Structural Properties and Tractability Results for Linear Synteny[*]

David Liben-Nowell
Department of Computer Science
Carleton College
Northfield, MN 55057 USA
dlibenno@carleton.edu

Jon Kleinberg
Department of Computer Science
Cornell University
Ithaca, NY 14853 USA
kleinber@cs.cornell.edu

Abstract

The syntenic dista nce between two species is the minimum number of fusions, fissions, and translocations required to transform one genome into the other. The linear syntenic distance, a restricted form of this model, has been shown to be close to the syntenic distance. Both models are computationally difficult to compute and have resisted efficient approximation algorithms with non-trivial performance guarantees. In this paper, we prove that many useful properties of syntenic distance carry over to linear syntenic distance. We also give a reduction from the general linear synteny problem to the question of whether a given instance can be solved using the maximum possible number of translocations. Our main contribution is an algorithm exactly computing linear syntenic distance in nested instances of the problem. This is the first polynomial time algorithm exactly solving linear synteny for a non-trivial class of instances. It is based on a novel connection between the syntenic distance and a scheduling problem that has been studied in the operations research literature.

1 Introduction

Computational models of the evolutionary distance between species have recently captured the attention of the theoretical computer science community. These models are often based on genome-level mutations that displace large pieces of genetic material, affecting the order of genes within chromosomes. The distance between two genomes $\mathcal{G}_1$ and $\mathcal{G}_2$ is then defined as the minimum number of these mutations required to transform $\mathcal{G}_1$ into $\mathcal{G}_2$. Examples of such models include the *reversal distance* [2, 4, 5, 6, 7, 8, 16, 17], where the transformation of interest is the inversion of a segment of a chromosome; the *transposition distance* [3, 12], in which a segment of a chromosome can be extracted and reinserted at some other location in the chromosome; and the combined *reversal and transposition distance* [11, 15].

Ferretti, Nadeau, and Sankoff [13] propose a somewhat different sort of measure of genetic distance, known as the *syntenic distance*. This model abstracts away from the order of the genes within chromosomes, and considers each chromosome as an unordered set of genes. The relevant transformations are *fusions*, in which two chromosomes join into one, *fissions*, in which one chromosome splits into two, and *translocations*, in which two chromosomes exchange subsets of their genes.

[*] Appears in *Journal of Discrete Algorithms*, Volume 2, Number 2, June 2004, pp. 207–228. A preliminary version of this paper appears in *Proceedings of the 11th Annual Conference on Combinatorial Pattern Matching (CPM '00)*. Minor changes have been made in this document; the last update was on 4 August 2005. Comments are welcome.

In practice, the order of genes within chromosomes is often unknown, and this model allows the computation of the distance between species regardless. Additional justification follows from the observation that interchromosomal evolutionary events may occur with relative rarity with respect to intrachromosomal events. (For some discussion of this and related models, see [10, 20].)

Work on the syntenic distance was initiated by Ferretti et al. [13], who give a heuristic for approximating this quantity, as well as empirical evidence of its success. Subsequent research has yielded a simple 2-approximation and a proof of NP-completeness [9], and has established a number of structural properties of the model [18].

The *linear synteny* problem is a restricted form of the general synteny problem that was defined by DasGupta et al. [9]. In attempting to determine the distance from genome $\mathcal{G}_1$ to genome $\mathcal{G}_2$, we consider only transformation sequences that take on the following form:

- First, the chromosomes of $\mathcal{G}_1$ are ordered and merged together in succession, as follows. The ith transformation is a fusion unless all of the genes contained in some chromosome C of $\mathcal{G}_2$ have already been merged; in this case, transformation i is a translocation that produces C and a chromosome containing all the other remaining merged genes.

- Then, after all the chromosomes of $\mathcal{G}_1$ have been merged, each succeeding transformation is a fission producing some chromosome C of $\mathcal{G}_2$, where all of the genes of C remain in the giant merged chromosome.

While linear syntenic distance is unmotivated biologically, its relation to the syntenic distance makes it worthy of study. DasGupta et al. prove that the linear distance between two species is not much larger than the unconstrained distance: if d is the syntenic distance for any instance, then the linear syntenic distance is at most $d + \log_{4/3}(d)$.

Our Results: Structural Properties of Linear Synteny. Although the additional constraints on the linear version of the problem seem to make it simpler to reason about, little work has made use of this model—possibly because many of the useful properties known for the unconstrained model were not known to carry over to the linear case.

In this paper, we prove a number of structural results for linear syntenic distance. Most are previously proven properties of the general model [9, 18] that we now extend to the linear case. We prove a *monotonicity property* for instances of the linear synteny problem, showing a natural ordering on problem instances. We give a method of *canonicalization* for linear move sequences: given an arbitrary move sequence σ solving an instance, we produce another sequence σ' such that (1) σ' is no longer than σ, (2) σ' solves the instance, and (3) in σ', all fusions precede all translocations. We also prove that *duality* holds in the linear model, i.e., that the measure is indeed symmetric. These properties, coupled with the additional structure imposed by the problem definition itself, make the linear problem much easier to consider.

Our Results: Solving Nested Linear Synteny. One of the most prominent features that the various measures of genomic distance share is that no efficient algorithms are known for any of them, and most have been shown to be NP-complete; see the hardness results for sorting by reversals in [6, 7] and for the syntenic distance in [9]. (A notable exception to this hardness is the version of the reversal distance when genes are *oriented*, in which the distance can be computed efficiently [16].) Much of the previous work on these distances has focused on approximation algorithms with good performance guarantees: this approach has yielded performance guarantees of 11/8 for the reversal distance [5], 3/2 for the transposition distance [3], and 2 for the syntenic distance [9, 13, 18].

In this paper, we present the first polynomial-time algorithm to solve a non-trivial class of instances of the linear syntenic distance problem. For two chromosomes C_i and C_j in $\mathcal{G}_1$, let S_i and S_j be the set of chromosomes in $\mathcal{G}_2$ from which C_i and C_j draw their genes. Call an instance *nested* if, for all chromosomes C_i and C_j in $\mathcal{G}_1$, either (1) S_i and S_j are disjoint, (2) $S_i \subseteq S_j$, or (3) $S_j \subseteq S_i$.

We give a polynomial-time algorithm that solves nested instances of the linear synteny problem, by developing a connection between the linear syntenic distance and a scheduling problem that has been studied in the operations research literature. Specifically, the scheduling problem to which we relate syntenic distance is the following. (Precise definitions will be given in Section 6.) Imagine a company that must undertake a sequence of tasks, each with an associated profit or loss. Moreover, there is a partial order specifying dependencies among the tasks. The company's goal is to perform the tasks in the order that minimizes its maximum cumulative "debt" at any point in time. When these dependencies have a *series-parallel* structure, polynomial-time solutions were given independently by Abdel-Wahab and Kameda [1] and Monma and Sidney [19].

It is intuitively natural that genome rearrangement problems should have a connection to scheduling; in seeking an optimal rearrangement sequence, one rapidly encounters the combinatorial problem of "sequencing" certain rearrangement events as parsimoniously as possible. Our polynomial-time result provides one of the first true formalizations of this intuition, and we hope that it suggests other applications in this area for the voluminous body of work on scheduling.

