> **Question:** Does G have a dominating set $W \subseteq V$ with $|W| \leq K$?
> **Parameter:** K.

Our main result relies on the following theorem from [5].

Theorem 1 DOMINATING SET *is complete for the class* $W[2]$.

3 Main Result

Theorem 2 PRECEDENCE CONSTRAINED K-PROCESSOR SCHEDULING *is* $W[2]$*-hard.*

Proof: We transform from DOMINATING SET. Let ($G = (V, E)$, k) be an instance to DOMINATING SET. Suppose $|V| = n$, and write $V = \{v_0, \ldots, v_{n-1}\}$.

Write $c = n^2 + 1$. Take $D = (k \cdot n) \cdot c + 2n$, and take $K = 2k + 1$.

We now define a directed acyclic graph $H = (W, F)$. H consists of the following parts:

The floor Take a path with length D: take vertices $\{a_1, \ldots, a_D\}$, and edges (a_i, a_{i+1}) for all i, $1 \leq i \leq D-1$.

The floor gadgets 'Parallel' to each floor vertex of the form $a_{n-1+\alpha \cdot c+in}$, $1 \leq i \leq n$, $0 \leq \alpha \leq kn-1$, we take a floor gadget vertex: take vertices $\{b_{n-1+\alpha \cdot c+in} \mid 1 \leq i \leq n,\ 0 \leq \alpha \leq kn-1\} = \mathcal{B}$, and add edges: (a_{i-1}, b_i) and (b_i, a_{i+1}) for all $b_i \in \mathcal{B}$.

The selector paths For each i, $1 \leq i \leq k$, we take a path of length $D-n+1$. This path will represent the ith vertex from a dominating set of G. Take vertices $\{c_{i,j} \mid 1 \leq i \leq k,\ 1 \leq j \leq D-n\}$, and edges $(c_{i,j}, c_{i,j+1})$ for all i, $1 \leq i \leq k$, j, $1 \leq j \leq D-n$.

The selector gadgets If $i \neq j$ and $(v_i, v_j) \notin E$, then we take a vertex, which is put 'parallel' to $c_{r,n-1+\alpha \cdot c+in-j}$, for all α, $1 \leq \alpha \leq k \cdot n$, r, $1 \leq r \leq k$. Take vertices $\{d_{r,n-1+\alpha \cdot c+in-j} \mid 1 \leq r \leq k,\ 1 \leq i \leq n,\ 1 \leq j \leq n,\ i \neq j,\ (v_i, v_j) \notin E,\ 1 \leq \alpha \leq kn\} = \mathcal{D}$, and for each vertex $d_{r,\beta} \in \mathcal{D}$, add edges $(c_{r,\beta-1}, d_{r,\beta})$ and $(d_{r,\beta}, c_{r,\beta+1})$.

Let $H = (W, F)$ be the directed acyclic graph (dag) resulting from this construction. Let $\prec \subseteq W \times W$ be the transitive closure of F, i.e, let $v \prec w$, if and only if there exists a path from v to w in H.

Claim 3 *Task set W with partial order $\prec$, deadline D, and number of processors K, is a yes-instance to* Precedence Constrained K-Processor Scheduling, *if and only if G has a dominating set of size at least k.*

Proof: $\Leftarrow$: Suppose $\{v_{\gamma_1}, \ldots, v_{\gamma_k}\} \subseteq V$ is a dominating set of size k of G. Consider the following schedule f of W:

$$\begin{aligned} a_i &= & i & \quad (1 \leq i \leq D) \\ b_i &= & i & \quad (b_i \in \mathcal{B}) \\ c_{i,j} &= & j + \gamma_i & \quad (1 \leq i \leq D-n) \\ d_{i,j} &= & j + \gamma_i & \quad (d_{i,j} \in \mathcal{D}) \end{aligned}$$

Clearly f satisfies the precedence constraints. To an integer i, not of the form $n-1+\alpha \cdot c+jn$ ($1 \leq j \leq n$, $1 \leq \alpha \leq kn$), one floor vertex, no floor gadget vertex, at most k selector path vertices, and at most k selector gadget vertices are mapped, so for such i, $|f^{-1}(i)| \leq 2k+1 = K$.

Look at i of the form $n-1+\alpha \cdot c + jn$ with $1 \leq j \leq n$, $1 \leq \alpha \leq kn$. As $\{v_{\gamma_1}, \ldots, v_{\gamma_k}\}$ is a dominating set of G, there are two cases:

Case 1: v_p is in the dominating set, i.e., $p = \gamma_q$, $1 \leq q \leq k$. As $d_{q,n-1+\alpha \cdot c+pn-p}$ does not exist in $\mathcal{D}$, at most $k-1$ selector gadget vertices are mapped to $i = n-1+\alpha \cdot n^2 + pn - p + \gamma_q$. The total number of vertices mapped to i hence is at most K. (The other vertices mapped to i are: at most one floor vertex, one floor gadget vertex, and k selector path vertices.)

Case 2: v_p is adjacent to vertex v_{γ_q}, $1 \leq q \leq k$. Now $d_{q,n-1+\alpha \cdot c+pn-\gamma_q}$ does not exist in $\mathcal{D}$, so again at most $k-1$ selector gadget vertices are mapped to i.

