A Space–Time Tradeoff for Permutation Problems*

Mikko Koivisto† Pekka Parviainen‡

Abstract

Many combinatorial problems—such as the traveling salesman, feedback arcset, cutwidth, and treewidth problem—can be formulated as finding a feasible permutation of n elements. Typically, such problems can be solved by dynamic programming in time and space $O^*(2^n)$, by divide and conquer in time $O^*(4^n)$ and polynomial space, or by a combination of the two in time $O^*(4^n 2^{-s})$ and space $O^*(2^s)$ for $s = n, n/2, n/4, \ldots$. Here, we show that one can improve the tradeoff to time $O^*(T^n)$ and space $O^*(S^n)$ with $TS < 4$ at any $\sqrt{2} < S < 2$. The idea is to find a small family of "thin" partial orders on the n elements such that every linear order is an extension of one member of the family. Our construction is optimal within a natural class of partial order families.

1 Introduction

Sequencing or *permutation problems* ask for a permutation on an n-element set so as to minimize a given cost function. Classical examples include the traveling salesman problem (TSP), the feedback arcset problem, and the treewidth problem; the present work is partly motivated by a generalization of the feedback arcset problem, the task of finding an optimal Bayesian network, which has recently attracted considerable interest in artificial intelligence and machine learning research [11, 14, 15, 16, 19]. Common to all the mentioned problems is that the cost of a permutation $\sigma_1\sigma_2\cdots\sigma_n$ decomposes into n local terms, the jth term depending only on the *sequence* $\sigma_{j-d+1}\sigma_{j-d+2}\cdots\sigma_j$ and, possibly, the *set* of the remaining $j-d$ preceding elements, $\{\sigma_1, \sigma_2, \ldots, \sigma_{j-d}\}$, for some constant $d \geq 0$. We call these *permutation problems of degree d*; TSP, for instance, is of degree 2, the jth term being the distance from city σ_{j-1} to city σ_j.

Thanks to Bellman [2], Held and Karp [10], and others [1, 12], we know that permutation problems can be solved by dynamic programming across the subsets of the n elements in time and space $O^*(2^n)$; throughout the paper O^* will suppress a factor polynomial in n. While there is an obvious interest in seeking faster algorithms, it is particularly the huge storage requirement of dynamic programming that determines the feasibility limit in practice. Thus, from the point of view of practical applications, the main concern is reducing the space requirement.

Some permutation problems are known to admit algorithms that run in time and space $O^*(C^n)$ with some $C < 2$. Recent examples include at least the treewidth problem [6, 8], the pathwidth problem [8], and TSP in bounded degree graphs [4, 7]. For so-called precedence constrained sequencing problems such improvements to dynamic programming were found already three decades ago: Schrage and Baker [18] introduced a labeling scheme for the constraints to reduce the number of states in dynamic programming; Lawler [13] provided a streamlined implementation that allows to bound the time and space requirement by the number of ideals of the precedence structure represented by a partial order.

But, in general, one may need to trade time for space. For instance, if given only *polynomial space*, it is not known if one can solve TSP in time $O^*(2^n)$. In fact, the best bound known is $O^*(4^n)$, achieved by a divide and conquer algorithm that considers all partitions of the n cities into two sets of about equal sizes, solves the TSP on the corresponding two subinstances recursively, and finally glues the two solutions and selects an optimum over all the $\binom{n}{\lfloor n/2 \rfloor}$ partitions; while this TSP algorithm is by Björklund and Husfeldt [3], the divide and conquer technique itself is well known, e.g., due to Savitch [17], Gurevich and Shelah [9], and others [6, 21]. Although algorithms faster than that, possibly requiring *moderately exponential space*, should be much more valuable in practice, there seem to be little prior work on interpolating between the two extremes of the space complexity. (We note that for partitioning problems Björklund et al. [5] devise such tradeoffs using the principle of inclusion and exclusion.)

What is, however, easily observed is that applying divide and conquer until the subproblems are of size s and then switching over to dynamic programming requires time $O^*(4^n 2^{-s})$ and space $O(2^s)$, for any $s = n/2, n/4, n/8, \ldots$. Yet, this scheme falls short if more

*Supported by the Academy of Finland, Grant 125637.

†Helsinki Institute for Information Technology HIIT, University of Helsinki. Email: mikko.koivisto@cs.helsinki.fi.

‡Helsinki Institute for Information Technology HIIT, University of Helsinki. Email: pekka.parviainen@cs.helsinki.fi.

space, say $O^*(2^{4/5n})$, is available. This is unfortunate, for reducing the space complexity by "only" a few orders of magnitude, with about an equivalent increase in the runtime, is what would make an algorithm feasible in practice, given the typical processing speed and memory size of modern computers.

Motivated by these concerns, which are particularly relevant in the Bayesian network application, we recently introduced a different approach, called the *pairwise scheme* [15]. The idea is to choose k pairs of elements (arbitrarily) and fix the mutual order of the elements within each pair, that is, a partial order on the n elements. For each of the 2^k possible partial orders, the fraction of permutations compatible with it can be scanned through by dynamic programming. In total, this takes time $O^*((3/2)^k 2^n)$ and space $O^*((3/4)^k 2^n)$ for any $k = 0, 1, \ldots, n/2$. This gives a reasonable scheme in moderately exponential space, $O^*(S^n)$ with $\sqrt{2} \leq S \leq 2$.

Here, we address some intriguing questions that remain. First, can one achieve the efficiency of the divide and conquer scheme, that is, time $O^*(T^n)$ and $O^*(S^n)$ with $TS \leq 4$, for any $\sqrt{2} \leq S \leq 2$? Note that the pairwise scheme gives $TS > 4$ for $S < 2$. Second, are there some T and S with $TS < 4$ such that permutation problems can be solved in time $O^*(T^n)$ and space $O^*(S^n)$? Cases $S = 1$ and $S = 2$ are well-known open problems, and it is tempting to conjecture that also in between the value 4 gives a lower bound. However, we will refute this conjecture by answering both questions in the affirmative.