2 Notational Preliminaries

Under the syntenic distance model, a *chromosome* is a subset of a set of *genes*, and a *genome* is an unordered collection of chromosomes.[1] A genome can be transformed by any of the following operations, for S, T, U, and V non-empty sets of genes:

- a *fusion* $(S, T) \longrightarrow U$, where $U = S \cup T$;
- a *fission* $S \longrightarrow (T, U)$, where $T \cup U = S$; and
- a *translocation* $(S, T) \longrightarrow (U, V)$, where $U \cup V = S \cup T$.

We sometimes refer to these operations as *transformations* or *moves*.

The *syntenic distance* $D(\mathcal{G}_1, \mathcal{G}_2)$ between genomes $\mathcal{G}_1$ and $\mathcal{G}_2$ is the minimum number of fusions, fissions, and translocations required to transform $\mathcal{G}_1$ into $\mathcal{G}_2$, disregarding any genes that appear in only one of the two genomes.

The *compact representation* of the syntenic distance problem [9, 13] makes the goal of each instance uniform and thus eases reasoning about move sequences. Consider an instance in which we are attempting to transform genome $\mathcal{A} = A_1, A_2, \ldots, A_k$ into genome $\mathcal{B} = B_1, B_2, \ldots, B_n$. In the compact representation, we relabel each gene ℓ contained in a chromosome of $\mathcal{A}$ by the indices of the chromosomes of $\mathcal{B}$ in which ℓ appears. Formally, we replace each chromosome A_i in $\mathcal{A}$ with

[1] We limit the genomes that we consider to those with *disjoint* chromosomes—i.e., without gene dupliation. For economy of notation, however, we allow non-disjoint chromosomes in the definition. (The compact representation defined below requires non-disjointness.)

$\bigcup_{\ell \in A_i} \{j : \ell \in B_j\}$, and attempt to transform these sets into the collection $\mathcal{G}_n = \{1\}, \{2\}, \ldots, \{n\}$. As an example of the compact representation (given in [13]), consider the instance

$$\begin{array}{rclll} \mathcal{A} & = & \{x, y\}, & \text{(Chromosome 1)} \\ & & \{p, q, r\}, & \text{(Chromosome 2)} \\ & & \{a, b, c\} & \text{(Chromosome 3)} \end{array} \qquad \begin{array}{rcll} \mathcal{B} & = & \{p, q, x\}, & \text{(Chromosome 1)} \\ & & \{a, b, r, y, z\} & \text{(Chromosome 2).} \end{array}$$

The compact representation of $\mathcal{A}$ with respect to $\mathcal{B}$ is $\{1, 2\}, \{1, 2\}, \{2\}$ and the compact representation of $\mathcal{B}$ with respect to $\mathcal{A}$ is $\{1, 2\}, \{1, 2, 3\}$.

Consider an instance $\mathcal{S}$ given in this compact representation, where $\mathcal{S} = S_1, S_2, \ldots, S_k$ and $\bigcup_i S_i = \{1, 2, \ldots, n\}$. We refer to k as the number of *sets* in $\mathcal{S}$, and n as the number of *elements*; the *syntenic distance* of $\mathcal{S}$ is given by $D(\mathcal{S}) = D(\mathcal{S}, \mathcal{G}_n)$. If $\mathcal{S}$ is the compact representation of $\mathcal{A}$ with respect to $\mathcal{B}$, then $D(\mathcal{S}) = D(\mathcal{A}, \mathcal{B})$ [9, 13]. In the remainder of this paper, we will consider only instances in the compact representation.

We will say that two sets S_i and S_j are *connected* if $S_i \cap S_j \neq \emptyset$, and that the sets are in the same *component*.

The *dual* of an instance $\mathcal{S} = S_1, S_2, \ldots, S_k$ is the instance $\mathsf{dual}(\mathcal{S}) = S'_1, S'_2, \ldots, S'_n$, where $j \in S'_i$ if and only if $i \in S_j$. (For an instance $\mathcal{S}$ that is the compact representation of a genome $\mathcal{A}$ with respect to a genome $\mathcal{B}$, the instance $\mathsf{dual}(\mathcal{S})$ is the compact representation of $\mathcal{B}$ with respect to $\mathcal{A}$.) DasGupta et al. [9] prove that $D(\mathcal{S}) = D(\mathsf{dual}(\mathcal{S}))$.

Let $\mathcal{A} = A_1, A_2, \ldots, A_k$ and $\mathcal{B} = B_1, B_2, \ldots, B_k$ be two collections of sets. If, for all i, we have $A_i \supseteq B_i$, then we say that $\mathcal{A}$ *dominates* $\mathcal{B}$.

Linear synteny. Consider an instance $\mathcal{S} = S_1, S_2, \ldots, S_k$. Formally, the *linear syntenic distance* problem is the restricted form of the synteny problem in which we consider only move sequences of the following form:

- Select one of the input sets S_{π_1}, and set the initial *merging set* $\Delta_1 := S_{\pi_1}$.
- The first $k - 1$ moves are fusions or translocations, restricted as follows:
  - The ith of these moves takes the current merging set Δ_i as input, along with one unused input set $S_{\pi_{i+1}}$, and produces a new merging set Δ_{i+1} as output.
  - If there is an element b that does not appear in any remaining unused input set—i.e., the element b appears only in $S_{\pi_{i+1}}$ and Δ_i—then the move is the translocation $(\Delta_i, S_{\pi_{i+1}}) \longrightarrow (\Delta_{i+1}, \{b\})$, where $\Delta_{i+1} := (\Delta_i \cup S_{\pi_{i+1}}) - \{b\}$. We say that b has been *emitted* or *produced* by this translocation.
  - If there is no such element b, then the ith move simply fuses the two sets: $(\Delta_i, S_{\pi_{i+1}}) \longrightarrow \Delta_{i+1}$, where $\Delta_{i+1} := \Delta_i \cup S_{\pi_{i+1}}$.

  If a set $S_j = \{b\}$ is the only set in a component (i.e., the element b appears only in S_j), then we do not merge it, and instead simply ignore this set. Call such an S_j a *lonely singleton.*
- Let Δ_k be the merging set after these $k - 1$ fusions and translocations have been completed. Each of the next $|\Delta_k| - 1$ moves are simple fissions.

  For the ith move, where $k \leq i \leq k + |\Delta_k| - 2$, let b be any element of Δ_i and let the move be the fission $\Delta_i \longrightarrow (\{b\}, \Delta_{i+1})$, where $\Delta_{i+1} := \Delta_i - \{b\}$.

$$\begin{array}{rcl}
\underline{\{1,2,5,6\}}, \{3\} & \longrightarrow & \{1,2,3,5,6\} \\
\underline{\{1,2,3,5,6\}}, \{1,2,3,4,5\} & \longrightarrow & \{3\}, \{1,2,4,5,6\} \\
\underline{\{1,2,4,5,6\}}, \{1,2,6\} & \longrightarrow & \{5\}, \{1,2,4,6\} \\
\underline{\{1,2,4,6\}}, \{1,2,4\} & \longrightarrow & \{1\}, \{2,4,6\} \\
\underline{\{2,4,6\}} & \longrightarrow & \{2\}, \{4,6\} \\
\underline{\{4,6\}} & \longrightarrow & \{4\}, \{6\}
\end{array}$$

Figure 1: The linear move sequence $\sigma^{\iota,\mathcal{S}}$ for the instance $\mathcal{S} = \{1,2,5,6\}$, $\{3\}$, $\{1,2,3,4,5\}$, $\{1,2,6\}$, $\{1,2,4\}$, where each merging set is underlined.

