ELSEVIER

Operations Research Letters 18 (1995) 93–97

operations research letters

$W[2]$-hardness of precedence constrained K-processor scheduling

Hans L. Bodlaender [a,1], Michael R. Fellows [b,*,2]

[a] Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
[b] Computer Science Department, University of Victoria, Victoria, British Columbia, Canada V8W 3P6

Received 1 May 1994; revised 1 April 1995

Abstract

It is shown that the Precedence Constrained K-Processor Scheduling problem is hard for the parameterized complexity class $W[2]$. 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. 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.

Keywords: Multiprocessor scheduling; Parameterized complexity; Partial orders

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 is interesting to consider the possibilities for efficient scheduling algorithms when the number of processors K is a fixed integer. Presently, a polynomial time algorithm is known only for the case of $K = 2$ [21], and the question of whether there exists a polynomial-time algorithm for each fixed K is a famous open problem (see e.g. Problem [OPEN 8] in [22]). If K is variable, then the problem is NP-hard. Many special cases have been investigated; see e.g. [24] for an overview.

The newly developed framework of parameterized computational complexity provides an important tool for the complexity analysis of problems for which important applications are served by a small range of parameter values. Examples of the use of this theory in the analysis of concrete problem complexities can be found in [1, 2, 5–9, 12, 13, 16–20, 23].

The central issue addressed by the theory is perhaps best illustrated by example. All of the following well-known problems are NP-complete, and all of them can be solved in polynomial time for any fixed parameter value K (for descriptions of the problems see [22]):

- K-Vertex Cover
- K-Dominating Set

* Corresponding author. E-mail: mfellows@csr.uvic.ca.

1 The research of this author was partially supported by the ESPRIT II Basic Research Actions of the EC under Contract No. 3075 (project ALCOM).

2 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.

SSDI 0167-6377(95)00031-3

- K-Min Cut Linear Arrangement
- K-Bandwidth

The NP-completeness of (all of) these problems is a determination that is insensitive to the following important qualitative difference in how the parameter contributes to overall problem complexity. Both K-Vertex Cover and K-Min Cut Linear Arrangement can be solved in *linear* time for any fixed K. If $K \leqslant 20$, this allows K-Vertex Cover to be solved in a practical manner for input of essentially any size (e.g. $n = 10^8$) [16]. This contrasts strikingly with the fact that the only known fixed-K algorithm for Dominating Set is simply the brute-force examination of all K-sets of vertices. The best known algorithm for K-Bandwidth similarly requires time $\Omega(n^K)$. Note that for $K = 6$ such an algorithm is essentially impractical even for $n = 100$.

In the framework of parameterized complexity theory, "good behaviour" is defined to be the existence of a constant c, such that for all fixed K, the problem can be solved in time $O(n^c)$. This is termed *fixed-parameter tractability*. In typical cases, the exponent constant c is small, just as with typical polynomial-time algorithms. It is important to note that the "good behaviour" toolkit of parameterized algorithm design techniques includes a number of powerful, general and distinctive methods, such as well-quasiordering, bounded treewidth and pathwidth methods [3, 4], and $f(K)$-bounded search trees and problem kernels [16]. A substantial number of well-known parameterized problems that are NP-complete (or harder) are now known to be fixed-parameter tractable by these various means [4, 16, 23].

We see that of the four NP-complete problems above, two are fixed-parameter tractable, and two apparently are not. The difference 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.

A proof that K-Dominating Set is not fixed-parameter tractable would immediately imply P $\neq$ NP. Thus it is reasonable to study the relative complexity of parameterized problems, by means of an appropriate notion of problem reduction. A framework of parameterized complexity classes natural for this program is described in the next section.

The main result of this paper is that the Precedence Constrained K-Processor Scheduling problem is hard for the parameterized complexity class $W[2]$, and is therefore unlikely to be fixed-parameter tractable. Fixed-parameter tractability for Precedence Constrained K-Processor Scheduling would imply fixed-parameter tractability for every parameterized problem in $W[2]$, including K-Dominating Set, K-Subset Sum, K-Subset Product and K-Length Permutation Group Factorization. It would also imply fixed-parameter tractability for all of the parameterized problems in $W[1] \subseteq W[2]$, including K-Independent Set, K-Clique, K-Square Tiling, K-Length Derivation in Context-Sensitive Grammars and the K-Step Halting Problem for Nondeterministic Turing Machines [12–14, 18]. (The last-named problem is perhaps intuitively the most resistant to any supposition of fixed-parameter tractability, due to its extremely generic and unstructured nature.)

