Algorithmica (2018) 80:2134–2159
https://doi.org/10.1007/s00453-017-0397-y

Optimal Composition Ordering Problems for Piecewise Linear Functions

Yasushi Kawase[1] · **Kazuhisa Makino**[2] · **Kento Seimi**[3]

Received: 25 January 2017 / Accepted: 14 November 2017 / Published online: 22 November 2017

Abstract In this paper, we introduce maximum composition ordering problems. The input is n real functions $f_1, \dots, f_n : \mathbb{R} \to \mathbb{R}$ and a constant $c \in \mathbb{R}$. We consider two settings: total and partial compositions. The maximum total composition ordering problem is to compute a permutation $\sigma : [n] \to [n]$ which maximizes $f_{\sigma(n)} \circ f_{\sigma(n-1)} \circ \cdots \circ f_{\sigma(1)}(c)$, where $[n] = \{1, \dots, n\}$. The maximum partial composition ordering problem is to compute a permutation $\sigma : [n] \to [n]$ and a nonnegative integer k $(0 \leq k \leq n)$ which maximize $f_{\sigma(k)} \circ f_{\sigma(k-1)} \circ \cdots \circ f_{\sigma(1)}(c)$. We propose $\mathrm{O}(n \log n)$ time algorithms for the maximum total and partial composition ordering problems for monotone linear functions f_i, which generalize linear deterioration and shortening models for the time-dependent scheduling problem. We also show that the maximum total composition ordering problem can be solved in polynomial time if f_i is of the form $\max\{a_i x + b_i, d_i, x\}$ for some constants a_i (≥ 0), b_i and d_i. As a corollary, we show that the two-valued free-order secretary problem can be solved in polynomial time. We finally prove that there exists no constant-factor approximation algorithm

An extended abstract of this paper appears in ISAAC 2016 [16].

✉ Yasushi Kawase
kawase.y.ab@m.titech.ac.jp

Kazuhisa Makino
makino@kurims.kyoto-u.ac.jp

Kento Seimi
kento.seimi@gmail.com

1 Tokyo Institute of Technology, Tokyo, Japan

2 Kyoto University, Kyoto, Japan

3 Tokyo, Japan

for the problems, even if f_i's are monotone, piecewise linear functions with at most two pieces, unless P = NP.

Keywords Function composition · Time-dependent scheduling · Ordering problem

1 Introduction

In this paper, we introduce *optimal composition ordering problems* and mainly study their time complexity. The input of the problems is n real functions $f_1, \ldots, f_n : \mathbb{R} \to \mathbb{R}$ and a constant $c \in \mathbb{R}$. In this paper, we assume that the input functions are piecewise linear, and the input length of a piecewise linear function is the sum of the sizes of junctions and coefficients of linear functions. We consider two settings: *total* and *partial* compositions. The maximum total composition ordering problem is to compute a permutation $\sigma : [n] \to [n]$ that maximizes $f_{\sigma(n)} \circ f_{\sigma(n-1)} \circ \cdots \circ f_{\sigma(1)}(c)$, where $[n] = \{1, \ldots, n\}$. The maximum partial composition ordering problem is to compute a permutation $\sigma : [n] \to [n]$ and a nonnegative integer k ($0 \leq k \leq n$) that maximize $f_{\sigma(k)} \circ f_{\sigma(k-1)} \circ \cdots \circ f_{\sigma(1)}(c)$. For example, if the input consists of $f_1(x) = 2x - 6$, $f_2(x) = \frac{1}{2}x + 2$, $f_3(x) = x + 2$, and $c = 2$, then the ordering σ such that $\sigma(1) = 2, \sigma(2) = 3$, and $\sigma(3) = 1$ is optimal for the maximum total composition ordering problem. In fact, $f_1 \circ f_3 \circ f_2(c) = f_1(f_3(f_2(c))) = f_1(f_3(c/2 + 2)) = f_1(c/2 + 4) = c + 2 = 4$ provides the optimal value of the problem. The ordering σ above and $k = 2$ is optimal for the maximum partial composition ordering problem, where $f_3 \circ f_2(c) = 5$. We also consider the maximum exact k-composition ordering problem, which is a problem to compute a permutation $\sigma : [n] \to [n]$ that maximizes $f_{\sigma(k)} \circ f_{\sigma(k-1)} \circ \cdots \circ f_{\sigma(1)}(c)$ for given n functions $f_1, \ldots, f_n : \mathbb{R} \to \mathbb{R}$, a constant $c \in \mathbb{R}$, and a nonnegative integer k ($0 \leq k \leq n$). We remark that the minimization versions are reducible to the maximization ones, which will be shown in the next section.

As we will see in this paper, the optimal composition ordering problems are natural and fundamental in many fields such as artificial intelligence, computer science, and operations research. However, to the best of the authors' knowledge, no one explicitly studies the problems from the algorithmic point of view. We below describe single machine time-dependent scheduling problems and the free-order secretary problem, which can be formulated as the optimal composition ordering problems.

1.1 Time-Dependent Scheduling

Consider machine scheduling problems with time-dependent processing times, called *time-dependent scheduling problems* [6,12].

Let J_i ($i \in [n]$) denote a job with a ready time $r_i \in \mathbb{R}$, a deadline $d_i \in \mathbb{R}$, and a processing time $p_i : \mathbb{R} \to \mathbb{R}$, where $r_i \leq d_i$ is assumed. Different from the classical setting, the processing time p_i is *not* a constant, but depends on the *starting* time of job J_i. The model has been studied to deal with learning and deteriorating effects, for example [13–15,20,21]. Here each p_i is assumed to satisfy $p_i(t) \leq s + p_i(t + s)$ for all t and $s \geq 0$, since we should be able to finish processing job J_i earlier if it starts earlier. Among time-dependent settings, we consider the single machine scheduling problem to minimize the makespan, where the input is the start time t_0 $(= 0)$ and a set of J_i $(i \in [n])$ above. The makespan denotes the time when all the jobs have finished processing, and we assume that the machine can handle only one job at a time and preemption is not allowed. We show that the problem can be viewed as the minimum total composition ordering problem.

For simplicity, let us first consider the simplest case, that is, each job has neither the ready time r_i nor the deadline d_i. Let $c = t_0$, and for each $i \in [n]$, define the function f_i by $f_i(t) = t + p_i(t)$. Note that job J_i has been finished processing at time $f_i(t)$ if it is started processing at time t. This implies that $f_{\sigma(n)} \circ f_{\sigma(n-1)} \circ \cdots \circ f_{\sigma(1)}(t_0)$ denotes the makespan of the scheduling problem when we fix the ordering σ of the jobs. Therefore, the problem is represented as the minimum total composition ordering problem. More generally, let us consider the case in which each job J_i also has both the ready time r_i and the deadline d_i with $d_i \geq r_i$. Define a function f_i by

$$f_i(t) = \begin{cases} r_i + p_i(r_i) & (t \leq r_i), \\ t + p_i(t) & (r_i < t \leq d_i - p_i(t)), \\ \infty & (t > d_i - p_i(t)). \end{cases}$$

Then the problem can be reduced to the minimum total composition ordering problem for $((f_i)_{i \in [n]}, c = t_0)$.

A number of restrictions on the processing time $p_i(t)$ has been studied in the literature (e.g., [3,5,17]).

In the *linear deterioration* model, the processing time p_i is restricted to be a monotone increasing linear function that satisfies $p_i(t) = a_i t + b_i$ for two positive constants a_i and b_i. Here a_i and b_i are respectively called the *deterioration rate* and the *basic processing time* of job J_i. Gawiejnowicz and Pankowska [13], Gupta and Gupta [14], Tanaev et al. [20], and Wajs [21] obtained the result that the time-dependent scheduling problem of this model (without the ready time r_i nor the deadline d_i) is solvable in $\mathrm{O}(n \log n)$ time by scheduling jobs in the nonincreasing order of the ratios b_i/a_i. As for the hardness results, it is known that the proportional deterioration model with ready time and deadline, the linear deterioration model with ready time, and the linear deterioration model with deadlines are all NP-hard [4,11].

Another model is called the *linear shortening* model introduced by Ho et al. [15]. In this model, the processing time p_i is restricted to be a monotone decreasing linear function that satisfies $p_i(t) = -a_i t + b_i$ with two constants a_i and b_i such that $1 > a_i > 0$, $b_i > 0$, and $a_j \left(\sum_{i=1}^n b_i - b_j\right) < b_j$. Here the assumptions on a_i and b_i make sense from the practical point of view (e.g., the processing time is nonnegative). They showed that the time-dependent scheduling problem of this model can be solved in $\mathrm{O}(n \log n)$ time by again scheduling jobs in the nonincreasing order of the ratios b_i/a_i. Cheng and Ding [4] showed some relationships between the linear deterioration and shortening model, from which the complexity results of the shortening model are obtained from the deterioration model, and vice versa. For example, they showed the linear shortening model $p_i(t) = -at + b_i$ with ready time r_i can be solved in $\mathrm{O}(n^6 \log n)$ time, and that the linear shortening model with deadline is NP-hard.

1.2 Free-Order Secretary Problem

The *free-order secretary problem* is another application of the optimal composition ordering problems, which is closely related to a branch of the problems such as the full-information secretary problem [9], knapsack and matroid secretary problems [1,2,19] and stochastic knapsack problems [7,8]. Imagine that an administrator wants to hire the best secretary out of n applicants for a position. Each applicant i has a nonnegative independent random variable X_i as his ability for the secretary. Here $X_1, \ldots, X_n$ are not necessarily based on the same probability distribution, and assume that the administrator knows all the probability distributions of X_i's before their interviews, where such information can be obtained by their curriculum vitae and/or results of some written examinations. The applicants are interviewed one-by-one, and the administrator can observe the value X_i during the interview of the applicant i. A decision on each applicant is to be made immediately after the interview. Once an applicant is rejected, he will never be hired. The interview process is finished if some applicant is chosen, where we assume that the last applicant is always chosen if he is interviewed since the administrator has to hire exactly one candidate. The objective is to find an optimal strategy for this interview process, i.e., to find an interview ordering together with the stopping rule that maximizes the expected value of the secretary hired.

Let $f_i(x) = \mathbf{E}[\max\{X_i, x\}]$. We now claim that our secretary problem can be represented by the maximum total composition ordering problem $((f_i)_{i\in[n]}, c = 0)$.

Let us first consider the best stopping rule for the interview to maximize the expected value for the secretary hired when the interview ordering is fixed in advance. Assume that the applicant i is interviewed in the ith place. Note that $\mathbf{E}[X_n]$ $(= f_n(0))$ is the expected value under the condition that all the applicants except for the last one are rejected, since the last applicant is hired. Consider the situation that all the applicants except for the last two ones are rejected. Then it is a best stopping rule that the applicant $n-1$ is hired if and only if $X_{n-1} \geq f_n(0)$ is satisfied (i.e., the applicant n is hired if and only if $X_{n-1} < f_n(0)$), where $f_{n-1} \circ f_n(0)$ is the expected value for the best stopping rule, under this situation. By applying backward induction, we have the following best stopping rule: we hire the applicant i $(< n)$ and stop the interview process, if $X_i \geq f_{i+1} \circ \cdots \circ f_n(0)$ (otherwise, the next applicant is interviewed), and we hire the applicant n if no applicant i $(< n)$ is hired. It turns out that $f_1 \circ \cdots \circ f_n(0)$ is the maximum expected value for the secretary hired, if the interview ordering is fixed such that the applicant i is interviewed in the ith place.

Therefore, the secretary problem (i.e., finding an interview ordering, together with a stopping rule) can be formulated as the maximum total composition ordering problem $((f_i)_{i\in[n]}, c = 0)$.