A linear move sequence can be completely determined by a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_k)$ of the input sets: the sets are merged in the order given by π, and the lexicographically smallest element is emitted whenever more than one element can be selected. (Which element is emitted in any translocation or fission does not affect the length of the move sequence.) Let $\sigma^{\pi,\mathcal{S}}$ denote the move sequence that results from using permutation π to order the input sets, and produces elements in this lexicographic order. We will use ι to denote the *identity permutation* $(1, 2, \ldots, k)$.

In Figure 1, we give an example of a linear move sequence $\sigma^{\iota,\mathcal{S}}$ for the instance $\mathcal{S} = \{1,2,5,6\}$, $\{3\}$, $\{1,2,3,4,5\}$, $\{1,2,6\}$, $\{1,2,4\}$. The first move is a fusion, because all elements appear at least one of the last three sets. The next three moves are translocations since some element does not appear in the remaining unused input sets when the move occurs. The last two moves are fissions.

The *linear syntenic distance* of an instance $\mathcal{S}$ is $\widetilde{D}(\mathcal{S}) := \min_\pi |\sigma^{\pi,\mathcal{S}}|$, the number of moves in the shortest linear move sequence for $\mathcal{S}$.

Note that if a linear move sequence performs α fusions in the first $k-1$ moves, then the move sequence contains $k-\alpha-1$ translocations. After the $k-1$ fusions and translocations are complete, there are $n-k+\alpha+1$ elements left in the merging set, since exactly one element is eliminated by each translocation. Therefore, $n-k+\alpha$ fissions must be performed to eliminate the remaining elements. Thus the length of the linear move sequence is $n+\alpha-1$ moves. (Every move either is a fusion or removes one element, and all but the last element must be removed.) We can therefore view the linear syntenic distance problem as the problem of maximizing the number of translocations in the sequence.

Computing the linear syntenic distance between two genomes is also known to be NP-hard [9]. The crucial theorem about linear syntenic distance is that it is not much larger than the general syntenic distance:

Theorem 2.1 (DasGupta et al. [9]) $D(\mathcal{S}) \leq \widetilde{D}(\mathcal{S}) \leq D(\mathcal{S}) + \log_{4/3}(D(\mathcal{S}))$. $\square$

We will say that an instance $\mathcal{S}$ with n elements and k sets is *linear exact* if $\widetilde{D}(\mathcal{S}) = \max(n,k)-1$. An instance is linear exact if and only if it can be solved using translocations whenever possible, i.e., fusions and fissions are only used to make up for differences in the number of elements and the number of sets.

3 Properties of Linear Synteny

In this section, we prove a number of structural properties for the linear syntenic distance. The majority of these are properties of the general model previously proven in [9, 18] which we now extend to the linear distance.

3.1 Monotonicity

We prove a *monotonicity property* for instances of the linear synteny problem: if an instance $\mathcal{S}$ dominates an instance $\mathcal{T}$, then $\widetilde{D}(\mathcal{S}) \geq \widetilde{D}(\mathcal{T})$. The same property was shown for $D(\cdot)$ in [18]. We actually prove a slightly stronger claim: for any permutation π, we have $|\sigma^{\pi,\mathcal{T}}| \leq |\sigma^{\pi,\mathcal{S}}|$. This stronger property will sometimes be useful in recovering an optimal move sequence for $\mathcal{T}$.

Lemma 3.1 *Let $\mathcal{S} = S_1, S_2, \ldots, S_k$ and $\mathcal{T} = T_1, T_2, \ldots, T_k$ be two instances such that $\mathcal{S}$ dominates $\mathcal{T}$, and let π be any permutation of $(1, 2, \ldots, k)$. Then $|\sigma^{\pi,\mathcal{T}}| \leq |\sigma^{\pi,\mathcal{S}}|$.*

Proof. If $\mathcal{T}$ contains more empty sets or lonely singletons than $\mathcal{S}$, then the length of the sequence $\sigma^{\pi,\mathcal{T}}$ is not affected by their presence since they do not need to be merged at all. In our analysis, we will count the merging of these sets against the length of $\sigma^{\pi,\mathcal{T}}$ anyway; this only increases our estimate of its length.

Let n and n' be the number of elements in $\mathcal{S}$ and $\mathcal{T}$, respectively. We assume that the elements are ordered such that, whenever possible, if move $\sigma_i^{\pi,\mathcal{S}}$ emits an element ℓ, then $\sigma_i^{\pi,\mathcal{T}}$ also emits element ℓ, and vice versa; any such reordering does not affect $|\sigma^{\pi,\mathcal{S}}|$ or $|\sigma^{\pi,\mathcal{T}}|$. To prove the lemma, it suffices to show the following:

($*$) For any translocation $\sigma_i^{\pi,\mathcal{S}}$ that emits an element b, either *(i)* the element b is emitted in some move in $\sigma_1^{\pi,\mathcal{T}}, \sigma_2^{\pi,\mathcal{T}}, \ldots, \sigma_i^{\pi,\mathcal{T}}$ or *(ii)* the instance $\mathcal{T}$ does not contain b.

If $\sigma^{\pi,\mathcal{S}}$ and $\sigma^{\pi,\mathcal{T}}$ contain α and α' fusions, respectively, then ($*$) implies that $\alpha' - \alpha \leq n - n'$ since each extra fusion in $\sigma^{\pi,\mathcal{T}}$ can be charged to an element that does not appear in $\mathcal{T}$. Then $|\sigma^{\pi,\mathcal{S}}| = n + \alpha - 1 \geq n' + \alpha' - 1 = |\sigma^{\pi,\mathcal{T}}|$, which proves the lemma.

To prove ($*$), suppose to the contrary that it does not hold, and let ℓ be the minimum index such that all of the following hold: (1) $\sigma_\ell^{\pi,\mathcal{S}}$ is a translocation emitting some element b; (2) the element b is not emitted in any of $\sigma_1^{\pi,\mathcal{T}}, \sigma_2^{\pi,\mathcal{T}}, \ldots, \sigma_\ell^{\pi,\mathcal{T}}$; and (3) the element b does appear somewhere in the genome $\mathcal{T}$.

Let χ_ℓ be the set of elements emitted during $\sigma_1^{\pi,\mathcal{S}}, \sigma_2^{\pi,\mathcal{S}}, \ldots, \sigma_{\ell-1}^{\pi,\mathcal{S}}$, and let χ'_ℓ be the set of elements emitted by $\sigma_1^{\pi,\mathcal{T}}, \sigma_2^{\pi,\mathcal{T}}, \ldots, \sigma_{\ell-1}^{\pi,\mathcal{T}}$. The current merging sets just before move ℓ are thus

$$\Delta = [S_{\pi_1} \cup S_{\pi_2} \cup \cdots \cup S_{\pi_\ell}] - \chi_\ell \qquad \Gamma = [T_{\pi_1} \cup T_{\pi_2} \cup \cdots \cup T_{\pi_\ell}] - \chi'_\ell.$$

The elements in the remaining unused input sets are

$$\overline{\Delta} = S_{\pi_{\ell+2}} \cup S_{\pi_{\ell+3}} \cup \cdots \cup S_{\pi_k} \qquad \overline{\Gamma} = T_{\pi_{\ell+2}} \cup T_{\pi_{\ell+3}} \cup \cdots \cup T_{\pi_k}.$$

Note that $\overline{\Delta} \supseteq \overline{\Gamma}$ since $\mathcal{S}$ dominates $\mathcal{T}$.

