Partial Optimality in the Linear Ordering Problem

David Stein [1] **Bjoern Andres** [1 2]

Abstract

The linear ordering problem consists in finding a linear order $<$ on a finite set A so as to minimize the sum of costs associated with pairs of elements a, b for which $a < b$. The problem is NP-hard and APX-hard. We introduce algorithms for solving the problem *partially* by deciding efficiently for some pairs (a, b) whether $a < b$ is in an optimal solution. To do so, we construct maps from the feasible set of orders to itself and establish efficiently testable conditions on the cost function of the problem for which these maps are improving. We examine the effectiveness and efficiency of these conditions and algorithms empirically, on two data sets.

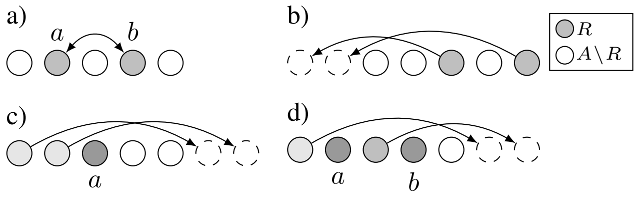

Figure 1. We solve instances of the NP-hard linear ordering *partially* by deciding efficiently for some pairs of elements a and b whether $a < b$ in an optimal solution to the problem. We do so by establishing and verifying conditions on the cost function which imply that certain transformations of feasible solutions are *improving maps* (Shekhovtsov, 2013). As transformations, we consider transpositions (a), groupings (b) and shiftings (c,d) of elements.

1. Introduction

We study the linear ordering problem, a combinatorial optimization problem whose feasible solutions are all linear orders on a finite set A. Given, for any pair $ab \in A^2$ with $a \neq b$, a cost $c_{ab} \in \mathbb{R}$, the objective is to find a strict linear order $<$ on A so as to minimize the sum of costs of those pairs ab for which $a < b$. Following Martí & Reinelt (2011), we state the problem in the form of a binary linear program:

Definition 1.1. Let $A \neq \emptyset$ and $P_A = \{ab \in A^2 \mid a \neq b\}$. The instance of the *linear ordering problem* with respect to A and $c \in \mathbb{R}^{P_A}$ has the form

$$\min_{x \in \{0,1\}^{P_A}} \sum_{ab \in P_A} c_{ab}\, x_{ab}$$

$$\text{subject to} \quad \forall a \in A\, \forall b \in A \setminus \{a\} \colon x_{ab} + x_{ba} = 1 \quad (1)$$

$$\forall a \in A\, \forall b \in A \setminus \{a\}\, \forall c \in A \setminus \{a, b\} \colon$$

$$x_{ab} + x_{bc} - 1 \leq x_{ac} \quad . \quad (2)$$

We let X_A denote the feasible set, i.e., the set of all $x \in \{0,1\}^{P_A}$ that satisfy (1) and (2), and let $\varphi_c \colon X_A \to \mathbb{R}$ denote the objective function with respect to A and $c \in \mathbb{R}^{P_A}$. We note that any feasible solution $x \in X_A$ defines a strict linear order $<_x$ on A such that for any $a \in A$ and any $b \in A \setminus \{a\}$ we have $a <_x b$ if and only if $x_{ab} = 1$.

In the field of machine learning, the linear ordering problem has two advantages as a model of ordering and ranking tasks: Firstly, its costs can be learned from independent classifications of pairs a, b as $a < b$ (Class 1) or $b < a$ (Class 0). For instance, one can learn how the cost $c_{ab} = f_\theta(y_a, y_b)$ associated with the decision $a < b$ depends (via a function f_θ) on features y_a, y_b of a and b. This requires only some labeled pairs. Not a single order needs to be given for learning. Secondly, solutions to the linear ordering problem are maximum a posteriori estimates under the assumption that decisions of pairs a, b occuring in the order $a < b$ are independent a priori and dependent a posteriori only by the fact that we associate zero probability with decisions violating (1) and (2) and equal (positive) probability with decisions satisfying these constraints. This is useful for estimating orders with little inductive bias.

The linear ordering problem is NP-hard (Garey & Johnson, 1979) and hard to approximate (Mishra & Sikdar, 2004). Here, we ask whether we can compute a partial solution to the problem efficiently, i.e., to decide efficiently for some pairs ab whether $a < b$ in an optimal solution. In order to find such *partial optimality*, we characterize and search for *improving maps*, a technique introduced by Shekhovtsov (2013); see also Shekhovtsov (2014); Shekhovtsov et al. (2015). Some technical proofs are deferred to Appendix A.

[1]Faculty of Computer Science, TU Dresden, Germany [2]Center for Scalable Data Analytics and AI Dresden/Leipzig. Correspondence to: Bjoern Andres <bjoern.andres@tu-dresden.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

In this article, we make the following contributions: Firstly, we establish four conditions of partial optimality for the linear ordering problem and define an algorithm for testing these conditions efficiently. Secondly, we show empirically, by quantitative experiments, that this algorithm is effective in fixing some variables of benchmark instances (Martí et al., 2012) of the linear ordering problem without compromising optimality, even though the entire problem is NP-hard and hard to approximate. Thirdly, we analyze quantitatively how the fraction of fixed variables depends on a parameter controlling the hardness of synthetic instances of the linear ordering problem we contribute. Fourthly, we show that the partial optimality we establish reduces the time for a branch-and-cut algorithm to find and certify optimal solutions to the entire problem to as little as 1/13 for the IO instances and as little as 1/33 for the Spec instances of the LOLIB benchmark.

2. Related Work

A common approach to learning and inferring orders is to learn an *embedding* $\xi\colon A \to \mathbb{R}$, often called a *scoring* or *ranking*, such that for any $a, b \in A$ and zero loss, we have $a \leq b$ if and only if $\xi(a) \leq \xi(b)$, and to infer an order on a set A by sorting its elements $a \in A$ by their score $\xi(a)$. Examples of this approach include Menon et al. (2022); Chang et al. (2020); Köppel et al. (2020); Zhang et al. (2022). A less common approach to learning and inferring orders is to learn costs $c\colon A^2 \to \mathbb{R}$ such that for any $a, b \in A$ and zero loss, we have $c_{ab} \leq 0$ if and only if $a \leq b$ (Lim et al., 2020), and to infer an order on a set A by solving the instance of the linear ordering problem with respect to A and c, cf. Definition 1.1. This approach is pursued by Tromble & Eisner (2009). An advantage of the second approach is that costs can be estimated from independent pairs whose order is known, e.g., by estimating for any $a, b \in A$ a probability $p_{ab} \in (0,1)$ of $a \leq b$, as discussed by Szczecinski & Sukheja (2023), and letting $c_{ab} = -\log p_{ab}$ and $c_{ba} = -\log(1 - p_{ab})$. A disadvantage of the second approach is that the linear ordering problem is NP-hard (Garey & Johnson, 1979) and APX-complete (Mishra & Sikdar, 2004).

Algorithms for the linear ordering problem fall into two categories. First are exact but not necessarily efficient algorithms such as branch-and-bound algorithms (Charon & Hudry, 2006; Kaas, 1981), branch-and-cut algorithms (Grötschel et al., 1984) and cutting plane algorithms (Mitchell & Borchers, 2000; 1996). Second are efficient but not necessarily exact heuristics (Ceberio et al. (2015); Martí & Reinelt (2011, Chapters 2-3)). To this end, He et al. (2022) learn a function $\chi\colon \mathbb{R}^{P_A} \to \mathbb{R}^A$ from the coefficients $c \in \mathbb{R}^{P_A}$ of an instance of the NP-hard linear ordering problem to the coefficients $\xi \in \mathbb{R}^A$ of an instance of the efficiently solvable sorting problem, and take the sorting of the elements $a \in A$ by their scores $\xi(a)$ as a feasible solution to the linear ordering problem. Heuristics like these are typically compared empirically in their application to a collection of instances of the linear ordering problem by Martí et al. (2012).

The algorithms we define here are different in that they solve the linear ordering problem partially, exactly and efficiently, e.g., by deciding for some pairs $ab \in P_A$ that $a < b$ in an optimal solution. Such *partial optimality* has been studied for various problems (Adams et al., 1998; Billionnet & Sutter, 1992; Hammer et al., 1984; Kappes et al., 2013; Kohli et al., 2008; Shekhovtsov, 2014; Shekhovtsov et al., 2015; Stein et al., 2023), notably for the correlation clustering problem (Alush & Goldberger, 2012; Lange et al., 2018; 2019; Stein et al., 2023). Our study of partial optimality for the linear ordering problem is analogous to prior work on partial optimality for the correlation clustering problem in two respects: Firstly, it is specific to a statement of the problem in the form of a binary linear program with transitivity constraints. Secondly, we do not consider reformulations in the form of an unconstrained optimization problem with infinite costs instead of constraints. While partial optimality for such reformulations might lead to different conditions on the linear order than the partial optimality we establish, we do not pursue this idea. Our study differs from prior work on correlation clustering in that the strict linear orders we consider are asymmetric whereas the equivalence relations that characterize the solutions to the correlation clustering problem are symmetric. Hence, the improving maps we define and the partial optimality conditions we establish are unrelated and novel.

3. Preliminaries

For clarity, we introduce basic notation and recall the definition and elementary property of improving maps. For any $r \in \mathbb{R}$, let $r^{\pm} := \max(0, \pm r)$. For any $c \in \mathbb{R}^{P_A}$, let $c^{\Delta} \in \mathbb{R}^{P_A}$ such that $\forall ab \in P_A\colon c^{\Delta}_{ab} = c_{ab} - c_{ba}$.

Definition 3.1. For any $\varphi\colon X \to \mathbb{R}$ and any $\sigma\colon X \to X$, we call σ *improving* for the problem $\min_{x \in X} \varphi(x)$ if and only if $\forall x \in X : \varphi(\sigma(x)) \leq \varphi(x)$.

Proposition 3.2. *Let $\varphi\colon X \to \mathbb{R}$ and $\sigma\colon X \to X$ improving. Moreover, let $X' \subseteq X$. If $\sigma(x) \in X'$ for every $x \in X$, there exists an optimal solution x^* to $\min_{x \in X} \varphi(x)$ such that $x^* \in X'$.*

Proof. Let x^* be an optimal solution to $\min_{x \in X} \varphi(x)$ such that $x^* \notin X'$. Then $\sigma(x^*)$ is also an optimal solution to $\min_{x \in X} \varphi(x)$ and $\sigma(x^*) \in X'$. □

Corollary 3.3. *Let $J \neq \emptyset$, $X \subseteq \{0,1\}^J$, $\varphi\colon X \to \mathbb{R}$ and $\sigma\colon X \to X$ improving. Moreover, let $j \in J$ and $b \in \{0,1\}$. If $\sigma(x)_j = b$ for every $x \in X$, there exists an optimal* *solution x^* to $\min_{x \in X} \varphi(x)$ such that $x^*_j = b$. In this case, the assignment of the value b to the variable x_j is said to be partially optimal by the improving map σ.*

4. Partial Optimality Conditions

4.1. Transposition

In this section, we state a condition on the cost function of the linear ordering problem under which a map is improving that swaps two elements $a, b \in A$, as depicted in Figure 1a, in all feasible solutions $x \in X_A$ where $a <_x b$. We begin by defining this map:

