Optimal Composition Ordering Problems for Piecewise Linear Functions

Yasushi Kawase* Kazuhisa Makino† Kento Seimi‡

September 12, 2018

Abstract

In this paper, we introduce maximum composition ordering problems. The input is n real functions $f_1, \ldots, f_n : \mathbb{R} \to \mathbb{R}$ 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 $[n] = \{1, \ldots, n\}$. The maximum partial composition ordering problem is to compute a permutation $\sigma : [n] \to [n]$ and a nonnegative integer k $(0 \le k \le n)$ 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 partial composition ordering problem can be solved in polynomial time if f_i is of form $\max\{a_i x + b_i, c_i\}$ for some constants $a_i\,(\ge 0)$, b_i and c_i. We finally prove that there exists no constant-factor approximation algorithm for the problems, even if f_i's are monotone, piecewise linear functions with at most two pieces, unless P=NP.

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 $(0 \le k \le n)$ 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 remark that the minimization versions are equivalent to the maximization ones.

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 $(0 \le k \le n)$.

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 the single machine time-dependent scheduling problems and the free-order secretary problem, which can be formulated as the optimal composition ordering problems.

*Tokyo Institute of Technology. E-mail: kawase.y.ab@m.titech.ac.jp
†Kyoto University. E-mail: makino@kurims.kyoto-u.ac.jp
‡The Toa Reinsurance Company, Limited. E-mail: kento.seimi@gmail.com

Time-dependent scheduling

Consider the machine scheduling problems with time-dependent processing times, called *time-dependent scheduling problems* [5,12].

Let J_i $(i = 1, \dots, n)$ 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* 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 any 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 $(i = 1, \dots, n)$ 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 seen 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 the 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 $((f_i)_{i\in[n]}, c = t_0)$. A number of restrictions on the processing time $p_i(t)$ has been studied in this literature (e.g., [3,6,16]).

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 $O(n \log n)$ time by scheduling jobs in the nonincreasing order of 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 a deadline are all NP-hard [4,11].

Another important 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 with $1 > a_i > 0$, $b_i > 0$. They showed that the time-dependent scheduling problem of this model can be solved in $O(n \log n)$ time by again scheduling jobs in the nonincreasing order of the ratios b_i/a_i.

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 $X_1, \dots, X_n$ 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 $f_i(x) = \mathbf{E}[\max\{X_i, x\}]$. For example, 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

$$f_i(x) = \sum_{j=1}^{m} p_i^j \max\{a_i^j, x\} = \max_{l=0,\ldots,m} \left\{ \sum_{j=1}^{l} p_i^j a_i^j + \sum_{j=l+1}^{m} p_i^j x \right\}.$$

Note that this f_i is a monotone piecewise linear function with at most $(m+1)$ pieces. 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 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. We show 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 $((f_i)_{i\in[n]}, c = 0)$.

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 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 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., $f_i(x) = \min\{a_i^1 x + b_i^1,\, a_i^2 x + b_i^2\}$ for some constants a_i^1, a_i^2, b_i^1, and b_i^2 with $a_i^1, a_i^2 > 0$. It turns out that all the other cases become intractable, even if all f_i's are monotone and consist of at most two pieces. Furthermore, the problems are inapproximable.

Theorem 3. (i) *For any 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, concave, and piecewise linear functions with at most two pieces, unless P=NP.*
(ii) *For any 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, convex, and piecewise linear functions with at most two pieces, unless P=NP.*

Note that f_i can be represented by $f_i(x) = \max\{a_i^1 x + b_i^1,\ a_i^2 x + b_i^2\}$ for some constants a_i^1, a_i^2, b_i^1, and b_i^2 with $a_i^1, a_i^2 > 0$ 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 monotone, convex, and piecewise linear function with at most two pieces such that one of the pieces is constant, 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. *Let $f_i(x) = \max\{a_i x + b_i, c_i\}$ for some constants a_i (≥ 0), b_i and c_i. Then the maximum partial composition ordering problem is 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. 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 can be transformed into the maximum total one as shown in Section 3.

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

Functions	Complexity	References
$f_i(x) = a_i x \ (a_i > 1)$	$\mathrm{O}(n)$	[17]
$f_i(x) = a_i x + b_i \quad (a_i > 1,\ b_i < 0)$	$\mathrm{O}(n \log n)$	[13, 14, 20, 21]
$f_i(x) = \min\{ax + b_i, r_i\} \quad (a > 1,\ b_i < 0)$	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]
$f_i(x) = \min\{a_i x + b_i, c_i\} \quad (a_i > 1)$	NP-hard	[4]
$f_i(x) = a_i x + b_i \quad (1 > a_i \geq 0,\ b_i < 0)$	$\mathrm{O}(n \log n)$	[15]
$f_i(x) = \begin{cases} a_i x + b_i & (x \geq d_i) \\ -\infty & (x < d_i) \end{cases} \quad (1 > a_i > 0)$	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]**
$f_i(x) = \max\{x, a_i x + b_i, c_i\} \quad (a_i \geq 0)$	$\mathbf{O(n^2)}$	**[Theorem 4]**
$f_i(x) = \max\{x, \min\{a_i^1 x + b_i^1, a_i^2 x + b_i^2\}\} \quad (a_i^1, a_i^2 > 0)$	**NP-hard**	**[Theorem 3]**
$f_i(x) = \max\{a_i^1 x + b_i^1, a_i^2 x + b_i^2\} \quad (a_i^1, a_i^2 > 0)$	**NP-hard**	**[Theorem 3]**