Since $\sigma_\ell^{\pi,\mathcal{S}}$ is a translocation emitting the element b, we know $b \in \Delta \cup S_{\pi_{\ell+1}}$ and $b \notin \overline{\Delta}$. From this and $\overline{\Delta} \supseteq \overline{\Gamma}$, we have that $b \notin \overline{\Gamma}$. So if $b \in \Gamma \cup T_{\pi_{\ell+1}}$, the move $\sigma_\ell^{\pi,\mathcal{T}}$ could emit b, but, by assumption, it does not. Then $b \notin \Gamma$, $b \notin T_{\pi_{\ell+1}}$, and $b \notin \overline{\Gamma}$.

The elements of $\mathcal{T}$ are all contained in χ'_ℓ, Γ, $T_{\pi_{\ell+1}}$, and $\overline{\Gamma}$. Thus either $b \in \chi'_\ell$ was emitted earlier in the sequence $\sigma^{\pi,\mathcal{T}}$, or b does not appear anywhere in the genome $\mathcal{T}$. This contradicts our assumption. $\square$

Corollary 3.2 (Linear Monotonicity) *If $\mathcal{S}$ dominates $\mathcal{T}$, then $\widetilde{D}(\mathcal{S}) \geq \widetilde{D}(\mathcal{T})$.*

Proof. Suppose $\mathcal{S}$ has k sets, and let π be a permutation of $(1, 2, \ldots, k)$ so that $\sigma^{\pi,\mathcal{S}}$ is optimal. Then we have $\widetilde{D}(\mathcal{S}) = |\sigma^{\pi,\mathcal{S}}| \geq |\sigma^{\pi,\mathcal{T}}| \geq \widetilde{D}(\mathcal{T})$ by Lemma 3.1. $\square$

3.2 Merging Set Expansion

In this section, we show the *merging set expansion* property for linear syntenic distance: for an instance $\mathcal{S}$ in which some set Δ is designated as the current merging set, if we add to Δ any of the elements that appear in $\mathcal{S}$, then the linear syntenic distance does not change.

We will subsequently consider instances $\mathcal{S} = \Delta, S_1, S_2, \ldots, S_k$ in which the set Δ is already designated as the merging set; we can consider such an instance by limiting our attention to permutations π where $\pi_1 = 1$.

Lemma 3.3 *Let $\mathcal{S} = \Delta, S_1, S_2, \ldots, S_k$ and $\mathcal{T} = \Delta \cup T, S_1, S_2, \ldots, S_k$ be two instances, for any set $T \subseteq S_1 \cup S_2 \cup \cdots \cup S_k$. For any permutation π of $(1, 2, \ldots, k+1)$ with $\pi_1 = 1$, we have $|\sigma^{\pi,\mathcal{S}}| = |\sigma^{\pi,\mathcal{T}}|$.*

Proof. Note that $\mathcal{T}$ dominates $\mathcal{S}$, so by Lemma 3.1, we have $|\sigma^{\pi,\mathcal{S}}| \leq |\sigma^{\pi,\mathcal{T}}|$.

For the other direction, assume that the elements are ordered such that, whenever possible, $\sigma_i^{\pi,\mathcal{S}}$ and $\sigma_i^{\pi,\mathcal{T}}$ emit the same element. Suppose that $|\sigma^{\pi,\mathcal{S}}| < |\sigma^{\pi,\mathcal{T}}|$. Then $\sigma^{\pi,\mathcal{S}}$ must do more translocations than $\sigma^{\pi,\mathcal{T}}$. Let ℓ be the index of the first move in which $\sigma_\ell^{\pi,\mathcal{S}}$ produces some element b by translocation, and $\sigma_\ell^{\pi,\mathcal{T}}$ cannot produce b. The element b cannot have been emitted earlier in $\sigma_\ell^{\pi,\mathcal{T}}$, because of our assumption that, whenever possible, the two move sequences produce the same element, and the fact that we have chosen ℓ to be the first time this cannot be done.

Therefore *(i)* $\sigma_\ell^{\pi,\mathcal{S}}$ emits b, but *(ii)* $\sigma_1^{\pi,\mathcal{T}}, \sigma_2^{\pi,\mathcal{T}}, \ldots, \sigma_\ell^{\pi,\mathcal{T}}$ cannot emit b. From *(i)*, we know that b appears in $\Delta \cup S_{\pi_1} \cup \cdots \cup S_{\pi_\ell}$ but not in $S_{\pi_{\ell+1}} \cup \cdots \cup S_{\pi_k}$. But clearly b also appears in $\Delta \cup T \cup S_{\pi_1} \cup \cdots \cup S_{\pi_\ell}$, and still does not appear in $S_{\pi_{\ell+1}} \cup \cdots \cup S_{\pi_k}$. Thus $\sigma_\ell^{\pi,\mathcal{T}}$ can produce b if it has not been emitted by a previous move, violating *(ii)*. $\square$

Corollary 3.4 (Merging Set Expansion) *Let $\mathcal{S} = \Delta, S_1, S_2, \ldots, S_k$ be an instance in which Δ is the current merging set. Let the instance $\mathcal{T} = \Delta \cup T, S_1, S_2, \ldots, S_k$ in which $\Delta \cup T$ is the merging set, for any set $T \subseteq S_1 \cup S_2 \cup \cdots \cup S_k$. Then $\widetilde{D}(\mathcal{S}) = \widetilde{D}(\mathcal{T})$.*

Proof. By linear monotonicity, $\widetilde{D}(\mathcal{S}) \leq \widetilde{D}(\mathcal{T})$.

Let π be a permutation of $(1, 2, \ldots, k+1)$ where $\pi_1 = 1$, i.e., in which Δ (or $\Delta \cup T$) is the initial merging set, so that $\sigma^{\pi,\mathcal{S}}$ is optimal. Then $\widetilde{D}(\mathcal{S}) = |\sigma^{\pi,\mathcal{S}}| = |\sigma^{\pi,\mathcal{T}}| \geq \widetilde{D}(\mathcal{T})$. $\square$

3.3 Linear Canonicalization

We now prove the existence of *canonical* optimal linear move sequences, in which all fusions occur before all translocations, for every instance $\mathcal{S}$. DasGupta et al. [9] proved the analogous result for the general syntenic distance.

Theorem 3.5 (Linear Canonicalization) *For any instance $\mathcal{S} = S_1, S_2, \ldots, S_k$, there exists a permutation π of $(1, 2, \ldots, k)$ such that $\sigma^{\pi,\mathcal{S}}$ is optimal and has all fusions preceding all translocations.*

Proof. Let π be a permutation of $(1, 2, \ldots, k)$ such that $\sigma^{\pi,\mathcal{S}}$ is optimal and has as many initial fusions as possible. Suppose that move $\sigma_i^{\pi,\mathcal{S}}$ is the last initial fusion and $\sigma_j^{\pi,\mathcal{S}}$ is the first non-initial fusion, for $j \geq i+2$. (If there is no non-initial fusion, we are done.)

Let $\pi' = (\pi_1, \ldots, \pi_{i+1}, \pi_{j+1}, \pi_{i+2}, \ldots, \pi_j, \pi_{j+2}, \ldots, \pi_k)$ be π modified so that π_{j+1} is immediately after π_{i+1}. We claim that $\sigma^{\pi',\mathcal{S}}$ is also optimal, and has one more initial fusion than $\sigma^{\pi,\mathcal{S}}$. This violates our choice of π and proves the theorem. Again we assume that the elements are ordered so that, whenever possible, the moves $\sigma_i^{\pi,\mathcal{S}}$ and $\sigma_i^{\pi',\mathcal{S}}$ emit the same element.