Definition 4.1. For any $A \neq \emptyset$ and any $\{a, b\} \in \binom{A}{2}$, define the *transposition map* $\tau_{\{a,b\}} \colon X_A \to X_A$ such that for all $x \in X_A$ and all $a'b' \in P_A$

$$\tau_{\{a,b\}}(x)_{a'b'} = \begin{cases} 1 - x_{a'b'} & \text{if } \{a', b'\} = \{a, b\} \\ x_{a'a} & \text{if } b' = b \wedge a' \neq a \\ x_{ab'} & \text{if } a' = b \wedge b' \neq a \\ x_{bb'} & \text{if } a' = a \wedge b' \neq b \\ x_{a'b} & \text{if } b' = a \wedge a' \neq b \\ x_{a'b'} & \text{if } \{a, b\} \cap \{a', b'\} = \emptyset \end{cases}$$

and the *conditional transposition map* $\tau^1_{\{a,b\}} \colon X_A \to X_A$ such that for all $x \in X_A$

$$\tau^1_{ab}(x) = \begin{cases} x & \text{if } x_{ab} = 1 \\ \tau_{\{a,b\}}(x) & \text{if } x_{ab} = 0 \end{cases} .$$

We proceed by calculating (Lemma 4.2) and tightly bounding (Lemma 4.3) the difference in cost incurred by this map:

Lemma 4.2. *Let $A \neq \emptyset$, $c \in \mathbb{R}^{P_A}$ and $x \in X_A$. Moreover, let $\{a, b\} \in \binom{A}{2}$ and $x' = \tau_{\{a,b\}}(x)$. Then:*

$$\begin{aligned} &\varphi_c(x') - \varphi_c(x) \\ &= c^\Delta_{ab}\,(1 - 2x_{ab}) + \sum_{d \in A \setminus \{a,b\}} \left(c^\Delta_{ad} + c^\Delta_{db}\right)(x_{bd} + x_{da} - 1) \quad . \end{aligned}$$

Lemma 4.3. *Let $A \neq \emptyset$, $\{a, b\} \in \binom{A}{2}$ and $e \in \mathbb{R}^{A \setminus \{a,b\}}$. Then:*

$$\max_{\substack{x \in X_A \\ x_{ab} = 0}} \sum_{d \in A \setminus \{a,b\}} (x_{bd} + x_{da} - 1)\, e_d = \sum_{d \in A \setminus \{a,b\}} e^+_d \quad . \tag{3}$$

Based on this bound, we establish a (necessary and) sufficient condition for a conditional transposition map to be improving, and note the resulting partial optimality:

Proposition 4.4. *Let $A \neq \emptyset$, $c \in \mathbb{R}^{P_A}$ and $ab \in P_A$. If (4) holds, τ^1_{ab} is improving. Moreover, there is an optimal solution x^* to the linear ordering problem with $x^*_{ab} = 1$.*

$$c^\Delta_{ab} + \sum_{d \in A \setminus \{a,b\}} \left(c^\Delta_{ad} + c^\Delta_{db}\right)^+ \leq 0 \tag{4}$$

Proof. Let $x' = \tau^1_{ab}(x)$. For all $x \in X_A$ with $x_{ab} = 1$, we have $\varphi_c(x') = \varphi_c(x)$. For all $x \in X_A$ with $x_{ab} = 0$, we have by Lemma 4.2:

$$\begin{aligned} &\varphi_c(x') - \varphi_c(x) \\ &= c^\Delta_{ab} + \sum_{d \in A \setminus \{a,b\}} (x_{da} + x_{bd} - 1)\left(c^\Delta_{ad} + c^\Delta_{db}\right) \\ &\leq c^\Delta_{ab} + \max_{\substack{x \in X_A \\ x_{ab} = 0}} \sum_{d \in A \setminus \{a,b\}} (x_{da} + x_{bd} - 1)\left(c^\Delta_{ad} + c^\Delta_{db}\right) \\ &\overset{(3)}{=} c^\Delta_{ab} + \sum_{d \in A \setminus \{a,b\}} \left(c^\Delta_{ad} + c^\Delta_{db}\right)^+ \\ &\overset{(4)}{\leq} 0 \end{aligned}$$

Thus, τ^1_{ab} is improving. Moreover, $\forall x \in X_A \colon x'_{ab} = 1$. □

Applying Proposition 4.4 with a and b swapped yields an analogous partial optimality condition:

Corollary 4.5. *Let $A \neq \emptyset$, $c \in \mathbb{R}^{P_A}$ and $ab \in P_A$. If (5) holds, there is an optimal solution x^* to the linear ordering problem such that $x^*_{ab} = 0$.*

$$c^\Delta_{ba} + \sum_{d \in A \setminus \{a,b\}} \left(c^\Delta_{bd} + c^\Delta_{da}\right)^+ \leq 0 \tag{5}$$

4.2. Grouping

In this section, we state a condition on the cost function of the linear ordering problem under which a map is improving that moves a subset $R \subseteq A$ to the front of the order, as depicted in Figure 1b, without changing the order of elements in R or $A \setminus R$. We begin by defining this map:

Definition 4.6. Let $A \neq \emptyset$ and $R \subseteq A$. The *grouping map* $\sigma_R \colon X_A \to X_A$ is such that for all $x \in X_A$ and $ab \in P_A$:

$$\sigma_R(x)_{ab} = \begin{cases} x_{ab} & \text{if } ab \in P_R \vee ab \in P_{A \setminus R} \\ 1 & \text{if } a \in R \wedge b \in A \setminus R \\ 0 & \text{if } a \in A \setminus R \wedge b \in R \end{cases} .$$

Lemma 4.7. *Let $A \neq \emptyset$, $c \in \mathbb{R}^{P_A}$ and $x \in X_A$. Moreover, let $R \subseteq A$ and $x' = \sigma_R(x)$. Then:*

$$\varphi_c(x') - \varphi_c(x) = \sum_{a \in R} \sum_{b \in A \setminus R} c^\Delta_{ab}\, x_{ba} \quad .$$

Next, we note a sufficient condition for the grouping map to be improving, and the resulting partial optimality:

Proposition 4.8. *Let $A \neq \emptyset$ and $c \in \mathbb{R}^{P_A}$. If there exists $R \subseteq A$ such that (6) holds, there is an optimal solution x^* to the linear ordering problem such that $\forall a \in R\ \forall b \in A \setminus R \colon x^*_{ab} = 1$.*

$$\forall a \in R\ \forall b \in A \setminus R \colon \quad c^\Delta_{ab} \leq 0 \tag{6}$$

Proof. Let $x' = \sigma_R(x)$. Firstly, by Lemma 4.7, $\forall x \in X_A$:

$$\begin{aligned}\varphi_c(x') - \varphi_c(x) &= \sum_{a \in R} \sum_{b \in A \setminus R} c^{\Delta}_{ab}\, x_{ba} \\ &\le \sum_{a \in R} \sum_{b \in A \setminus R} c^{\Delta,+}_{ab} \overset{(6)}{=} 0 \ .\end{aligned}$$

Secondly, $\forall a \in R\ \forall b \in A \setminus R$: $x'_{ab} = 1$. □

If Proposition 4.8 is satisfied for a set R, the linear ordering problem with respect to A and c decomposes into two independent sub-problems. More specifically, with $X'_{AR} := \{x \in X_A \mid \forall a \in R\ \forall b \in A \setminus R\colon x_{ab} = 1\}$:

$$\begin{aligned}&\min_{x \in X'_{A,R}} \varphi_c(x) \\ &= \min_{x \in X_R} \varphi_{c|_{P_R}}(x) + \min_{x \in X_{A\setminus R}} \varphi_{c|_{P_{A\setminus R}}}(x) + \sum_{a \in R} \sum_{b \in A \setminus R} c_{ab} \ .\end{aligned}$$

Moreover, given solutions

$$x' \in \operatorname*{argmin}_{x \in X_R} \varphi_{c|_{P_R}}(x) \ , \quad x'' \in \operatorname*{argmin}_{x \in X_{A\setminus R}} \varphi_{c|_{P_{A\setminus R}}}(x) \ ,$$

an optimal solution to $\min_{x \in X'_{A,R}} \varphi_c(x)$ is given by the $x \in X'_{A,R}$ such that

$$x_{a'b'} = \begin{cases} x'_{a'b'} & \text{if } a'b' \in P_R \\ x''_{a'b'} & \text{if } a'b' \in P_{A\setminus R} \\ 1 & \text{if } a' \in R \wedge b' \in A \setminus R \\ 0 & \text{if } a' \in A \setminus R \wedge b' \in R \end{cases} \ .$$

Example 1. Let $\epsilon > \delta > 0$ and $A = A_1 \cup A_2$, $A_1 \cap A_2 = \emptyset$ such that $c^{\Delta}_{ab} = -\delta$ for all $a \in A_1$ and $b \in A_2$ and $|c^{\Delta}_{ab}| \le \epsilon$ for all $ab \in P_{A_1} \cup P_{A_2}$. Proposition 4.8 leads to the partially optimal assignment $x_{ab} = 1$ for all $a \in A_1$ and $b \in A_2$, since $c^{\Delta}_{ab} = -\delta < 0$ for all $a \in A_1$ and $b \in A_2$. I.e., the problem decomposes into independent sub-problems thanks to partial optimality. Note also that the same partially optimal assignment can be found by Proposition 4.4 under the additional condition $\delta \ge \frac{|A|-2}{|A|-1}\epsilon$. This shows that partial optimality due to Proposition 4.8 is not subsumed under partial optimality due to Proposition 4.4.

4.3. Conditional Grouping

In this section, we generalize Proposition 4.8. More specifically, we state a condition on the cost function of the linear ordering problem under which a map is improving that moves a subset $R \subseteq A$ to the front of the order, as depicted in Figure 1b, without changing the order of elements in R or $A \setminus R$, but does so only for those feasible solutions $x \in X_A$ where $b <_x a$ for given elements $a \in R$ and $b \in A \setminus R$. We begin by defining this map:

Definition 4.9. For any $R \subseteq A$, any $a \in R$ and any $b \in A \setminus R$, define the *conditional grouping map* $\sigma^{ab}_R\colon X_A \to X_A$ such that

$$\forall x \in X_A\colon \quad \sigma^{ab}_R(x) = \begin{cases} x & \text{if } x_{ab} = 1 \\ \sigma_R(x) & \text{if } x_{ab} = 0 \end{cases} \ .$$

Lemma 4.10. *Let $A \neq \emptyset$, $c \in \mathbb{R}^{P_A}$, $ab \in P_A$ and $R \subseteq A$ such that $a \in R$ and $b \in A \setminus R$. If $x \in X_A$ such that $x_{ab} = 0$, and $x' = \sigma_R(x)$, then*

$$\varphi_c(x') - \varphi_c(x) \le -c^{\Delta,-}_{ab} + \sum_{a' \in R} \sum_{b' \in A \setminus R} c^{\Delta,+}_{a'b'} \ .$$

We note a sufficient condition for the conditional grouping map to be improving, and the resulting partial optimality:

Proposition 4.11. *Let $A \neq \emptyset$, $c \in \mathbb{R}^{P_A}$, $ab \in P_A$ and $R \subseteq A$ such that $a \in R$ and $b \in A \setminus R$. If (7) holds, then there is an optimal solution x^* to the linear ordering problem such that $x^*_{ab} = 1$.*

$$c^{\Delta,-}_{ab} \ge \sum_{a' \in R} \sum_{b' \in A \setminus R} c^{\Delta,+}_{a'b'} \tag{7}$$

Proof. Let $\sigma^{ab}_R\colon X_A \to X_A$ and $x' = \sigma^{ab}_R(x)$. For all $x \in X_A$ with $x_{ab} = 1$: $\varphi_c(x') = \varphi_c(x)$. And for all $x \in X_A$ with $x_{ab} = 0$, by Lemma 4.10:

$$\varphi_c(x') - \varphi_c(x) = -c^{\Delta,-}_{ab} + \sum_{a' \in R} \sum_{b' \in A \setminus R} c^{\Delta,+}_{a'b'} \overset{(7)}{\le} 0 \ .$$