In addition, let us assume that X_i is an m-valued random variable that takes the value $a_{i,j}$ with probability $p_{i,j} \geq 0$ $(j = 1, \ldots, m)$. Here we assume that $a_{i,1} \geq \cdots \geq a_{i,m} \geq 0$ and $\sum_{j=1}^{m} p_{i,j} = 1$. Then we have

$$f_i(x) = \sum_{j=1}^{m} p_{i,j} \max\{a_{i,j}, x\}$$

$$
\begin{aligned}
&= \begin{cases} \sum_{j=1}^{m} p_{i,j} a_{i,j} & \text{if } x \leq a_{i,m}, \\ \sum_{j=1}^{l} p_{i,j} a_{i,j} + \sum_{j=l+1}^{m} p_{i,j} x & \text{if } a_{i,l+1} < x \leq a_{i,l} \quad (l=1,\ldots,m-1), \\ x & \text{if } x > a_{i,1}, \end{cases} \\
&= \max_{l=0}^{m} \left\{ \sum_{j=1}^{l} p_{i,j} a_{i,j} + \sum_{j=l+1}^{m} p_{i,j} x \right\}.
\end{aligned}
$$

Note that this f_i is a monotone convex piecewise linear function with at most $(m+1)$ pieces.

1.3 Main Results Obtained in This Paper

In this paper, we consider the computational issues for the optimal composition ordering problems, when all f_i's are monotone and almost linear.

We first show that the problems become tractable if all f_i's are monotone and linear, i.e., $f_i(x) = a_i x + b_i$ for $a_i \geq 0$.

Theorem 1 *The maximum partial and total composition ordering problems for monotone nondecreasing linear functions are both solvable in* $\mathrm{O}(n \log n)$ *time.*

Recall that the algorithm for the linear shortening model (resp., the linear deterioration model) for the time-dependent scheduling problem is easily generalized to the case when all a_i's satisfy $a_i < 1$ (resp., $a_i > 1$). The best composition ordering is obtained as the nondecreasing order of the ratios b_i/a_i. This idea can be extended to the maximum partial composition ordering problem in the mixed case (i.e., some $a_i > 1$ and some $a_{i'} < 1$) of Theorem 1. However, we cannot extend it to the maximum total composition ordering problem. In fact, we do not know if there exists such a simple criterion on the maximum total composition ordering. We instead present an efficient algorithm that chooses the best ordering among linearly many candidates.

We also provide a dynamic-programming based polynomial-time algorithm for the exact k-composition setting.

Theorem 2 *The maximum exact k-composition ordering problem for monotone nondecreasing linear functions is solvable in* $\mathrm{O}(k \cdot n^2)$ *time.*

We next consider the monotone, piecewise linear case. It can be directly shown from the time-dependent scheduling problem that the maximum total composition ordering problem is NP-hard, even if all f_i's are monotone, concave, and piecewise linear functions with at most two pieces, i.e., $f_i(x) = \min\{a_i x + b_i,\ c_i x + d_i\}$ for some constants a_i, b_i, c_i, and d_i with $a_i, c_i > 0$. Furthermore, the maximum total (partial) composition ordering problem is inapproximable even if all f_i's are monotone piecewise linear functions with at most two pieces and either all f_i's are convex or all are concave.

Theorem 3 *For all positive real number* α (≤ 1)*, there exists no* α*-approximation algorithm for the maximum total (partial) composition ordering problem even if all* f_i*'s are monotone piecewise linear functions with at most two pieces and either all* f_i*'s are concave or all are convex, unless P = NP. Moreover, for the concave case, the inapproximability holds even if there is only one function which consists of two pieces.*

Table 1 The current status on the time complexity of the maximum total composition ordering problem

Functions	Complexity	References
$f_i(x) = a_i x + b_i \quad (a_i > 1,\ b_i < 0)$	$\mathrm{O}(n \log n)$	[13,14,20,21]
$f_i(x) = a_i x + b_i \quad (1 > a_i \geq 0,\ b_i < 0)$	$\mathrm{O}(n \log n)$	[15]
$f_i(x) = \min\{ax + b_i, r_i\} \quad (1 > a > 0,\ b_i < 0)$	$\mathrm{O}(n^6 \log n)$	[4]
$f_i(x) = \min\{ax + b_i, r_i\} \quad (a > 1,\ b_i < 0)$	NP-hard	[4]
$f_i(x) = \min\{a_i x + b_i, d_i\} \quad (a_i > 1)$	NP-hard	[4]
$f_i(x) = \begin{cases} \min\{a_i x, r_i\} & (x \geq d_i) \\ -\infty & (x < d_i) \end{cases} \quad (a_i > 1)$	NP-hard	[11]
$f_i(x) = \begin{cases} a_i x + b_i & (x \geq d_i) \\ -\infty & (x < d_i) \end{cases} \quad (1 > a_i > 0)$	NP-hard	[4]
$f_i(x) = a_i x + b_i \quad (a_i \geq 0)$	$\mathbf{O(n \log n)}$	[**Theorem** 1]
$f_i(x) = \max\{x, a_i x + b_i\} \quad (a_i \geq 0)$	$\mathbf{O(n \log n)}$	[**Theorem** 1]
$f_i(x) = \max\{x, a_i x + b_i, d_i\} \quad (a_i \geq 0)$	$\mathbf{O(n^2)}$	[**Theorem** 4]
$f_i(x) = \min\{a_i x + b_i, c_i x + d_i\} \quad (a_i, c_i > 0)$	**NP-hard**	[**Theorem** 3]
$f_i(x) = \max\{a_i x + b_i, c_i x + d_i\} \quad (a_i, c_i > 0)$	**NP-hard**	[**Theorem** 3]

Here the bold letters represent our results, and the results for the minimum and/or partial versions are described as the ones for the maximum total composition ordering problem, since the minimum and partial versions all can be transformed into the maximum total one as shown in Sects. 2 and 3

Here f_i can be represented by $f_i(x) = \max\{a_i x + b_i,\ c_i x + d_i\}$ for some constants a_i, b_i, c_i, and d_i with $a_i, c_i > 0$ if f_i is a monotone, convex, and piecewise linear function with at most two pieces.

As for the positive side, if each f_i is a special case of monotone, convex, and piecewise linear function, then we have the following result, which implies that the two-valued free-order secretary problem can be solved in $\mathrm{O}(n^2)$ time.

Theorem 4 *For $i \in [n]$, let $f_i(x) = \max\{a_i x + b_i, d_i\}$ and let $\overline{f}_i(x) = \max\{a_i x + b_i, d_i, x\}$ for some constants a_i (≥ 0), b_i and d_i. Then the maximum partial composition ordering problem $((f_i)_{i \in [n]}, c)$ and the maximum total composition ordering problem $((\overline{f}_i)_{i \in [n]}, c)$ are both solvable in $\mathrm{O}(n^2)$ time.*

We summarize the current status on the time complexity of the maximum total composition ordering problem in Table 1.

1.4 The Organization of the Paper

The rest of the paper is organized as follows. In Sect. 2, we show that the minimum and/or partial versions of the optimal composition ordering problem can be formulated as the maximum total composition ordering problem. In Sect. 3, we prove the partial composition part of Theorems 1 and 4, and in Sect. 4, we prove the total composition part of Theorems 1 and 2. Finally, Sect. 5 provides a proof of Theorem 3.

2 Properties of Function Composition

In this section, we present two basic properties of the optimal composition ordering problems, which imply that the *maximum total* composition ordering problem represents all the other composition ordering problems, i.e., the minimum partial, the minimum total, and the maximum partial ones. Let us start with the lemma that the minimization problems are equivalent to the maximization ones.

For a function $f : \mathbb{R} \to \mathbb{R}$, define a function $\tilde{f} : \mathbb{R} \to \mathbb{R}$ by

$$\tilde{f}(x) := -f(-x).$$

For example, if $f(x) = 2x - 3$, then we have $\tilde{f}(x) = 2x + 3$. By the definition, we have $\tilde{\tilde{f}} = f$, and $\tilde{f}$ inherits several properties for f, e.g., linearity and monotonicity.

Lemma 5 *Let c be a real, and for $i = 1, \ldots, n$, let $f_i : \mathbb{R} \to \mathbb{R}$ be real functions. Then we have the following two statements.*

(a) *A permutation $\sigma : [n] \to [n]$ is optimal for the maximum total composition ordering problem $((f_i)_{i\in[n]}, c)$ if and only if it is optimal for the minimum total composition ordering problem $((\tilde{f}_i)_{i\in[n]}, -c)$.*
(b) *A permutation $\sigma : [n] \to [n]$ and an integer k with $0 \leq k \leq n$ form an optimal solution for the maximum partial composition ordering problem $((f_i)_{i\in[n]}, c)$ if and only if they form an optimal solution for the minimum partial composition ordering problem $((\tilde{f}_i)_{i\in[n]}, -c)$.*

Proof The lemma holds since

$$f_{\sigma(k)} \circ f_{\sigma(k-1)} \circ \cdots \circ f_{\sigma(1)}(c) = -\tilde{f}_{\sigma(k)} \circ \tilde{f}_{\sigma(k-1)} \circ \cdots \circ \tilde{f}_{\sigma(1)}(-c),$$

for all permutation $\sigma : [n] \to [n]$ and an integer k with $0 \leq k \leq n$. □

Due to the lemma, this paper deals with the maximum composition ordering problems only.

We next show the relationships between total and partial compositions. For a function $f : \mathbb{R} \to \mathbb{R}$, define a function $\overline{f} : \mathbb{R} \to \mathbb{R}$ by

$$\overline{f}(x) := \max\{f(x), x\}.$$

Lemma 6 *Let c be a real, and for $i = 1, \ldots, n$, let $f_i : \mathbb{R} \to \mathbb{R}$ be monotone nondecreasing real functions. Then the objective value of the maximum partial composition ordering problem $((f_i)_{i\in[n]}, c)$ is equal to the one of the maximum total composition ordering problem $((\overline{f}_i)_{i\in[n]}, c)$. Moreover, we have the following relationships for the optimal solutions.*

(a) *If a permutation $\sigma : [n] \to [n]$ and an integer k with $0 \leq k \leq n$ form an optimal solution for the maximum partial composition ordering problem $((f_i)_{i\in[n]}, c)$, then σ is also optimal for the maximum total composition ordering problem $((\overline{f}_i)_{i\in[n]}, c)$.*

(b) *Let $\sigma : [n] \to [n]$ denote an optimal permutation for the maximum total composition ordering problem $((\overline{f}_i)_{i\in[n]}, c)$. Let k denote the number of i's such that*

$$\overline{f}_{\sigma(i)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) > \overline{f}_{\sigma(i-1)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c), \tag{1}$$

and $\tau : [n] \to [n]$ denote a permutation such that $\tau(j)$ $(j \le k)$ is equal to the jth $\sigma(i)$ that satisfies (1). Then (τ, k) is optimal for the maximum partial composition ordering problem $((f_i)_{i\in[n]}, c)$.

Proof Let $\sigma : [n] \to [n]$ be a permutation and k be a nonnegative integer. Then we have

$$f_{\sigma(k)} \circ \cdots \circ f_{\sigma(1)}(c) \le \overline{f}_{\sigma(k)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) \le \overline{f}_{\sigma(n)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) \tag{2}$$

by $\overline{f}(x) \ge f(y)$ and $\overline{f}(x) \ge x$ for all reals x, y such that $x \ge y$. This implies that the objective value of the maximum partial composition ordering problem $((f_i)_{i\in[n]}, c)$ is at most the one of the maximum total composition ordering problem $((\overline{f}_i)_{i\in[n]}, c)$.

On the other hand, for a permutation $\sigma : [n] \to [n]$, let τ and k be defined as the statement in the lemma. Then we have

$$f_{\tau(k)} \circ \cdots \circ f_{\tau(1)}(c) = \overline{f}_{\tau(k)} \circ \cdots \circ \overline{f}_{\tau(1)}(c) = \overline{f}_{\sigma(n)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) \tag{3}$$

by the definition of τ, which implies that the objective value of the maximum partial composition ordering problem $((f_i)_{i\in[n]}, c)$ is at least the one of the maximum total composition ordering problem $((\overline{f}_i)_{i\in[n]}, c)$. Therefore, the objective values of the two problems are same.

Moreover, this together with (2) and (3) implies (a) and (b) in the lemma. □

From Lemmas 5 and 6, it is enough to consider the maximum total composition ordering problem. However, the properties of the functions f_i are not always inherited. For example, the partial composition ordering problem for linear functions does not correspond to the total one for the linear functions.