First we claim that $\sigma^{\pi',\mathcal{S}}$ has $i+1$ initial fusions. Clearly, the first i moves of the two sequences are identical, since they merge exactly the same sets (and exactly the same sets remain unmerged). Thus we need only prove that $\sigma_{i+1}^{\pi',\mathcal{S}}$ is a fusion. Suppose that it were a translocation, i.e., there is some element ℓ that appears only in the sets $S_{\pi_1}, S_{\pi_2}, \ldots, S_{\pi_{i+1}}, S_{\pi_{j+1}}$. If $\ell \in S_{\pi_{j+1}}$, then the move $\sigma_j^{\pi,\mathcal{S}}$ would be a translocation, since the last occurrence of the element ℓ is in the set $S_{\pi_{j+1}}$. If $\ell \notin S_{\pi_{j+1}}$, and instead $\ell \in S_{\pi_1} \cup S_{\pi_2} \cup \cdots \cup S_{\pi_{i+1}}$, there would be a translocation somewhere in $\sigma_1^{\pi,\mathcal{S}}, \sigma_2^{\pi,\mathcal{S}}, \ldots, \sigma_i^{\pi,\mathcal{S}}$, since ℓ does not appear outside the first $i+1$ sets. Neither of these occur, so there is no such ℓ, and $\sigma_{i+1}^{\pi',\mathcal{S}}$ is a fusion.

For optimality, we claim that every element emitted by translocation in $\sigma^{\pi,\mathcal{S}}$ is emitted in $\sigma^{\pi',\mathcal{S}}$. Note that, for all $j+2 \leq r \leq k$, we have $\pi_r = \pi'_r$, which implies that any element b emitted by a move in $\sigma_{j+1}^{\pi,\mathcal{S}}, \sigma_{j+2}^{\pi,\mathcal{S}}, \ldots, \sigma_{k-1}^{\pi,\mathcal{S}}$ is also emitted by some move in $\sigma_{j+1}^{\pi',\mathcal{S}}, \sigma_{j+2}^{\pi',\mathcal{S}}, \ldots, \sigma_{k-1}^{\pi',\mathcal{S}}$ unless b were previously emitted in the sequence $\sigma^{\pi',\mathcal{S}}$. For earlier moves, suppose that $\sigma_r^{\pi,\mathcal{S}}$ produces an element b by translocation, for some $1 \leq r \leq j$. That is, the element b appears in $S_{\pi_1}, S_{\pi_2}, \ldots, S_{\pi_{r+1}}$ and not in $S_{\pi_{r+2}}, S_{\pi_{r+3}}, \ldots, S_{\pi_k}$. Obviously having already merged $S_{\pi_{j+1}}$ changes neither the presence of b in the current merging set nor the absence of b in the unused input sets. Thus $\sigma_{r+1}^{\pi',\mathcal{S}}$ is a translocation emitting b, unless the element b were previously emitted in $\sigma^{\pi',\mathcal{S}}$. □

3.4 Duality

Finally, we show the *duality* property $\widetilde{D}(\mathcal{S}) = \widetilde{D}(\mathsf{dual}(\mathcal{S}))$ for the linear syntenic distance; this property was proven for the unconstrained syntenic distance by DasGupta et al. [9].

Proposition 3.6 *Let $\mathcal{S} = S_1, S_2, \ldots, S_k$ and $\mathcal{T} = T_1, T_2, \ldots, T_k$ be two instances. If $\mathcal{S}$ dominates $\mathcal{T}$, then $\mathsf{dual}(\mathcal{S})$ dominates $\mathsf{dual}(\mathcal{T})$.*

Proof. Suppose not. Let $\mathsf{dual}(\mathcal{S}) = S'_1, S'_2, \ldots, S'_{n_1}$ and $\mathsf{dual}(\mathcal{T}) = T'_1, T'_2, \ldots, T'_{n_2}$, where $n_2 \leq n_1$.

Since $\mathsf{dual}(\mathcal{S})$ does not dominate $\mathsf{dual}(\mathcal{T})$, there is some set i such that $S'_i \not\supseteq T'_i$. That is, there is some element $\ell \in T'_i$ but $\ell \notin S'_i$. By the definition of the dual, this means that the element $i \in T_\ell$ but $i \notin S_\ell$. This violates the assumption that $\mathcal{S}$ dominates $\mathcal{T}$. □

In our proof of linear duality, we will consider the following special class of instances:

Definition 3.7 *For $\alpha \leq \min(n, k) - 1$, the instance $\mathcal{K}_{\alpha,n,k}$ consists of the k sets $S_1, S_2, \ldots, S_k$:*

— *For $1 \leq i \leq k - \alpha$, we have $S_i = \{1, 2, \ldots, n\}$.*

— *For $k - \alpha + 1 \leq i \leq k$, we have $S_i = \{1, 2, \ldots, n - i + k - \alpha\}$.*

Note the following facts:

- $\widetilde{D}(\mathcal{K}_{\alpha,n,k}) = n + k - \alpha - 3$. Merging the sets in the stated order requires $k - \alpha - 2$ fusions, $\alpha + 1$ translocations, and $n - \alpha - 2$ fissions, or $n + k - \alpha - 3$ moves total.

  In the first $m - 1$ moves of any linear move sequence, only elements that appear in at most m sets can be emitted [18]. Here, there are only α elements that appear in at most $k - 1$ sets, so the first $k - 2$ moves can emit at most α elements. Thus there are at least $k - 2 - \alpha$ fusions in any linear move sequence for $\mathcal{K}_{\alpha,n,k}$, and $\widetilde{D}(\mathcal{K}_{\alpha,n,k}) \geq n + k - \alpha - 3$.

- $\mathsf{dual}(\mathcal{K}_{\alpha,k,n}) = \mathcal{K}_{\alpha,n,k}$. We can verify this straightforwardly: the first $n - \alpha$ elements appear in all sets, the element $n - \alpha + 1$ appears in all but S_k, etc.

Theorem 3.8 (Linear Duality) *For all $\mathcal{S}$, $\widetilde{D}(\mathcal{S}) = \widetilde{D}(\mathsf{dual}(\mathcal{S}))$.*

Proof. Suppose not, i.e., suppose that $\widetilde{D}(\mathcal{S}) < \widetilde{D}(\mathsf{dual}(\mathcal{S}))$. Let n and k be the number of elements and sets in $\mathcal{S}$, respectively.

Relabel the sets and elements of $\mathcal{S}$ such that the move sequence $\sigma^{\iota,\mathcal{S}}$ is optimal, canonical, and produces elements in the order $n, n-1, \ldots, 1$. Note that this relabeling does not change $\widetilde{D}(\mathcal{S})$. Let $\widetilde{D}(\mathcal{S}) = n + k - \alpha - 3$.