Moreover, $x'_{ab} = 1$. □

Corollary 4.12. *Let $A \neq \emptyset$, $c \in \mathbb{R}^{P_A}$, $ab \in P_A$ and $R \subseteq A$ such that $b \in R$ and $a \in A \setminus R$. If (8) holds, there is an optimal solution x^* to the linear ordering problem with $x^*_{ab} = 0$.*

$$c^{\Delta,-}_{ba} \ge \sum_{a' \in R} \sum_{b' \in A \setminus R} c^{\Delta,+}_{a'b'} \tag{8}$$

4.4. Shifting

In this section, we state conditions on the cost function of the linear ordering problem under which a map is improving that shifts all elements before an element $a \in A$ to the end of the order, as depicted in Figure 1c. We begin by defining this map:

Definition 4.13. For any $a \in A$, define the *shift map* $s_a\colon X_A \to X_A$ such that for all $x \in X_A$ and all $a'b' \in P_A$:

$$s_a(x)_{a'b'} = \begin{cases} 1 & \text{if } a' = a \\ 0 & \text{if } b' = a \\ 0 & \text{if } a' <_x a <_x b' \\ 1 & \text{if } b' <_x a <_x a' \\ x_{a'b'} & \text{otherwise} \end{cases} \ .$$

Lemma 4.14. *Let $c \in \mathbb{R}^{P_A}$, $x \in X_A$, $a \in A$ and $x' = s_a(x)$. Then*

$$\varphi_c(x') - \varphi_c(x) = \sum_{d <_x a} \sum_{d' \in A \setminus \{d\}} c^{\Delta}_{d'd} \enspace .$$

Next, we state a sufficient condition for the shift map to be improving, along with the resulting partial optimality:

Proposition 4.15. *Let $a \in A$ and $c \in \mathbb{R}^{P_A}$. If (9) holds, then there exists an optimal solution x^* to the linear ordering problem such that $\forall b \in A \setminus \{a\}\colon x^*_{ab} = 1$.*

$$\forall d \in A \setminus \{a\}\colon \sum_{d' \in A \setminus \{d\}} c^{\Delta}_{d'd} \leq 0 \tag{9}$$

Proof. Let $x' = s_a(x)$. Firstly, by Lemma 4.14:

$$\begin{aligned} \varphi_c(x') - \varphi_c(x) &= \sum_{d <_x a} \sum_{d' \in A \setminus \{d\}} c^{\Delta}_{d'd} \\ &\leq \sum_{d \in A \setminus \{a\}} \max\left(0, \sum_{d' \in A \setminus \{d\}} c^{\Delta}_{d'd}\right) = 0 \enspace . \end{aligned}$$

Secondly, for any $b \in A \setminus \{a\}$, we have $x'_{ab} = 1$. □

Corollary 4.16. *Let $a \in A$ and $c \in \mathbb{R}^{P_A}$. If (10) holds, there is an optimal solution x^* to the linear ordering problem such that $\forall b \in A \setminus \{a\}\colon x^*_{ab} = 0$.*

$$\forall d \in A \setminus \{a\}\colon \sum_{d' \in A \setminus \{d\}} c^{\Delta}_{dd'} \leq 0 \tag{10}$$

4.5. Generalized Shifting

In this section, we state conditions on the cost function of the linear ordering problem under which a map is improving that shifts all elements before the maximum of two elements $a, b \in A$ to the end of the order, except the minimum of a and b. An example is depicted in Figure 1d. We begin by defining this map:

Definition 4.17. For any distinct $a, b \in A$, define the *generalized shift map* $s_{\{a,b\}}\colon X_A \to X_A$ such that for all $x \in X_A$ and all $a'b' \in P_A$:

$$s_{\{a,b\}}(x)_{a'b'} = \begin{cases} 1 & \text{if } a' \in \{a,b\} \wedge b' \notin \{a,b\} \\ 0 & \text{if } b' \in \{a,b\} \wedge a' \notin \{a,b\} \\ 1 & \text{if } b' <_x \max_x\{a,b\} <_x a' \\ 0 & \text{if } a' <_x \max_x\{a,b\} <_x b' \\ x_{a'b'} & \text{otherwise} \end{cases} \enspace .$$

Lemma 4.18. *Let $A \neq \emptyset$, $c \in \mathbb{R}^{P_A}$ and $x \in X_A$. Moreover, let $a, b \in A$ distinct and $x' = s_{\{a,b\}}(x)$. If $x_{ab} = 1$ we have*

$$\begin{aligned} \varphi_c(x') - \varphi_c(x) = &\sum_{d <_x \min_x\{a,b\}} \left(c^{\Delta}_{ad} + c^{\Delta}_{bd}\right) + \sum_{\substack{d >_x \min_x\{a,b\} \\ d <_x \max_x\{a,b\}}} c^{\Delta}_{bd} \\ &+ \sum_{d >_x \max_x\{a,b\}} \sum_{d' \in A \setminus \{a,b,d\}} c^{\Delta}_{dd'} \enspace . \end{aligned}$$

If $x_{ab} = 0$ we have

$$\begin{aligned} \varphi_c(x') - \varphi_c(x) = &\sum_{d <_x \min_x\{a,b\}} \left(c^{\Delta}_{ad} + c^{\Delta}_{bd}\right) + \sum_{\substack{d >_x \min_x\{a,b\} \\ d <_x \max_x\{a,b\}}} c^{\Delta}_{ad} \\ &+ \sum_{d >_x \max_x\{a,b\}} \sum_{d' \in A \setminus \{a,b,d\}} c^{\Delta}_{dd'} \enspace . \end{aligned}$$

Next, we state a sufficient condition for the generalized shift map to be improving, along with the resulting partial optimality:

Proposition 4.19. *Let $A \neq \emptyset$, $c \in \mathbb{R}^{P_A}$ and $\{a,b\} \in \binom{A}{2}$. If (11) and (12) hold, then there is an optimal solution x^* to the linear ordering problem such that $x^*_{ad} = x^*_{bd} = 1, \forall d \in A \setminus \{a,b\}$.*

$$\sum_{d \in A \setminus \{a,b\}} \max\left(c^{\Delta}_{ad} + c^{\Delta}_{bd}, c^{\Delta}_{bd}, \sum_{d' \in A \setminus \{a,b,d\}} c^{\Delta}_{dd'}\right) \leq 0 \tag{11}$$

$$\sum_{d \in A \setminus \{a,b\}} \max\left(c^{\Delta}_{ad} + c^{\Delta}_{bd}, c^{\Delta}_{ad}, \sum_{d' \in A \setminus \{a,b,d\}} c^{\Delta}_{dd'}\right) \leq 0 \tag{12}$$

Proof. Let $x' = s_{\{a,b\}}(x)$. Firstly, by Lemma 4.18, for $x \in X_A$ such that $x_{ab} = 1$:

$$\begin{aligned} &\varphi_c(x') - \varphi_c(x) \\ &= \sum_{d <_x \min_x\{a,b\}} \left(c^{\Delta}_{ad} + c^{\Delta}_{bd}\right) + \sum_{\substack{d >_x \min_x\{a,b\} \\ d <_x \max_x\{a,b\}}} c^{\Delta}_{bd} \\ &\quad + \sum_{d >_x \max_x\{a,b\}} \sum_{d' \in A \setminus \{a,b,d\}} c^{\Delta}_{dd'} \\ &\leq \sum_{d \in A \setminus \{a,b\}} \max\left(c^{\Delta}_{ad} + c^{\Delta}_{bd}, c^{\Delta}_{bd}, \sum_{d' \in A \setminus \{a,b,d\}} c^{\Delta}_{dd'}\right) \\ &\overset{(11)}{\leq} 0 \enspace . \end{aligned}$$

Secondly, for x such that $x_{ab} = 0$:

$$\begin{aligned}
&\varphi_c(x') - \varphi_c(x) \\
&= \sum_{d <_x \min_x\{a,b\}} (c^\Delta_{ad} + c^\Delta_{bd}) + \sum_{\substack{d >_x \min_x\{a,b\} \\ d <_x \max_x\{a,b\}}} c^\Delta_{ad} \\
&\quad + \sum_{d >_x \max_x\{a,b\}} \sum_{d' \in A \setminus \{a,b,d\}} c^\Delta_{dd'} \\
&\leq \sum_{d \in A \setminus \{a,b\}} \max \left(c^\Delta_{ad} + c^\Delta_{bd}, c^\Delta_{ad}, \sum_{d' \in A \setminus \{a,b,d\}} c^\Delta_{dd'} \right) \\
&\overset{(12)}{\leq} 0 \ .
\end{aligned}$$

Thirdly, $\forall d \in A \setminus \{a,b\}\colon x'_{ad} = x'_{bd} = 1$. □

Corollary 4.20. *Let $A \neq \emptyset$, $c \in \mathbb{R}^{P_A}$ and $a \in A$. If (13) and (14) hold, there is an optimal solution x^* to the linear ordering problem such that $x^*_{ad} = x^*_{bd} = 0$, $\forall d \in A \setminus \{a,b\}$.*

$$\sum_{d \in A \setminus \{a,b\}} \max \left(c^\Delta_{da} + c^\Delta_{db}, c^\Delta_{db}, \sum_{d' \in A \setminus \{a,b,d\}} c^\Delta_{d'd} \right) \leq 0 \quad (13)$$

$$\sum_{d \in A \setminus \{a,b\}} \max \left(c^\Delta_{da} + c^\Delta_{db}, c^\Delta_{da}, \sum_{d' \in A \setminus \{a,b,d\}} c^\Delta_{d'd} \right) \leq 0 \quad (14)$$

5. Efficient Algorithms

In this section, we define and discuss algorithms for testing all partial optimality conditions introduced in Section 4.

Transposition and Shifting. Propositions 4.4, 4.15 and 4.19 as well as Corollaries 4.16 and 4.20 are tested by enumeration: For Proposition 4.4, testing (4) for a fixed pair ab takes time $\mathcal{O}(|A|)$. Thus, testing it for all pairs takes time $\mathcal{O}(|A|^3)$. Testing Proposition 4.4 is equivalent to testing Corollary 4.5. Thus, testing one of these is sufficient. For Proposition 4.15 and Corollary 4.16, we compute the right-hand side of (9) for every $d \in A$ in time $\mathcal{O}(|A|^2)$. Then, we test for every $a \in A$ whether (9) is satisfied for every $d \in A \setminus \{a\}$, also in time $\mathcal{O}(|A|^2)$. Thus, testing Proposition 4.15 and Corollary 4.16 for every $a \in A$ takes time $\mathcal{O}(|A|^2)$ in total. Regarding Proposition 4.19 and Corollary 4.20, computing the left-hand sides of (11), (12), (13) or (14) for a fixed pair $\{a,b\}$ takes time $\mathcal{O}(|A|^2)$. Thus, testing Proposition 4.19 and Corollary 4.20 for all pairs takes time $\mathcal{O}(|A|^4)$.

Grouping. Regarding Proposition 4.8, we search for candidates $R \subseteq A$ by introducing the set $E^+ = \{ab \in P_A \mid c^\Delta_{ab} > 0\}$ of positive-cost pairs. Then, a subset $R \subseteq A$ fulfills (6) if and only if

$$E^+ \cap (R \times (A \setminus R)) = \emptyset \ . \quad (15)$$

To find such R, we iterate over all initializations $R = \{a\}$ with $a \in A$. For each such initialization, we add elements b to R with $ab \in E^+ \cap (R \times (A \setminus R))$. For any initialization, the algorithm terminates with R fulfilling (15). If $R \neq A$ we take R as a candidate and exploit the condition to obtain two smaller problems. If $R = A$, we proceed with the next initialization. See Algorithm 1. In practice, we add elements to R by breadth-first search. Moreover, we sort the initializations $R = \{a\}$ by increasing out-degree in the graph (A, E^+). For each initialization, this algorithm takes time $\mathcal{O}(|A|^2)$. In total, it takes time $\mathcal{O}(|A|^3)$.