The organization of the paper

The rest of the paper is organized as follows. In Section 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 Section 3, we prove the partial composition part of Theorem 1 and Theorem 4, and in Section 4, we prove the total composition part of Theorem 1 and Theorem2. Finally, Section 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 namely, 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

$$\tilde{f}(x) := -f(-x). \tag{1}$$

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. For any permutation $\sigma : [n] \to [n]$ and an integer k with $0 \leq k \leq n$, we have

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

which proves the lemma. □

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

$$\overline{f}(x) := \max\{f_i(x), x\}. \tag{2}$$

Lemma 6. *Let c be a real, and for $i = 1, \ldots, n$, let $f_i : \mathbb{R} \to \mathbb{R}$ be 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 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 $((\overline{f}_i)_{i \in [n]}, c)$. Then let k denote the number of i's such that*

$$\overline{f}_{\sigma(i)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) > \overline{f}_{\sigma(i-1)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c), \tag{3}$$

and $\tau : [n] \to [n]$ denote a permutation such that $\tau(j)$ $(j \leq k)$ is equal to the jth $\sigma(i)$ that satisfies (3). 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

$$f_{\sigma(k)} \circ \cdots \circ f_{\sigma(1)}(c) \leq \overline{f}_{\sigma(k)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) \leq \overline{f}_{\sigma(n)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) \tag{4}$$

by $\overline{f}(x) \geq f(x)$ and $\overline{f}(x) \geq x$. 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

$$f_{\tau(k)} \circ \cdots \circ f_{\tau(1)}(c) = \overline{f}_{\tau(k)} \circ \cdots \circ \overline{f}_{\tau(1)}(c) = \overline{f}_{\sigma(n)} \circ \cdots \circ \overline{f}_{\sigma(1)}(c) \tag{5}$$

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 (4) and (5) 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 the 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 $\overline{f}_i(x) = \max\{f_i(x), x\}$. Recall that 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 $((\overline{f}_i)_{i\in[n]}, c)$. Let us start with the maximum partial composition ordering problem for monotone linear functions $f_i(x) = a_i x + b_i$ ($a_i \geq 0$), i.e., the total composition ordering problem for $\overline{f}_i(x) = \max\{a_i x + b_i, x\}$ ($a_i \geq 0$).

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

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

Note that the relation $\preceq$ is not total relation in general, here a relation $\preceq$ is called total if $f \preceq g$ or $g \preceq f$ for any f, g. 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 less than $f_2 \circ f_1(-2)\,(= -9)$.

However, if two consecutive functions are total, 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 any $x \in \mathbb{R}$.*

It follows from the lemma that, for monotone functions f_i, there exists a maximum total composition $f_n \circ f_{n-1} \circ \cdots \circ f_1$ 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 as the more general form in Lemma 10.

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

$$\begin{aligned} f_i \preceq f_j &\iff f_i \circ f_j(x) \leq f_j \circ f_i(x) \text{ for any } x \in \mathbb{R} \\ &\iff a_i(a_j x + b_j) + b_i \leq a_j(a_i x + b_i) + b_j \text{ for any } x \in \mathbb{R} \\ &\iff b_i(1 - a_j) \leq b_j(1 - a_i). \end{aligned} \tag{6}$$

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

The totality of the relation is proven in Lemma 15, when all functions are of the form $\max\{ax + b, x\}$ with $a \geq 0$.

We further note that the relation $\preceq$ is transitive for linear functions $f(x) = ax + b$ with $a > 1$, since (6) 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 $\overline{f}_1 \prec \overline{f}_2$, $\overline{f}_2 \prec \overline{f}_3$, 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 $a \geq 0$. These show 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 which can be used even for *non-transitive* relations.

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 $i \leq j$ implies $f_{\sigma(i)} \preceq f_{\sigma(j)}$ for any $i, j \in [n]$, then σ is an optimal solution for the maximum total composition ordering problem for $((f_i)_{i\in[n]}, c)$.*

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. Here, the inversion number denotes the number of pairs (i, j) with $i < j$ and $\sigma'(i) > \sigma'(j)$. Then we show that σ' is the identity permutation by contradiction. 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. $\square$

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. We analyze the relation $\preceq$ in terms of the following γ and δ, and provide an efficient algorithm.

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

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

Note that $\gamma(f)$ is the solution of the equation $f(x) = x$ if $\gamma(f) \neq -\infty, +\infty$. In the rest of the paper, we assume without loss of generality that no f_i is identity (i.e., $f_i(x) = x$), since we can ignore identity function for both the total and partial composition problems.

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, there exists an integer k such that $0 \leq k \leq n$, $\delta(f_{\sigma(1)}) = \cdots = \delta(f_{\sigma(k)}) = -1$, $\delta(f_{\sigma(k+1)}) = \cdots = \delta(f_{\sigma(n)}) = +1$, $\gamma(f_{\sigma(1)}) \leq \cdots \leq \gamma(f_{\sigma(k)})$, and $\gamma(f_{\sigma(k+1)}) \leq \cdots \leq \gamma(f_{\sigma(n)})$.

We prove that the lexicographic order satisfies the condition in Lemma 10 and thus, the order is the optimal solution.

Lemma 12. *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 $i \leq j$ implies $\overline{f}_{\sigma(i)} \preceq \overline{f}_{\sigma(j)}$ for any $i, j \in [n]$.*

Before proving Lemma 12, we discuss algorithms for the maximum partial composition ordering problem.

3.1 Algorithms

By Lemma 12, the maximum total composition ordering problem $((\overline{f}_i)_{i \in [n], c})$ such that f_i's are monotone nondecreasing linear functions can be solved by computing the lexicographic order with respect to $(\delta(f_i), \gamma(f_i))$. Therefore, it can be solved in $\mathrm{O}(n \log n)$ time. This is our algorithm for 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.