1.1 The problem and main results. For a unified treatment of various permutation problems, we abstract the notions of *minimizing* over permutations and *decomposing* the cost of a permutation into local terms by letting the costs take values in some semiring R with addition $\oplus$ and multiplication $\odot$. Thus, the cost $f(\sigma)$ of a permutation σ, that is, a sequence $\sigma_1\sigma_2\cdots\sigma_n$ of n different elements from $N = \{1, 2, \ldots, n\}$, decomposes into a product of n local costs:

$$f(\sigma) = \bigodot_{j=1}^{n} f_j\big(\{\sigma_1, \sigma_2, \ldots, \sigma_j\}, \sigma_{j-d+1}\cdots\sigma_{j-1}\sigma_j\big)\ ,$$

and the task is compute the sum $\bigoplus_\sigma f(\sigma)$ over all permutations σ. Here, if $d > j$ we read $\sigma_{j-d+1}\cdots\sigma_{j-1}\sigma_j$ as $\sigma_1\cdots\sigma_{j-1}\sigma_j$, and if $d = 0$ the sequence is void. We call any problem of this form, with the f_j specified by the problem input, a *permutation problem of degree* d, or of *bounded degree* when d is some constant.

The traveling salesman problem, for example, is a permutation problem of degree 2 in the min–sum semiring of reals, with $f_1(A, x) = 0$, and $f_j(A, x, y)$, for $j > 1$, equaling the weight of edge xy in an input graph with vertex set N, indifferent of A. Strictly speaking, this formulation corresponds to the problem of computing the minimum total weight over Hamiltonian *paths*, not cycles; one can fix this by minor modifications.

The feedback arcset problem is a permutation problem of degree 1 in the min–sum semiring, with $f_j(A, x)$ equaling the number (or total weight) of edges from $A \setminus \{x\}$ to x in the input graph.

For yet another example, the cutwidth problem is a permutation problem of degree 0 in the min–max semiring, with $f_j(A)$ equaling the number of edges with one endpoint in A and the other in $N \setminus A$, where N is the vertex set of the input graph.

Finally, the treewidth problem is a permutation problem of degree 1 in the min–max semiring, with $f_j(A, x)$ equaling the number of vertices in $N \setminus A$ that have a neighbor in the unique component of the induced subgraph $G[A]$ containing x, where G is the input graph with vertex set N; see, e.g., Bodlaender et al. [6].

As the algorithms we consider operate on semirings, we take the time requirement of an algorithm as the total number of semiring additions and multiplications it performs, and the space requirement as the maximum number of semiring elements that need to be stored at any point during the execution of the algorithm. Here it is reasonable to assume that the (black-box) functions f_j can be evaluated in time and space polynomial in n.

While we are interested in schemes that yield a good time bound at any space bound of choice, we find it convenient to gauge the efficiency of an algorithm by a single, best-case number: we define the *time–space product of an algorithm* as the infimum of the product TS such that the algorithm (with worst case input) runs in time $O^*(T^n)$ and space $O^*(S^n)$. We further define the *time–space product of a permutation problem* as the infimum of the time–space product over all algorithms solving the problem; as above, T and S will refer to the time and space contributions also in the sequel.

THEOREM 1.1. *The time–space product of any permutation problem of bounded degree is less than* 3.93.

We prove this by a combinatorial construction that generalizes and improves upon the pairwise scheme [15]. We find it handy to view a permutation $\sigma = \sigma_1\sigma_2\cdots\sigma_n$ as a linear order,[1] with $\sigma_i\sigma_j \in \sigma$ if and only if $i \leq j$.

[1] A *partial order* P on *baseset* M is a subset of $M \times M$ such that for all $x, y, z \in M$ it holds that $xx \in P$ (reflexive), $xy \in P$ and $yx \in P$ implies $y = x$ (antisymmetry), and $xy \in P$ and $yz \in P$ implies $xz \in P$ (transitivity); P is a *linear order* (or, total order) if, in addition, $xy \in P$ or $yx \in P$ (comparability). Another partial order Q on M is an *extension* of P if $P \subseteq Q$. Note that a partial order fully specifies its baseset.

We make use of Lawler's observation (see the references [18, 13, 20] and Section 2) that the sum of $f(\sigma)$ over the linear extensions σ of a fixed partial order P can be computed by dynamic programming in time and space proportional to the number of ideals of P; an *ideal* of P is a set of elements I such that if $y \in I$ and $xy \in P$, then $x \in I$; we denote by $\mathcal{I}(P)$ the set of ideals of P. In our setting no particular partial order is given, but the challenge is to find a "small" family of "thin" partial orders such their linear extensions together *exactly cover* the $n!$ linear orders on the ground set N, that is, any linear order on N is an extension of exactly one member of the family. This motivates the definition of the *time–space product of a family $\mathcal{P}$ of partial orders* on N as the nth root of the product

$$\theta(\mathcal{P}) = \left(\sum_{P \in \mathcal{P}} |\mathcal{I}(P)| \right) \left(\max_{P \in \mathcal{P}} |\mathcal{I}(P)| \right) .$$

To prove Theorem 1.1, we will construct a sequence of families $\mathcal{P}_n$ such that $\theta(\mathcal{P}_n)^{1/n} < 3.93$ for any sufficiently large n.

In the present pursuit, we will study a subclass of series–parallel partial orders (see, e.g., Steiner [20] and references therein), namely, parallel compositions of bucket orders; we postpone formal definitions of such partial orders and the associated families to later sections. This restriction not only suffices for proving the bound in Theorem 1.1, but also enables showing that the bound is the best one can achieve with such partial orders. Curiously enough, the optimum is achieved with partial orders that are parallel compositions of $n/26$ bucket orders of type $13 * 13$, that is, 13 elements in the first bucket and 13 in the second. More generally, we may take k such bucket orders (leaving the remaining $n - 26k$ elements unordered) and obtain the following tradeoff in the space complexity range $1.452 < S \leq 2$.

Theorem 1.2. *Let k be an integer at most $n/26$. Then any permutation problem of bounded degree can be solved in time $O^*(\alpha^k 2^n)$ and space $O^*(\beta^k 2^n)$, with*

$$\begin{aligned} \alpha &= \binom{26}{13}(2^{14}-1)/2^{26} < 2.54 \times 10^3 , \\ \beta &= (2^{14}-1)/2^{26} < 2.45 \times 10^{-4} . \end{aligned}$$

With $k = \lfloor n/26 \rfloor$, Theorem 1.2 implies Theorem 1.1.