3 Maximum Partial Composition Ordering Problem

In this section, we discuss tractable results for the maximum partial composition ordering problem for monotone and almost-linear functions. By Lemma 6, we deal with the problem as the maximum total composition ordering problem for functions $\overline{f}_i$ $(i \in [n])$, where $\overline{f}_i(x) = \max\{f_i(x), x\}$. Let us start with the maximum partial composition ordering problem for monotone linear functions $f_i(x) = a_i x + b_i$ $(a_i \ge 0)$, i.e., the total composition ordering problem for $\overline{f}_i(x) = \max\{a_i x + b_i, x\}$ $(a_i \ge 0)$.

The following binary relation $\preceq$ plays an important role in the problem.

Definition 7 For two functions $f, g : \mathbb{R} \to \mathbb{R}$, we write $f \preceq g$ (or $g \succeq f$) if $f \circ g(x) \le g \circ f(x)$ for all $x \in \mathbb{R}$, $f \simeq g$ if $f \preceq g$ and $f \succeq g$ (i.e., $f \circ g(x) = g \circ f(x)$ for all $x \in \mathbb{R}$), and $f \prec g$ (or $g \succ f$) if $f \preceq g$ and $f \not\succeq g$.

Note that the relation $\preceq$ is not a *total relation* in general. Here a relation $\preceq$ is called total if any given two functions f_i and f_j are comparable (i.e., $f_i \preceq f_j$ or $f_j \preceq f_i$). For example, let $f_1(x) = \max\{2x, 3x\}$ and $f_2(x) = \max\{2x-1, 3x+1\}$. Then $f_1 \circ f_2(0)\ (=3)$ is greater than $f_2 \circ f_1(0)\ (=1)$, but $f_1 \circ f_2(-2)\ (=-10)$ is smaller than $f_2 \circ f_1(-2)\ (=-9)$.

However, if two consecutive functions are comparable, then we have the following easy but useful lemma.

Lemma 8 *Let $f_1, \ldots, f_n$ be monotone nondecreasing functions. If $f_i \preceq f_{i+1}$, then it holds that $f_n \circ \cdots \circ f_{i+2} \circ f_{i+1} \circ f_i \circ f_{i-1} \circ \cdots \circ f_1(x) \geq f_n \circ \cdots \circ f_{i+2} \circ f_i \circ f_{i+1} \circ f_{i-1} \circ \cdots \circ f_1(x)$ for all $x \in \mathbb{R}$.*

It follows from the lemma that, for monotone functions f_i, there exists a maximum total composition ordering $f_n \circ f_{n-1} \circ \cdots \circ f_1$ that satisfies $f_1 \preceq f_2 \preceq \cdots \preceq f_n$, if the relation is total. Moreover, if the relation $\preceq$ is in addition transitive (i.e., $f \preceq g$ and $g \preceq h$ imply $f \preceq h$), then it is not difficult to see that $f_1 \preceq f_2 \preceq \cdots \preceq f_n$ becomes a sufficient condition that $f_n \circ f_{n-1} \circ \cdots \circ f_1$ is a maximum total composition ordering, where the proof is given in a more general form in Lemma 10.

Fortunately, the relation is total if all functions are linear or of the form $\max\{ax + b, x\}$ with $a \geq 0$.

Lemma 9 *The relation $\preceq$ is total for linear functions.*

Proof Let $f_i(x) = a_i x + b_i$ and $f_j(x) = a_j x + b_j$. Then we have

$$\begin{aligned} f_i \preceq f_j &\iff f_i \circ f_j(x) \leq f_j \circ f_i(x) \text{ for all } x \in \mathbb{R} \\ &\iff a_i(a_j x + b_j) + b_i \leq a_j(a_i x + b_i) + b_j \text{ for all } x \in \mathbb{R} \\ &\iff b_i(1 - a_j) \leq b_j(1 - a_i). \end{aligned} \tag{4}$$

Since the last inequality consists of only constants, we have $f_i \preceq f_j$ or $f_i \succeq f_j$. □

When all functions are of the form $\max\{ax + b, x\}$ with $a \geq 0$, the totality of the relation is proven in Lemma 13.

We further note that the relation $\preceq$ is transitive for linear functions $f(x) = ax + b$ with $a > 1$, since (4) is equivalent to $b_i/(1-a_i) \leq b_j/(1-a_j)$, and hence the ordering $b_1/(1-a_1) \leq b_2/(1-a_2) \leq \cdots \leq b_n/(1-a_n)$ gives an optimal solution for the maximum total composition ordering problem. Therefore, it can be solved efficiently by sorting the elements by $b_i/(1-a_i)$. The same statement holds when all linear functions have slope less than 1. This idea is used for the linear deterioration and linear shortening models for time-dependent scheduling problems. However, in general, this is not the case, i.e., the relation $\preceq$ does not satisfy transitivity. Let $f_1(x) = 2x + 1$, $f_2(x) = 2x - 1$, and $f_3(x) = x/2$. Then we have $f_1 \prec f_2$, $f_2 \prec f_3$, and $f_3 \prec f_1$, which implies that the transitivity is not satisfied for linear functions, and also $\overline{f}_1 \prec \overline{f}_2$, $\overline{f}_2 \preceq \overline{f}_3$, and $\overline{f}_3 \prec \overline{f}_1$ hold, implying that the transitivity is not satisfied for the functions of the form $\max\{ax + b, x\}$ with $a \geq 0$. This shows that the maximum total and partial composition ordering problems are not trivial, even when all functions are monotone and linear.

We first show the following key lemma.

Lemma 10 *For monotone nondecreasing functions $f_i : \mathbb{R} \to \mathbb{R}$ ($i \in [n]$), if a permutation $\sigma : [n] \to [n]$ satisfies that $f_{\sigma(i)} \preceq f_{\sigma(j)}$ for all $i, j \in [n]$ such that $i < j$, then σ is an optimal solution for the maximum total composition ordering problem $((f_i)_{i\in[n]}, c)$ for all c.*

Proof Without loss of generality, we may assume that σ is the identity permutation. Let σ' be an optimal solution for the maximum total composition ordering problem such that it has the minimum inversion number, where the inversion number denotes the number of pairs (i, j) with $i < j$ and $\sigma'(i) > \sigma'(j)$. Then we show by contradiction that σ' is the identity permutation. Assume that $\sigma'(l) > \sigma'(l+1)$ for some l. Then consider the following permutation:

$$\tau(i) = \begin{cases} \sigma'(i) & (i \neq l,\ l+1), \\ \sigma'(l+1) & (i = l), \\ \sigma'(l) & (i = l+1). \end{cases}$$

Since $\sigma'(l+1) < \sigma'(l)$ implies $f_{\sigma'(l+1)} \preceq f_{\sigma'(l)}$ by the condition of the identity σ, Lemma 8 implies that τ is also optimal for the problem. Since τ has an inversion number smaller than the one for σ', we derive a contradiction. Therefore, σ' is the identity. □

As mentioned above, if the relation $\preceq$ is in addition transitive (i.e., $\preceq$ is a total preorder), then such a σ always exists.

To efficiently solve the maximum partial composition ordering problem for the linear functions, we show that for $\overline{f}_i(x) = \max\{a_i x + b_i, x\}$ ($a_i \geq 0$), (i) there exists a permutation σ which satisfies the condition in Lemma 10 and (ii) the permutation σ can be computed efficiently. Let us analyze the relation $\preceq$ in terms of the following function γ.

Definition 11 For a linear function $f(x) = ax + b$, we define

$$\gamma(f) = \begin{cases} \frac{b}{1-a} & (a \neq 1), \\ +\infty & (a = 1 \text{ and } b < 0), \\ -\infty & (a = 1 \text{ and } b \geq 0). \end{cases}$$

Note that $\gamma(f)$ is the solution of the equation $f(x) = x$ if $\gamma(f) \neq -\infty, +\infty$. Such points x have a name, they are called *fixpoint* (or *fixed point*).

In the rest of the paper, we assume without loss of generality that no f_i is the identity (i.e., $f_i(x) = x$), since we can ignore identity functions for both the total and partial composition problems.

Lemma 12 *Let $f_i(x) = a_i x + b_i$ and $f_j(x) = a_j x + b_j$ be monotone nondecreasing functions (i.e., $(a_i, b_i), (a_j, b_j) \neq (1, 0)$, $a_i, a_j \geq 0$). Then we have the following statements;*

(a) *if $a_i, a_j = 1$, then $f_i \simeq f_j$,*
(b) *if $a_i, a_j \geq 1$ and $a_i \cdot a_j > 1$, then $f_i \preceq f_j \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$,*

(c) *if* $a_i, a_j < 1$, *then* $f_i \preceq f_j \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$,
(d) *if* $a_i \geq 1$, $a_j < 1$, *then* $f_i \preceq f_j \Leftrightarrow \gamma(f_i) \geq \gamma(f_j)$ *and* $f_i \succeq f_j \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$.

Proof (a): It immediately follows from $f_i \circ f_j(x) = f_j \circ f_i(x) = x + b_i + b_j$.
(b): If $a_i, a_j > 1$, then the lemma holds, since we have the following equivalences (4)$\Leftrightarrow \frac{b_i}{1-a_i} \leq \frac{b_j}{1-a_j} \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$. If $a_i > 1$ and $a_j = 1$, then the lemma holds, since we have the following equivalences (4)$\Leftrightarrow 0 \leq b_j(1-a_i) \Leftrightarrow b_j < 0 \Leftrightarrow \gamma(f_j) = +\infty \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$. Otherwise (i.e., $a_i = 1$ and $a_j > 1$), we have (4)$\Leftrightarrow b_i(1-a_j) \leq 0 \Leftrightarrow b_i > 0 \Leftrightarrow \gamma(f_i) = -\infty \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$, which prove the lemma.
(c): The lemma holds, since we have the following equivalences (4)$\Leftrightarrow \frac{b_i}{1-a_i} \leq \frac{b_j}{1-a_j} \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$.
(d): If $a_i > 1$, the lemma holds since we have the following equivalences (4)$\Leftrightarrow \frac{b_i}{1-a_i} \geq \frac{b_j}{1-a_j} \Leftrightarrow \gamma(f_i) \geq \gamma(f_j)$. On the other hand, if $a_i = 1$, then $f_i \preceq f_j \Leftrightarrow b_i(1-a_j) \leq 0 \Leftrightarrow b_i < 0 \Leftrightarrow \gamma(f_i) = +\infty \Leftrightarrow \gamma(f_i) \geq \gamma(f_j)$, and $f_i \succeq f_j \Leftrightarrow b_i(1-a_j) \geq 0 \Leftrightarrow b_i > 0 \Leftrightarrow \gamma(f_i) = -\infty \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$. □

Lemma 13 *For monotone nondecreasing linear functions* $f_i(x) = a_i x + b_i$ *and* $f_j(x) = a_j x + b_j$, *we have the following statements;*

(a) *if* $a_i, a_j \geq 1$ *and* $\gamma(f_i) \leq \gamma(f_j)$, *then* $\overline{f}_i \preceq \overline{f}_j$,
(b) *if* $a_i, a_j < 1$ *and* $\gamma(f_i) \leq \gamma(f_j)$, *then* $\overline{f}_i \preceq \overline{f}_j$,
(c) *if* $a_i < 1$, $a_j \geq 1$, *and* $\gamma(f_i) \leq \gamma(f_j)$, *then* $\overline{f}_i \simeq \overline{f}_j$,
(d) *if* $a_i \geq 1$, $a_j < 1$, *and* $\gamma(f_i) \leq \gamma(f_j)$, *then* $\overline{f}_i \succeq \overline{f}_j$.