Next, for $i \in [n]$, let $f_i(x) = a_i x + b_i$ be a monotone nondecreasing linear function and let $h_i(x) = \max\{f_i(x), c_i\}$ for some constant c_i. We give an efficient algorithm for the maximum partial composition ordering problem $((h_i)_{i \in [n]}, c)$, which is the tractability result of Theorem 4. As mentioned in the introduction, the problem includes the two-valued free-order secretary problem, and it is a generalization of the maximum partial composition ordering problem for monotone linear functions.

By Lemma 6, we instead consider the maximum total composition ordering problem for the functions

$$\overline{h}_i(x) = \max\{a_i x + b_i, c_i, x\} \text{ for } a_i \in \mathbb{R}_+,\ b_i, c_i \in \mathbb{R}, \tag{7}$$

where $\mathbb{R}_+$ is the set of nonnegative real numbers.

Lemma 13. *Let $c \in \mathbb{R}$, and let $\overline{h}_i$ $(i \in [n])$ be a function defined as (7). Then there exists an optimal solution σ for the maximum total composition ordering problem $((\overline{h}_i)_{i\in[n]}, c)$ such that no $i\ (> 1)$ satisfies $\overline{f}_{\sigma(i)} \circ \overline{h}_{\sigma(i-1)} \circ \cdots \circ \overline{h}_{\sigma(1)}(c) < c_{\sigma(i)}$, where $\overline{f}_i(x) = \max\{a_i x + b_i, x\}$.*

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 $\overline{h}_{\sigma(i^*)} \circ \cdots \circ \overline{h}_{\sigma(1)}(c) = \overline{h}_{\sigma(i^*)}(c) = c_{\sigma(i^*)}$. It holds that $c_{\sigma(i)} < c_{\sigma(i^*)}$ for any i with $0 \leq i < i^*$, since $c_{\sigma(i)} \leq \overline{h}_{\sigma(i)} \circ \cdots \circ \overline{h}_{\sigma(1)}(c) \leq \overline{h}_{\sigma(i^*-1)} \circ \cdots \circ \overline{h}_{\sigma(1)}(c) < c_{\sigma(i^*)}$, where $c_{\sigma(0)} = c$ is assumed. Thus, we have $\overline{h}_{\sigma(n)} \circ \cdots \circ \overline{h}_{\sigma(1)}(c) = \overline{h}_{\sigma(n)} \circ \cdots \circ \overline{h}_{\sigma(i^*)}(c) \leq \overline{h}_{\sigma(i^*-1)} \circ \cdots \circ \overline{h}_{\sigma(1)} \circ \overline{h}_{\sigma(n)} \circ \cdots \circ \overline{h}_{\sigma(i^*)}(c)$. 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 $\bar{h}_i$ $(i \neq i^*)$ is not explicitly used by the definition of i^* and $c_{\sigma(i)} < c_{\sigma(i^*)}$ for any $i\ (< i^*)$, which completes the proof. □

It follows from Lemma 13 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 $((f_i)_{i\in[n]}, c)$ and $((f_i)_{i\in[n]\setminus\{k\}}, c_k)$ for all $k \in [n]$.

Therefore, we have an $\mathrm{O}(n^2 \log n)$-time algorithm by directly applying Theorem 1 to the problems. 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 order. This implies that the problem can be solved in $\mathrm{O}(n^2)$ time by first computing the lexicographic order with respect to $(\delta(f_i), \gamma(f_i))$. This is our algorithm for Theorem 4.

3.2 The proof of Lemma 12

In this subsection, we prove Lemma 12. We first consider the relationship between two linear functions. The proof can be found in Appendix.

Lemma 14. *Let $f_i(x) = a_i x + b_i$ and $f_j(x) = a_j x + b_j$ be (non-identity) 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 $f_i \succeq f_j \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$.*

By this lemma, the relation $\preceq$ is total preorder for the both cases $a_1, a_2, \ldots, a_n \geq 1$ and $a_1, a_2, \ldots, a_n < 1$. Moreover, the permutation $\sigma : [n] \to [n]$ such that $\gamma(f_{\sigma(1)}) \leq \cdots \leq \gamma(f_{\sigma(n)})$ is optimal for the cases. This result matches the results in the time-dependent scheduling problem of the linear deterioration model (when $a_1, a_2, \ldots, a_n \geq 1$) and the linear shortening model (when $a_1, a_2, \ldots, a_n < 1$).

Next we characterize the relationship between two functions of the form $\max\{a_i x + b_i, x\}$, the proof can be found in Appendix.

Lemma 15. *For (non-identity) 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 $\overline{f}_i \succeq \overline{f}_j$.*

Note that Lemma 15 implies that the relation $\preceq$ is total for the functions of the form $\max\{ax + b, x\}$ with $a \geq 0$. Moreover, it is easy to check that the lexicographic order with respect to $(\delta(f_i), \gamma(f_i))$ satisfies the condition in Lemma 10, i.e., $i < j$ implies $\overline{f}_{\sigma(i)} \preceq \overline{f}_{\sigma(j)}$ for the permutation σ that is compatible with the ordering. Therefore, we have Lemma 12.

4 Maximum Total Composition and Exact k-composition Ordering Problems

In this section we prove the total composition part of Theorem 1 and Theorem 2. Different from the case when each function is of form $\max\{a_i x + b_i, x\}$, the binary relation $\preceq$ for linear functions does not satisfy the condition in Lemma 10. 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. Our main result is the following lemma.