Conditional Grouping. For Proposition 4.11 and a fixed $ab \in P_A$, we find candidates $R \subseteq A$ that satisfy (7) with a maximum margin by minimizing the right-hand side, i.e.,

$$\min_{\substack{R \subseteq A \\ a \in R, b \notin R}} \sum_{a' \in R} \sum_{b' \in A \setminus R} c^{\Delta,+}_{a'b'} \ .$$

For each pair ab, this problem can be cast in the form of a min-ab-cut problem with non-negative costs (see Section 6 for details). Therefore, Proposition 4.11 can be tested by solving $\mathcal{O}(|A|^2)$ max-flow problems. In our code, we solve these by means of a c++ implementation of the push-relabel algorithm of Goldberg & Tarjan (1988).

6. Combining Partial Optimality

Partially optimal assignments of zeros and ones to the variables x due to Propositions 4.8, 4.15 and 4.19 and Corollaries 4.16 and 4.20 hold simultaneously, as these assignments are such that the linear ordering problem decomposes into independent sub-problems that can be solved independently, without compromising optimality. In this section, we show that also assignments due to Proposition 4.4 can be applied simultaneously (Proposition 6.3), and that assignments due to Proposition 4.11 or Corollary 4.12 cannot be applied simultaneously (Example 2). We then generalize Proposition 4.11 to arrive at a criterion that we can apply iteratively

Algorithm 1 Seeded Component Growing

```
1:  input: $A \neq \emptyset$, $c : P_A \to \mathbb{R}$
2:  initialize: $E^+ = \{ab \in P_A \mid c^\Delta_{ab} > 0\}$, queue $Q = A$
3:  repeat
4:      $a := Q.pop$
5:      $R := \{a\}$
6:      repeat
7:          choose $ab \in (R \times A \setminus R) \cap E^+$
8:          $R := R \cup \{b\}$
9:      until $(R \times A \setminus R) \cap E^+ = \emptyset$
10:     if $R \neq A$: then
11:         return $R$
12:     end if
13: until $Q = \emptyset$
```

(Proposition 6.5). Finally, we define an algorithm for iteratively testing all partial optimality conditions.

We now show that assignments due to Proposition 4.4 can be applied simultaneously. For clarity, we introduce some notation. If we decide for a subset $P \subset P_A$ of pairs $ab \in P$ that $a < b$, these decisions need not be consistent in the sense that there exists a feasible solution $x \in X_A$ such that for all $ab \in P$ we have $x_{ab} = 1$. Those $P \subseteq P_A$ that are consistent are the edge sets of an acyclic digraph $G = (A', P)$ with $A' \subseteq A$. We call any such digraph a *partial 1-assignment* and let $\mathcal{G}_A$ denote the set of all such graphs. Moreover, we let $X_{A'P} = \{x \in X_A \mid \forall ab \in P \colon x_{ab} = 1\}$ denote the set of those feasible solutions to the linear ordering problem that are consistent with (A', P).

Lemma 6.1. *For any $(A', P) \in \mathcal{G}_A$, we have $X_{A'P} \neq \emptyset$.*

Lemma 6.2. *For any $(A', P) \in \mathcal{G}_A$ and any $x \in X_A$, there exists a composition σ of functions $\{\tau^1_{ab}\}_{ab \in P}$, such that $\sigma(x)_{ab} = 1$ for all $ab \in P$.*

Proposition 6.3. *Let $A \neq \emptyset$, $c \in \mathbb{R}^{P_A}$ and $(A', P) \in \mathcal{G}_A$. If (16) holds, then there exists an optimal solution x^* to the linear ordering problem such that $\forall ab \in P \colon x^*_{ab} = 1$.*

$$\forall ab \in P \colon \quad c^\Delta_{ab} + \sum_{d \in A \setminus \{a,b\}} \left(c^\Delta_{ad} + c^\Delta_{db}\right)^+ \leq 0 \qquad (16)$$

Proof. By Lemma 6.2 there is a composition σ of functions $\{\tau^1_{ab}\}_{ab \in P}$ such that $\sigma(x)_{ab} = 1$ for all $ab \in P$. Firstly, each τ^1_{ab} is improving by (16), i.e., for all $ab \in P$ and all $x \in X_A$ we have $\varphi_c(\tau^1_{ab}(x)) \leq \varphi_c(x)$. Let $(a_i b_i \mid 1 \leq i \leq n)$ be this sequence in P, such that $\circ_{i=1}^n \tau^1_{a_i b_i} = \sigma$ and $x^{(k)} = \circ_{i=1}^k \tau^1_{a_i b_i}(x)$ and $x^{(0)} = x$. Then the difference in objective values is given by $\varphi_c(x^{(n)}) - \varphi_c(x) = \sum_{i=0}^{n-1} \left(\varphi_c(x^{(i+1)}) - \varphi_c(x^{(i)})\right) \leq 0$. □

Next, we show: Partially optimal assignments due to Proposition 4.11 or Corollary 4.12 need not hold simultaneously. I.e., if $a_1 b_1 \in P_A$ and $R_1 \subseteq A$ such that (7) holds true and $a_2 b_2 \in P_A$ and $R_2 \subseteq A$ such that (7) holds true, there need not exist an optimal solution x^* to the linear ordering problem such that $x^*_{a_1 b_1} = 1 = x^*_{a_2 b_2} = 1$:

Example 2. Let $A = \{1, 2, 3\}$ and $c_{12} = -2$, $c_{21} = -1$, $c_{13} = 3$, $c_{31} = 1$, $c_{23} = -2$ and $c_{32} = -1$.

$$\begin{array}{ll}
1 < 2 < 3 & \varphi_c(x) = c_{12} + c_{13} + c_{23} = -2 + 3 - 2 = -1 \\
2 < 1 < 3 & \varphi_c(x) = c_{21} + c_{23} + c_{13} = -1 - 2 + 3 = 0 \\
2 < 3 < 1 & \varphi_c(x) = c_{23} + c_{21} + c_{31} = -2 - 1 + 1 = -2 \\
1 < 3 < 2 & \varphi_c(x) = c_{13} + c_{12} + c_{32} = 3 - 2 - 1 = 0 \\
3 < 1 < 2 & \varphi_c(x) = c_{31} + c_{32} + c_{12} = 1 - 1 - 2 = -2 \\
3 < 2 < 1 & \varphi_c(x) = c_{32} + c_{21} + c_{31} = -1 - 1 + 1 = -1
\end{array}$$

Both $2 < 3 < 1$ and $3 < 1 < 2$ are optimal. Proposition 4.11 is true for $a_1 = 1$, $b_1 = 2$ and $R_1 = \{1, 3\}$:

$$c^{\Delta,-}_{12} = 1 \geq \sum_{a' \in \{1,3\}} \sum_{b' \in \{2\}} c^{\Delta,+}_{a'b'} = 0 + 1 = 1 \ .$$

Proposition 4.11 is true for $a_2 = 2$, $b_2 = 3$ and $R_1 = \{2\}$:

$$c^{\Delta,-}_{23} = 1 \geq \sum_{a' \in \{2\}} \sum_{b' \in \{1,3\}} c^{\Delta,+}_{a'b'} = 0 + 1 = 1 \ .$$

The two optimal solutions $3 < 1 < 2$ and $2 < 3 < 1$ satisfy $x^*_{12} = 1$ and $x^*_{23} = 1$, respectively, but there is no optimal solution x^* satisfying both $x^*_{12} = 1$ and $x^*_{23} = 1$.

Next, we generalize Proposition 4.11 to arrive at a criterion that we can apply iteratively.

Lemma 6.4. *Let $(A', P) \in \mathcal{G}_A$ and $R \subseteq A$. If*

$$\forall ab \in P \colon \quad a \in R \vee b \notin R$$

then $\sigma_R(X_{A'P}) \subseteq X_{A'P}$.

Proposition 6.5. *Let $A \neq \emptyset$, $c \in \mathbb{R}^{P_A}$, $(A', P) \in \mathcal{G}_A$, $ab \in P_A \setminus P$ and $R \subseteq A$ such that $a \in R$ and $b \in A \setminus R$. If (17) and (18) hold, there exists an optimal solution x^* to the linear ordering problem such that $x^*_{ab} = 1$ and $x \in X_{A'P}$.*

$$\forall a'b' \in P \colon a' \in R \vee b' \notin R \qquad (17)$$

$$c^{\Delta,-}_{ab} \geq \sum_{a' \in R} \sum_{b' \in A \setminus R} c^{\Delta,+}_{a'b'} \qquad (18)$$

Proof. σ^{ab}_R is improving by the same argument as in the proof of Proposition 4.11. By Lemma 6.4 and (18), we have $\forall a'b' \in P \colon \sigma^{ab}_R(x)_{a'b'} = 1$. □

Algorithmically, we search for subsets $R \subseteq A$ that satisfy (18) by setting the cost $c_{b'a'}$ for every $a'b' \in P$ to a large positive constant such that the edge $b'a'$ cannot be in a minimum ab-cut. See Algorithm 2. In order to test all partial optimality conditions iteratively, we operate as follows. In a first step, we test all conditions that potentially decompose the linear ordering problem into sub-problems that can be solved independently without compromising optimality, namely Propositions 4.8, 4.15 and 4.19 and Corollaries 4.16 and 4.20, in this order. As soon as one of these conditions can be applied, we decompose the problem and process the sub-problems independently. If none of the aforementioned conditions can be verified, we proceed to a second step in which we test Propositions 6.3 and 6.5 until no more variables can be fixed.

7. Numerical Experiments

In this section, we analyze the algorithms defined in Section 5 empirically on two collections of instances of the linear ordering problem. We report the fraction of variables fixed by partial optimality conditions as well as the runtime. Additionally, we implement a branch-and-cut algorithm for the linear ordering problem using the state-of-the-art integer programming software Gurobi (Gurobi Optimization, LLC, 2023). In this algorithm, we separate violated inequalities of the form (1) and (2). We compare the time until termination of this algorithm for the linear ordering problem and the linear ordering problem with variables fixed according to the partial optimality conditions we establish. We test combinations of partial optimality conditions as described in Section 6, and we also test each condition separately. The complete c++ code for reproducing these experiments is provided in Stein (2024). All experiments are conducted on a single core of a Ryzen 9 7900X equipped with 64 GB of RAM.

Synthetic Instances. For a systematic study, we synthesize instances of the linear ordering problem with $|A|$ between 5 and 200, w.r.t. a given order $<$ on A, and w.r.t. a parameter $\alpha \in [0, 1]$. Costs c_{ab} of pairs ab are drawn from two normal distributions with means $-1 + \alpha$ and $1 - \alpha$ and standard deviation $\sigma = 0.2$, depending on whether $a < b$ or $b < a$.

In Figure 2, we report the percentage of variables fixed by partial optimality (a,c) and the runtime (b,d), as a function of α (a,b) and as a function of n (c,d) when applying combinations of partial optimality conditions as described in Section 6. It can be seen from (a) that for $\alpha = 0.4$ nearly all variables are fixed by partial optimality, and for $\alpha = 1.0$ nearly none are, for any n. It can be seen from (c) for $\alpha = 0.65$ and $\alpha = 0.7$ that the percentage of fixed variables drops with increasing n. It can be seen from (b) that for any n, the runtime is a non-trivial function of α. It can be seen from (d) that the runtime is approximately polynomial in n and does not exceed four minutes for $n = 200$.