$\Rightarrow$: Suppose $f : W \rightarrow \{1, \ldots, D\}$ is a schedule, fulfilling the required properties. First, as the length of the floor path equals the deadline D, it follows that we have for all i, $1 \leq i \leq D$:

$$f(a_i) = i$$

For floor gadget vertices, only one possibility is now left:

$$f(b_i) = i$$

Call the interval $[n-1+(i-1)c+1, n-1+ic]$ the ith range ($1 \leq i \leq kn$). We say that the ith range is *polluted* by the jth selector path, when there exist an integer in this range to which no vertex on this jth selector path is mapped, i.e., when there exists an x, $n-1+(i-1)c+1 \leq x \leq n-1+ic$, with $f^{-1}(x) \cap \{c_{j,j'} \mid 1 \leq j' \leq D-n+1\} = \emptyset$. As each selector path has length $D-n+1$, it can pollute only $n-1$ ranges. The total number of polluted ranges hence is at most $kn-k$, so there is at least one range that is not polluted, say the δth range $[n-1+(\delta-1)c+1, n-1+\delta c]$. We now define numbers $\gamma_1, \ldots, \gamma_k$, such that

$$f(c_{i,n-1+(\delta-1)c+1-\gamma_i}) = n-1+(\delta-1)c+1$$

Note that by the discussion above, $\gamma_1, \ldots, \gamma_k$ are uniquely defined. It easily follows that for all selector path vertices, the following holds:

$$j \leq f(c_{i,j}) \leq j+n-1$$

So, $\{\gamma_1, \ldots, \gamma_k\} \subseteq \{0, \ldots, n-1\}$.

Now, we show that for all q, v_q belongs to the set $\{v_{\gamma_1}, \ldots, v_{\gamma_k}\}$, or is adjacent to a vertex in this set. As shorthand notation, we write $z = n-1+(\delta-1)c+qn$. Look at $X = f^{-1}(z)$. Note that the set X contains one floor vertex, one floor gadget vertex, and k selector path vertices. So, it can contain at most $k-1$ selector gadget vertices. So, there is an l, $1 \leq l \leq k$, such that X does not contain any vertex of the form $d_{l,\epsilon}$. We claim that $d_{l,z-\gamma_l}$ does not exist in $\mathcal{D}$: Note that $f(c_{l,z-\gamma_l-1}) = z-1$, $f(c_{l,z-\gamma_l+1}) = z+1$. So, $d_{l,z-\gamma_l}$ does not exist in $\mathcal{D}$, otherwise it would be mapped to z. As $d_{l,n-1+(\delta-1)c+qn-\gamma_l}$ does not exist in $\mathcal{D}$, we have that $\gamma_l = q$, or $(v_{\gamma_l}, v_q) \in E$. It follows that $\{v_{\gamma_1}, \ldots, v_{\gamma_k}\}$ is a dominating set of G. $\square$

The theorem follows directly by Claim 3, and Theorem 1. $\square$

4 Conclusions

The main result of this paper indicates that PRECEDENCE CONSTRAINED K-PROCESSOR SCHEDULING is unlikely to be fixed-parameter tractable. In practical terms, this means, that even if the problem were found to be polynomial-time solvable for fixed numbers of processors, the problem still is likely to be impractically hard for small values of K.

We feel that this result is a nice example of the use of a powerful and interesting new tool for the complexity analysis of practical problems.

Acknowledgement

We would like to thank Gene Lawler for stimulating our work on this problem by asking, some years ago, whether it might be fixed-parameter tractable by means of RS posets.

References

[1] K. Abrahamson, R. Downey and M. Fellows. Fixed-Parameter Intractability II. *Proceedings of the 10th Symposium on Theoretical Aspects of Computer Science (STACS)*, pp. 374-385, Springer-Verlag, Lecture Notes in Computer Science, 1993.

[2] H. Bodlaender, M. Fellows and M. Hallett. Beyond NP-completeness for problems of bounded width: hardness for the W hierarchy. To appear, *Proceedings of the 26th ACM Symposium on the Theory of Computing*, 1994.

[3] L. Cai, J. Chen, R. Downey and M. Fellows. The parameterized complexity of short computations and factorization. University of Victoria, Technical Report, Department of Computer Science, July, 1993.

[4] R. Downey, P. Evans and M. Fellows. Parameterized learning complexity. *Proc. Sixth ACM Workshop on Computational Learning Theory (COLT)*, pp. 51–57, ACM Press, 1993.

[5] R. Downey and M. Fellows. Fixed-parameter intractability (extended abstract). In *Proceedings of the Seventh Annual Conference on Structure in Complexity Theory*, pp. 36–49, IEEE Computer Society Press, Los Alamitos, CA, 1992. Final version to appear in *SIAM J. Comp.*

[6] R. Downey, M. Fellows, B. Kapron, M. Hallett and H.T. Wareham. The parameterized complexity of some problems in logic and linguistics. To appear in *Proceedings of Symposium on Logical Foundations of Computer Science (LFCS'94)*, Springer-Verlag, Lecture Notes in Computer Science, 1994.

[7] M. Fujii, T. Kasami, and K. Ninomiya. Optimal sequencing of two equivalent processors. *SIAM J. Appl. Math.*, 17 (1969), 784–789. Erratum. *SIAM J. Appl. Math.* 20 (1971), 141.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, New York, 1979.

[9] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. Sequencing and scheduling: Algorithms and complexity. In S. G. et al, editor, *Handbooks in OR & MS, Vol. 4*, pages 445–522. Elsevier Science Publishers, 1993.