Lemma 16. *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 an optimal solution for the maximum total composition ordering problem $((f_i)_{i\in[n]}, c)$ is*

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

for some t.

Before proving Lemma 16, we discuss algorithms for the maximum total composition and the exact k-composition ordering problems.

4.1 Algorithm for Total Composition

In this subsection, we prove the total composition part of Theorem 1, i.e., we provide an efficient algorithm for the maximum total composition ordering problem $((f_i(x) = a_i x + b_i)_{i\in[n]}, c)$, where $a_i \geq 0$. Let $\sigma : [n] \to [n]$ be a permutation compatible with the lexicographic order with respect to $(\delta(f_i), \gamma(f_i))$. Then 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 by Lemma 16. 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 $k = 1, \ldots, n$ and let $a = \prod_{i=1}^{n} a_i$. Then it is not difficult to see that $d_{k+1} = a_{\sigma(k)} \cdot (d_k - a \cdot c) - b_{\sigma(k)} \cdot (a - 1) + a \cdot c$, and hence the problem is solvable in $\mathrm{O}(n \log n)$ time.

4.2 exact k-composition

In this subsection, we prove Theorem 2, i.e., we provide an efficient algorithm for the maximum exact k-composition ordering problem $((f_i(x) = a_i x + b_i)_{i\in[n]}, c)$, where $a_i \geq 0$. We use a dynamic programming to find the optimal value.

For simplicity, we relabel the indices of functions so that the lexicographic order of $(\delta(f_i), \gamma(f_i))$ is monotone increasing. 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 $i \leq \sigma(1) < \sigma(2) < \cdots < \sigma(l) \leq i + j - 1$ if $i + j - 1 \leq n$, and $i \leq \sigma(1) < \cdots < \sigma(p) \leq n,\ 1 \leq \sigma(p+1) < \cdots < \sigma(l) \leq i + j - 1 - n$ for some p ($0 \leq p \leq l$) if $i + j - 1 > n$. We claim that the optimal value for the problem is $\max_{i=1}^{n} m(i, n, k)$.

Let $\sigma^* : [n] \to [n]$ be an optimal permutation for the problem. By Lemma 16, we can assume that $i^* \leq \sigma^*(1) < \cdots < \sigma^*(p) \leq n,\ 1 \leq \sigma^*(p+1) < \cdots < \sigma^*(k) \leq i^* - 1$ for some i^* and p. Therefore, we have $f_{\sigma^*(k)} \circ \cdots \circ f_{\sigma^*(1)}(c) \leq m(i^*, n, k) \leq \max_{i=1}^{n} m(i, n, k)$ and, thus, $\max_{i=1}^{n} m(i, n, k)$ is the optimal value for the problem.

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

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

To evaluate $\max_{i=1}^{n} m(i, n, k)$, 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. The detailed algorithm for the maximum exact k-composition problem is shown in Algorithm 1.

4.3 The proof of Lemma 16

In this subsection, we prove Lemma 16. To overcome the difficulty that the binary relation $\preceq$ for linear functions does not satisfy the condition in Lemma 10, we discuss relationships among three or four functions.

Algorithm 1 Maximum Exact k-Composition

sort the input functions according to the lexicographic order of $(\delta(f_i), \gamma(f_i))$
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** $m(i,j,l) \leftarrow f_{(i+j \mod n)}(m(i,j-1,l-1))$
      else $m(i,j,l) \leftarrow \max\{m(i,j-1,l), f_{(i+j \mod n)}(m(i,j-1,l-1))\}$
    end for
  end for
end for
return $\max_{i=1}^{n} m(i,n,k)$

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. The proof can be found in Appendix.

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

(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 $\gamma(f_j \circ f_i) \geq \gamma(f_j)$ $(> \gamma(f_i))$,

(e) If $\gamma(f_i) < \gamma(f_j)$, $a_i < 1$, $a_j \geq 1$, and $a_i \cdot a_j < 1$, then $\gamma(f_j \circ f_i) \leq \gamma(f_i)$ $(< \gamma(f_j))$,

(f) If $\gamma(f_i) < \gamma(f_j)$, $a_i \geq 1$, $a_j < 1$, and $a_i \cdot a_j \geq 1$, then $\gamma(f_j \circ f_i) \leq \gamma(f_i)$ $(< \gamma(f_j))$,

(g) If $\gamma(f_i) < \gamma(f_j)$, $a_i \geq 1$, $a_j < 1$, and $a_i \cdot a_j < 1$, then $\gamma(f_j \circ f_i) \geq \gamma(f_j)$ $(> \gamma(f_i))$.

By Lemmas 14 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*

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

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*

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

Proof. We only prove Lemma 18 since Lemma 19 can be proved in a similar way. Let $g(x) = f_3 \circ f_2(x)$. If $a_2 \cdot a_3 \geq 1$, then $\gamma(g) \leq \gamma(f_3) \leq \gamma(f_1)$ holds by (a) and (f) in Lemma 17, and $g \circ f_1(x) \leq f_1 \circ g(x)$ holds by (a) and (b) in Lemma 14. 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 by (a) and (g) in Lemma 17, and $f_4 \circ g(x) \leq g \circ f_4(x)$ holds by (c) in Lemma 14. 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 we have $\delta(g_1) = -\delta(g_2) = \delta(g_3) = -\delta(g_4)$. 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 (d) in Lemma 14 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

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

by Lemmas 18 and 19. This again contradicts the assumption on σ. □

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*

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

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*

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

Proof. We only prove Lemma 21 since Lemma 22 can be prove in a similar way. If $a_2 \cdot a_3 \geq 1$, then $\gamma(f_3 \circ f_2) \leq \gamma(f_3) \leq \gamma(f_1)$ by (a) and (f) in Lemma 17, and it implies $f_3 \circ f_2 \circ f_1(x) \leq f_1 \circ f_3 \circ f_2(x)$ by (a) and (b) in Lemma 14. 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 (d) in Lemma 14.