In Figure 3, we report the percentage of fixed variables and

Algorithm 2 Algorithm for Proposition 6.5

```
input: $A \neq \emptyset$, $c : P_A \to \mathbb{R}$, $(A', P) \in \mathcal{G}_A$, $ab \in P_A$
initialize:
        $E = \{ab \in P_A \mid c^{\Delta}_{ab} > 0 \vee ba \in P\}$
        $c' : E \to \mathbb{R}^+ : ab \mapsto \begin{cases} \infty & \text{if } ba \in P \\ c^{\Delta,+}_{ab} & \text{if } ba \notin P \end{cases}$
if $ab \notin P \wedge ba \notin P$ then
    $(\text{flow}, \text{cut}) := \text{min-}ab\text{-cut}((A, E), c')$
    for $a'b' \in \text{cut}$ do
        if $c^{\Delta,-}_{a'b'} \geq \text{flow} \wedge a'b' \notin P \wedge b'a' \notin P$ then
            $(A', P) := (A' \cup \{a', b'\}, P \cup \{a'b'\})$
            $(A', P) := \text{transitiveClosure}((A', P))$
        end if
    end for
end if
```

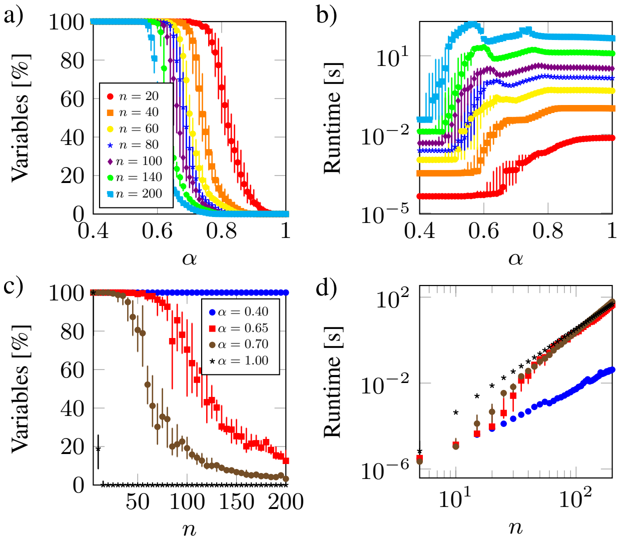

Figure 2. We report above for the synthetic dataset the percentage of fixed variables and runtime when applying a combination of partial optimality conditions as described in Section 6.

runtime for Proposition 4.8, Proposition 4.4 and Proposition 4.11 separately. It can be seen from this figure that each conditions can fix some variables. Propositions 4.8 and 4.11 are more effective empirically than Proposition 4.4. While Proposition 4.11 fixes at least as many variables as Proposition 4.8, for all n, testing Proposition 4.11 takes three orders of magnitude longer than testing Proposition 4.8.

In Appendix B (Figure B.1) we additionally show the results when applying conditions separately as a function of n.

In Figure 4a we report the speed-up when solving the linear ordering problem to optimality when applying our partial optimality conditions to the instances of the synthetic dataset for $\alpha \in \{0.65, 0.7\}$. We observe a speed-up of about 3 to 4 for $\alpha = 0.65$ and a speed-up of about 1 to 3 for $\alpha = 0.7$.

LOLIB. Next, we consider the instances of the classes IO, Spec and SGB from LOLIB (Martí et al., 2012). IO contains instances from Grötschel et al. (1984). Spec contains diverse instances from Christof & Reinelt (1996); Christof (1997); Goemans & Hall (1996). SGB contains instances from the Stanford GraphBase (Knuth, 1993) the costs of which are drawn uniformly from $[0, 25000]$.

Figure 5 shows the percentage of variables fixed by partial optimality and runtime when applying combinations of partial optimality conditions as described in Section 6, for IO (a-b), Spec (c-d) and SGB (e-f). It can be seen form this figure that between 10% and 90% of variables are fixed for instances in IO, between 0% and 70% for instances in Spec, and about 3% for instances in SGB. It can also be seen that the runtime is less than 5 seconds for all but one instance of Spec. This is explained by the fact that this instance has a larger number of elements, $|A| = 452$, than all other instances for which $11 \leq |A| \leq 163$.

In Appendix B (Figures B.2 to B.4) we additionally show these metrics when applying our conditions separately.

In Figure 4b, we report the speed-up in solving the linear ordering problem to optimality when applying partial optimality conditions. For the IO instances of the LOLIB benchmark, we observe a maximum speed-up factor of 13.18 (i.e. 1/13.18 the time) and a minimum speed-up factor of 0.77. For those Spec instances of the LOLIB benchmark that can be solved to optimality within 60 minutes, we observe a maximum speed-up factor of 33.74 and a minimum speed-up factor of 0.69.

8. Conclusion

We introduce partial optimality conditions for the APX-hard linear ordering problem, along with efficient algorithms for finding partial optimality. We examine the effectiveness and efficiency of these conditions and algorithms numerically, on two datasets. We find Propositions 4.4, 4.8 and 4.11 to be effective while other conditions contribute only marginally to the fraction of fixed variables. Prospects for future work include the exploitation of sparsity of non-zero cost coefficients, and an extension to the partial ordering problem.

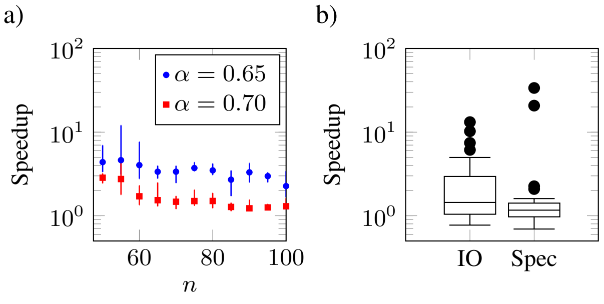

Figure 4. Above, we report the speed-up obtained by applying our our partial optimality conditions when solving the linear ordering ILP. In a) we show these for the synthetic instances as a function of the number of elements and for $\alpha \in \{0.65, 0.70\}$. In b) we show these for the problem class IO of LOLIB and for those instances of the class Spec of LOLIB that can be solved within 1h.

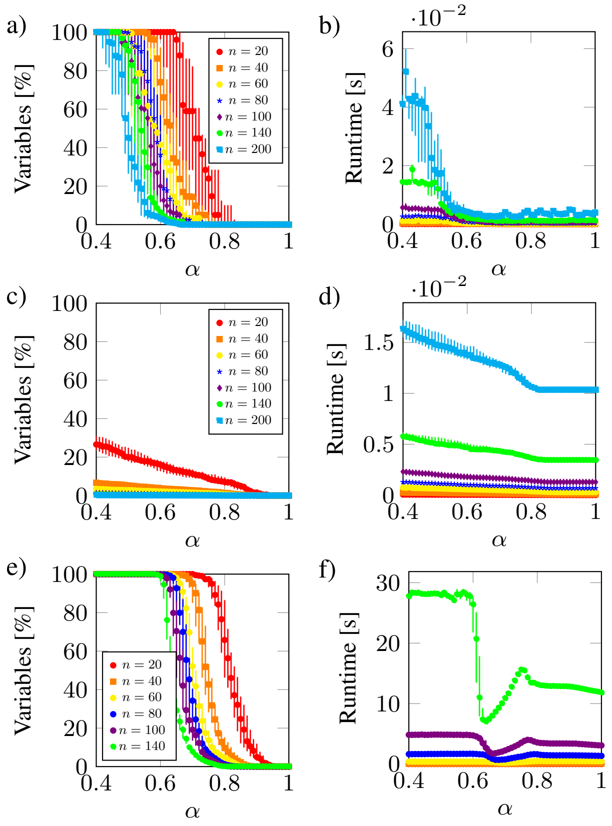

Figure 3. We report above for the synthetic dataset the percentage of fixed variables and runtime when applying each partial optimality condition separately. a)-b) shows these for Proposition 4.8, c)-d) for Proposition 4.4 and e)-f) for Proposition 4.11 Conditions that do not fix variables on the initial problem are not shown.

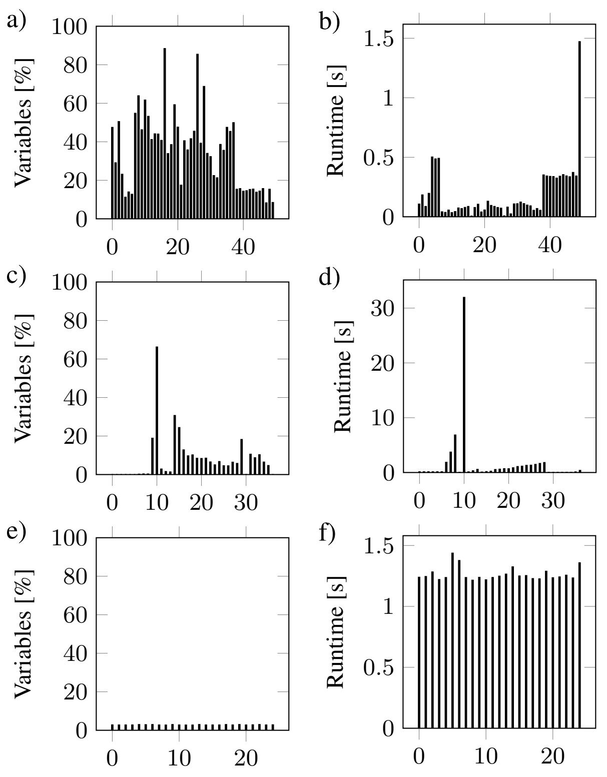

Figure 5. We report above for the instances IO (a-b), Spec (c-d) and SGB (e-f) from the LOLIB data set the percentage of fixed variables and runtime when applying a combination of partial optimality conditions as described in Section 6.

Acknowledgements

The authors acknowledge funding by the Federal Ministry of Education and Research of Germany, from grant 16LW0079K.

Impact Statement

This theoretical paper presents work whose goal is to advance the field of Machine Learning, more specifically linear ordering. As for all advances in this field, there are many potential societal consequences of our work, regarding the application of linear ordering algorithms for ranking, also some with negative impact, e.g., social ranking.

References

Adams, W. P., Lassiter, J. B., and Sherali, H. D. Persistency in 0-1 polynomial programming. *Mathematics of Operations Research*, 23(2):359–389, 1998. doi: 10.1287/moor.23.2.359.

Alush, A. and Goldberger, J. Ensemble segmentation using efficient integer linear programming. *Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1966–1977, 2012. doi: 10.1109/TPAMI.2011.280.

Billionnet, A. and Sutter, A. Persistency in quadratic 0–1 optimization. *Mathematical Programming*, 54(1):115–119, 1992. doi: 10.1007/BF01586044.

Ceberio, J., Mendiburu, A., and Lozano, J. A. The linear ordering problem revisited. *European Journal of Operational Research*, 241(3):686–696, 2015. doi: 10.1016/j.ejor.2014.09.041.

Chang, W.-C., Yu, F. X., Chang, Y.-W., Yang, Y., and Kumar, S. Pre-training tasks for embedding-based large-scale retrieval. In *ICLR*, 2020. URL https://openreview.net/forum?id=rkg-mA4FDr.

Charon, I. and Hudry, O. A branch-and-bound algorithm to solve the linear ordering problem for weighted tournaments. *Discrete Applied Mathematics*, 154(15):2097–2116, 2006. doi: 10.1016/j.dam.2005.04.020.

Christof, T. *Low-Dimensional 0/1-Polytopes and Branch-and-Cut in Combinatorial Optimization*. PhD thesis, 1997.

Christof, T. and Reinelt, G. Combinatorial optimization and small polytopes. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, 4(1):1–53, 1996. doi: 10.1007/BF02568602.

Garey, M. and Johnson, D. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Mathematical Sciences Series. Freeman, 1979.

Goemans, M. X. and Hall, L. A. The strongest facets of the acyclic subgraph polytope are unknown. In *IPCO*, 1996. doi: 10.1007/3-540-61310-2_31.

Goldberg, A. V. and Tarjan, R. E. A new approach to the maximum-flow problem. *Journal of the ACM*, 35(4): 921–940, 1988. doi: 10.1145/48014.61051.