Notice that the element $n - i$ does not appear outside the first $k - \alpha + i$ sets, since otherwise the $(k - \alpha + i - 1)$th move could not produce element $n - i$. Therefore, we have that $\mathcal{K}_{\alpha,n,k}$ dominates $\mathcal{S}$. Thus $\mathsf{dual}(\mathcal{K}_{\alpha,n,k})$ dominates $\mathsf{dual}(\mathcal{S})$ by Proposition 3.6. Linear monotonicity, along with the fact that $\mathsf{dual}(\mathcal{K}_{\alpha,k,n}) = \mathcal{K}_{\alpha,n,k}$, then gives us

$$\widetilde{D}(\mathcal{S}) \quad = \quad \widetilde{D}(\mathcal{K}_{\alpha,n,k}) \quad = \quad \widetilde{D}(\mathsf{dual}(\mathcal{K}_{\alpha,k,n})) \quad \geq \quad \widetilde{D}(\mathsf{dual}(\mathcal{S})).$$

This contradicts the assumption and proves the theorem. □

4 From General Linear to Exact Linear Synteny

In this section, we give a reduction from the general linear to the exact linear synteny problem, a conceptually simpler problem. We first define an augmentation to instances:

Definition 4.1 *For an instance $\mathcal{S} = S_1, S_2, \ldots, S_k$, and for any $1 \leq i \leq k$, let*

$$\mathcal{S}^{i\uparrow\delta} := S_1, S_2, \ldots, \hat{S}_i, \ldots, S_k,$$

where $\hat{S}_i = S_i \cup \{a_1, a_2, \ldots, a_\delta\}$, and, for all $1 \leq \ell \leq \delta$ and $1 \leq j \leq k$, we have $a_\ell \notin S_j$.

The intuition behind this instance is that we have augmented S_i with extra elements that will be emitted during would-be fusions. This new instance can be thought of as the original with δ fusion "coupons" that can be used to turn fusions into translocations. For some choices of i and δ, this increases the number of elements and translocations without a corresponding increase in distance.

Theorem 4.2 *Let $\mathcal{S}$ be an instance with n elements. Suppose $\sigma^{\pi,\mathcal{S}}$ is a move sequence solving $\mathcal{S}$ such that $|\sigma^{\pi,\mathcal{S}}| = n + \alpha - 1$. Then $|\sigma^{\pi,\mathcal{S}^{\pi_1\uparrow\delta}}| = n + \max(\alpha, \delta) - 1$.*

Proof. There are α fusions in $\sigma^{\pi,\mathcal{S}}$. When the jth of these fusions occurs, $\sigma^{\pi,\mathcal{S}^{\pi_1\uparrow\delta}}$ could emit the element a_j (since a_j does not appear in any of the unused input sets, and is in the merging set as of the first move). Every translocation in the original move sequence remains a translocation in the new sequence since we have the same remaining unused input sets at every point.

Thus we can eliminate fusions from the move sequence using a-elements, until we run out of fusions or a-elements with which to eliminate them. Thus we have $\alpha - \delta$ fusions left if there are too many fusions, and therefore $|\sigma^{\pi,\mathcal{S}^{\pi_1\uparrow\delta}}| = n + \delta + \max(\alpha - \delta, 0) - 1 = n + \max(\alpha, \delta) - 1$. □

Consider an instance $\mathcal{S} = S_1, S_2, \ldots, S_k$ for which we somehow know that there is an optimal linear move sequence $\sigma^{\pi,\mathcal{S}}$ where $\pi_1 \in \Gamma$, for some set $\Gamma \subseteq \{1, 2, \ldots, k\}$. For any instance $\mathcal{S}$ with k sets, of course, we can take $\Gamma = \{1, 2, \ldots, k\}$, but for certain classes of instances we can prove that there is an optimal move sequence with $\pi_1 \in \Gamma$ for a much smaller set Γ. The algorithm that we will develop in Section 7.3 loops over each possible value of $\pi_1 \in \Gamma$, so a smaller set Γ will yield a better running time.

Proposition 4.3 *Let $\mathcal{S}$ be an instance with n elements and k sets, and let Γ be some subset of $\{1, 2, \ldots, k\}$. If there exists an optimal move sequence $\sigma^{\pi,\mathcal{S}}$ solving $\mathcal{S}$ such that $\pi_1 \in \Gamma$, then*

$$\left[\exists i \in \Gamma \quad \widetilde{D}(\mathcal{S}^{i\uparrow\delta}) = n + \delta - 1\right] \iff \widetilde{D}(\mathcal{S}) \leq n + \delta - 1.$$

Proof. Immediate from monotonicity and Theorem 4.2. □

5 Nested Synteny

In this section, we consider the special class of instances in which all non-disjoint sets are totally ordered by the subset relation:

Definition 5.1 (Nested Synteny) *An instance $\mathcal{S} = S_1, S_2, \ldots, S_k$ is* nested *if, for all $1 \leq i \leq k$ and $1 \leq j \leq k$, either (1) $S_i \cap S_j = \emptyset$, (2) $S_i \subseteq S_j$, or (3) $S_j \subseteq S_i$.*

In each component of a nested instance of synteny, call the set containing all elements in the component the *root* of the component. If there are multiple copies of this set in some component, we will identify the root as the copy with the smallest index.

Lemma 5.2 *If $\mathcal{S} = S_1, S_2, \ldots, S_k$ and, for some $1 \leq i \leq k$ and $1 \leq j \leq k$, we have $S_i \subseteq S_j$, then there exists an optimal linear move sequence solving $\mathcal{S}$ in which S_j is merged before S_i.*

Proof. Suppose π is a permutation of $(1, 2, \ldots, k)$ such that $\sigma^{\pi,\mathcal{S}}$ is optimal and i appears before j in π. (If there is no such π, then we are done.) Let $\pi_x = i$.

If we have $x \geq 3$, then let $\mathcal{T} = T_1, T_2, \ldots, T_{k'-3}, S_i, S_j, \Delta$ be the instance resulting after the completion of the first $x-2$ moves $\sigma_1^{\pi,\mathcal{S}}, \sigma_2^{\pi,\mathcal{S}}, \ldots, \sigma_{x-2}^{\pi,\mathcal{S}}$, where Δ is the merging set after these $x-2$ moves. (For ease of reference, we have named the sets of $\mathcal{T}$ in this particular order, but of course S_i, S_j, and Δ may be at a different point in the sequence of sets of the instance.) The next move $\sigma_{x-1}^{\pi,\mathcal{S}}$ would merge S_i. We have two cases for this move:

- $\sigma^{\pi,\mathcal{S}}_{x-1}$ is a fusion, $(\Delta, S_i) \longrightarrow \Delta \cup S_i$.

  We define the following instances, where the last set is designated as the current merging set:

$$\begin{aligned}
\mathcal{U}_1 &:= T_1, T_2, \ldots, T_{k'-3}, S_j, \Delta \cup S_i \\
\mathcal{U}_2 &:= T_1, T_2, \ldots, T_{k'-3}, S_j, \Delta \cup S_j \\
\mathcal{U}_3 &:= T_1, T_2, \ldots, T_{k'-3}, S_i, \Delta \cup S_j.
\end{aligned}$$

  By Corollary 3.4, since $\Delta \cup S_j \supseteq \Delta \cup S_i$ and obviously $S_j \subseteq T_1 \cup T_2 \cup \cdots \cup T_{k'-3} \cup S_j$, we have $\widetilde{D}(\mathcal{U}_1) = \widetilde{D}(\mathcal{U}_2)$. By linear monotonicity, we have $\widetilde{D}(\mathcal{U}_2) \geq \widetilde{D}(\mathcal{U}_3)$, since $S_i \subseteq S_j$. Therefore $\widetilde{D}(\mathcal{U}_1) \geq \widetilde{D}(\mathcal{U}_3)$.