If $a_1 \cdot a_2 \geq 1$, then $\gamma(f_2 \circ f_1) \geq \gamma(f_1) \geq \gamma(f_3)$ by (a) and (d) in Lemma 17, and it implies $f_3 \circ f_2 \circ f_1(x) \leq f_2 \circ f_1 \circ f_3(x)$ by (a) and (b) in Lemma 14. 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 (d) in Lemma 14.

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 (d) in Lemma 17, and $\gamma(f_3 \circ (f_2 \circ f_1)) \leq \gamma(f_3)$ by (f) in Lemma 17 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 (a) in Lemma 17. □

By Lemmas 14, 17, 20, 21, and 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 Lemma 24 can be proved in a similar way. 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 14, 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 14 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 (d) in Lemma 14. 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 (d) in Lemma 14. 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 Lemma 16.

5 Negative Results

In the previous sections, we show 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 they 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 [4], which can be shown by considering the time-dependent scheduling problem.

For our reductions, we use the following NP-complete problems (see [10, 18]).

Partition: Given n positive integers $a_1, \dots, a_n$ with $\sum_{i=1}^{n} a_i = 2T$, ask whether exists a subset $I \subseteq [n]$ such that $\sum_{i \in I} a_i = T$.

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

We use Partition problem for concave case and ProductPartition for convex case.

5.1 Monotone increasing, concave, piecewise linear functions with at most two pieces

In this section, we 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

$$f_i(x) = \min\{a_i^1 x + b_i^1,\ a_i^2 x + b_i^2\} \tag{8}$$

for some reals a_i^1, a_i^2, b_i^1 and b_i^2 with $a_i^1, a_i^2 > 0$.

Proof for Theorem 3 (i). We show that Partition can be reduced to the problem. Let $a_1, \dots, a_n$ denote positive integers with $\sum_{i=1}^{n} a_i = 2T$. We construct $n+2$ functions f_i ($i = 1, \dots, n+2$) as follows:

$$f_i(x) = \begin{cases} x + a_i & \text{if } i = 1, \dots, n, \\ \min\left\{2x,\ \frac{1}{2}x + \frac{3}{2}T\right\} & \text{if } i = n+1, \\ 6\alpha T(x - (3T - \frac{1}{2})) + (3T - \frac{1}{2}) & \text{if } i = n+2. \end{cases}$$

It is clear that all f_i's are monotone, concave, and piecewise linear with at most two pieces. Moreover, we note that all f_i's ($i = 1, \dots, n+1$) satisfy $f_i(x) \geq x$ if $0 \leq x \leq 3T$, and $f_{n+2}(x) \leq x$ if $x \leq 3T - 1/2$. We claim that $3T$ is the optimal value for the maximum partial (total) composition ordering problem $((f_i)_{i \in [n+1]}, c = 0)$ if there exists a partition $I \subseteq [n]$ such that $\sum_{i \in I} a_i = T$, and the optimal value is at most $3T - 1/2$ if $\sum_{i \in I} a_i \neq T$ for any partition $I \subseteq [n]$. This implies that the optimal value for the maximum partial (total) composition ordering problem $((f_i)_{i \in [n+2]}, c = 0)$ is at least $3\alpha T$ if $\sum_{i \in I} a_i = T$ for some $I \subseteq [n]$, and at most $3T$ if $\sum_{i \in I} a_i \neq T$ for any partition $I \subseteq [n]$, since $f_{n+2}(3T) = 3\alpha T + 3T - 1/2 > 3\alpha T$ and $f_{n+2}(x) \leq x$ if $x \leq 3T - 1/2$. 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 $I = \{\sigma(i) : i = 1, \dots, l-1\}$ and $q = \sum_{i \in I} a_i$. Note that $\sum_{i=l+1}^{n+1} a_{\sigma(i)} = \sum_{i \notin I} a_i = 2T - q$. Consider the function composition by σ:

$$\begin{aligned} &f_{\sigma(n+1)} \circ \cdots \circ f_{\sigma(l+1)} \circ f_{\sigma(l)} \circ f_{\sigma(l-1)} \circ \cdots \circ f_{\sigma(1)}(0) \\ &\quad = f_{\sigma(n)} \circ \cdots \circ f_{\sigma(l+1)} \circ f_{n+1}(q) \\ &\quad = f_{\sigma(n)} \circ \cdots \circ f_{\sigma(l+1)} \left(\min\left\{2q,\ \frac{1}{2}q + \frac{3}{2}T\right\}\right) \\ &\quad = \min\left\{2q,\ \frac{1}{2}q + \frac{3}{2}T\right\} + 2T - q = \min\left\{q,\ -\frac{1}{2}q + \frac{3}{2}T\right\} + 2T. \end{aligned}$$

Note that $\min\left\{q,\ -\frac{1}{2}q + \frac{3}{2}T\right\} \leq T$ holds, where the equality holds only when $q = T$. This implies that

$$f_{\sigma(n+1)} \circ \cdots \circ f_{\sigma(l+1)} \circ f_{\sigma(l)} \circ f_{\sigma(l-1)} \circ \cdots \circ f_{\sigma(1)}(0) \begin{cases} = 3T & (q = T), \\ \leq 3T - 1/2 & (q \neq T) \end{cases} \tag{9}$$

since q is an integer, which proves the claim. $\square$

By Lemma 6, we have the following corollary. We also have the following corollary.