Proof (a): We prove that $\overline{f}_j \circ \overline{f}_i(x) \geq \overline{f}_i \circ \overline{f}_j(x)$ holds for all x. We separately consider three cases $x < \gamma(f_i)$, $\gamma(f_i) \leq x \leq \gamma(f_j)$, and $\gamma(f_j) < x$ (see Fig. 1a).
Case a-1: If $x < \gamma(f_i)$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(x) = x$ and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(x) = x$ by $x < \gamma(f_i) \leq \gamma(f_j)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_i \circ \overline{f}_j(x)$.
Case a-2: If $\gamma(f_i) \leq x \leq \gamma(f_j)$, then it holds that $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(x) = f_i(x)$ and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(f_i(x))$ by $\gamma(f_i) \leq x \leq \gamma(f_j)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) \geq \overline{f}_i \circ \overline{f}_j(x)$, since $\overline{f}_j(y) \geq y$ for all y.
Case a-3: If $\gamma(f_j) < x$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(f_j(x)) = f_i(f_j(x))$ by $\gamma(f_i) \leq \gamma(f_j) < x \leq f_j(x)$, and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(f_i(x)) = f_j(f_i(x))$ by $\gamma(f_i) \leq \gamma(f_j) < x \leq f_i(x)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) \geq \overline{f}_i \circ \overline{f}_j(x)$ by Lemma 12 (a) and (b).
(b): We prove that $\overline{f}_j \circ \overline{f}_i(x) \geq \overline{f}_i \circ \overline{f}_j(x)$ holds for all x. We separately consider four cases $x < f_j^{-1}(\gamma(f_i))$, $f_j^{-1}(\gamma(f_i)) \leq x < \gamma(f_i)$, $\gamma(f_i) \leq x < \gamma(f_j)$, and $\gamma(f_j) \leq x$ (see Fig. 1b).
Case b-1: If $x < f_j^{-1}(\gamma(f_i))$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(f_j(x)) = f_i(f_j(x))$ by $x \leq f_j(x) \leq \gamma(f_i) \leq \gamma(f_j)$, and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(f_i(x)) = f_j(f_i(x))$ by $x \leq f_i(x) \leq \gamma(f_i) \leq \gamma(f_j)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) \geq \overline{f}_i \circ \overline{f}_j(x)$ by Lemma 12 (c).

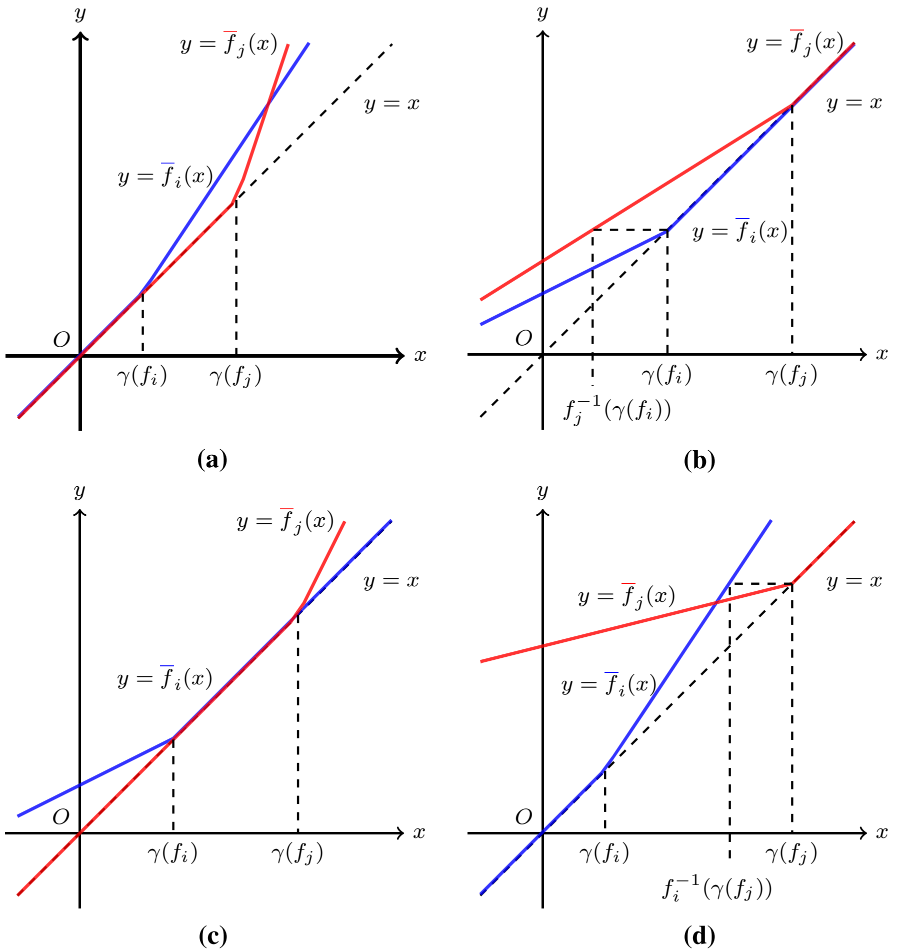

Fig. 1 Typical situations for functions $\overline{f}_i$ and $\overline{f}_j$ **a** $a_i, a_j \geq 1,\ \gamma(f_i) \leq \gamma(f_j)$, **b** $0 \leq a_i, a_j < 1,\ \gamma(f_i) \leq \gamma(f_j)$, **c** $0 \leq a_i < 1, a_j \geq 1,\ \gamma(f_i) \leq \gamma(f_j)$, **d** $a_i \geq 1,\ 0 \leq a_j < 1,\ \gamma(f_i) \leq \gamma(f_j)$

Case b-2: If $f_j^{-1}(\gamma(f_i)) \leq x < \gamma(f_i)$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(f_j(x)) = f_j(x)$ and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(f_i(x)) = f_j(f_i(x))$ by $x \leq f_i(x) \leq \gamma(f_i) \leq f_j(x) \leq \gamma(f_j)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) \geq \overline{f}_i \circ \overline{f}_j(x)$, since $f_i(x) \geq x$ and f_j is monotone nondecreasing.
Case b-3: If $\gamma(f_i) \leq x < \gamma(f_j)$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(f_j(x)) = f_j(x)$ and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(x) = f_j(x)$ by $\gamma(f_i) \leq x \leq f_j(x) < \gamma(f_j)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_i \circ \overline{f}_j(x)$.
Case b-4: If $\gamma(f_j) \leq x$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(x) = x$ and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(x) = x$ by $\gamma(f_i) \leq \gamma(f_j) \leq x$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_i \circ \overline{f}_j(x)$.
(c): We prove that $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_i \circ \overline{f}_j(x)$ holds for all x. We separately consider three cases $x < \gamma(f_i)$, $\gamma(f_i) \leq x < \gamma(f_j)$, and $\gamma(f_j) \leq x$ (see Fig. 1c).

Case c-1: If $x < \gamma(f_i)$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(x) = f_i(x)$ and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(f_i(x)) = f_i(x)$ by $x \le f_i(x) \le \gamma(f_i) \le \gamma(f_j)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_i \circ \overline{f}_j(x)$.
Case c-2: If $\gamma(f_i) \le x < \gamma(f_j)$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(x) = x$ and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(x) = x$ by $\gamma(f_i) \le x < \gamma(f_j)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_i \circ \overline{f}_j(x)$.
Case c-3: If $\gamma(f_j) \le x$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(f_j(x)) = f_j(x)$ and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(x) = f_j(x)$ by $\gamma(f_i) \le \gamma(f_j) \le x \le f_j(x)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_i \circ \overline{f}_j(x)$.
(d): We prove that $\overline{f}_j \circ \overline{f}_i(x) \le \overline{f}_i \circ \overline{f}_j(x)$ holds for all x. We separately consider four cases $x < \gamma(f_i)$, $\gamma(f_i) \le x < f_i^{-1}(\gamma(f_j))$, $f_i^{-1}(\gamma(f_j)) \le x < \gamma(f_j)$, and $\gamma(f_j) \le x$ (see Fig. 1d).
Case d-1: If $x < \gamma(f_i)$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(f_j(x))$ and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(x) = f_j(x)$ by $x < \gamma(f_i) \le \gamma(f_j)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) \le \overline{f}_i \circ \overline{f}_j(x)$, since $\overline{f}_i(y) \ge y$ for all y.
Case d-2: If $\gamma(f_i) \le x < f_i^{-1}(\gamma(f_j))$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(f_j(x)) = f_i(f_j(x))$ by $\gamma(f_i) \le x \le f_j(x) \le \gamma(f_j)$, and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(f_i(x)) = f_j(f_i(x))$ by $\gamma(f_i) \le x \le f_i(x) \le \gamma(f_j)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) \le \overline{f}_i \circ \overline{f}_j(x)$ by Lemma 12 (d).
Case d-3: If $f_i^{-1}(\gamma(f_j)) \le x < \gamma(f_j)$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(f_j(x)) = f_i(f_j(x))$ by $\gamma(f_i) \le f_i^{-1}(\gamma(f_j)) \le x \le f_j(x) \le \gamma(f_j)$ and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(f_i(x)) = f_i(x)$ by $\gamma(f_i) \le f_i^{-1}(\gamma(f_j)) \le x \le \gamma(f_j) \le f_i(x)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) \le \overline{f}_i \circ \overline{f}_j(x)$, since $f_j(x) \ge x$ and f_i is monotone nondecreasing.
Case d-4: If $\gamma(f_j) \le x$, then we have $\overline{f}_i \circ \overline{f}_j(x) = \overline{f}_i(x) = f_i(x)$ and $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_j(f_i(x)) = f_i(x)$ by $\gamma(f_i) \le \gamma(f_j) \le x \le f_i(x)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_i \circ \overline{f}_j(x)$. □

Note that Lemma 13 implies that the relation $\preceq$ is total for the functions of the form $\max\{ax + b, x\}$ with $a \ge 0$ because at least one of $\overline{f}_i \preceq \overline{f}_j$ or $\overline{f}_i \succeq \overline{f}_j$ holds for any monotone nondecreasing linear functions f_i and f_j. Moreover, it implies that the following permutation σ satisfies the condition in Lemma 10.

For a linear function $f(x) = ax + b$, let

$$\delta(f) = \begin{cases} +1 & (a \ge 1), \\ -1 & (a < 1). \end{cases}$$

Let $\sigma : [n] \to [n]$ denote a permutation that is *compatible* with the lexicographic ordering with respect to $(\delta(f_i), \gamma(f_i))$, i.e., $(\delta(f_{\sigma(i)}), \gamma(f_{\sigma(i)}))$ is lexicographically smaller than or equal to $(\delta(f_{\sigma(j)}), \gamma(f_{\sigma(j)}))$ if $i < j$. Namely, σ is a permutation such that $\delta(f_{\sigma(1)}) = \cdots = \delta(f_{\sigma(k)}) = -1$, $\delta(f_{\sigma(k+1)}) = \cdots = \delta(f_{\sigma(n)}) = +1$, $\gamma(f_{\sigma(1)}) \le \cdots \le \gamma(f_{\sigma(k)})$, and $\gamma(f_{\sigma(k+1)}) \le \cdots \le \gamma(f_{\sigma(n)})$ where k is the number of functions f_i with slope smaller than 1. Then we have the following lemma by Lemma 13.

Lemma 14 *For monotone nondecreasing linear functions f_i $(i \in [n])$, let σ denote a permutation compatible with the lexicographic order with respect to $(\delta(f_i), \gamma(f_i))$. Then $\overline{f}_{\sigma(i)} \preceq \overline{f}_{\sigma(j)}$ holds for any $i, j \in [n]$ such that $i \leq j$.*

By Lemmas 10 and 14, the lexicographic order with respect to $(\delta(f_i), \gamma(f_i))$ is an optimal solution for the maximum total composition ordering problem for the functions $\overline{f}_i$ such that f_i's are monotone nondecreasing linear functions. As we noted at the beginning of this section, the maximum partial composition ordering problem for monotone linear functions f_i $(i \in [n])$ can be formulated as the maximum total composition ordering problem for functions $\overline{f}_i$ $(i \in [n])$. Therefore, the maximum partial composition ordering problem can be solved in $\mathrm{O}(n \log n)$ time, which proves the partial composition part of Theorem 1. We remark that the time complexity $\mathrm{O}(n \log n)$ of the problem is the best possible in the comparison model. We also remark that the optimal value for the maximum partial composition ordering problem for $f_i(x) = a_i x + b_i$ $(a_i \geq 0)$ forms a piecewise linear function (in c) with at most $(n+1)$ pieces.

We next extend this tractability result to Theorem 4. For $i \in [n]$, let $h_i(x) = a_i x + b_i$ be a monotone nondecreasing linear function, and let $f_i(x) = \max\{h_i(x), d_i\}$ for a constant d_i. We consider the maximum partial composition ordering problem for f_i's and the maximum total composition ordering problem for $\overline{f}_i$'s where

$$\overline{f}_i(x) = \max\{a_i x + b_i, d_i, x\}. \tag{5}$$

Here, by Lemma 6, it is sufficient to consider the maximum total composition ordering problem $((\overline{f}_i)_{i \in [n]}, c)$.