  The fusion $\sigma^{\pi,\mathcal{S}}_{x-1} = (\Delta, S_i) \longrightarrow \Delta \cup S_i$ produces the instance $\mathcal{U}_1$. If we instead complete the fusion $(\Delta, S_j) \longrightarrow \Delta \cup S_j$, then the resulting instance is $\mathcal{U}_3$. Therefore, fusing S_j instead of fusing S_i yields an instance that is no harder, and making this move instead cannot increase the distance.

- $\sigma^{\pi,\mathcal{S}}_{x-1}$ is a translocation, $(\Delta, S_i) \longrightarrow (\Delta \cup S_i - \{b\}, \{b\})$ for some $b \in \Delta \cup S_i$ and $b \notin T_1 \cup T_2 \cup \ldots \cup T_{k'-3} \cup S_j$.

  Consider the following instances, where the last set is again designated as the merging set:

$$\begin{aligned}
\mathcal{V}_1 &:= T_1, T_2, \ldots, T_{k'-3}, S_j, \Delta \cup S_i - \{b\} \\
\mathcal{V}_2 &:= T_1, T_2, \ldots, T_{k'-3}, S_j, \Delta \cup S_j - \{b\} \\
\mathcal{V}_3 &:= T_1, T_2, \ldots, T_{k'-3}, S_i, \Delta \cup S_j - \{b\}.
\end{aligned}$$

  The move $\sigma^{\pi,\mathcal{S}}_{x-1} = (\Delta, S_i) \longrightarrow (\Delta \cup S_i - \{b\}, \{b\})$ produces $\mathcal{V}_1$. By successively applying Corollaries 3.4 and 3.2, as in the fusion case, we have that $\widetilde{D}(\mathcal{V}_1) = \widetilde{D}(\mathcal{V}_2) \geq \widetilde{D}(\mathcal{V}_3)$.

  If instead of doing $\sigma^{\pi,\mathcal{S}}_{x-1}$, we merge S_j instead, we can still complete a translocation: we have $b \in \Delta \cup S_j \supseteq \Delta \cup S_i$, and $b \notin S_i \subseteq S_j$, so this move is $(\Delta, S_j) \longrightarrow (\Delta \cup S_j - \{b\}, \{b\})$. The result of this move is $\mathcal{V}_3$, so by the above, making this move instead can only decrease the length of the sequence.

In either case, we have shown how to merge S_j before S_i without increasing the length of the move sequence, for any $x \geq 3$. The proof for $x \in \{1, 2\}$ is strictly analogous (by considering the previous merging set Δ to be $S_{\pi_{(2-x)}}$, the other set merged in the first move of $\sigma^{\pi,\mathcal{S}}$). ◻

Corollary 5.3 *For any instance $\mathcal{S} = S_1, S_2, \ldots, S_k$, there exists an optimal move sequence $\sigma^{\pi,\mathcal{S}}$ solving the instance such that for all $1 \leq i < j \leq k$, we have $S_{\pi_i} \not\subset S_{\pi_j}$.*

Proof. The transformation described in Lemma 5.2 can be applied to the lexicographically minimum violating pair $\langle i, j \rangle$, and does not create any new lexicographically smaller violations. Thus repeatedly fixing the first violation in an arbitrary optimal linear move sequence eventually leads to an optimal sequence with no violations. ◻

Corollary 5.4 *If $\mathcal{S}$ is a nested instance with roots $R_1, R_2, \ldots, R_p$, then there exists an optimal move sequence $\sigma^{\pi,\mathcal{S}}$ solving $\mathcal{S}$ such that $\pi_1 \in \{R_1, R_2, \ldots, R_p\}$.* ◻

Note that if $\mathcal{S}$ is nested, then so is $\mathcal{S}^{R_q \uparrow \delta}$ since we are only adding extra elements to the root of some component.

6 The Minimum Loan Problem

We now formally define the MIN LOAN problem and review previous results.

Definition 6.1 *Let $T = \{T_1, T_2, \ldots, T_n\}$ be a set of* tasks. *Let $v : T \longrightarrow \mathbb{Z}$ be a function giving the* profit *of each task. Then, for π a permutation of $(1, 2, \ldots, n)$, the quantity*

$$V_\pi(i) := \sum_{j=1}^{i} v(T_{\pi_j})$$

is the cumulative profit *of the first i tasks under π.*

Note that if tasks have negative profits (i.e., costs), then the cumulative profit can also be negative. We will say that a permutation π *respects* a partial order $\prec$ on T if, for all $i < j$, we have $T_{\pi_j} \not\prec T_{\pi_i}$. This gives rise to the following scheduling problem:

Definition 6.2 (MIN LOAN) *Let $T = \{T_1, T_2, \ldots, T_n\}$ be a set of* tasks. *Let $\prec$ be a partial order on T defining a* precedence relation *among the tasks. Let $v : T \longrightarrow \mathbb{Z}$ be a function giving the* profit *of each task. Then $\langle T, \prec, v \rangle$ is an instance of the* minimum loan *problem: find*

$$\max_{\pi} \min_{0 \leq i \leq n} V_\pi(i)$$

for π respecting $\prec$.

(Abdel-Wahab and Kameda [1] define this problem in terms of the cumulative *cost* of the tasks rather than the cumulative profit.) Notice that for any permutation π, we have $V_\pi(0) = 0$, so the optimum value for any instance of the MIN LOAN problem is always non-positive.

The intuition for this problem is the following: suppose that there is a company with a set of jobs that it has chosen to undertake. Each job will result in either a profit or a loss. The jobs must respect some precedence constraints, e.g., the engines must be built before the cars can be assembled. The *minimum loan* is the minimum amount of initial funding necessary to be able complete all of the jobs without ever running out of money. (Alternatively, this is the maximum amount of debt for the company at the worst financial moment.)

The MIN LOAN problem is NP-complete in general [14, p. 238], but Abdel-Wahab and Kameda [1] give an $O(n^2)$ algorithm when $\prec$ is *series-parallel*. Monma and Sidney [19] independently give a polynomial-time algorithm for this case as well. A partial order is series-parallel when its associated DAG is a series-parallel graph, according to the following rules:

- a graph with two nodes with an edge from one to the other is series-parallel;
- if G is series-parallel, then so is the graph that results from adding a node to the middle of any edge in G; and
- if G is series-parallel, then so is the graph that results from duplicating any edge in G.

We will not go into the details of the algorithms of Abdel-Wahab and Kameda or Monma and Sidney here; rather, we will use them in a black box fashion in our approach to the nested linear synteny problem in the following section.

7 Minimum Loans and Exact Linear Synteny

In this section, we establish a connection between a given nested instance of the linear syntenic distance problem and a class of series-parallel instances of the Min Loan problem. Throughout this section, we will consider a nested instance $\mathcal{S} = S_1, S_2, \ldots, S_k$ with $n \geq k$ elements and p components. Let the roots of the components be $S_{R_1}, S_{R_2}, \ldots, S_{R_p}$.

Our p Min Loan instances will have different profit functions, but will all consider the same set of tasks and the same precedence constraints.

Definition 7.1 *Define $T = \{T_1, T_2, \ldots, T_k, T_{\mathit{first}}, T_{\mathit{last}}\}$ to be a set of tasks.*

Intuitively, for each $1 \leq j \leq k$, the completion of the task T_j denotes the merging of the set S_j with the current merging set. The tasks T_{first} and T_{last} are dummy tasks required to make our precedence relation series-parallel, and have no meaning in terms of a linear move sequence solving $\mathcal{S}$. (In each of our Min Loan instances, the profit associated with T_{first} and T_{last} will be zero; thus their presence does not in any way affect the optimal minimum loan.)