Corollary 25. *The maximum total composition ordering problem is NP-hard, even if all f_i's are represented by $f_i(x) = \max\{x, \min\{a_i^1 x + b_i^1, a_i^2 x + b_i^2\}\}$ for some reals a_i^1, a_i^2, b_i^1 and b_i^2 with $a_i^1, a_i^2 > 0$.*

5.2 Monotone increasing, convex, piecewise linear functions with at most two pieces

In this section, we 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

$$f_i(x) = \max\{a_i^1 x + b_i^1,\ a_i^2 x + b_i^2\} \tag{10}$$

for some reals a_i^1, a_i^2, b_i^1 and b_i^2 with $a_i^1, a_i^2 > 0$. Before showing the intractability of the problems, we present two basic properties for the function composition.

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

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

We are now ready to prove the intractability.

Proof for Theorem 3 (ii). We show that ProductPartition can be reduced to them.

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

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

It is clear that all f_i's are monotone, convex, and piecewise linear with at most two pieces. Moreover, we note that $f_i \geq x$ holds for all functions f_i, which together with Lemma 6 implies that the maximum partial and total composition ordering problems are equivalent for the functions f_i. Therefore, we deal with the total setting only. We now claim that $2T^2$ is the optimal value for the maximum partial composition ordering problem $((f_i)_{i\in[n+1}, c=0)$ if there exists a desired partition $I \subseteq [n]$ for ProductPartition, i.e., $\prod_{i\in I} a_i = T$, and at most $2T^2 - (T/(T+1))^2$ otherwise. This implies that the optimal value for the maximum total composition ordering problem $((f_i)_{i\in[n+2]}, c=0)$ is at least $2\alpha T^2$ 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 any $I \subseteq [n]$, since $f_{n+2}(2T^2) > 2\alpha T^2$ and $f_{n+2}(x) \leq x$ if $x \leq 2T^2 - (T/(T+1))^2$. 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 $\prod_{i=l+1}^{n+1} a_{\sigma(i)} = \prod_{i\notin I} a_i = pT^2$. Consider the function composition by σ:

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

where (13) follows from (11) and (12), and (14) follows from (11) and $a_{\sigma(i)} > 1$ for all $i \geq l+1$. We also note that (14) is satisfied by equality if and only if $T^2(1-p) + 2T \geq T^2$, i.e., $p \leq 2/T$. Thus, we have

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

since $1/p$ is an integer, which proves the claim. □

By Lemma 6, we also have the following result.

Corollary 26. *The maximum total composition ordering problem is NP-hard, even if all f_i's are represented by $f_i(x) = \max\{x, a_i^1 x + b_i^1, a_i^2 x + b_i^2\}$ for some reals a_i^1, a_i^2, b_i^1 and b_i^2 with $a_i^1, a_i^2 > 0$.*

References

[1] Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. A knapsack secretary problem with applications. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 16–28, 2007.

[2] Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 434–443, 2007.

[3] Jin-Yi Cai, Pu Cai, and Yixin Zhu. On a scheduling problem of time deteriorating jobs. *Journal of Complexity*, 14(2):190–209, 1998.

[4] T. C. E. Cheng and Q. Ding. The complexity of scheduling starting time dependent tasks with release times. *Information Processing Letters*, 65(2):75–79, 1998.

[5] T. C. E. Cheng, Q Ding, and B.M.T Lin. A concise survey of scheduling with time-dependent processing times. *European Journal of Operational Research*, 152(1):1–13, 2004.

[6] T. C. E. Cheng, Qing Ding, Mikhail Y. Kovalyov, Aleksander Bachman, and Adam Janiak. Scheduling jobs with piecewise linear decreasing processing times. *Naval Research Logistics*, 50(6):531–554, 2003.

[7] B.C. Dean, M.X. Goemans, and J. Vondrák. Adaptivity and approximation for stochastic packing problems. In *Proceedings of the sixteenth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 395–404. Society for Industrial and Applied Mathematics, 2005.

[8] B.C. Dean, M.X. Goemans, and J. Vondrák. Approximating the stochastic knapsack problem: the benefit of adaptivity. *Mathematics of Operations Research*, 33(4):945–964, 2008.

[9] Thomas S. Ferguson. Who solved the secretary problem? *Statical Science*, 4(3):282–289, 1989.

[10] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman New York, 1979.

[11] S. Gawiejnowicz. Scheduling deteriorating jobs subject to job or machine availability constraints. *European Journal of Operational Research*, 180(1):472–478, 2007.

[12] Stanisław Gawiejnowicz. *Time-Dependent Scheduling.* Springer, 2008.

[13] Stanisław Gawiejnowicz and Lidia Pankowska. Scheduling jobs with varying processing times. *Information Processing Letters*, 54(3):175–178, 1995.

[14] Jatinder N.D. Gupta and Sushil K. Gupta. Single facility scheduling with nonlinear processing times. *Computers & Industrial Engineering*, 14(4):387–393, 1988.

[15] Kevin I-J. Ho, Joseph Y-T. Leung, and W-D. Wei. Complexity of scheduling tasks with time-dependent execution times. *Information Processing Letters*, 48(6):315–320, 1993.

[16] O. I. Melnikov and Y. M. Shafransky. Parametric problem of scheduling theory. *Cybernetics*, 15:352–357, 1980.