The range where a time–space product less than 4 is achieved can be extended to $\sqrt{2} < S < 2$ by replacing 13 above by larger numbers, $14, 15, \ldots$; we omit detailed calculations and refer to Figure 1, which shows a selection of different space–time tradeoff schemes. The $1 * 1$ scheme is the aforementioned pairwise scheme [15], while the naïve $m * (n-m)$ scheme, $m \geq n/2$, tries out

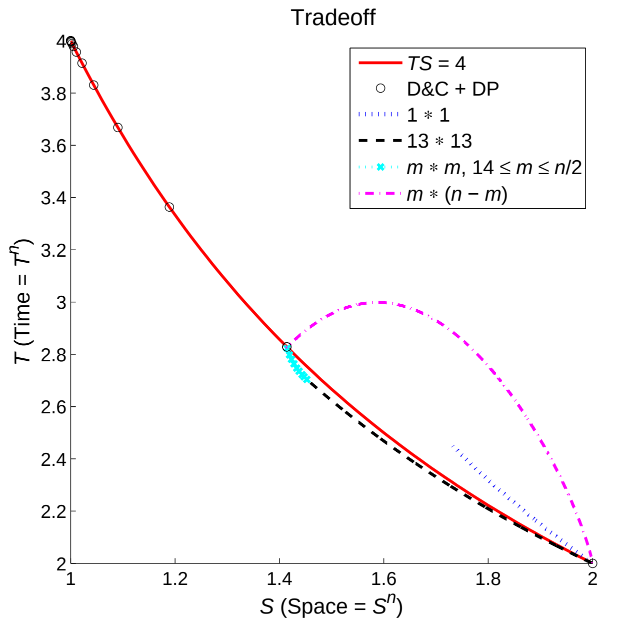

Figure 1: Space–time tradeoff schemes for permutation problems. The time requirement $O^*(T^n)$ is shown as a function of the space requirement $O^*(S^n)$, for $1 \leq S \leq 2$.

every partition of the n elements into two sets of sizes m and $n - m$, for each solving the constrained problem in time and space $O^*(2^m)$.

Our complexity bounds readily apply to the mentioned, well-known permutation problems. The bounds are the best we know of, except for the treewidth problem, for which Fomin and Villanger [8] have presented an algorithm running in time and space $O(1.7549^n)$ and another algorithm running in time $O(2.6151^n)$ and polynomial space. Thus, for the treewidth problem, our bounds become interesting (i.e., not dominated by either of the two) when $1.5048 < S < 1.7549$.

2 Dynamic programming over partial orders

We begin with the basic dynamic programming algorithm for permutation problems. For simplicity, we consider permutation problems of degree 2; we trust the reader can generalize this to any degree d. For any $A \subseteq N$ and $x \in A$, define $g(A, x)$ as the sum of

$$\bigodot_{j=1}^{|A|} f_j\big(\{\sigma_1, \sigma_2, \ldots, \sigma_j\}, \sigma_{j-1}\sigma_j\big) ,$$

over all permutations $\sigma_1\sigma_2\cdots\sigma_{|A|}$ of the elements in A with $\sigma_{|A|} = x$. Note that the sum of $f(\sigma)$ over all permutations σ of the elements in N equals $\bigoplus_{x \in N} g(N, x)$. Because multiplication distributes over addition in a semiring, we have the recurrence

$$\begin{aligned} g(\{x\},x) &= f_1(\{x\},x)\ , \\ g(A,x) &= \bigoplus_{y\in A\setminus\{x\}} g(A\setminus\{x\},y)\odot f_{|A|}(A,yx) \\ &\quad \text{for } A\subseteq N \text{ and } |A|\geq 2\ . \end{aligned}$$

Thus, a straightforward dynamic programming algorithm computes $g(N,x)$ for all $x \in N$, and hence the sum of $f(\sigma)$ over all permutations on N, in time and space $O^*(2^n)$.

We then generalize the above algorithm to compute the sum of the cost over all linear extensions of a given partial order P on N. For $A \subseteq N$ denote by $P[A]$ the induced partial order $P \cap (A \times A)$. For any ideal $A \in \mathcal{I}(P)$ and element $x \in A$, define $g_P(A,x)$ as the sum of

$$\bigodot_{j=1}^{|A|} f_j\big(\{\sigma_1,\sigma_2,\ldots,\sigma_j\},\sigma_{j-1}\sigma_j\big)\ ,$$

over all linear extensions σ of $P[A]$ with $\sigma_{|A|} = x$. Note that if P is the trivial order $\{xx : x \in N\}$, then g_P equals the above defined g.

Consider $g_P(A,x)$. If $A\setminus\{x\}$ is not an ideal of P, then $g_P(A,x) = 0$, since the sum is empty. Suppose therefore that $A\setminus\{x\}$ is an ideal of P. Then, any linear extension σ of $P[A]$ with $\sigma_{|A|} = x$ determines a linear extension $\sigma' = \sigma[A\setminus\{x\}]$ of $P[A\setminus\{x\}]$ with some $y = \sigma'_{|A|-1}$. Thus, the recurrence takes the form

$$\begin{aligned} g_P(\{x\},x) &= f_1(\{x\},x) \\ &\quad \text{for } \{x\}\in\mathcal{I}(P)\ , \\ g_P(A,x) &= \bigoplus_{y\in A\setminus\{x\}} g_P(A\setminus\{x\},y)\odot f_{|A|}(A,yx) \\ &\quad \text{for } A, A\setminus\{x\}\in\mathcal{I}(P) \text{ and } |A|\geq 2\ . \end{aligned}$$

That is, the formulas are the same as in the basic version but applied only for ideals A of P where $x \in A$ is a maximal element. Again, a straightforward dynamic programming algorithm shows the following.

Proposition 2.1. *The sum of the costs $f(\sigma)$ over all linear extensions σ of a given partial order P on N can be computed in time and space $O^*(|\mathcal{I}(P)|)$.*

To extend the sum over all linear orders, we generally need to consider more than one partial order. Recall that a family of partial orders $\mathcal{P}$ *exact covers* the linear orders on N if every linear order on N is an extension of exactly one partial order in $\mathcal{P}$. Given such an exact cover, the sum over linear orders can be computed by simply computing the sums over the linear extensions of each partial order in $\mathcal{P}$ separately, and finally taking the sum of the $|\mathcal{P}|$ results. Thus, by Proposition 2.1, we have the following.