Although we do not solve the problem [OPEN 8] from [22], 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 of the theory of fixed parameter intractability (for more details, see [14]). 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, $1 \leqslant i \leqslant D$: $|f^{-1}(i)| \leqslant K$?
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 $L_K = \{x \mid (x,K) \in L\}$. We say that a parameterized problem L is *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 *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 [14], 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:

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

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 $|W| \leqslant K$?
Parameter: K.

Our main result relies on the following theorem from [14].

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, $1 \leqslant i \leqslant D-1$.

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

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

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, $1 \leqslant \alpha \leqslant k \cdot n$, r, $1 \leqslant r \leqslant k$. Take vertices $\{d_{r,n-1+\alpha \cdot c+in-j} \mid 1 \leqslant r \leqslant k,\ 1 \leqslant i \leqslant n,\ 1 \leqslant j \leqslant n,\ i \neq j,\ (v_i, v_j) \notin E,\ 1 \leqslant \alpha \leqslant kn\} = \mathscr{D}$, and for each vertex $d_{r,\beta} \in \mathscr{D}$, 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:

$$f(a_i) = i \quad (1 \leqslant i \leqslant D),$$

$$f(b_i) = i \quad (b_i \in \mathscr{B}),$$

$$f(c_{i,j}) = j + \gamma_i \quad (1 \leqslant j \leqslant D-n),$$

$$f(d_{i,j}) = j + \gamma_i \quad (d_{i,j} \in \mathscr{D}).$$

Clearly f satisfies the precedence constraints. To an integer i, not of the form $n-1+\alpha\cdot c+jn$ $(1\leqslant j\leqslant n,$ $1\leqslant\alpha\leqslant kn)$, 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, $|f^{-1}(i)|\leqslant 2k+1=K$.

Look at i of the form $n-1+\alpha\cdot c+pn$ with $1\leqslant p\leqslant n$, $1\leqslant\alpha\leqslant kn$. 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., $p=\gamma_q$, $1\leqslant q\leqslant k$. As $d_{q,n-1+\alpha\cdot c+pn-p}$ does not exist in $\mathscr{D}$, 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}, $1\leqslant q\leqslant k$. Now $d_{q,n-1+\alpha\cdot c+pn-\gamma_q}$ does not exist in $\mathscr{D}$, 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, $1\leqslant i\leqslant D$,

$$f(a_i)=i.$$

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 $(1\leqslant i\leqslant kn)$. We say that the ith range is *polluted* by the jth selector path, when there exists an integer in this range to which no vertex on this jth selector path is mapped, i.e., when there exists an x, $n-1+(i-1)c+1\leqslant x\leqslant n-1+ic$, with $f^{-1}(x)\cap\{c_{j,j'}\mid 1\leqslant j'\leqslant D-n+1\}=\emptyset$. 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:

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

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, $1\leqslant l\leqslant k$, such that X does not contain any vertex of the form $d_{l,\varepsilon}$. We claim that $d_{l,z-\gamma_l}$ does not exist in $\mathscr{D}$: 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 $\mathscr{D}$, otherwise it would be mapped to z. As $d_{l,n-1+(\delta-1)c+qn-\gamma_l}$ does not exist in $\mathscr{D}$, 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. □

The theorem follows directly by Claim 3, and Theorem 1. □

We may also conclude from Theorem 2 that there is no fully polynomial-time approximation scheme (see [22]) to compute an approximation to the minimum number of processors required to meet an instance deadline, unless the W hierarchy collapses. This follows by a result of Cai and Chen [10] showing that minimization problems with a fully polynomial-time approximation scheme are necessarily fixed-parameter tractable.

Acknowledgement and dedication

We dedicate this paper to the memory of Gene Lawler, who stimulated our work by asking, some years ago, whether well-quasiordering techniques might apply to this problem.

References

[1] K. Abrahamson, R. Downey and M. Fellows, "Fixed-parameter intractability II", *Proc. 10th Symp. on Theoretical Aspects of Computer Science (STACS)*, Lecture Notes in Computer Science, Springer, Berlin, 1993, pp. 374–385.

[2] K. Abrahamson, R. Downey and M. Fellows, "Fixed-parameter tractability and completeness IV: on completeness

for W[P] and PSPACE analogs", to appear in *Ann. Pure Appl. Logic*.

[3] K. Abrahamson and M. Fellows, "Finite automata, bounded treewidth and well-quasiordering", in: N. Robertson and P. Seymour (eds.), *Graph Structure Theory: Proc. Joint Summer Res. Conf. on Graph Minors*, Seattle, June 1991, Contemporary Mathematics, Vol. 147, Amer. Mathematical Soc., Providence, RI, 1993, pp. 539–564.