[17] Gur Mosheiov. Scheduling jobs under simple linear deterioration. *Computers & Operations Research*, 21(6):653–659, 1994.

[18] C. T. Ng, M.S. Barketau, T. C. E. Cheng, and Mikhail Y. Kovalyov. "Product partition" and related problems of scheduling and systems reliability: Computational complexity and approximation. *European Journal of Operational Research*, 207:601–604, 2010.

[19] Shayan Oveis Gharan and Jan Vondrák. On variants of the matroid secretary problem. In *Proceedings of the 19th Annual European Symposium on Algorithms*, pages 335–346, 2011.

[20] V. S. Tanaev, V. S. Gordon, and Y. M. Shafransky. *Scheduling Theory: Single-Stage Systems.* Kluwer Academic Publishers, 1994.

[21] W. Wajs. Polynomial algorithm for dynamic sequencing problem. *Archiwum Automatyki i Telemechaniki*, 31(3):209–213, 1986.

Appendix: Omitted Proofs

Proof of Lemma 14

Lemma 14. *Let $f_i(x) = a_i x + b_i$ and $f_j(x) = a_j x + b_j$ be (non-identity) 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 (6) $\Leftrightarrow \frac{b_i}{1-a_i} \leq \frac{b_j}{1-a_j} \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$. If $a_i > 1$ and $a_j = 1$, then the lemma holds, since we have the following equivalences (6) $\Leftrightarrow 0 \leq b_j(1-a_i) \Leftrightarrow b_j < 0 \Leftrightarrow \gamma(f_j) = +\infty \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$. Otherwise (i.e., $a_i = 1$ and $a_j > 1$), we have (6) $\Leftrightarrow b_i(1-a_j) \leq 0 \Leftrightarrow b_i > 0 \Leftrightarrow \gamma(f_i) = -\infty \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$, which prove the lemma.

(*c*): The lemma holds, since we have the following equivalences (6) $\Leftrightarrow \frac{b_i}{1-a_i} \leq \frac{b_j}{1-a_j} \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$.

(*d*): If $a_i > 1$, the lemma holds since we have the following equivalences (6) $\Leftrightarrow \frac{b_i}{1-a_i} \geq \frac{b_j}{1-a_j} \Leftrightarrow \gamma(f_i) \geq \gamma(f_j)$. 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 $f_i \succeq f_j \Leftrightarrow b_i(1-a_j) \geq 0 \Leftrightarrow b_i > 0 \Leftrightarrow \gamma(f_i) = -\infty \Leftrightarrow \gamma(f_i) \leq \gamma(f_j)$. □

Proof of Lemma 15

Lemma 15. *For (non-identity) 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 any 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 Figure 1 (a)).

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 any 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 (a) and (b) in Lemma 14.

(*b*): We prove that $\overline{f}_j \circ \overline{f}_i(x) \geq \overline{f}_i \circ \overline{f}_j(x)$ holds for any 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 Figure 1 (b)).

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 (c) in Lemma 14.

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 $\gamma(f_i) \le x < \gamma(f_j)$, 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 $\gamma(f_i) \le x \le f_j(x) < \gamma(f_j)$. Thus, we obtain $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_i \circ \overline{f}_j(x)$.

Case *b*-4: If $\gamma(f_j) \le x$, 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 $\gamma(f_i) \le \gamma(f_j) \le x$. 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 any x. We separately consider three cases $x < \gamma(f_i)$, $\gamma(f_i) \le x < \gamma(f_j)$, and $\gamma(f_j) \le x$ (see Figure 1 (c)).

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 any 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 Figure 1 (d)).

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 any 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 (d) in Lemma 14.

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 $\overline{f}_j \circ \overline{f}_i(x) = \overline{f}_i \circ \overline{f}_j(x)$.

□

Proof of Lemma 17

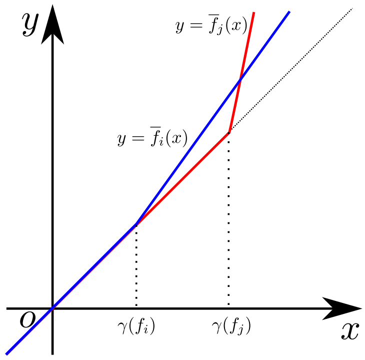
Lemma 17. *For monotone nondecreasing linear functions $f_i(x) = a_i x + b_i$ and $f_j(x) = a_j x + b_j$ ($a_i, a_j \ge 0$), we have the following statements.*

- (a) *If $\gamma(f_i) = \gamma(f_j)$, then $\gamma(f_i) = \gamma(f_j) = \gamma(f_j \circ f_i)$,*
- (b) *If $\gamma(f_i) < \gamma(f_j)$ and $a_i, a_j \ge 1$, then $\gamma(f_i) \le \gamma(f_j \circ f_i) \le \gamma(f_j)$,*
- (c) *If $\gamma(f_i) < \gamma(f_j)$ and $a_i, a_j < 1$, then $\gamma(f_i) \le \gamma(f_j \circ f_i) \le \gamma(f_j)$,*
- (d) *If $\gamma(f_i) < \gamma(f_j)$, $a_i < 1$, $a_j \ge 1$, and $a_i \cdot a_j \ge 1$, then $\gamma(f_j \circ f_i) \ge \gamma(f_j)$ ($> \gamma(f_i)$),*
- (e) *If $\gamma(f_i) < \gamma(f_j)$, $a_i < 1$, $a_j \ge 1$, and $a_i \cdot a_j < 1$, then $\gamma(f_j \circ f_i) \le \gamma(f_i)$ ($< \gamma(f_j)$),*
- (f) *If $\gamma(f_i) < \gamma(f_j)$, $a_i \ge 1$, $a_j < 1$, and $a_i \cdot a_j \ge 1$, then $\gamma(f_j \circ f_i) \le \gamma(f_i)$ ($< \gamma(f_j)$),*
- (g) *If $\gamma(f_i) < \gamma(f_j)$, $a_i \ge 1$, $a_j < 1$, and $a_i \cdot a_j < 1$, then $\gamma(f_j \circ f_i) \ge \gamma(f_j)$ ($> \gamma(f_i)$).*