Proposition 2.2. *Let $\mathcal{P}$ be a family a partial orders that exactly covers the linear orders on N. Then, given $\mathcal{P}$, the sum of the costs $f(L)$ over all linear orders on N can be computed in time $O^*\big(\sum_{P\in\mathcal{P}} |\mathcal{I}(P)|\big)$ and space $O^*\big(\max_{P\in\mathcal{P}} |\mathcal{I}(P)|\big)$.*

We summarize in terms of the time–space product: Let $(\mathcal{P}_n)$ be a sequence of partial order families such that each $\mathcal{P}_n$ exactly covers the linear orders on $\{1,2,\ldots,n\}$. Then the time–space product of the bounded-degree permutation problem is at most $\lim_n \theta(\mathcal{P}_n)^{1/n}$ (supposing the limit exists).

3 Parallel Bucket Orders: Upper Bound

We will consider parallel compositions of bucket orders, defined as follows. A partial order P is the *parallel composition* of partial orders $P_1, P_2, \ldots, P_k$ if the P_i are pairwise disjoint and their union is P, that is, $\{P_1, P_2, \ldots, P_k\}$ is a partition of P; given P, the partition becomes unique if each *component* P_i is required to be *connected*, that is, P_i does not further partition into two nonempty parts. A partial order B on baseset M is a *bucket order* if M can be partitioned into nonempty sets $B_1, B_2, \ldots, B_\ell$, called *buckets*, such that $xy \in B$ if and only if $x = y$ or $x \in B_i$ and $y \in B_j$ for some $i < j$; the bucket sequence $B_1B_2\cdots B_\ell$ is unique; the bucket order is said to be of *length* ℓ and *type* $|B_1| * |B_2| * \cdots * |B_\ell|$. For example, if $B_1 = \{1,3\}$ and $B_2 = \{2\}$, then the bucket order $B_1B_2 = \{11, 22, 33, 12, 32\}$ on $\{1,2,3\}$ is of length 2 and type $2 * 1$.

As already mentioned, parallel compositions of bucket orders belong to the class of series–parallel partial orders. Simple rules are known for calculating the number of ideals of a series–parallel partial order; see, e.g., Equations 3.1 and 3.2 in Steiner [20] and references therein. The following two lemmas are mere applications of these rules.

Lemma 3.1. *The number of ideals of a bucket order $B = B_1B_2\cdots B_\ell$ is given by $|\mathcal{I}(B)| = 1 - \ell + 2^{|B_1|} + 2^{|B_2|} + \cdots + 2^{|B_\ell|}$.*

Lemma 3.2. *Let P be the parallel composition partial orders $P_1, P_2, \ldots, P_k$. Then the number of ideals of P is given by $|\mathcal{I}(P)| = |\mathcal{I}(P_1)|\,|\mathcal{I}(P_2)|\cdots|\mathcal{I}(P_k)|$.*

To define a partial order family that exactly covers the linear orders on the ground set N, we introduce a notion of reordering. We say that two bucket orders are *reorderings* of each other if they have the same baseset and they are of the same type (and length). For example, the bucket order $\{1,3\}\{2\}$ is a reordering of $\{1,2\}\{3\}$ but not of $\{2\}\{1,3\}$. Further, we say that two parallel composition of bucket orders are *reorderings* of each other if their connected components can be labeled as $P_1, P_2, \ldots, P_k$ and $Q_1, Q_2, \ldots, Q_k$ such that P_i is a reordering of Q_i for all i. If P is a parallel composition of bucket orders, we denote by $\mathcal{P}(P)$ the family of partial orders that are reorderings of P. We call $\mathcal{P}(P)$ the *equivalence class* of P (w.r.t. the reordering relation).

PROPOSITION 3.1. *Let P be a parallel composition of bucket orders. Then $\mathcal{P}(P)$ exactly covers the linear orders on the baseset of P.*

Proof. Let $P_1, P_2, \ldots, P_k$ be the connected components of P with basesets $N_1, N_2, \ldots, N_k$, respectively. Let $\sigma = \sigma_1\sigma_2 \ldots, \sigma_n$ be a linear order on the baseset of P. It suffices to show that there is a unique partial order Q equivalent to P such that σ is an extension of Q.

For each $i = 1, 2, \ldots, k$, we construct a bucket order Q_i on N_i as follows. Let $m_1 * m_2 * \cdots * m_\ell$ be the type of P_i. For $j = 1, 2, \ldots, \ell$ denote $s_j = m_1 + m_2 + \cdots + m_j$, $s_0 = 0$, and $m = s_\ell$. Let $\sigma' = \sigma'_1\sigma'_2 \cdots \sigma'_m$ be the induced order $\sigma[N_i]$. Now, let $Q_i = C_1C_2 \cdots C_\ell$ with $C_j = \{\sigma'_t : s_{j-1} < t \leq s_j\}$. Note that, on one hand, σ' is an extension of Q_i, and on the other hand, any other reordering of Q_i must contain a pair xy with $yx \in \sigma'$.

Finally, let Q be the parallel composition of the bucket orders Q_i. Note that σ is an extension of Q, since if $xy \in Q$, then $xy \in Q_i$ for some i, and hence, $xy \in \sigma$. □

By basic combinatorial arguments we find the number of reorderings of a given parallel composition of bucket orders:

LEMMA 3.3. *The number of reorderings of a bucket order of type $m_1 * m_2 * \cdots m_\ell$ is given by $(m_1 + m_2 + \cdots + m_\ell)!/(m_1!m_2! \cdots m_\ell!)$.*

LEMMA 3.4. *The number of reorderings of the parallel composition of bucket orders $P_1, P_2, \ldots, P_k$ is given by $p_1p_2 \cdots p_k$, where p_i is the number of reorderings of P_i.*