Lemma 15 *Let $c \in \mathbb{R}$, and let $\overline{f}_i$ $(i \in [n])$ be a function defined as (5). Then there exists an optimal solution σ for the maximum total composition ordering problem $((\overline{f}_i)_{i \in [n]}, c)$ such that no $i > 1$ satisfies $\overline{h}_{\sigma(i)} \circ \overline{f}_{\sigma(i-1)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) < d_{\sigma(i)}$, where $\overline{h}_i(x) = \max\{a_i x + b_i, x\}$.*

Proof Let σ denote an optimal solution for the problem. Assume that there exists an index i that satisfies the condition in the lemma. Let i^* denote the largest such i. Then by the definition of i^*, we have

$$\begin{aligned}
\overline{f}_{\sigma(i^*)} \circ \overline{f}_{\sigma(i^*-1)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) &= \max\{\overline{h}_{\sigma(i^*)} \circ \overline{f}_{\sigma(i^*-1)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c), d_{\sigma(i^*)}\} \\
&= d_{\sigma(i^*)} \\
&= \max\{\overline{h}_{\sigma(i^*)}(c), d_{\sigma(i^*)}\} = \overline{f}_{\sigma(i^*)}(c).
\end{aligned}$$

Here, the third equality holds since $\overline{h}_{\sigma(i^*)}(c) \leq \overline{h}_{\sigma(i^*)} \circ \overline{f}_{\sigma(i^*-1)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) < d_{\sigma(i^*)}$ by the definition of i^* and $c \leq \overline{f}_{\sigma(i^*-1)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c)$. Also, it holds that $d_{\sigma(i)} < d_{\sigma(i^*)}$ for all $i < i^*$, since

$$\begin{aligned}
d_{\sigma(i)} &\leq \overline{f}_{\sigma(i)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) \\
&\leq \overline{f}_{\sigma(i^*-1)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) \\
&\leq \overline{h}_{\sigma(i^*)} \circ \overline{f}_{\sigma(i^*-1)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) < d_{\sigma(i^*)}.
\end{aligned}$$

Thus, we have

$$\begin{aligned}\overline{f}_{\sigma(n)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) &= \overline{f}_{\sigma(n)} \circ \cdots \circ \overline{f}_{\sigma(i^*)}(c)\\ &\leq \overline{f}_{\sigma(i^*-1)} \circ \cdots \circ \overline{f}_{\sigma(1)} \circ \overline{f}_{\sigma(n)} \circ \cdots \circ \overline{f}_{\sigma(i^*)}(c).\end{aligned}$$

This implies that $(\sigma(i^*), \ldots \sigma(n), \sigma(1) \ldots, \sigma(i^*-1))$ is also an optimal permutation for the problem. Moreover, in the composition according to this permutation, the constant part of $\overline{f}_i$ $(i \neq i^*)$ is not explicitly used by the definition of i^* and $d_{\sigma(i)} < d_{\sigma(i^*)}$ for all i $(< i^*)$, which completes the proof. □

Proof of Theorem 4 It follows from Lemma 15 that an optimal solution for the problem can be obtained by solving the following $n+1$ instances of the maximum partial composition ordering problem for monotone nondecreasing linear functions:

$$((h_i)_{i \in [n] \setminus \{1\}}, d_1), \ldots, ((h_i)_{i \in [n] \setminus \{n\}}, d_n), \text{ and } ((h_i)_{i \in [n]}, c).$$

Therefore, we have an $O(n^2 \log n)$-time algorithm by directly applying Theorem 1 to the problem. Moreover, we note that the maximum partial composition ordering problem for monotone nondecreasing linear functions can be solved in linear time, if we know the lexicographic ordering with respect to $(\delta(h_i), \gamma(h_i))$. This implies that the problem can be solved in $O(n^2)$ time by computing the lexicographic order with respect to $(\delta(h_i), \gamma(h_i))$ only once. □

4 Maximum Total Composition Ordering Problem

In this section we prove the total composition part of Theorems 1 and 2.

We start with several lemmas to prove the theorems. Note that no f_i is the identity function.

Lemma 16 *For any real c and linear function $f(x) = ax+b$, the following statements hold.*

(a) *If $a > 1$, then $f(c) > c \Leftrightarrow \gamma(f) < c$, $f(c) < c \Leftrightarrow \gamma(f) > c$, and $f(c) = c \Leftrightarrow \gamma(f) = c$.*
(b) *If $a < 1$, then $f(c) > c \Leftrightarrow \gamma(f) > c$, $f(c) < c \Leftrightarrow \gamma(f) < c$, and $f(c) = c \Leftrightarrow \gamma(f) = c$.*
(c) *If $a = 1$, then $f(c) > c \Leftrightarrow \gamma(f) = -\infty$, $f(c) < c \Leftrightarrow \gamma(f) = +\infty$.*

Proof (a) Note that $f(\gamma(f)) = \gamma(f)$ and $f(x) - x$ is monotone increasing if $a > 1$. Therefore the equivalences hold.
(b) Note that $f(\gamma(f)) = \gamma(f)$ and $f(x) - x$ is monotone decreasing if $a < 1$. Therefore the equivalences hold.
(c) This statement directly follows from the definition of γ.

□

The following lemma shows the relationships between $\gamma(f_i)$, $\gamma(f_j)$, $\gamma(f_j \circ f_i)$, and $\gamma(f_i \circ f_j)$ for monotone linear functions.

Lemma 17 *For monotone nondecreasing linear functions $f_i(x) = a_i x + b_i$ and $f_j(x) = a_j x + b_j$ ($a_i, a_j \geq 0$), we have the following statements.*

(a) *If $\gamma(f_i) = \gamma(f_j)$, then $\gamma(f_i) = \gamma(f_j) = \gamma(f_j \circ f_i)$ or $f_j \circ f_i$ is the identity function,*
(b) *If $\gamma(f_i) < \gamma(f_j)$ and $a_i, a_j \geq 1$, then $\gamma(f_i) \leq \gamma(f_j \circ f_i) \leq \gamma(f_j)$,*
(c) *If $\gamma(f_i) < \gamma(f_j)$ and $a_i, a_j < 1$, then $\gamma(f_i) \leq \gamma(f_j \circ f_i) \leq \gamma(f_j)$,*
(d) *If $\gamma(f_i) < \gamma(f_j)$, $a_i < 1$, $a_j \geq 1$, and $a_i \cdot a_j \geq 1$, then $\gamma(f_i \circ f_j) \geq \gamma(f_j)$ ($> \gamma(f_i)$),*
(e) *If $\gamma(f_i) < \gamma(f_j)$, $a_i < 1$, $a_j \geq 1$, and $a_i \cdot a_j < 1$, then $\gamma(f_i \circ f_j) \leq \gamma(f_i)$ ($< \gamma(f_j)$),*
(f) *If $\gamma(f_i) < \gamma(f_j)$, $a_i \geq 1$, $a_j < 1$, and $a_i \cdot a_j \geq 1$, then $\gamma(f_i \circ f_j) \leq \gamma(f_i)$ ($< \gamma(f_j)$),*
(g) *If $\gamma(f_i) < \gamma(f_j)$, $a_i \geq 1$, $a_j < 1$, and $a_i \cdot a_j < 1$, then $\gamma(f_i \circ f_j) \geq \gamma(f_j)$ ($> \gamma(f_i)$).*

Proof (a) Let $d = \gamma(f_i) = \gamma(f_j)$. If $d = +\infty$, then $a_i = a_j = 1$ and $b_i, b_j < 0$. Thus, $\gamma(f_j \circ f_i) = \gamma(x + b_i + b_j) = +\infty$. If $d = -\infty$, then $a_i = a_j = 1$ and $b_i, b_j > 0$. Thus, $\gamma(f_j \circ f_i) = \gamma(x + b_i + b_j) = -\infty$. Otherwise (i.e., $a_i, a_j \neq 1$), we have $f_i(x) = a_i(x - d) + d$ and $f_j(x) = a_j(x - d) + d$. Therefore, $f_j \circ f_i(x) = a_i a_j (x - d) + d$ and hence the claim holds.

(b) If $a_i = a_j = 1$, then we have $\gamma(f_i) = -\infty \leq \gamma(f_j \circ f_i) \leq +\infty = \gamma(f_j)$. Thus we assume that $a_i \cdot a_j > 1$. By Lemma 16 (a) and (c) and by $\gamma(f_i) < \gamma(f_j)$, we have

$$f_j \circ f_i(\gamma(f_i)) = f_j(\gamma(f_i)) \leq \gamma(f_i), \tag{6}$$
$$f_j \circ f_i(\gamma(f_j)) \geq f_j(\gamma(f_j)) = \gamma(f_j). \tag{7}$$

We obtain $\gamma(f_i) \leq \gamma(f_j \circ f_i)$ by $\gamma(f_i) = -\infty \leq \gamma(f_j \circ f_i)$ if $a_i = 1$ and by (6) and Lemma 16 (a) if $a_i > 1$. Also, we get $\gamma(f_j \circ f_i) \leq \gamma(f_j)$ by $\gamma(f_j \circ f_i) \leq +\infty = \gamma(f_j)$ if $a_j = 1$ and by (7) and Lemma 16 (a) if $a_j > 1$.

(c) By Lemma 16 (b) and $\gamma(f_i) < \gamma(f_j)$, we have

$$f_j \circ f_i(\gamma(f_i)) = f_j(\gamma(f_i)) \geq \gamma(f_i), \tag{8}$$
$$f_j \circ f_i(\gamma(f_j)) \leq f_j(\gamma(f_j)) = \gamma(f_j). \tag{9}$$

Therefore, we obtain $\gamma(f_i) \leq \gamma(f_j \circ f_i) \leq \gamma(f_j)$ where the first inequality holds by (8) and by Lemma 16 (b), and the second inequality holds by (9) and by Lemma 16 (b).

(d) Note that $a_j > 1$ since $a_i \cdot a_j \geq 1$. By Lemma 16 (b) and $\gamma(f_i) < \gamma(f_j)$, we have

$$f_i \circ f_j(\gamma(f_j)) = f_i(\gamma(f_j)) < \gamma(f_j).$$

Therefore, we obtain $\gamma(f_i \circ f_j) \geq \gamma(f_j)$ by Lemma 16 (a) and (c).

(e) By Lemma 16 (a)and (c) and by $\gamma(f_i) < \gamma(f_j)$, we have

$$f_i \circ f_j(\gamma(f_i)) \leq f_i(\gamma(f_i)) = \gamma(f_i).$$

Therefore, we obtain $\gamma(f_i \circ f_j) \leq \gamma(f_i)$ by Lemma 16 (b).

(f) Note that $a_i > 1$ and $a_j > 0$ since $a_i \cdot a_j \geq 1$. By Lemma 16 (b) and $\gamma(f_i) < \gamma(f_j)$, we have

$$f_i \circ f_j(\gamma(f_i)) > f_i(\gamma(f_i)) = \gamma(f_i)$$

since f_i and f_j are monotone increasing. Therefore, we obtain $\gamma(f_i \circ f_j) \leq \gamma(f_i)$ by Lemma 16 (a) and (c).

(g) By Lemma 16 (a) and (c) and by $\gamma(f_i) < \gamma(f_j)$, we have

$$f_i \circ f_j(\gamma(f_j)) = f_i(\gamma(f_j)) \geq \gamma(f_j).$$

Therefore, we obtain $\gamma(f_i \circ f_j) \geq \gamma(f_j)$ by Lemma 16 (b).

□

By Lemmas 12 and 17, we have the following inequalities for compositions of four functions.