Grötschel, M., Jünger, M., and Reinelt, G. A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32(6):1195–1220, 1984. doi: 10.1287/opre.32.6.1195.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL https://www.gurobi.com.

Hammer, P. L., Hansen, P., and Simeone, B. Roof duality, complementation and persistency in quadratic 0–1 optimization. *Mathematical Programming*, 28(2):121–155, 1984. doi: 10.1007/BF02612354.

He, Y., Gan, Q., Wipf, D., Reinert, G. D., Yan, J., and Cucuringu, M. GNNRank: Learning global rankings from pairwise comparisons via directed graph neural networks. In *ICML*, 2022. URL https://proceedings.mlr.press/v162/he22b.html.

Kaas, R. A branch and bound algorithm for the acyclic subgraph problem. *European Journal of Operational Research*, 8(4):355–362, 1981. doi: 10.1016/0377-2217(81)90005-9.

Kappes, J. H., Speth, M., Reinelt, G., and Schnörr, C. Towards efficient and exact map-inference for large scale discrete computer vision problems via combinatorial optimization. In *CVPR*, 2013. doi: 10.1109/CVPR.2013.229.

Knuth, D. E. *The Stanford GraphBase: a platform for combinatorial computing*. Association for Computing Machinery, New York, NY, USA, 1993.

Kohli, P., Shekhovtsov, A., Rother, C., Kolmogorov, V., and Torr, P. On partial optimality in multi-label mrfs. In *ICML*, 2008. doi: 10.1145/1390156.1390217.

Köppel, M., Segner, A., Wagener, M., Pensel, L., Karwath, A., and Kramer, S. Pairwise learning to rank by neural networks revisited: Reconstruction, theoretical analysis and practical performance. In *Machine Learning and Knowledge Discovery in Databases*, 2020. doi: 10.1007/978-3-030-46133-1_15.

Lange, J., Andres, B., and Swoboda, P. Combinatorial persistency criteria for multicut and max-cut. In *CVPR*, 2019. doi: 10.1109/CVPR.2019.00625.

Lange, J.-H., Karrenbauer, A., and Andres, B. Partial optimality and fast lower bounds for weighted correlation clustering. In *ICML*, 2018. URL https://proceedings.mlr.press/v80/lange18a.html.

Lim, K., Shin, N.-H., Lee, Y.-Y., and Kim, C.-S. Order learning and its application to age estimation. In *ICLR*, 2020. URL https://openreview.net/forum?id=HygsuaNFwr.

Martí, R. and Reinelt, G. *The Linear Ordering Problem. Exact and Heuristic Methods in Combinatorial Optimization*. Springer, 2011. doi: 10.1007/978-3-642-16729-4.

Martí, R., Reinelt, G., and Duarte, A. A benchmark library and a comparison of heuristic methods for the linear ordering problem. *Computational Optimization and Applications*, 51(3):1297–1317, 2012. doi: 10.1007/s10589-010-9384-9.

Menon, A., Jayasumana, S., Rawat, A. S., Kim, S., Reddi, S., and Kumar, S. In defense of dual-encoders for neural ranking. In *ICML*, 2022. URL https://proceedings.mlr.press/v162/menon22a.html.

Mishra, S. and Sikdar, K. On approximability of linear ordering and related np-optimization problems on graphs. *Discrete Applied Mathematics*, 136(2-3):249–269, 2004. doi: 10.1016/S0166-218X(03)00444-X.

Mitchell, J. E. and Borchers, B. Solving real-world linear ordering problems using a primal-dual interior point cutting plane method. *Annals of Operations Research*, 62 (1):253–276, 1996. doi: 10.1007/BF02206819.

Mitchell, J. E. and Borchers, B. Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. In Frenk, H., Roos, K., Terlaky, T., and Zhang, S. (eds.), *High Performance Optimization*, pp. 349–366. 2000. doi: 10.1007/978-1-4757-3216-0_14.

Shekhovtsov, A. *Exact and Partial Energy Minimization in Computer Vision*. PhD thesis, Center for Machine Perception, Czech Technical University, Prague, 2013.

Shekhovtsov, A. Maximum persistency in energy minimization. In *CVPR*, 2014. doi: 10.1109/CVPR.2014.152.

Shekhovtsov, A., Swoboda, P., and Savchynskyy, B. Maximum persistency via iterative relaxed inference with graphical models. In *CVPR*, 2015. doi: 10.1109/CVPR.2015.7298650.

Stein, D. Partial optimality in the linear ordering problem - code, 2024. URL https://github.com/dsteindd/partial-optimality-in-the-linear-ordering-problem.

Stein, D., Di Gregorio, S., and Andres, B. Partial optimality in cubic correlation clustering. In *ICML*, 2023. URL https://proceedings.mlr.press/v202/stein23a.html.

Szczecinski, L. and Sukheja, H. Rankability and linear ordering problem: Probabilistic insight and algorithms. *Computers & Operations Research*, 159:106347, 2023. doi: 10.1016/j.cor.2023.106347.

Tromble, R. and Eisner, J. Learning linear ordering problems for better translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2009. doi: 10.3115/1699571.1699644.

Zhang, H., Gong, Y., Shen, Y., Lv, J., Duan, N., and Chen, W. Adversarial retriever-ranker for dense text retrieval. In *ICLR*, 2022. URL https://openreview.net/forum?id=MR7XubKUFB.

A. Proofs

Proof of Lemma 4.2. By plugging in we obtain

$$
\begin{aligned}
&\varphi_c(x') - \varphi_c(x) \\
=& \sum_{a'b' \in P_A} c_{a'b'} \left(x'_{a'b'} - x_{a'b'}\right) \\
=& \sum_{\substack{a'b' \in P_A \\ \{a',b'\} \cap \{a,b\} \neq \emptyset}} c_{a'b'} \left(x'_{a'b'} - x_{a'b'}\right) \\
=& c_{ab}\left(x'_{ab} - x_{ab}\right) + c_{ba}\left(x'_{ba} - x_{ba}\right) \\
&+ \sum_{b' \in A \setminus \{a,b\}} c_{ab'}\left(x'_{ab'} - x_{ab'}\right) + \sum_{a' \in A \setminus \{a,b\}} c_{a'a}\left(x'_{a'a} - x_{a'a}\right) \\
&+ \sum_{b' \in A \setminus \{a,b\}} c_{bb'}\left(x'_{bb'} - x_{bb'}\right) + \sum_{a' \in A \setminus \{a,b\}} c_{a'b}\left(x'_{a'b} - x_{a'b}\right) \\
=& c_{ab}\left(1 - 2x_{ab}\right) + c_{ba}\left(1 - 2x_{ba}\right) \\
&\quad + \sum_{d \in A \setminus \{a,b\}} c_{ad}\left(x_{bd} - x_{ad}\right) + \sum_{d \in A \setminus \{a,b\}} c_{da}\left(x_{db} - x_{da}\right) \\
&\quad + \sum_{d \in A \setminus \{a,b\}} c_{bd}\left(x_{ad} - x_{bd}\right) + \sum_{d \in A \setminus \{a,b\}} c_{db}\left(x_{da} - x_{db}\right) \\
=& \left(c_{ab} - c_{ba}\right)\left(1 - 2x_{ab}\right) \\
&\quad + \sum_{d \in A \setminus \{a,b\}} \left(x_{da} + x_{bd} - 1\right)\left(c_{ad} - c_{da} - c_{bd} + c_{db}\right) \\
=& c^{\Delta}_{ab}\left(1 - 2x_{ab}\right) + \sum_{d \in A \setminus \{a,b\}} \left(x_{da} + x_{bd} - 1\right)\left(c^{\Delta}_{ad} + c^{\Delta}_{db}\right)
\end{aligned}
$$

This concludes the proof. □

Proof of Lemma 4.3. First, let $g_1 \colon \{x \in X_A \mid x_{ab} = 0\} \to \{0,1\}^{A \setminus \{a,b\}}$ be such that $g_1(x)_d = x_{ad} + x_{db}$, $\forall d \in A \setminus \{a,b\}$. On the one hand, we have $g_1(x)_d \in \{0,1\}$. On the other hand, g_1 is surjective. To see the latter, let $y \in \{0,1\}^{A \setminus \{a,b\}}$. Then, let x' be a feasible vector such that $\neg(b <_{x'} d <_{x'} a)$, $\forall d \in y^{-1}(1)$ and $b <_{x'} d <_{x'} a$, $\forall d \in y^{-1}(0)$. Then, we have that $g_1(x') = y$. Therefore,

$$
\begin{aligned}
&\max_{\substack{x \in X_A \\ x_{ab}=0}} \sum_{d \in A \setminus \{a,b\}} \left(x_{bd} + x_{da} - 1\right) e_d \\
&\quad = \max_{y \in \{0,1\}^{A \setminus \{a,b\}}} \sum_{d \in A \setminus \{a,b\}} e_d(1 - y_d) = \sum_{d \in A \setminus \{a,b\}} e^{+}_d
\end{aligned}
$$

This concludes the proof. □

Proof of Corollary 4.5. By Proposition 4.4 there is an optimal solution x^* to the linear ordering problem such that $x^*_{ba} = 1$ since (5) is fulfilled. Moreover, we have that $x^*_{ab} = 1 - x^*_{ba} = 0$. □

Proof of Lemma 4.7. By plugging in we obtain

$$
\begin{aligned}
&\varphi_c(x') - \varphi_c(x) \\
&= \sum_{ab \in P_R} c_{ab}\left(x'_{ab} - x_{ab}\right) + \sum_{ab \in P_{A \setminus R}} c_{ab}\left(x'_{ab} - x_{ab}\right) \\
&+ \sum_{a \in R} \sum_{b \in A \setminus R} c_{ab}\left(x'_{ab} - x_{ab}\right) + \sum_{a \in A \setminus R} \sum_{b \in R} c_{ab}\left(x'_{ab} - x_{ab}\right) \\
&= \sum_{a \in R} \sum_{b \in A \setminus R} c_{ab}\left(1 - x_{ab}\right) + \sum_{b \in A \setminus R} \sum_{a \in R} c_{ba}\left(0 - x_{ba}\right) \\
&= \sum_{a \in R} \sum_{b \in A \setminus R} c^{\Delta}_{ab} x_{ba} \ .
\end{aligned}
$$

This concludes the proof. □

Proof of Lemma 4.10. We have by Lemma 4.7 that

$$
\begin{aligned}
&\varphi_c(x') - \varphi_c(x) \\
&= \sum_{a' \in R} \sum_{b' \in A \setminus R} c^{\Delta}_{a'b'} x_{b'a'} \\
&= c^{\Delta}_{ab} x_{ba} + \sum_{a' \in R} \sum_{\substack{b' \in A \setminus R \\ a'b' \neq ab}} c^{\Delta}_{a'b'} x_{b'a'} \\
&= c^{\Delta}_{ab} + \sum_{a' \in R} \sum_{\substack{b' \in A \setminus R \\ a'b' \neq ab}} c^{\Delta}_{a'b'} x_{b'a'} \\
&\leq c^{\Delta}_{ab} + \sum_{a' \in R} \sum_{\substack{b' \in A \setminus R \\ a'b' \neq ab}} c^{\Delta,+}_{a'b'} \\
&= c^{\Delta}_{ab} - c^{\Delta,+}_{ab} + \sum_{a' \in R} \sum_{b' \in A \setminus R} c^{\Delta,+}_{a'b'} \\
&= -\, c^{\Delta,-}_{ab} + \sum_{a' \in R} \sum_{b' \in A \setminus R} c^{\Delta,+}_{a'b'}
\end{aligned}
$$

This concludes the proof. □

Proof of Corollary 4.12. By Proposition 4.11 there is an optimal solution x^* to the linear ordering problem such that $x^*_{ba} = 1$ since (7) is fulfilled. Moreover, we have that $x^*_{ab} = 1 - x^*_{ba} = 0$. □