Armed with Lemmas 3.1–3.4, we can calculate the time–space product of the equivalence class of any parallel composition of bucket orders. Let P be the parallel composition of bucket orders $P_1, P_2, \ldots, P_k$. Then we have

$$\begin{aligned}\sum_{Q \in \mathcal{P}(P)} |\mathcal{I}(Q)| &= |\mathcal{P}(P)||\mathcal{I}(P)| = \prod_{i=1}^{k} |\mathcal{P}(P_i)||\mathcal{I}(P_i)| , \\ \max_{Q \in \mathcal{P}(P)} |\mathcal{I}(Q)| &= |\mathcal{I}(P)| = \prod_{i=1}^{k} |\mathcal{I}(P_i)| ,\end{aligned}$$

and thus

$$\theta(\mathcal{P}(P)) = |\mathcal{P}(P)||\mathcal{I}(P)|^2 = \prod_{i=1}^{k} |\mathcal{P}(P_i)||\mathcal{I}(P_i)|^2 .$$

We complete this section with a specific family of partial orders that yields a good time–space tradeoff; in the next section, we show that the time–space product of this family is the minimum over all equivalence classes of parallel compositions of bucket orders. For natural numbers n and k with $26k \leq n$ define the family $\mathcal{P}_{n,k}$ as follows. First, for $i = 1, 2, \ldots, k$, let $N_i = \{x \in N : 26(i-1) < x \leq 26i\}$, and let P_i be an arbitrary $13 * 13$ bucket order on N_i. Then, let $N_0 = \{x \in N : 26k < x \leq n\}$, and let P_0 be the bucket order $\{xx : x \in N_0\}$. Finally, let $\mathcal{P}_{n,k}$ the equivalence class of the parallel composition of $P_0, P_1, \ldots, P_k$.

Straightforward application of the formulas presented in the section gives following result, which, by Propositions 2.2 and 3.1, implies Theorem 1.2.

LEMMA 3.5. ($13 * 13$ SCHEME) *Let n and k be natural numbers with $26k \leq n$. Then*

$$\sum_{Q \in \mathcal{P}_{n,k}} |\mathcal{I}(Q)| = \alpha^k 2^n \quad \textit{and} \quad \max_{Q \in \mathcal{P}_{n,k}} |\mathcal{I}(Q)| = \beta^k 2^n ,$$

where

$$\begin{aligned}\alpha &= \binom{26}{13}(2^{14}-1)/2^{26} < 2.539055 \times 10^3 , \\ \beta &= (2^{14}-1)/2^{26} < 2.441258 \times 10^{-4} .\end{aligned}$$

We see that Theorem 1.2 in turn implies Theorem 1.1: If n is divisible by 26, then $\theta(\mathcal{P}_{n,\lfloor n/26 \rfloor})^{1/n} = 4(\alpha\beta)^{1/26} = 3.9271\ldots$. In general, for any $\epsilon > 0$ we have $\theta(\mathcal{P}_{n,\lfloor n/26 \rfloor})^{1/n} < 4(\alpha\beta)^{1/26}(\alpha\beta)^{-1/n} < 4(\alpha\beta)^{1/26} + \epsilon$ for sufficiently large n. This means that with this sequence of partial order families the infimum of the time–space product TS is $4(\alpha\beta)^{1/26} < 3.93$. It may be worth noting that, actually, the “first” family that already yields a time–space product less than 4 is the one where the number 13 above is replaced by 5, that is, with $\lfloor n/10 \rfloor$ parallel bucket orders of type $5 * 5$.

4 Parallel Bucket Orders: Lower Bound

In this section we argue that the $13 * 13$ scheme (Lemma 3.5) is actually optimal in the sense that it minimizes the time–space product within the class of partial order families in question. To this end, let P be the parallel composition of k bucket orders on $N = \{1, 2, \ldots, n\}$. To lower-bound the product $\theta(\mathcal{P}(P))$, define $\psi(m)$ as the minimum of $|\mathcal{P}(B)||\mathcal{I}(B)|^2$ over all bucket orders B on m elements. Before we calculate $\psi(m)$ below, we first note the bound

$$\begin{aligned}\theta(\mathcal{P}(P)) &\geq \prod_{i=1}^{k}\psi(|N_h|)\\ &\geq \min\left\{\psi(m)^{n/m} : m = 1,2,\dots,n\right\}. \qquad (4.1)\end{aligned}$$

Here the first inequality follows by the definition of ψ and the second by $\psi(m) > 0$ and the following elementary observation.

Lemma 4.1. *Let $s_1, s_2, \dots, s_k \geq 1$ be numbers that sum up to s, and let $\phi(r) > 0$ for any r. Then $\phi(s_1)\phi(s_2)\cdots\phi(s_k) \geq \min\left\{\phi(s_i)^{s/s_i} : i = 1,2,\dots,k\right\}$.*

Proof. Suppose the contrary. Then $\prod_i \phi(s_i)^{s_j}$ is (strictly) less than $\phi(s_j)^s$ for all $j = 1,2,\dots,k$. Taking products on both sides yields the contradiction that $\prod_j\prod_i \phi(s_i)^{s_j} = \prod_i \phi(s_i)^s$ is less than $\prod_j \phi(s_j)^s$. $\square$

It remains to calculate $\psi(m)$ and show that $\psi(m)^{1/m}$ is minimized at $m = 26$. We begin by calculating $|\mathcal{P}(B)||\mathcal{I}(B)|^2$ for a bucket order $B = B_1B_2\cdots B_\ell$. If each B_j consists of m_j elements, then $|\mathcal{P}(B)||\mathcal{I}(B)|^2$ is given by

$$\begin{aligned}\theta(m_1, m_2, \dots, m_\ell) &= \frac{(m_1 + m_2 + \cdots + m_\ell)!}{m_1!m_2!\cdots m_\ell!}\\ &\quad\times(1 - \ell + 2^{m_1} + 2^{m_2} + \cdots + 2^{m_\ell})^2 .\end{aligned}$$

We next show that $\theta(m_1, m_2, \dots, m_\ell)$ is minimized subject to $m_1 + m_2 + \cdots + m_\ell = m$ either at $\ell = 1$, $m_1 = m$ or at $\ell = 2$, $m_1 = \lceil m/2\rceil$, $m_2 = \lfloor m/2\rfloor$; this will allow us to express $\psi(m)$ as $\min\{4^n, \theta(\lceil m/2\rceil, \lfloor m/2\rfloor)\}$.

We consider first the case $\ell = 2$ and show that $\theta(m_1, m_2)$ is minimized when m_1 and m_2 are as close to each other as possible, formalized in the following "balancing lemma."