Lemma 18 *For monotone nondecreasing linear functions* $f_i(x) = a_i x + b_i$ $(i = 1, 2, 3, 4)$*, if* $a_1, a_3 \geq 1$*,* $a_2, a_4 < 1$ *and* $\gamma(f_1) \geq \gamma(f_2) \geq \gamma(f_3) \geq \gamma(f_4)$*, then we have*

$$f_4 \circ f_3 \circ f_2 \circ f_1(x) \leq \max\{f_4 \circ f_1 \circ f_3 \circ f_2(x),\ f_3 \circ f_2 \circ f_4 \circ f_1(x)\} \quad (\forall x \in \mathbb{R}).$$

Lemma 19 *For monotone nondecreasing linear functions* $f_i(x) = a_i x + b_i$ $(i = 1, 2, 3, 4)$*, if* $a_1, a_3 < 1$*,* $a_2, a_4 \geq 1$ *and* $\gamma(f_1) \geq \gamma(f_2) \geq \gamma(f_3) \geq \gamma(f_4)$*, then we have*

$$f_4 \circ f_3 \circ f_2 \circ f_1(x) \leq \max\{f_4 \circ f_1 \circ f_3 \circ f_2(x),\ f_3 \circ f_2 \circ f_4 \circ f_1(x)\} \quad (\forall x \in \mathbb{R}).$$

Proof We only prove Lemma 18 since the proof of Lemma 19 is almost the same. Let $g(x) = f_3 \circ f_2(x)$. We assume that g is not the identity function since otherwise the lemma is clear. If $a_2 \cdot a_3 \geq 1$, then $\gamma(g) \leq \gamma(f_3) \leq \gamma(f_1)$ holds by Lemma 17 (a) and (f), and $g \circ f_1(x) \leq f_1 \circ g(x)$ holds by Lemma 12 (a) and (b). Thus, we have $f_4 \circ f_3 \circ f_2 \circ f_1(x) \leq f_4 \circ f_1 \circ f_3 \circ f_2(x)$. On the other hand, if $a_2 \cdot a_3 < 1$, then $\gamma(g) \geq \gamma(f_2) \geq \gamma(f_4)$ holds Lemma 17 (a) and (g), and $f_4 \circ g(x) \leq g \circ f_4(x)$ holds by Lemma 12 (c) . Thus, we have $f_4 \circ f_3 \circ f_2 \circ f_1(x) \leq f_3 \circ f_2 \circ f_4 \circ f_1(x)$. □

By Lemmas 18 and 19, we obtain the following lemma.

Lemma 20 *There exists an optimal permutation* σ *for the maximum total composition ordering problem for monotone nondecreasing functions* f_i $(i \in [n])$ *such that at most two* i*'s satisfy* $\delta(f_{\sigma(i)}) \cdot \delta(f_{\sigma(i+1)}) = -1$.

Proof Let σ be an optimal solution with the minimum number of i's satisfying $\delta(f_{\sigma(i)})\cdot\delta(f_{\sigma(i+1)}) = -1$. Assume that σ contains at least three such i's. Let i_1, i_2 and i_3 denote the three smallest such i's with $i_1 < i_2 < i_3$, and i_4 denote the fourth smallest such i if exists; otherwise we define $i_4 = n$. Let $g_1(x) = f_{\sigma(i_1)} \circ \cdots \circ f_{\sigma(1)}(x)$, $g_2(x) = f_{\sigma(i_2)} \circ \cdots \circ f_{\sigma(i_1+1)}(x)$, $g_3(x) = f_{\sigma(i_3)} \circ \cdots \circ f_{\sigma(i_2+1)}(x)$, and $g_4(x) = f_{\sigma(i_4)} \circ \cdots \circ f_{\sigma(i_3+1)}(x)$. Then it is easy to see that $\delta(g_1) = -\delta(g_2) = \delta(g_3) = -\delta(g_4)$. We claim that $\gamma(g_1) \geq \gamma(g_2) \geq \gamma(g_3) \geq \gamma(g_4)$. Assume that $\gamma(g_j) < \gamma(g_{j+1})$ for some $j \in \{1, 2, 3\}$. Then it follows from Lemma 12 (d) that $g_{j+1} \circ g_j(x) \leq g_j \circ g_{j+1}(x)$ holds, which contradicts the assumption on σ. Therefore we have

$$\begin{aligned} f_{\sigma(n)} \circ \cdots \circ f_{\sigma(1)}(x) &= f_{\sigma(n)} \circ \cdots \circ f_{\sigma(i_4+1)} \circ g_4 \circ g_3 \circ g_2 \circ g_1(x) \\ &\leq \max \left\{ \begin{array}{l} f_{\sigma(n)} \circ \cdots \circ f_{\sigma(i_4+1)} \circ g_4 \circ g_1 \circ g_3 \circ g_2(x), \\ f_{\sigma(n)} \circ \cdots \circ f_{\sigma(i_4+1)} \circ g_3 \circ g_2 \circ g_4 \circ g_1(x) \end{array} \right\} \end{aligned}$$

by Lemmas 18 and 19. This again contradicts the assumption on σ. $\square$

Next, we provide inequalities for compositions of three functions.

Lemma 21 *For monotone nondecreasing linear functions $f_i(x) = a_i x + b_i$ $(i = 1, 2, 3)$, if $a_1, a_3 \geq 1$, $a_2 < 1$, $a_1 \cdot a_2 \cdot a_3 \geq 1$ and $\gamma(f_1) \geq \gamma(f_2) \geq \gamma(f_3)$, then we have*

$$f_3 \circ f_2 \circ f_1(x) \leq \max\{f_2 \circ f_1 \circ f_3(x),\ f_1 \circ f_3 \circ f_2(x)\} \quad (\forall x \in \mathbb{R}).$$

Lemma 22 *For monotone nondecreasing linear functions $f_i(x) = a_i x + b_i$ $(i = 1, 2, 3)$, if $a_1, a_3 < 1$, $a_2 \geq 1$, $a_1 \cdot a_2 \cdot a_3 < 1$ and $\gamma(f_1) \geq \gamma(f_2) \geq \gamma(f_3)$, then we have*

$$f_3 \circ f_2 \circ f_1(x) \leq \max\{f_2 \circ f_1 \circ f_3(x),\ f_1 \circ f_3 \circ f_2(x)\} \quad (\forall x \in \mathbb{R}).$$

Proof We only prove Lemma 21 since the proof of Lemma 22 is almost the same. We assume that both of $f_2 \circ f_1$ and $f_3 \circ f_2$ are not the identity function since otherwise the lemma is clear. If $a_2 \cdot a_3 \geq 1$, then $\gamma(f_3 \circ f_2) \leq \gamma(f_3) \leq \gamma(f_1)$ by Lemma 17 (a) and (f), and it implies $f_3 \circ f_2 \circ f_1(x) \leq f_1 \circ f_3 \circ f_2(x)$ by Lemma 12 (a) and (b). If $a_2 \cdot a_3 < 1$ and $\gamma(f_3 \circ f_2) \geq \gamma(f_1)$, then $f_3 \circ f_2 \circ f_1(x) \leq f_1 \circ f_3 \circ f_2(x)$ by Lemma 12 (d).

If $a_1 \cdot a_2 \geq 1$, then $\gamma(f_2 \circ f_1) \geq \gamma(f_1) \geq \gamma(f_3)$ by Lemma 17 (a) and (d), and it implies $f_3 \circ f_2 \circ f_1(x) \leq f_2 \circ f_1 \circ f_3(x)$ by Lemma 12 (a) and (b). If $a_1 \cdot a_2 < 1$ and $\gamma(f_2 \circ f_1) \leq \gamma(f_3)$, then $f_3 \circ f_2 \circ f_1(x) \leq f_2 \circ f_1 \circ f_3(x)$ by Lemma 12 (d).

Otherwise, we have $a_2 \cdot a_3 < 1$, $a_1 \cdot a_2 < 1$, $\gamma(f_3 \circ f_2) < \gamma(f_1)$, and $\gamma(f_2 \circ f_1) > \gamma(f_3)$. Then we have $\gamma((f_3 \circ f_2) \circ f_1) \geq \gamma(f_1)$ by Lemma 17 (d), and $\gamma(f_3 \circ (f_2 \circ f_1)) \leq \gamma(f_3)$ by Lemma 17 (f) since $a_1 \cdot a_2 \cdot a_3 \geq 1$. Therefore $\gamma(f_1) = \gamma(f_2) = \gamma(f_3)$, This together with $\gamma(f_3 \circ f_2) < \gamma(f_1)$ contradicts Lemma 17 (a). $\square$

By Lemmas 12, 20–22, we get the following lemmas.

Lemma 23 *If $\prod_{i=1}^n a_i \geq 1$, then there exists an optimal permutation σ such that, for some two integers s, t $(0 \leq s \leq t \leq n)$, $\delta(f_{\sigma(t+1)}) = \cdots = \delta(f_{\sigma(n)}) = \delta(f_{\sigma(1)}) = \cdots = \delta(f_{\sigma(s)}) = -1$, $\delta(f_{\sigma(s+1)}) = \cdots = \delta(f_{\sigma(t)}) = 1$, $\gamma_{\sigma(t+1)} \leq \cdots \leq \gamma_{\sigma(n)} \leq \gamma_{\sigma(1)} \leq \cdots \leq \gamma_{\sigma(s)}$, and $\gamma_{\sigma(s+1)} \leq \cdots \leq \gamma_{\sigma(t)}$.*

Lemma 24 *If $\prod_{i=1}^n a_i < 1$, then there exists an optimal permutation σ such that, for some two integers s, t $(0 \leq s \leq t \leq n)$, $\delta(f_{\sigma(t+1)}) = \cdots = \delta(f_{\sigma(n)}) = \delta(f_{\sigma(1)}) = \cdots = \delta(f_{\sigma(s)}) = 1$, $\delta(f_{\sigma(s+1)}) = \cdots = \delta(f_{\sigma(t)}) = -1$, $\gamma_{\sigma(t+1)} \leq \cdots \leq \gamma_{\sigma(n)} \leq \gamma_{\sigma(1)} \leq \cdots \leq \gamma_{\sigma(s)}$, and $\gamma_{\sigma(s+1)} \leq \cdots \leq \gamma_{\sigma(t)}$.*

Proof We only prove Lemma 23 since the proof of Lemma 24 is almost the same. By Lemma 20, there exists an optimal permutation σ and two integers s, t $(0 \leq s \leq t \leq n)$ such that $\delta(f_{\sigma(1)}) = \cdots = \delta(f_{\sigma(s)}) = -\delta(f_{\sigma(s+1)}) = \cdots = -\delta(f_{\sigma(t)}) = \delta(f_{\sigma(t+1)}) = \cdots = \delta(f_{\sigma(n)})$. By Lemma 12, we have

$$\gamma_{\sigma(1)} \leq \cdots \leq \gamma_{\sigma(s)}, \ \gamma_{\sigma(s+1)} \leq \cdots \leq \gamma_{\sigma(t)}, \ \gamma_{\sigma(t+1)} \leq \cdots \leq \gamma_{\sigma(n)}.$$

This implies that the lemma holds when $s = 0$ or $t = n$. For $0 < s \leq t < n$, we separately consider the following two cases.
Case 1 If $\delta(f_{\sigma(s+1)}) = \cdots = \delta(f_{\sigma(t)}) = +1$, let $g = f_{\sigma(n-1)} \circ \cdots \circ f_{\sigma(2)}$. Then Lemma 12 and the optimality of σ imply $\gamma(f_{\sigma(1)}) \geq \gamma(g) \geq \gamma(f_{\sigma(n)})$, since $-\delta(f_{\sigma(1)}) = \delta(g) = -\delta(f_{\sigma(n)}) = +1$. This proves the lemma.
Case 2 If $\delta(f_{\sigma(s+1)}) = \cdots = \delta(f_{\sigma(t)}) = -1$, then let $h_1 = f_{\sigma(s)} \circ \cdots \circ f_{\sigma(1)}$, $h_2 = f_{\sigma(t)} \circ \cdots \circ f_{\sigma(s+1)}$ and $h_3 = f_{\sigma(n)} \circ \cdots \circ f_{\sigma(t+1)}$. If $\gamma(h_1) < \gamma(h_2)$, then $h_3 \circ h_2 \circ h_1(x) \leq h_3 \circ h_1 \circ h_2(x)$ by Lemma 12 (d). If $\gamma(h_2) < \gamma(h_3)$, then $h_3 \circ h_2 \circ h_1(x) \leq h_2 \circ h_3 \circ h_1(x)$ by Lemma 12 (d). Otherwise (i.e., $\gamma(h_1) \geq \gamma(h_2) \geq \gamma(h_3)$), we have

$$h_3 \circ h_2 \circ h_1(x) \leq \max\{h_2 \circ h_1 \circ h_3(x), \ h_1 \circ h_3 \circ h_2(x)\}$$

by Lemma 21. In either case, we can obtain a desired optimal solution by modifying σ. □

By Lemmas 23 and 24, we obtain polynomial time algorithm for the maximum total composition ordering problem for monotone nondecreasing linear functions. The outline of the algorithm is shown in Algorithm 1.