Proof of Lemma 4.14. We have that

$$
\begin{aligned}
&\varphi_c(x') - \varphi_c(x) \\
&= \sum_{b' \in A \setminus \{a\}} c_{ab'}\left(1 - x_{ab'}\right) + \sum_{a' \in A \setminus \{a\}} c_{a'a}(0 - x_{a'a}) \\
&\quad + \sum_{a' <_x a} \sum_{b' >_x a} c_{a'b'}(0 - x_{a'b'}) \\
&\quad + \sum_{a' >_x a} \sum_{b' <_x a} c_{a'b'}(1 - x_{a'b'}) \\
&= \sum_{d \in A \setminus \{a\}} c^{\Delta}_{ad} x_{da} + \sum_{a' <_x a} \sum_{b' >_x a} c^{\Delta}_{b'a'} x_{a'b'} \\
&= \sum_{d <_x a} c^{\Delta}_{ad} + \sum_{a' <_x a} \sum_{b' >_x a} c^{\Delta}_{b'a'}
\end{aligned}
$$

$$= \sum_{d <_x a} c^\Delta_{ad} + \sum_{a' <_x a} \sum_{b' >_x a} c^\Delta_{b'a'} + \underbrace{\sum_{a' <_x a} \sum_{\substack{b' <_x a \\ b' \neq a'}} c^\Delta_{b'a'}}_{=0}$$
$$= \sum_{d <_x a} c^\Delta_{ad} + \sum_{d <_x a} \sum_{d' \in A \setminus \{a,d\}} c^\Delta_{d'd}$$
$$= \sum_{d <_x a} \sum_{d' \in A \setminus \{d\}} c^\Delta_{d'd}$$

This concludes the proof. □

Proof of Corollary 4.16. By Proposition 4.15 and (10), there is an optimal solution $\hat{x}^*$ to the linear ordering problem with respect to A and $-c$ such that $\hat{x}^*_{ab} = 1, \forall b \in A \setminus \{a\}$. Then, $x^* = 1 - \hat{x}^*$ is an optimal solution to the linear ordering problem with respect to A and c such hat $x^*_{ab} = 0$, $\forall b \in A \setminus \{a\}$. □

Proof of Lemma 4.18. If $x_{ab} = 1$ we have that

$$\varphi_c(x') - \varphi_c(x)$$
$$= \sum_{d <_x \min_x\{a,b\}} c^\Delta_{ad} + \sum_{\substack{d <_x \max_x\{a,b\} \\ d \notin \{a,b\}}} c^\Delta_{bd}$$
$$+ \sum_{d >_x \max_x\{a,b\}} \sum_{\substack{d' <_x \max_x\{a,b\} \\ d' \notin \{a,b\}}} c^\Delta_{dd'}$$
$$= \sum_{d <_x \min_x\{a,b\}} (c^\Delta_{ad} + c^\Delta_{bd}) + \sum_{\substack{d >_x \min_x\{a,b\} \\ d <_x \max_x\{a,b\}}} c^\Delta_{bd}$$
$$+ \sum_{d >_x \max_x\{a,b\}} \sum_{d' \in A \setminus \{a,b,d\}} c^\Delta_{dd'}$$

Analogously, for x such that $x_{ab} = 0$, we arrive at

$$\varphi_c(x') - \varphi_c(x) =$$
$$= \sum_{d <_x \min_x\{a,b\}} (c^\Delta_{ad} + c^\Delta_{bd}) + \sum_{\substack{d > \min_x\{a,b\} \\ d < \max_x\{a,b\}}} c^\Delta_{ad}$$
$$+ \sum_{d >_x \max_x\{a,b\}} \sum_{d' \in A \setminus \{a,b,d\}} c^\Delta_{dd'}$$

This concludes the proof. □

Proof of Corollary 4.20. If (13) and (14) hold, then by Proposition 4.19 there is an optimal solution $\hat{x}^*$ to the linear ordering problem with respect to A and $-c$ such that $\hat{x}^*_{ad} = \hat{x}^*_{bd} = 1, \forall d \in A \setminus \{a,b\}$. Then, $x^* = 1 - \hat{x}^*$ is an optimal solution to the linear ordering problem with respect to A and c such that $x^*_{ad} = x^*_{bd} = 0, \forall d \in A \setminus \{a,b\}$. □

Proof of Lemma 6.1. Let $G = (A', P) \in \mathcal{G}_A$ be any partial 1-assignment. We prove, that there is at least one extension $x \in X_A$ such that $\forall ab \in P \colon x_{ab} = 1$. Let us define $G' = (A, P)$. Because G and thus G' is cycle-free there exists a topological order on G'. Let $\xi \colon \{0, 1, ..., |A| - 1\} \to A$ be any such topological order on G'. Moreover, let $x \in \{0,1\}^{P_A}$ be defined such that

$$\forall ab \in P_A \colon x_{ab} = 1 \Leftrightarrow \xi^{-1}(a) < \xi^{-1}(b) \ .$$

First, we show that $x \in X_A$. Suppose, there is a transitivity constraint (2) which is violated, i.e., for $a \in A$, $b \in A \setminus \{a\}$ and $c \in A \setminus \{a,b\}$ we have that

$$\begin{aligned} & x_{ab} + x_{bc} - x_{ac} > 1 \\ \Leftrightarrow\ & x_{ab} = 1 \wedge x_{bc} = 1 \wedge x_{ac} = 0 \\ \Leftrightarrow\ & \left(\xi^{-1}(a) < \xi^{-1}(b)\right) \\ & \wedge \left(\xi^{-1}(b) < \xi^{-1}(c)\right) \\ & \wedge \left(\xi^{-1}(a) > \xi^{-1}(c)\right) \\ \Rightarrow\ & \left(\xi^{-1}(a) < \xi^{-1}(c)\right) \wedge \left(\xi^{-1}(a) > \xi^{-1}(c)\right) \ . \end{aligned}$$

This is a contradiction. Therefore, all transitivity constraints must be fulfilled. Moreover, we have $x_{ab} + x_{ba} = 1$ for all $ab \in P_A$ by definition of $x \in \{0,1\}^A$. In total we have that $x \in X_A$. Second, for any $ab \in P$ we have that $\xi^{-1}(a) < \xi^{-1}(b)$ by definition of a topological order and hence $x_{ab} = 1$ by definition of x. □

Lemma A.1. *Let $A \neq \emptyset$, $(A', P) \in \mathcal{G}_A$ is a path and $d \in A'$ be the unique element, which has no predecessors in (A', P). Then, for any $x \in X_A$ there exists a composition σ of functions $\{\tau^1_{ab}\}_{ab \in P}$, such that $\sigma(x)_{da} = 1$ for all $a \in A' \setminus \{d\}$.*

Proof of Lemma A.1. We prove this by induction over $|A'|$. If $|A'| = 2$, i.e., $A' = \{a,b\}$ and $P = \{ab\}$, then τ^1_{ab} is such that the statement holds. Now, let $|A'| = n + 1$ and the statement be true for all $|A'| \leq n$. Let $x \in X_A$, $a \in A'$ such that $x_{ab} = 1$ for all $b \in A' \setminus \{a\}$. We denote by $\mathcal{P}_G(a,b)$ the set of paths in (A', P) from a to b. Let $A'' = \{b \in A' \mid \mathcal{P}_G(b,a) \neq \emptyset\}$ and $P' = P \cap (A'' \times A'')$. Then $(A'', P') \in \mathcal{G}_A$ is a path as well and we have that $d \in A''$.

If $A'' \neq A'$, then $|A''| \leq n$. By induction hypothesis, there is a composition σ of functions $\{\tau^1_{ab}\}_{ab \in P'} \subset \{\tau^1_{ab}\}_{ab \in P}$ such that $\sigma(x)_{db} = 1$ for all $b \in A''$. Moreover, we can describe the composition σ by a bijection $\xi \colon A \to A$ with $\xi|_{A \setminus A''} = \mathrm{id}|_{A \setminus A''}$ such that $x'_{a'b'} = x_{\xi^{-1}(a')\xi^{-1}(b')}$ for all $a'b' \in S_A$. Thus $\xi(a) = d$, and $x'_{db} = x_{\xi^{-1}(d)\xi^{-1}(b)} = x_{ab} = 1$ for all $b \in A' \setminus A''$.

If $A'' = A'$, then a has no successors in (A', P). Let $ba \in P$ be the single in-edge of a in (A'', P'). Then, we have that $x_{ba} = 0$ by definition of $a \in A'$. We first apply τ^1_{ba}, i.e.,

$x' = \tau^1_{ba}(x)$. Subsequently, we have that $x'_{bc} = x_{ac} = 1$ for all $c \in A'' \setminus \{a, b\}$ and, moreover, $x'_{ba} = x_{ab} = 1$. By the same argument as before, let $P'' = P' \setminus \{ba\}$, $A''' = A' \setminus \{a\}$ and $(A''', P'') \in \mathcal{G}_A$ is a path. Thus, we can find by the induction hypothesis a composition σ' of functions $\{\tau^1_{ab}\}_{ab \in P''} \subset \{\tau^1_{ab}\}_{ab \in P}$ such that $x''_{dc} = \sigma'(x')_{dc} = 1$ for all $c \in A' \setminus \{d\}$. $\square$

Proof of Lemma 6.2. We prove this statement by induction over $|A'|$. For $A' = \{a, b\}$ and $P = \{ab\}$ the statement is fulfilled by τ^1_{ab}. Now, let $|A'| = n + 1$ and the statement be true for all $|A'| \le n$. Let $x \in X_A$, $a \in A'$ such that $x_{ab} = 1$ for all $b \in A' \setminus \{a\}$. Moreover, let $b \in A'$ be any element in A' with no predecessors in (A', P) and $\mathcal{P}_G(b, a) \neq \emptyset$. Let $(A'', P') \in \mathcal{P}_G(b, a)$ be any such path. By Lemma A.1 there is a composition σ' of functions $\{\tau^1_{ab}\}_{ab \in P'} \subset \{\tau^1_{ab}\}_{ab \in P}$ such that $x' = \sigma'(x)_{bc} = 1$ for all $c \in A'' \setminus \{b\}$ and, moreover, for all $c \in A' \setminus \{b\}$. Now, we can apply the induction hypothesis on $A''' = A' \setminus \{b\}$ and $P'' = P \setminus \{a'b' \in P \mid a' = b\}$, which concludes the proof. $\square$

Proof of Lemma 6.4. Let $x \in X_{A'P}$ and $x' = \sigma_R(x)$. For every $ab \in E$:

$$\begin{aligned}
& x'_{ab} = 1 \\
\Leftrightarrow\ & \left(x_{ab} = 1 \wedge ab \in P_R \cup P_{A \setminus R}\right) \vee (a \in R \wedge b \in A \setminus R) \\
\Leftrightarrow\ & ab \in P_R \cup P_{A \setminus R} \vee (a \in R \wedge b \in A \setminus R) \\
\Leftrightarrow\ & \neg\, (a \notin R \wedge b \in R) \\
\Leftrightarrow\ & a \in R \vee b \notin R
\end{aligned}$$

Therefore, we have that $x' \in X_{A'P}$. $\square$

B. Additional Experiments

In the following figures, we report the percentage of fixed variables, as well as corresponding runtimes, for applying partial optimality conditions separately. Conditions not shown in the respective figure do not fix any variables on the initial instances.

Synthetic Dataset. Figure B.1 shows these after applying Proposition 4.8, Proposition 4.4 and Proposition 4.11 separately and for varying instance size n.

LOLIB Dataset. Figure B.2 shows these for applying Proposition 4.8, Proposition 4.4, and Proposition 4.11 to instances of IO.

Figure B.3 shows these for applying Proposition 4.15 and Corollary 4.16 to instances of SGB.