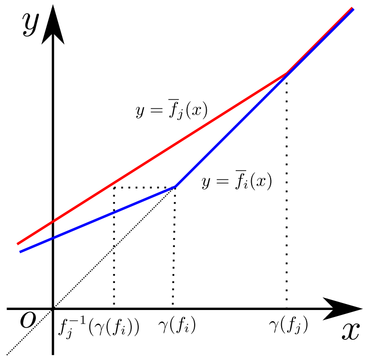
proof. To prove the theorem, we use the following facts for a real c and a linear function $f(x) = ax + b$:

- (i) If $a > 1$, then $f(c) > c \Leftrightarrow \gamma(f) < c$, $f(c) < c \Leftrightarrow \gamma(f) > c$, and $f(c) = c \Leftrightarrow \gamma(f) = c$.
- (ii) If $a < 1$, then $f(c) > c \Leftrightarrow \gamma(f) > c$, $f(c) < c \Leftrightarrow \gamma(f) < c$, and $f(c) = c \Leftrightarrow \gamma(f) = c$.
- (iii) If $a = 1$, then $f(c) \ge c \Leftrightarrow \gamma(f) = -\infty$, $f(c) < c \Leftrightarrow \gamma(f) = +\infty$.

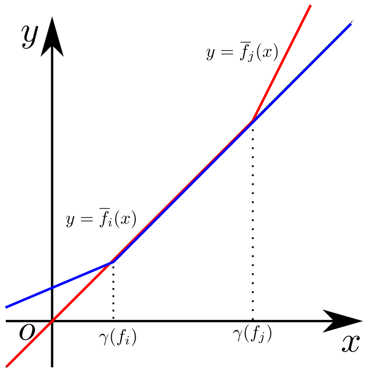
- (a) Let $d = \gamma(f_i) = \gamma(f_j)$. If $d = +\infty$, then $a_i = a_j = 1$ and $b_i, b_j < 0$. Thus, $\gamma(f_j \circ f_i) = \gamma(x + b_i + b_j) = +\infty$. If $d = -\infty$, then $a_i = a_j = 1$ and $b_i, b_j \ge 0$. Thus, $\gamma(f_j \circ f_i) = \gamma(x + b_i + b_j) = -\infty$. Otherwise (i.e., $a_i, a_j \ne 1$), 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 $\gamma(f_j \circ f_i) = d$.



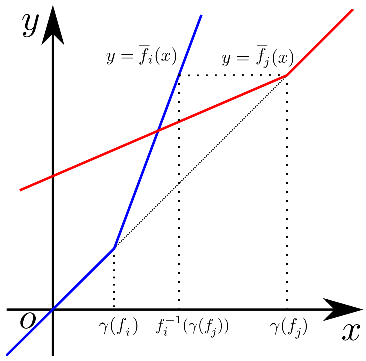
(a) $a_i, a_j \geq 1,\ \gamma(f_i) \leq \gamma(f_j)$



(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) $a_i \geq 1,\ 0 \leq a_j < 1,\ \gamma(f_i) \leq \gamma(f_j)$

Figure 1: Typical situations for the functions $\overline{f}_i$ and $\overline{f}_j$.

(b) By (i) and (iii) and $\gamma(f_i) < \gamma(f_j)$, we have

$$f_j \circ f_i(\gamma(f_i)) = f_j(\gamma(f_i)) \leq \gamma(f_i), \tag{15}$$
$$f_j \circ f_i(\gamma(f_j)) \geq f_j(\gamma(f_j)) = \gamma(f_j). \tag{16}$$

Therefore, we obtain $\gamma(f_i) \leq \gamma(f_j \circ f_i) \leq \gamma(f_j)$ where the first inequality holds by (15) and by (i) and (iii), and the second inequality holds by (16) and by (i) and (iii),

(c) By (ii) and $\gamma(f_i) < \gamma(f_j)$, we have

$$f_j \circ f_i(\gamma(f_i)) = f_j(\gamma(f_i)) \geq \gamma(f_i), \tag{17}$$
$$f_j \circ f_i(\gamma(f_j)) \leq f_j(\gamma(f_j)) = \gamma(f_j). \tag{18}$$

Therefore, we obtain $\gamma(f_i) \leq \gamma(f_j \circ f_i) \leq \gamma(f_j)$ where the first inequality holds by (17) and by (ii), and the second inequality holds by (18) and by (ii).

(d) By (ii) and $\gamma(f_i) < \gamma(f_j)$, we have

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

Therefore, we obtain $\gamma(f_i \circ f_j) \geq \gamma(f_j)$ by (i) and (iii).

(e) By (i) and (iii) and $\gamma(f_i) < \gamma(f_j)$, 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 (ii).

(f) By (ii) and $\gamma(f_i) < \gamma(f_j)$, we have

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

Therefore, we obtain $\gamma(f_i \circ f_j) \leq \gamma(f_i)$ by (i) and (iii).

(g) By (i), (iii), and $\gamma(f_i) < \gamma(f_j)$, 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 (ii).

$\square$