Proof of the total composition part of Theorem 1 By Lemmas 23 and 24, the total composition ordering problem for monotone nondecreasing linear functions can be computed as follows. Let $\sigma : [n] \to [n]$ be a permutation which satisfies $\delta(f_{\sigma(1)}) = \cdots = \delta(f_{\sigma(r)}) = -1$, $\delta(f_{\sigma(r+1)}) = \cdots = \delta(f_{\sigma(n)}) = 1$, $\gamma(f_{\sigma(1)}) \leq \cdots \leq \gamma(f_{\sigma(r)})$, and $\gamma(f_{\sigma(r+1)}) \leq \cdots \leq \gamma(f_{\sigma(n)})$. Then Lemmas 23 and 24 implies that there exists an optimal solution of the form

$$(\sigma(t), \sigma(t+1), \ldots, \sigma(n), \sigma(1), \sigma(2), \ldots, \sigma(t-1))$$

for some t. Therefore, the problem can be computed in polynomial time by checking n permutations above. To reduce the time complexity, let $d_k = f_{\sigma(k-1)} \circ \cdots \circ f_{\sigma(1)} \circ f_{\sigma(n)} \circ \cdots \circ f_{\sigma(k)}(c)$ for $k = 1, \ldots, n$. Let $a = \prod_{i=1}^n a_i$. Then, if $a_{\sigma(k)} > 0$, we have

Algorithm 1: Maximum Total Composition

$I_- \leftarrow \{i : \delta(f_i) = -1\}$, $I_+ \leftarrow \{i : \delta(f_i) = 1\}$;
sort I_- and I_+ according to the order induced by γ_i;
let $I_- = \{\sigma(1), \dots, \sigma(r)\}$, $I_+ = \{\sigma(r+1), \dots, \sigma(n)\}$ such that $\gamma_{\sigma(1)} \le \cdots \le \gamma_{\sigma(r)}$ and $\gamma_{\sigma(r+1)} \le \cdots \le \gamma_{\sigma(n)}$.;
$s \leftarrow c, a \leftarrow 1$;
for $i = 1$ *to* n **do**
  $s \leftarrow f_{\sigma(i)}(s)$;
  $a \leftarrow a \cdot a_i$;
end
$s^* \leftarrow s, t \leftarrow 1$;
for $i = 1$ *to* $n - 1$ **do**
  $s \leftarrow a_{\sigma(i)} \cdot (s - c) - b_{\sigma(i)} \cdot (a - 1) + c$;
  if $s > s^*$ **then** $s^* \leftarrow s, t \leftarrow i + 1$;
end
return $(\sigma(t), \sigma(t+1), \dots, \sigma(n), \sigma(1), \sigma(2), \dots, \sigma(t-1))$;

$$\begin{aligned}
d_{k+1} &= f_{\sigma(k)} \circ \cdots \circ f_{\sigma(1)} \circ f_{\sigma(n)} \circ \cdots \circ f_{\sigma(k+1)}(c) \\
&= f_{\sigma(k)} \circ f_{\sigma(k-1)} \circ \cdots \circ f_{\sigma(1)} \circ f_{\sigma(n)} \circ \cdots \circ f_{\sigma(k)}(f_{\sigma(k)}^{-1}(c)) \\
&= a_{\sigma(k)} \left(d_k - a \cdot c + \frac{c - b_{\sigma(k)}}{a_{\sigma(k)}} \cdot a \right) + b_{\sigma(k)} \\
&= a_{\sigma(k)} \cdot (d_k - a \cdot c) - b_{\sigma(k)} \cdot (a - 1) + a \cdot c.
\end{aligned}$$

Also, if $a_{\sigma(k)} = 0$, we have $d_{k+1} = b_{\sigma(k)} = a_{\sigma(k)} \cdot (d_k - a \cdot c) - b_{\sigma(k)} \cdot (a - 1) + a \cdot c$ since $a = 0$. Hence Algorithm 1 solves the problem in $\mathrm{O}(n \log n)$ time. □

In the rest of this section, we prove Theorem 2. We use dynamic programming to find the optimal value.

Proof of Theorem 2 Without loss of generality, we may assume that the indices of functions are $\delta(f_1) = \cdots = \delta(f_r) = -1$, $\delta(f_{r+1}) = \cdots = \delta(f_n) = 1$, $\gamma(f_1) \le \cdots \le \gamma(f_r)$, and $\gamma(f_{r+1}) \le \cdots \le \gamma(f_n)$. We use dynamic programming to solve the problem. Let $m(i, j, l)$ be the maximum value of $f_{\sigma(l)} \circ \cdots \circ f_{\sigma(1)}(c)$ for a permutation σ such that $i \le \sigma(1) < \sigma(2) < \cdots < \sigma(l) \le i + j - 1$ if $i + j - 1 \le n$, and $i \le \sigma(1) < \cdots < \sigma(p) \le n,\ 1 \le \sigma(p+1) < \cdots < \sigma(l) \le i + j - 1 - n$ for some p $(0 \le p \le l)$ if $i + j - 1 > n$. We claim that the optimal value for the problem is $\max_{i \in [n]} m(i, n, k)$.

Let $\sigma^* : [n] \to [n]$ be an optimal permutation for the problem. By Lemmas 23 and 24, we can assume that $i^* \le \sigma^*(1) < \cdots < \sigma^*(p) \le n,\ 1 \le \sigma^*(p+1) < \cdots < \sigma^*(k) \le i^* - 1$ for some i^* and p because $(\sigma^*(1), \dots, \sigma^*(k))$ is an optimal solution for the maximum total composition ordering problem $((f_{\sigma^*(i)})_{i=1}^k, c)$. Therefore, we have $f_{\sigma^*(k)} \circ \cdots \circ f_{\sigma^*(1)}(c) \le m(i^*, n, k) \le \max_{i \in [n]} m(i, n, k)$ and thus $\max_{i \in [n]} m(i, n, k)$ is the optimal value for the problem.

For each i, j, l, the value $m(i, j, l)$ satisfies the following relation:

$$m(i,j,l) = \begin{cases} c & (l = 0), \\ f_{\sigma(j)}(m(i, j-1, l-1)) & (l \geq 1, j = l), \\ \max\{m(i, j-1, l), f_{\sigma(j)}(m(i, j-1, l-1))\} & (l \geq 1, j > l). \end{cases}$$

To evaluate $\max_{i \in [n]} m(i, n, k)$, our algorithm calculate the values of $m(i, j, l)$ for $0 \leq i, j \leq n$ and $0 \leq l \leq k$. Therefore, we can obtain the optimal value for the problem in $\mathrm{O}(k \cdot n^2)$ time. $\square$

A formal description of the algorithm for the maximum exact k-composition problem is presented as Algorithm 2.

Algorithm 2: Maximum Exact k-Composition

$I_- \leftarrow \{i : \delta(f_i) = -1\}$, $I_+ \leftarrow \{i : \delta(f_i) = 1\}$;
sort I_- and I_+ according to the order induced by γ_i;
let $I_- = \{\sigma(1), \ldots, \sigma(k)\}$, $I_+ = \{\sigma(k+1), \ldots, \sigma(n)\}$ such that $\gamma_{\sigma(1)} \leq \cdots \leq \gamma_{\sigma(k)}$ and $\gamma_{\sigma(k+1)} \leq \cdots \leq \gamma_{\sigma(n)}$;
for $l = 0$ *to* k **do**
    for $j = l$ *to* n **do**
        for $i = 1$ *to* n **do**
            if $l = 0$ **then** $m(i, j, l) \leftarrow c$;
            else if $j = l$ **then** $m(i, j, l) \leftarrow f_{\sigma(i+j \bmod n)}(m(i, j-1, l-1))$;
            else $m(i, j, l) \leftarrow \max\{m(i, j-1, l), f_{\sigma(i+j \bmod n)}(m(i, j-1, l-1))\}$;
        end
    end
end
return $\max_{i \in [n]} m(i, n, k)$

---

5 Negative Results for the Optimal Composition Ordering Problems

In the previous sections, we showed that both the total and partial composition ordering problems can be solved efficiently if all f_i's are monotone linear. It turns out that this cannot be generalized to nonlinear functions f_i. In this section, we show the optimal composition ordering problems are in general intractable, even if all f_i's are monotone increasing, piecewise linear functions with at most two pieces. We remark that the maximum total composition ordering problem is known to be NP-hard, even if all f_i's are monotone increasing, *concave*, piecewise linear functions with at most two pieces, which can be shown by considering the time-dependent scheduling problem [4].

5.1 The Concave Part

We first consider the case in which all f_i's are monotone increasing, concave, piecewise linear functions with at most two pieces, that is, f_i is given as $f_i(x) = \min\{a_i x +$

$b_i,\ c_i x + d_i\}$ for some reals a_i, b_i, c_i, and d_i with $a_i, c_i > 0$. We also prove that the maximum partial and total composition ordering problem is NP-hard, even if f_i is of the form $f_i(x) = \min\{a_i x + b_i,\ d_i\}$ or $f_i(x) = a_i x + b_i$ for some reals a_i, b_i, and d_i with $a_i > 0$. For our reductions, we use the PARTITION problem, which is known to be NP-complete [10].

PARTITION: Given n positive integers $a_1, \ldots, a_n$ with $\sum_{i\in[n]} a_i = 2T$, ask whether exists a subset $I \subseteq [n]$ such that $\sum_{i\in I} a_i = T$.

Proof for the concave part of Theorem 3 We show that PARTITION can be reduced to the problem. Let $a_1, \ldots, a_n$ denote positive integers with $\sum_{i\in[n]} a_i = 2T$ and let p be a real such that $0 \le p < 1$. Let α (≤ 1) be a positive real number. We construct $n+2$ functions f_i $(i = 1, \ldots, n+2)$ as follows:

$$f_i(x) = \begin{cases} x + a_i & \text{if } i = 1, \ldots, n, \\ \min\{2x,\ p(x-T) + 2T\} & \text{if } i = n+1, \\ \frac{3T}{\alpha(1-p)}(x - (3T - 1 + p)) + (3T - 1 + p) & \text{if } i = n+2. \end{cases}$$

Note that $f_{n+1}(x) = \min\{2x,\ 2T\}$ when $p = 0$ and $f_{n+1}(x) = \min\left\{2x,\ \frac{1}{2}x + \frac{3}{2}T\right\}$ when $p = 1/2$. We prove that there exists no α-approximation algorithm for the maximum partial (total) composition ordering problem for $((f_i)_{i\in[n+2]}, c = 0)$, unless P = NP.

It is clear that all f_i's are monotone, concave, and piecewise linear with at most two pieces. We note that $f_{n+2}(x) \le x$ if $x \le 3T - 1 + p$. We claim that $3T$ is the optimal value for the maximum partial (total) composition ordering problem for $((f_i)_{i\in[n+1]}, c = 0)$ if there exists a partition $I \subseteq [n]$ such that $\sum_{i\in I} a_i = T$, and the optimal value is at most $3T - 1 + p$ if $\sum_{i\in I} a_i \ne T$ for all partition $I \subseteq [n]$. This implies that the optimal value for the maximum partial (total) composition ordering problem for $((f_i)_{i\in[n+2]}, c = 0)$ is at least $3T/\alpha$ if $\sum_{i\in I} a_i = T$ for some $I \subseteq [n]$, and at most $3T$ if $\sum_{i\in I} a_i \ne T$ for any partition $I \subseteq [n]$, since $f_{n+2}(3T) = 3T/\alpha + 3T - 1 + p > 3T/\alpha$ and $f_{n+2}(x) \le x < 3T$ if $x \le 3T - 1 + p$. This implies that there exists no α-approximation algorithm for the problems unless P = NP.