Figure B.4 shows these for applying Proposition 4.8, Proposition 4.15, Proposition 4.4, Corollary 4.16 and Proposition 4.11 to instances of Spec.

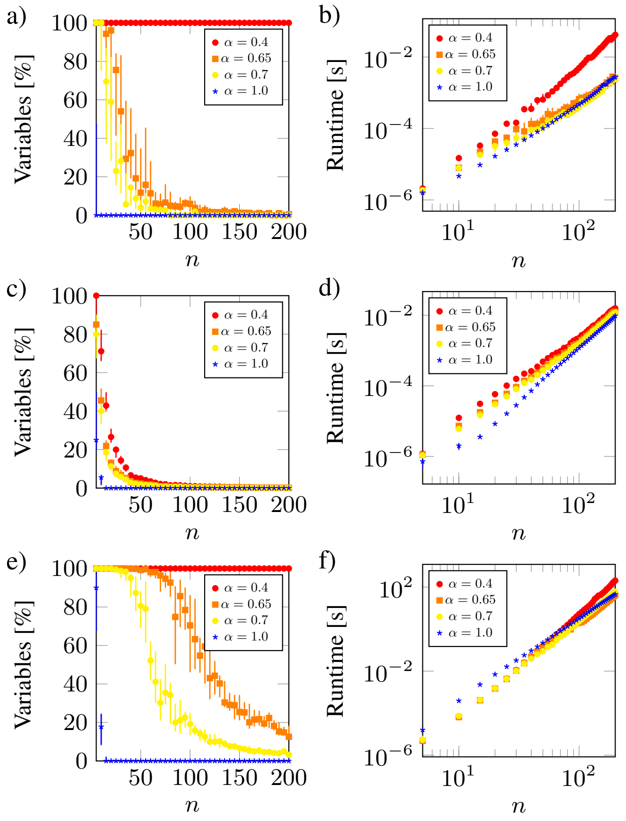

Figure B.1. We report above for the synthetic dataset the percentage of fixed variables and runtime as a function of the instance size when applying each partial optimality condition separately. a-b) shows these for Proposition 4.8, c-d) for Proposition 4.4 and e) -f) for Proposition 4.11. Conditions that do not fix variables are not shown.

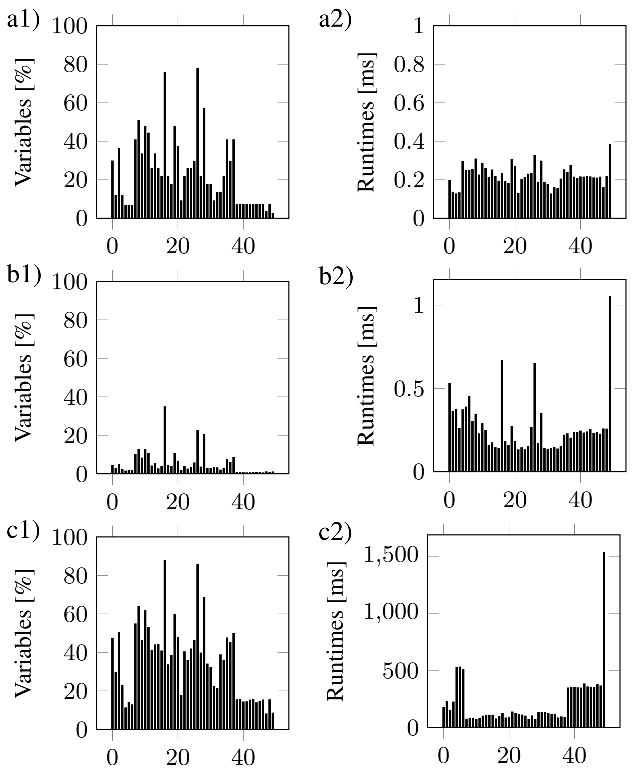

Figure B.2. We report above the percentage of fixed variables after applying each condition separately, as well as the corresponding runtimes, for the problem class IO of LOLIB. a1)-a2) show these for Proposition 4.8, b1)-b2) show these for Proposition 4.4 and c1)-c2) show these for Proposition 4.11. Conditions, which yield zero fixed variables are not shown.

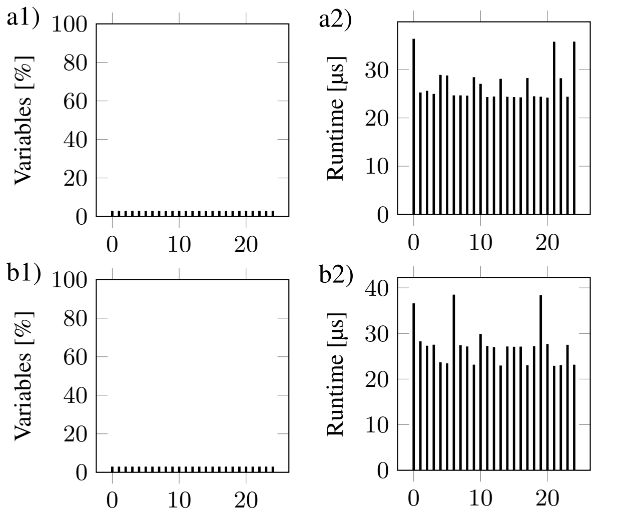

Figure B.3. We report above the percentage of fixed variables after applying each condition separately, as well as the corresponding runtimes, for the problem class SGB of LOLIB. a1)-a2) show these for Proposition 4.15 and b1)-b2) for Corollary 4.16. Conditions, which fix zero variables are not shown.

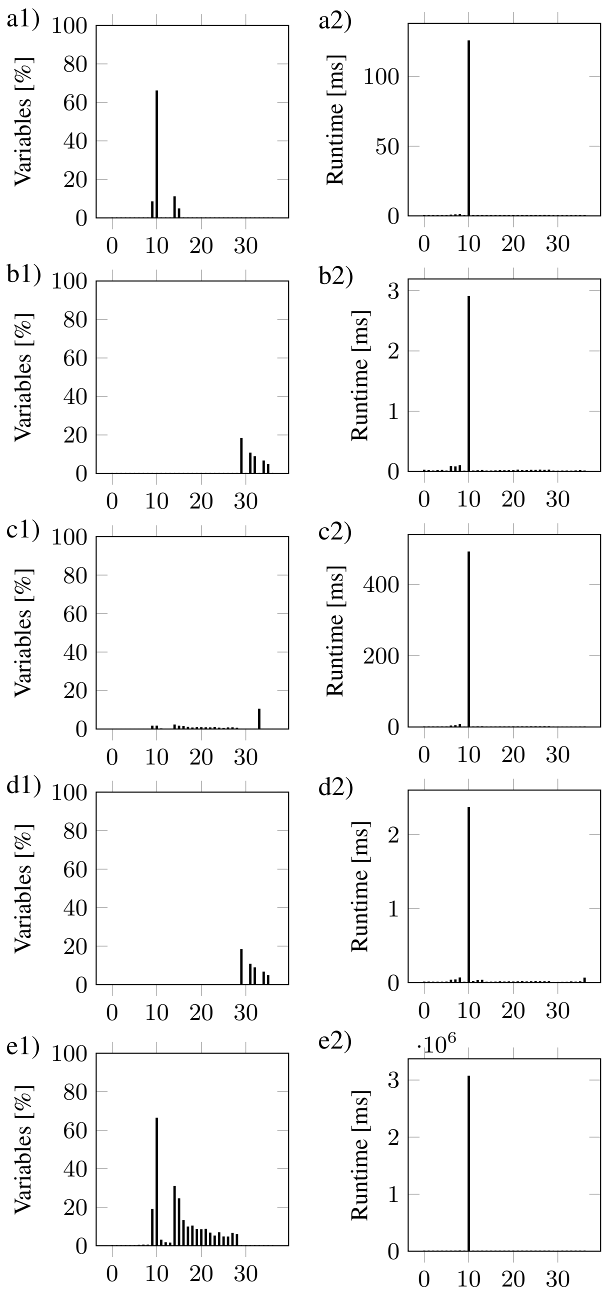

Figure B.4. We report above the percentage of fixed variables after applying each condition separately, as well as the corresponding runtimes, for Spec from LOLIB. a1)-a2) show these for Proposition 4.8, b1)-b2) for Proposition 4.15, c1)-c2) show these for Proposition 4.4, d1)-d2) for Corollary 4.16 and e1)-e2) for Proposition 4.11. Conditions, which fix variables are not shown.

C. LOLIB Data

j	IO	$\lvert A\rvert$	Spec	$\lvert A\rvert$	SGB	$\lvert A\rvert$
0	N-be75eec	50	N-EX1	50	N-sgb75.01	75
1	N-be75np	50	N-EX2	50	N-sgb75.02	75
2	N-be75oi	50	N-EX3	50	N-sgb75.03	75
3	N-be75tot	50	N-EX4	50	N-sgb75.04	75
4	N-stabu70	60	N-EX5	50	N-sgb75.05	75
5	N-stabu74	60	N-EX6	50	N-sgb75.06	75
6	N-stabu75	60	N-atp111	111	N-sgb75.07	75
7	N-t59b11xx	44	N-atp134	134	N-sgb75.08	75
8	N-t59d11xx	44	N-atp163	163	N-sgb75.09	75
9	N-t59f11xx	44	N-atp24	24	N-sgb75.10	75
10	N-t59i11xx	44	N-atp452	452	N-sgb75.11	75
11	N-t59n11xx	44	N-atp48	48	N-sgb75.12	75
12	N-t65b11xx	44	N-atp66	66	N-sgb75.13	75
13	N-t65d11xx	44	N-atp76	76	N-sgb75.14	75
14	N-t65f11xx	44	N-econ36	36	N-sgb75.15	75
15	N-t65i11xx	44	N-econ43	43	N-sgb75.16	75
16	N-t65l11xx	44	N-econ47	47	N-sgb75.17	75
17	N-t65n11xx	44	N-econ58	58	N-sgb75.18	75
18	N-t65w11xx	44	N-econ59	59	N-sgb75.19	75
19	N-t69r11xx	44	N-econ61	61	N-sgb75.20	75
20	N-t70b11xx	44	N-econ62	62	N-sgb75.21	75
21	N-t70d11xx	44	N-econ64	64	N-sgb75.22	75
22	N-t70d11xxb	44	N-econ67	67	N-sgb75.23	75
23	N-t70f11xx	44	N-econ68	68	N-sgb75.24	75
24	N-t70i11xx	44	N-econ71	71	N-sgb75.25	75
25	N-t70k11xx	44	N-econ72	72		
26	N-t70l11xx	44	N-econ73	73		
27	N-t70n11xx	44	N-econ76	76		
28	N-t70u11xx	44	N-econ77	77		
29	N-t70w11xx	44	N-pal11	11		
30	N-t70x11xx	44	N-pal13	13		
31	N-t74d11xx	44	N-pal19	19		
32	N-t75d11xx	44	N-pal23	23		
33	N-t75e11xx	44	N-pal27	27		
34	N-t75i11xx	44	N-pal31	31		
35	N-t75k11xx	44	N-pal43	43		
36	N-t75n11xx	44	N-pal55	55		
37	N-t75u11xx	44				
38	N-tiw56n54	56				
39	N-tiw56n58	56				
40	N-tiw56n62	56				
41	N-tiw56n66	56				
42	N-tiw56n67	56				
43	N-tiw56n72	56				
44	N-tiw56r54	56				
45	N-tiw56r58	56				
46	N-tiw56r66	56				
47	N-tiw56r67	56				
48	N-tiw56r72	56				
49	N-usa79	79				

Table C.1. Above, we report for each index j in Figures 5 and B.2 to B.4 the corresponding LOLIB problem and instance size $|A|$ for IO (left), Spec (middle) and SGB (right).