Lemma 4.2. *If m_1 and m_2 are positive integers with $m_1 + m_2 = m$, then $\theta(m_1, m_2) \geq \theta(\lceil m/2\rceil, \lfloor m/2\rfloor)$.*

Proof. We consider even and odd m separately.

Suppose $m = 2a$ is even. Let $c \leq a-1$ be a positive integer. We will show that $\theta(a + c, a - c)/\theta(a, a) \geq 1$. To this end, observe first that

$$(2^{a+c} + 2^{a-c} - 1)/(2^a + 2^a - 1) \geq 2^{c-1}.$$

Thus,

$$\frac{\theta(a + c, a - c)}{\theta(a, a)} \geq \frac{a!a!}{(a + c)!(a - c)!}\, 4^{c-1} =: \rho(a, c) .$$

Next, note that $\rho(a, c)$ grows with a for any fixed c; to see this, observe that the ratio $\rho(a, c)/\rho(a-1, c)$ equals $a^2/[(a+c)(a-c)] > 1$ for $0 < c < a$. Thus, for any fixed c it would suffice to show that $\rho(c+1, c) \geq 1$. What we, in fact, can do is to show that $\rho(c + 1, c)$ grows with c and that $\rho(3, 2) > 1$, which leaves the case $c = 1$ open for a moment. Here, the former claim is proved by

$$\frac{\rho(c + 1, c)}{\rho(c, c - 1)} = \frac{4(c + 1)^2}{(2c + 1)2c} > \frac{4(c + 1)^2}{(2c + 2)^2} = 1 ,$$

and the latter claim by calculation: $\rho(3, 2) = 6/5 > 1$. Finally, the case of $c = 1$ is handled by

$$\begin{aligned}\frac{\theta(a + 1, a - 1)}{\theta(a, a)} &= \frac{a}{a + 1}\Big(\frac{5 \cdot 2^{a-1} - 1}{4 \cdot 2^{a-1} - 1}\Big)^2\\ &> \frac{2}{2 + 1}\Big(\frac{5}{4}\Big)^2\\ &= \frac{50}{48} > 1 .\end{aligned}$$

Suppose then that $m = 2a + 1$ is odd. Again, let $c \leq a - 1$ be a positive integer. To show that $\theta(a + 1 + c, a - c)/\theta(a + 1, a) \geq 1$ we will repeat the line of argumentation given above for even m. To this end, observe

$$(2^{a+1+c} + 2^{a-c} - 1)/(2^{a+1} + 2^a - 1) \geq 2^{c+1}/3 .$$

Thus,

$$\begin{aligned}\frac{\theta(a + 1 + c, a - c)}{\theta(a + 1, a)} &\geq \frac{(a + 1)!a!}{(a + 1 + c)!(a - c)!}\,\frac{4^{c+1}}{9}\\ &=: \rho'(a, c) .\end{aligned}$$

Next, note that $\rho'(a, c)$ grows with a for any fixed c; to see this, observe that the ratio $\rho'(a, c)/\rho'(a-1, c)$ equals $(a+1)a/[(a+1+c)(a-c)] > 1$ for $0 < c < a$. Thus, for any fixed c it would suffice to show that $\rho'(c+1, c) \geq 1$. What we, in fact, can do is to show that $\rho'(c+1, c)$ grows with c and that $\rho'(3, 2) > 1$, which leaves the case $c = 1$ open for a moment. Here, the former claim is proved by

$$\frac{\rho'(c + 1, c)}{\rho'(c, c - 1)} = \frac{4(c + 2)(c + 1)}{(2c + 2)(2c + 1)} > \frac{4(c + 2)(c + 1)}{(2c + 4)(2c + 2)} = 1 ,$$

and the latter by calculation: $\rho'(3, 2) = 64/45 > 1$. Finally, the case of $c = 1$ is handled by

$$\begin{aligned}\frac{\theta(a + 2, a - 1)}{\theta(a + 1, a)} &= \frac{a}{a + 2}\Big(\frac{9 \cdot 2^{a-1} - 1}{6 \cdot 2^{a-1} - 1}\Big)^2\\ &> \frac{2}{2 + 2}\Big(\frac{3}{2}\Big)^2\\ &= \frac{18}{16} > 1 . \; \square\end{aligned}$$

At first glance, one might think that the uniform distribution should minimize $\theta(m_1, m_2, \dots, m_\ell)$ also for

$\ell > 2$. However, this is in fact not the case.[2] Therefore, the proof technique we use for Lemma 4.2 or other (often powerful) convexity arguments seem not applicable. Instead, we are able to prove the following "shortening lemma", which states that for any bucket order of length $\ell + 1 \geq 3$ there is another bucket order of length $\ell \geq 2$ that yields a smaller time–space product.

LEMMA 4.3. *Let $\ell \geq 2$ and let $m_1 \geq m_2 \geq \cdots \geq m_{\ell+1} \geq 0$ and $c_1 \geq c_2 \geq \cdots \geq c_\ell \geq 0$ be integers such that $m_{\ell+1} = c_1 + c_2 + \cdots + c_\ell$ and $c_1 - c_\ell \leq 1$. Then $\theta(m_1, m_2, \ldots, m_{\ell+1}) > \theta(m_1 + c_1, m_2 + c_2, \ldots, m_\ell + c_\ell)$.*

Proof. Put $a_i := m_i + c_i$ for $i = 1, 2, \ldots, \ell$. Also, denote $b := m_{\ell+1}$ for brevity. We will show that $\theta(a_1, a_2, \ldots, a_\ell)/\theta(m_1, m_2, \ldots, m_\ell, b) < 1$.

We begin with the case $b = 1$. Then it suffices to show that

$$\begin{aligned}&\frac{\theta(m_1+1, m_2, \ldots, m_\ell)}{\theta(m_1, m_2, \ldots, m_\ell, 1)}\\ &= \frac{1}{m_1+1}\left(\frac{2^{m_1+1} + 2^{m_2} + \cdots + 2^{m_\ell} + 1 - \ell}{2^{m_1} + 2^{m_2} + \cdots + 2^{m_\ell} + 2 - \ell}\right)^2\\ &< 1 .\end{aligned}$$

To see that this holds, we consider a few cases to show that the squared term is always less than $m_1 + 1$. Because the squared term is always less than 4, we are done for $m_1 \geq 3$. Now, if $m_1 = 2$, then the squared term is at most $[(2^3 + 2^1 - 1)/(2^2 + 2^1 + 2^1 - 2)]^2 = (9/6)^2 = 9/4 < 3$, Finally, if $m_1 = 1$, then the squared term is at most $[(2^2 + 2^1 - 1)/(2^1 + 2^1 + 2^1 - 2)]^2 = (5/4)^2 = 25/16 < 2$.