Let $\sigma : [n+1] \to [n+1]$ denote a permutation and let l be an index such that $\sigma(l) = n+1$. Then define $I = \{\sigma(i) : i = 1, \ldots, l-1\}$ and $q = \sum_{i\in I} a_i$. Note that $\sum_{i=l+1}^{n+1} a_{\sigma(i)} = \sum_{i\in[n]\setminus I} a_i = 2T - q$. Consider the function composition by σ:

$$\begin{aligned} f_{\sigma(n+1)} \circ \cdots \circ f_{\sigma(l+1)} \circ f_{\sigma(l)} \circ f_{\sigma(l-1)} \circ \cdots \circ f_{\sigma(1)}(0) \\ &= f_{\sigma(n)} \circ \cdots \circ f_{\sigma(l+1)} \circ f_{n+1}(q) \\ &= f_{\sigma(n)} \circ \cdots \circ f_{\sigma(l+1)} (\min\{2q,\ p(q-T) + 2T\}) \\ &= \min\{2q,\ p(q-T) + 2T\} + 2T - q \\ &= 2T + \min\{q,\ (2-p)T - (1-p)q\}. \end{aligned}$$

Note that $\min\{q,\ (2-p)T - (1-p)q\} \le T$ holds, where the equality holds only when $q = T$. In addition, $\min\{q,\ (2-p)T - (1-p)q\} \le T - 1$ if $q \le T - 1$ and $\min\{q,\ (2-p)T - (1-p)q\} \le T - (1-p)$ if $q \ge T + 1$. Thus we have

$$f_{\sigma(n+1)} \circ \cdots \circ f_{\sigma(l+1)} \circ f_{\sigma(l)} \circ f_{\sigma(l-1)} \circ \cdots \circ f_{\sigma(1)}(0) \begin{cases} = 3T & (q = T), \\ \leq 3T - (1-p) & (q \neq T). \end{cases}$$

This proves the concave part of Theorem 3. □

5.2 The Convex Part

We next consider the case in which all f_i's are monotone increasing, convex, piecewise linear functions with at most two pieces, that is, f_i is given as

$$f_i(x) = \max\{a_i x + b_i,\ c_i x + d_i\} \tag{10}$$

for some reals a_i, b_i, c_i and d_i with $a_i, c_i > 0$. Before showing the intractability of the problems, we present two basic properties for the function composition.

For an integer $i \in [n]$, let $g_i(x) = a_i(x-d) + d$. Then we have

$$g_n \circ g_{n-1} \circ \cdots \circ g_1(x) = (x-d)\prod_{i=1}^{n} a_i + d. \tag{11}$$

Thus, $\prod_{i=1}^{n} a_i > 0$ implies the following inequalities:

$$\begin{aligned} g_n \circ g_{n-1} \circ \cdots \circ g_1(x) &< d \text{ if } x < d, \\ g_n \circ g_{n-1} \circ \cdots \circ g_1(x) &= d \text{ if } x = d, \\ g_n \circ g_{n-1} \circ \cdots \circ g_1(x) &> d \text{ if } x > d. \end{aligned} \tag{12}$$

For our reductions, we use the ProductPartition problem, which is known to be NP-complete [10,18].

ProductPartition: Given n positive integers $a_1, \ldots, a_n$ (> 1) and a positive integer T such that $\prod_{i=1}^{n} a_i = T^2$, ask whether there exists a subset $I \subseteq [n]$ such that $\prod_{i \in I} a_i = T$.

We are now ready to prove the intractability.

Proof for the convex part of Theorem 3 We show that ProductPartition can be reduced to them.

Let $a_1, \ldots, a_n$ (> 1) and T denote positive integers with $\prod_{i=1}^{n} a_i = T^2$. Let α (≤ 1) be a positive real number. We construct $n+2$ functions f_i $(i = 1, \ldots, n+2)$ as follows:

$$f_i(x) = \begin{cases} \max\left\{\frac{1}{a_i}(x - T^2) + T^2,\ a_i(x - T^2) + T^2\right\} & \text{if } i = 1, \ldots, n, \\ x + 2T & \text{if } i = n+1, \\ \frac{4T^2}{\alpha}\left(x - 2T^2 + 1\right) - 2T^2 + 1 & \text{if } i = n+2. \end{cases}$$

We prove that there exists no α-approximation algorithm for the maximum partial (total) composition ordering problem for $((f_i)_{i \in [n+2]}, c = 0)$, unless P = NP.

It is clear that all f_i's are monotone, convex, and piecewise linear with at most two pieces. Moreover, we note that $f_i(x) \geq x$ holds for each function f_i ($i = 1, \ldots, n+1$), and hence all the functions $f_1, \ldots, f_{n+1}$ are used in an optimal solution even for the partial setting. We now claim that $2T^2$ is the optimal value for the maximum partial composition ordering problem for $((f_i)_{i \in [n+1]}, c = 0)$ if there exists a desired partition $I \subseteq [n]$ for ProductPartition, i.e., $\prod_{i \in I} a_i = T$, and at most $2T^2 - 1$ otherwise. This implies that the optimal value for the maximum total composition ordering problem for $((f_i)_{i \in [n+2]}, c = 0)$ is at least $2T^2/\alpha$ if $\prod_{i \in I} a_i = T$ for an $I \subseteq [n]$, and at most $2T^2$ if $\prod_{i \in I} a_i \neq T$ for all $I \subseteq [n]$, since $f_{n+2}(2T^2) = 4T^2/\alpha - 2T^2 + 1 > 2T^2/\alpha$ and $f_{n+2}(x) \leq x < 2T^2$ if $x \leq 2T^2 - 1$. Thus, there exists no α-approximation algorithm for the problems unless P = NP.

Let $\sigma : [n+1] \to [n+1]$ denote a permutation with $\sigma(l) = n+1$. Then define $I = \{\sigma(i) : i = 1, \ldots, l-1\}$ and $p = \frac{1}{\prod_{i \in I} a_i}$. Note that $\prod_{i=l+1}^{n+1} a_{\sigma(i)} = \prod_{i \in [n] \setminus I} a_i = pT^2$. Consider the function composition by σ:

$$\begin{aligned}
&f_{\sigma(n+1)} \circ \cdots \circ f_{\sigma(l+1)} \circ f_{\sigma(l)} \circ f_{\sigma(l-1)} \circ \cdots \circ f_{\sigma(1)}(0) \\
&\quad = f_{\sigma(n)} \circ \cdots \circ f_{\sigma(l+1)} \circ f_{n+1}(T^2(1-p)) \qquad (13) \\
&\quad = f_{\sigma(n)} \circ \cdots \circ f_{\sigma(l+1)}(T^2(1-p) + 2T) \\
&\quad = \begin{cases} pT^2(T^2(1-p) + 2T - T^2) + T^2 & (pT \leq 2), \\ \frac{1}{pT^2}(T^2(1-p) + 2T - T^2) + T^2 & (pT > 2) \end{cases} \qquad (14) \\
&\quad = \begin{cases} 2T^2 - (pT^2 - T)^2 & (pT \leq 2), \\ T^2 + \frac{2}{pT} - 1 & (pT > 2) \end{cases}
\end{aligned}$$

where (13) follows from (11) and (12), and (14) follows from (11) and $a_{\sigma(i)} > 1$ for all $i \geq l+1$. Thus, we have

$$f_{\sigma(n+1)} \circ \cdots \circ f_{\sigma(1)}(0) \begin{cases} = 2T^2 & (pT = 1), \\ \leq 2T^2 - 1 & (pT \neq 1) \end{cases}$$

since pT^2 is an integer, which proves the claim. □

6 Conclusions

In this paper, we have introduced optimal composition ordering problems and studied their computational complexity. We have proposed polynomial time algorithms for the maximum total, exact k, and partial composition ordering problems when the input functions are monotone linear. We have also provided a polynomial time algorithm for the maximum total composition ordering problem when the input functions are of the form $\max\{a_i x + b_i, d_i, x\}$ for some constants a_i (≥ 0), b_i and d_i. This implies that the two-valued free-order secretary problem can be solved in polynomial time. As for the negative side, we have proved that there exists no constant-factor approximation algorithm for the problems, even if input functions are monotone piecewise linear functions with at most two pieces, unless P = NP.

In closing, we propose several open problems and future directions. An open problem is to determine the time complexity of optimal composition ordering problems for linear functions without monotone nondecreasing assumption. Another one is to determine the time complexity of optimal composition ordering problems for monotone convex piecewise linear functions with at most two pieces when there is only one (or a few) function(s) which consists of two pieces. Also, the time complexity of three-valued free-order secretary problem is open. It would be interesting in future work to simplify our proofs.

Acknowledgements The first author is supported by JSPS KAKENHI Grant Numbers 26887014 and JP16K16005. The second author is supported by supported by JSPS KAKENHI Grant Numbers JP24106002, JP25280004, JP26280001, and JST CREST Grant Number JPMJCR1402, Japan.

References

1. Babaioff, M., Immorlica, N., Kempe, D., Kleinberg, R.: A knapsack secretary problem with applications. In: Proceedings of APPROX/RANDOM 2007, pp. 6–28 (2007)
2. Babaioff, M., Immorlica, N., Kleinberg, R.: Matroids, secretary problems, and online mechanisms. in: Proceedings of SODA 2007, pp. 434–443 (2007)
3. Cai, J.Y., Cai, P., Zhu, Y.: On a scheduling problem of time deteriorating jobs. J. Complex. **14**(2), 190–209 (1998)
4. Cheng, T.C.E., Ding, Q.: The complexity of scheduling starting time dependent tasks with release times. Inf. Process. Lett. **65**(2), 75–79 (1998)
5. Cheng, T.C.E., Ding, Q., Kovalyov, M.Y., Bachman, A., Janiak, A.: Scheduling jobs with piecewise linear decreasing processing times. Naval Res. Logist. **50**(6), 531–554 (2003)
6. Cheng, T.C.E., Ding, Q., Lin, B.M.T.: A concise survey of scheduling with time-dependent processing times. EJOR **152**(1), 1–13 (2004)
7. Dean, B., Goemans, M., Vondrák, J.: Adaptivity and approximation for stochastic packing problems. In: Proceedings of SODA 2005, pp. 395–404 (2005)
8. Dean, B., Goemans, M., Vondrák, J.: Approximating the stochastic knapsack problem: the benefit of adaptivity. Math. Oper. Res. **33**(4), 945–964 (2008)
9. Ferguson, T.S.: Who solved the secretary problem? Stat. Sci. **4**(3), 282–289 (1989)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York (1979)
11. Gawiejnowicz, S.: Scheduling deteriorating jobs subject to job or machine availability constraints. EJOR **180**(1), 472–478 (2007)
12. Gawiejnowicz, S.: Time-Dependent Scheduling. Springer, Berlin (2008)
13. Gawiejnowicz, S., Pankowska, L.: Scheduling jobs with varying processing times. Inf. Process. Lett. **54**(3), 175–178 (1995)
14. Gupta, J.N., Gupta, S.K.: Single facility scheduling with nonlinear processing times. Comput. Ind. Eng. **14**(4), 387–393 (1988)
15. Ho, K.I.J., Leung, J.Y.T., Wei, W.D.: Complexity of scheduling tasks with time-dependent execution times. Inf. Process. Lett. **48**(6), 315–320 (1993)
16. Kawase, Y., Makino, K., Seimi, K.: Optimal composition ordering problems for piecewise linear functions. In: Proceedings of ISAAC 2016, pp. 42:1–42:13 (2016)
17. Melnikov, O.I., Shafransky, Y.M.: Parametric problem of scheduling theory. Cybernetics **15**, 352–357 (1980)
18. Ng, C.T., Barketau, M., Cheng, T.C.E., Kovalyov, M.Y.: “Product partition” and related problems of scheduling and systems reliability: computational complexity and approximation. EJOR **207**, 601–604 (2010)

19. Oveis Gharan, S., Vondrák, J.: On variants of the matroid secretary problem. In: Proceedings of ESA 2011, pp. 335–346 (2011)
20. Tanaev, V.S., Gordon, V.S., Shafransky, Y.M.: Scheduling Theory: Single-Stage Systems. Kluwer Academic Publishers, Dordrecht (1994)
21. Wajs, W.: Polynomial algorithm for dynamic sequencing problem. Archiwum Automatyki i Telemechaniki **31**(3), 209–213 (1986)