[4] H. Bodlaender, "A tourist guide through treewidth", Technical Report RUU-CS-92-12, Utrecht University, 1992.

[5] H. Bodlaender, R. Downey, M. Fellows, M. Hallett and H. Wareham, "Parameterized complexity analysis in computational biology", to appear in *Comput. Appl. Biosci.*

[6] H. Bodlaender, R. Downey, M. Fellows and H. Wareham, "The parameterized complexity of sequence alignment and consensus", in: *Proc. 5th Symp. on Combinatorial Pattern Matching (CPM)*, Lecture Notes in Computer Science, Vol. 807, Springer, Berlin, 1994, pp. 15–30.

[7] H. Bodlaender, R. Downey, M. Fellows and H. Wareham, "The parameterized complexity of the longest common subsequence problem", to appear in *Theoret. Comput. Sci.*

[8] H. Bodlaender and J. Engelfriet, "Domino treewidth", Technical Report UU-CS-1994-11, Department of Computer Science, Utrecht University, Utrecht, Netherlands, 1994.

[9] H. Bodlaender, M. Fellows and M. Hallett, "Beyond NP-completeness for problems of bounded width: hardness for the *W* hierarchy", in: *Proc. ACM Symp. on the Theory of Computing (STOC)*, 1994, pp. 449–458.

[10] L. Cai and J. Chen, "Fixed-parameter tractability and approximability of *NP*-hard optimization problems", in: *Proc. 2nd Israel Symp. on Theory of Computing and Systems*, 1993, pp. 118–126.

[11] L. Cai, J. Chen, R. Downey and M. Fellows, "On the structure of parameterized problems in NP", in: *Proc. 11th Symp. on Theoretical Aspects of Computer Science (STACS'94)*, Lecture Notes in Computer Science, Vol. 775, Springer, Berlin, 1994, pp. 509–520.

[12] L. Cai, J. Chen, R. Downey and M. Fellows, "The parameterized complexity of short computation and factorization", to appear in *Arch. Math. Logic*.

[13] R. Downey, P. Evans and M. Fellows, "Parameterized learning complexity", in: *Proc. 6th ACM Workshop on Computational Learning Theory (COLT)*, ACM, New York, 1993, pp. 51–57.

[14] R. Downey and M. Fellows, "Fixed-parameter intractability" (extended abstract), in: *Proc. 7th Ann. Conf. on Structure in Complexity Theory*, IEEE Computer Soc. Press, Los Alamitos, CA, 1992, pp. 36–49; final version to appear in *SIAM J. Comput.*

[15] R. Downey and M. Fellows, "Fixed-parameter tractability and completeness III: some structural aspects of the *W*-hierarchy", in: K. Ambos-Spies, S. Homer and U. Schöning (eds.), *Proc. 1992 Dagstuhl Workshop on Structural Complexity, Complexity Theory: Current Research*, Cambridge Univ. Press, Cambridge, 1993, pp. 191–226.

[16] R. Downey and M. Fellows, "Parameterized computational feasibility", in: P. Clote and J. Remmel (eds.), *Proc. 2nd Cornell Workshop on Feasible Mathematics, Feasible Mathematics II*, Birkhauser, Boston, 1995, pp. 219–244.

[17] R. Downey and M. Fellows, "Fixed-parameter tractability and completeness II: completeness for W[1]", *Theoret. Comput. Sci. A* **141**, 109–131 (1995).

[18] R. Downey, M. Fellows, B. Kapron, M. Hallett and H.T. Wareham, "The parameterized complexity of some problems in logic and linguistics", in: *Proc. Symp. on Logical Foundations of Computer Science (LFCS)*, Lecture Notes in Computer Science, Vol. 813, Springer, Berlin, 1994, pp. 89–100.

[19] M. Fellows, M. Hallett and H. Wareham, "DNA physical mapping: three ways difficult", in: *Algorithms - ESA '93 (Proc. 1st European Symp. on Algorithms)*, Lecture Notes in Computer Science Vol. 726, Springer, Berlin, 1993, pp. 157–168.

[20] M. Fellows and N. Koblitz, "Fixed-parameter complexity and cryptography", in: *Proc. 10th Internat. Symp. on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC'93)*, Lecture Notes in Computer Science, Vol. 673, Springer, Berlin, 1993, pp. 121–131.

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

[22] 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.

[23] H. Kaplan, R. Shamir and R. Tarjan, "Tractablity of parameterized completion problems on chordal and interval graphs", in: *Proc. IEEE Symp. on Foundations of Computer Science (FOCS)*, 1994.

[24] 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. (eds.), *Handbooks in OR & MS, Vol. 4*, Elsevier, Amsterdam, 1993, pp. 445–522.