Then, for any $b \geq 1$, we notice the bound

$$\frac{m_1!m_2!\cdots m_\ell!b!}{a_1!a_2!\cdots a_\ell!} \leq \frac{b!}{(b+1)^b} \tag{4.2}$$

that follows since the denominator contains the factorials in the numerator, except for $b!$, plus b other terms all greater or equal to $b + 1$.

Next suppose $2 \leq b \leq \ell$. Under this assumption $a_i = m_i + 1$ for $i = 1, 2, \ldots, b$ and $a_i = m_i$ for $i = b + 1, b + 2, \ldots, \ell$. So we find that

$$\begin{aligned}&2^{a_1} + 2^{a_2} + \cdots + 2^{a_\ell} + 1 - \ell\\ &\leq 2\big(2^{m_1} + 2^{m_2} + \cdots + 2^{m_\ell} + 2^b - \ell\big) .\end{aligned} \tag{4.3}$$

To see this, subtract the terms on the left from the ones on the right to get

$$\begin{aligned}&2^{m_{b+1}} + 2^{m_{b+2}} + \cdots + 2^{m_\ell} + 2^{b+1} - \ell - 1\\ &\geq (\ell - b + 2)2^b - \ell - 1 \geq 0 .\end{aligned}$$

Here the last inequality follows because $(\ell - b + 2)2^b$ clearly grows with b for $1 \leq b \leq \ell$, and at $b = \ell$ we have $2^{\ell+1} \geq \ell + 1$, which holds for all $\ell > 0$. Combining the bounds (4.2) and (4.3) yields

$$\frac{\theta(a_1, a_2, \ldots, a_\ell)}{\theta(m_1, m_2, \ldots, m_\ell, b)} \leq \frac{4b!}{(b+1)^b} < 1$$

for $b \geq 2$, since $(4 \cdot 2!)/(2 + 1)^2 = 8/9$ and it is easy to verify that $4b!/(b + 1)^b$ decreases when b grows.

It remains to consider the case $b > \ell$. We will first examine the cases $b = 3$ and $b = 4$, and then the remaining case $b \geq 5$.

Suppose $b = 3$; hence, $\ell = 2$. Thus $a_1 = m_1 + 2$ and $a_2 = m_2 + 1$, and so

$$\begin{aligned}\frac{\theta(a_1, a_2)}{\theta(m_1, m_2, b)} &= \frac{3!}{(m_1+2)(m_1+1)(m_2+1)}\\ &\qquad\times\left(\frac{2^{m_1+2} + 2^{m_2+1} - 1}{2^{m_1} + 2^{m_2} + 2^3 - 2}\right)^2 .\end{aligned}$$

Now, if $m_1 = 3$, then $m_2 = 3$, and the above ratio evaluates to $6/80(47/22)^2 < 1$. Otherwise $m_1 \geq 4$, and the ratio can be bounded from above by $6/(6 \cdot 5 \cdot 4)4^2 = 4/5 < 1$.

Next suppose $b = 4$. This means $a_i \leq m_i + 2$ for all $i = 1, 2, \ldots, \ell$. Using the bound (4.2) yields

$$\frac{\theta(a_1, a_2, \ldots, a_\ell)}{\theta(m_1, m_2, \ldots, m_\ell, b)} \leq \frac{4^2 \cdot 4!}{(4+1)^4} = \frac{384}{625} < 1 .$$

Finally, suppose $b \geq 5$. Observe $a_i \leq m_i + \lceil b/\ell \rceil \leq m_i + (b + \ell - 1)/\ell \leq m_i + (b + 1)/2$. Thus,

$$\begin{aligned}&2^{a_1} + 2^{a_2} + \cdots + 2^{a_\ell} + 1 - \ell\\ &\leq 2^{(b+1)/2}\big(2^{m_1} + 2^{m_2} + \cdots + 2^{m_\ell} + 2^b - \ell\big) ;\end{aligned} \tag{4.4}$$

note that $2^b - \ell$ is positive, since $b > \ell$. Combining the bounds (4.2) and (4.4) yields

$$\frac{\theta(a_1, a_2, \ldots, a_\ell)}{\theta(m_1, m_2, \ldots, m_\ell, b)} \leq \frac{2^{b+1}b!}{(b+1)^b} < 1 .$$

Here the last inequality follows because at $b = 5$ we have $2^{5+1}5!/(5 + 1)^5 = 80/81 < 1$ and because this bound decreases when b grows. To verify the latter claim, observe that the bound at b divided by the bound at $b-1$ equals $2[b/(b+1)]^b \leq 2[(b+1)/b]\mathrm{e}^{-1} \leq 8/(3\mathrm{e}) < 1$ for any $b \geq 3$. $\square$

Combining the shortening lemma (Lemma 4.3) with the balancing lemma (Lemma 4.2) immediately yields the following summary.

LEMMA 4.4. *Let $m \geq 1$ be an integer. Then $\psi(m)$ equals the smaller of $\theta(m) = 4^m$ and $\theta(\lceil m/2 \rceil, \lfloor m/2 \rfloor)$.*

[2] A counter example is $\theta(3, 3, 3) \approx 4.536 > 4.421 \approx \theta(4, 4, 1)$.

We are now ready to show that the time–space product of the $13 * 13$ scheme is the smallest one can achieve with parallel compositions of bucket orders.

Proposition 4.1. (Lower bound) *Let P be a parallel composition of bucket orders on $\{1, 2, \ldots, n\}$. Then the time-space product $\theta(\mathcal{P}(P))^{1/n}$ is at least $4(\alpha\beta)^{1/26} \geq 3.9271$, where α and β are as defined in Lemma 3.5.*

Proof. By the bound (4.1), it suffices to show that $\psi(m)^{1/m}$ is minimized at $m = 26$. By calculation, using Lemma 4.4, we find that this is indeed the case when $1 \leq m \leq 149$ (results not shown).

It remains to show that $\psi(m)^{1/m} \geq \psi(26)^{1/26} = 3.9271\ldots$ for all $m \geq 150$. Because this clearly holds if $\psi(m) = \theta(m)$, we may, by Lemma 4.4, without any loss in generality assume $\psi(m) = \theta(\lceil m/2 \rceil, \lfloor m/2 \rfloor)$. To this end, define $\upsilon(m) := \binom{m}{\lfloor m/2 \rfloor}^{1/m}$ and observe

$$\begin{aligned} \psi(m)^{1/m} &= \upsilon(m)\big(2^{\lceil m/2 \rceil} + 2^{\lfloor m/2 \rfloor} - 1\big)^{2/m} \\ &\geq \upsilon(m)\big(2^{m/2+1} - 1\big)^{2/m} \\ &\geq 2\upsilon(m)\ . \end{aligned}$$

Next we show that $\upsilon(m)$ grows with m, by proving $\upsilon(2a-1)/\upsilon(2a) \leq 1$ and $\upsilon(2a)/\upsilon(2a+1) \leq 1$ for any $a = 1, 2, \ldots$ (actually, for any $m \geq 150$ would do). The former is shown by

$$\left(\frac{\upsilon(2a-1)}{\upsilon(2a)}\right)^{2a-1} = \binom{2a}{a}^{\frac{1}{2a}} \frac{a}{2a} \leq \left(2^{2a}\right)^{\frac{1}{2a}} \frac{1}{2} = 1\ ;$$

the latter is shown by

$$\begin{aligned} \left(\frac{\upsilon(2a)}{\upsilon(2a+1)}\right)^{2a+1} &= \binom{2a}{a}^{\frac{1}{2a}} \frac{a+1}{2a+1} \\ &\leq \left(\frac{2^{2a}}{\mathrm{e}}\right)^{\frac{1}{2a}} \frac{1}{2}\left(1 + \frac{1}{2a+1}\right) \\ &\leq \mathrm{e}^{-\frac{1}{2a}} \mathrm{e}^{\frac{1}{2a+1}} \\ &< \mathrm{e}^0 = 1\ , \end{aligned}$$

where the first inequality holds for $a \geq 2$, whereas in the case $a = 1$ we replace e by 2 and obtain the bound $2\sqrt{2}/3 < 1$.

Now it suffices to verify that $2\upsilon(150) = 3.92778\ldots > 3.9271$. $\square$

Acknowledgements The authors are grateful to Fedor Fomin, Petteri Kaski, Saket Saurabh, and Yngve Villanger for valuable discussions about the divide and conquer technique for permutation problems.

References

[1] S. Arnborg, D. G. Corneil, and A. Proskurowski, *Complexity of finding embeddings in a k-tree*, SIAM J. Alg. Disc. Meth., 8 (1987), pp. 277–284.

[2] R. Bellman, *Dynamic programming treatment of the travelling salesman problem*, J. Assoc. Comput. Mach., 9 (1962), pp. 61–63.

[3] A. Björklund and T. Husfeldt, *Exact algorithms for exact satisfiability and number of perfect matchings*, Algorithmica, 52 (2008), pp. 226–249.

[4] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto, *The travelling salesman problem in bounded degree graphs*, in Proc. of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008), pp. 198–209. Springer LNCS 5125, 2008.

[5] ———, *Computing the Tutte polynomial in vertex-exponential time*, in Proc. of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008), pp. 677–686. IEEE Computer Society, 2008.

[6] H. Bodlaender, F. Fomin, A. Koster, D. Kratsch, and D. Thilikos, *On exact algorithms for treewidth*, in Proc. of the 14th Annual European Symposium on Algorithms (ESA 2006), pp. 672–683, 2006.

[7] D. Eppstein, *The traveling salesman problem for cubic graphs*, J. Graph Algorithms Appl., 11 (2003), pp. 61–81.

[8] F. Fomin and Y. Villanger, *Treewidth computation and extremal combinatorics*, in Proc. of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008), pp. 210–221. Springer LNCS 5125, 2008.

[9] Y. Gurevich and S. Shelah, *Expected computation time for Hamiltonian path problem*, SIAM J. Comput., 16 (1987), pp. 486–502.

[10] M. Held and R. Karp, *A dynamic programming approach to sequencing problems*, J. Soc. Indust. Appl. Math., 10 (1962), pp. 196–210.

[11] M. Koivisto and K. Sood, *Exact Bayesian structure discovery in Bayesian networks*, Journal of Machine Learning Research, 5 (2004), pp. 549–573.

[12] E. Lawler, *A comment on minimum feedback arc sets*, IEEE Trans. on Circuit Theory, pp. 296–297, 1964.

[13] ———, *Efficient implementation of dynamic programming algorithms for sequencing problems*, Technical Report BW 106/79, Stiching Matematisch Centrum, Amsterdam, 1979.

[14] S. Ott and S. Miyano, *Finding optimal gene networks using biological constraints*, Genome Informatics, 14 (2003), pp. 124–133.

[15] P. Parviainen and M. Koivisto, *Exact structure discovery in Bayesian networks with less space*, in Proc. of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009).

[16] E. Perrier, S. Imoto, and S. Miyano, *Finding optimal Bayesian network given a super-structure*, Journal of Machine Learning Research, 9 (2008), pp. 2251–2286.

[17] W. Savitch, *Relationships between nondeterministic and deterministic tape complexities*, Journal of Computer and System Sciences, 4 (1970), pp. 177–192.

[18] L. Schrage and K. R. Baker, *Dynamic programming solution for sequencing problems with precedence constraints*, Operations Research, 26 (1978), pp. 444–449.

[19] T. Silander and P. Myllymäki, *A simple approach for finding the globally optimal Bayesian network structure*, in Proc. of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI 2006), pp 445–452. AUAI Press, 2006.

[20] G. Steiner, *On the complexity of dynamic programming for sequence problems with precedence constraints*, Annals of Operations Research, 26 (1990), pp. 103–123.

[21] V. Vassilevska and R. Williams, *Finding, minimizing, and counting weighted subgraphs*, in Proc. of the 41st ACM Symposium on Theory of Computing (STOC 2009), pp. 455–464. ACM Press, 2009.