Definition 7.2 *Let $\prec$ be the smallest relation such that, for all $1 \leq j \leq k$ and $1 \leq \ell \leq k$, we have*

1. *$T_{\mathit{first}} \prec T_{\mathit{last}}$, $T_{\mathit{first}} \prec T_j$, and $T_j \prec T_{\mathit{last}}$;*
2. *$T_j \prec T_\ell$ if $S_j \supset S_\ell$; and*
3. *$T_j \prec T_\ell$ if $S_j = S_\ell$ and $j \leq \ell$.*

Lemma 7.3 *The precedence relation $\prec$ is series-parallel.*

Proof. For a nested instance $\mathcal{S}$, we have a forest of inclusion constraints, with the roots of the trees corresponding to the roots of the components in $\mathcal{S}$. With the inclusion of T_{first}, this becomes a tree; with T_{last} included, every leaf of the tree points to T_{last}.

We outline the method for constructing $\prec$ using the rules of series-parallel relations. Start with a single edge from T_{first} to T_{last}. Now if any node T_i in the graph has $\gamma \geq 1$ children in the inclusion tree, duplicate $\gamma - 1$ times the edge from T_i to T_{last} and add a node to each such edge, and label the nodes as the children of T_i. Iterating this process yields the relation $\prec$. □

Note that, regardless of the profit function, the only feasible solutions to any Min Loan problem with $\prec$ as the precedence constraints must complete T_{first} and T_{last} as the first and last tasks in the sequence, respectively. Throughout the remainder of this section, we will only consider sequences that respect $\prec$, and we will henceforth omit reference to T_{first} and T_{last}. (The presence of these tasks is solely to make $\prec$ series-parallel.) From now on, all references to permutations π of the tasks will be permutations of the tasks $T_1, T_2, \ldots, T_k$.

We will consider p different instances of Min Loan, using the following p profit functions:

Definition 7.4 *For any $1 \leq i \leq k$ and $1 \leq q \leq p$, the q-*profit *of task T_i is the following:*

$$v^q(T_i) \quad := \quad \begin{cases} \left| S_i - \bigcup_{j : T_i \prec T_j} S_j \right| - 1 & \text{if } i \neq R_q \\[2ex] \left| S_i - \bigcup_{j : T_i \prec T_j} S_j \right| & \text{if } i = R_q. \end{cases}$$

Let $v^q(T_{\mathit{first}}) := 0$ and $v^q(T_{\mathit{last}}) := 0$.

Let $V_\pi^q(i) := \sum_{j=1}^{i} v^q(T_{\pi_j})$ be the *cumulative q-profit* of the first i steps under permutation π.

Intuitively, the profit of a task T_j is one less than the number of elements that can be emitted via translocation once set S_j is merged; we must decrease the profit of each task T_j by one to account for the element that is emitted during the merging of set S_j. We consider p different profit functions $v^1, v^2, \ldots, v^p$ since the merging of the first two sets only requires a single element to be emitted, and one of the roots $R_1, R_2, \ldots, R_p$ will be the first task completed. If the tasks are ordered so that cumulative profit is always non-negative, then merging the sets in that order means that there is always an element in the merging set that can be emitted at every stage of the linear move sequence. Then each set can be merged via a translocation, and the instance is linear exact.

We write $\mathsf{opt}(T, \prec, v^q)$ to denote the optimum value for the Min Loan instance $\langle T, \prec, v^q \rangle$. In this section, we will prove the following result, and then apply it to give an efficient algorithm for nested instances of the linear syntenic distance problem:

Theorem 7.5 *Let $\mathcal{S}$ be a nested instance with n elements, $k \leq n$ sets, and p components with roots $R_1, R_2, \ldots, R_p$. Then $\mathcal{S}$ is linear exact if and only if, for some $1 \leq q \leq p$, we have $\mathsf{opt}(T, \prec, v^q) = 0$.*

Proof. Immediate from Lemma 7.11, which establishes the forward implication, and Lemma 7.14, which proves the reverse implication. □

7.1 From Linear Synteny to Minimum Loans

First we prove that, given a linear exact nested instance $\mathcal{S}$ of the linear syntenic distance problem, there is some $1 \leq q \leq p$ so that the optimum value $\mathsf{opt}(T, \prec, v^q)$ is zero.

Definition 7.6 *For a permutation π of $(1, 2, \ldots, k)$:*

1. *Define $x_\pi^{\mathcal{S}}(i) := 1$ if $\sigma_i^{\pi,\mathcal{S}}$ is a translocation, and $x_\pi^{\mathcal{S}}(i) := 0$ otherwise.*
2. *Let $\chi_\pi^{\mathcal{S}}(i)$ be the set of elements emitted by translocation during $\sigma_1^{\pi,\mathcal{S}}, \sigma_2^{\pi,\mathcal{S}}, \ldots, \sigma_i^{\pi,\mathcal{S}}$.*

Proposition 7.7 *For all instances $\mathcal{S}$ with k sets, for every π a permutation of $(1, 2, \ldots, k)$, and for all $0 \leq i \leq k-1$, we have $|\chi_\pi^{\mathcal{S}}(i)| \leq i$.*

Proof. At most one element can be emitted per move. □

Proposition 7.8 *For all instances $\mathcal{S} = S_1, S_2, \ldots, S_k$, for every π a permutation of $(1, 2, \ldots, k)$ that respects $\prec$, and for all $1 \leq i \leq k$, we have*

$$S_{\pi_i} - \bigcup_{j:T_{\pi_i} \prec T_{\pi_j}} S_{\pi_j} \quad = \quad S_{\pi_i} - \bigcup_{j>i} S_{\pi_j}.$$

Proof. If $T_{\pi_i} \prec T_{\pi_j}$, then $j > i$ since π respects $\prec$. Thus $\bigcup_{j:T_{\pi_i} \prec T_{\pi_j}} S_{\pi_j} \subseteq \bigcup_{j>i} S_{\pi_j}$, and we have $S_{\pi_i} - \bigcup_{j:T_{\pi_i} \prec T_{\pi_j}} S_{\pi_j} \supseteq S_{\pi_i} - \bigcup_{j>i} S_{\pi_j}$.

For the other direction, consider an arbitrary element $b \in S_{\pi_i} - \bigcup_{j:T_{\pi_i} \prec T_{\pi_j}} S_{\pi_j}$, so that $b \in S_{\pi_i}$ and $b \notin \bigcup_{j:T_{\pi_i} \prec T_{\pi_j}} S_{\pi_j}$. We claim that $b \notin \bigcup_{j>i} S_{\pi_j}$, either: for every $j > i$, either $T_{\pi_i} \prec T_{\pi_j}$ or $S_{\pi_i} \cap S_{\pi_j} = \emptyset$, since $\mathcal{S}$ is nested and π respects $\prec$. □

Proposition 7.9 *For all nested instances $\mathcal{S} = S_1, S_2, \ldots, S_k$ containing p components with roots $R_1, R_2, \ldots, R_p$, and for some $1 \le q \le p$, there exists an optimal linear move sequence $\sigma^{\pi,\mathcal{S}}$ so that $\pi_1 = R_q$ and π respects $\prec$.*

Proof. Let $\sigma^{\pi',\mathcal{S}}$ be an optimal move sequence respecting the subset precedence relation (by Corollary 5.3), and let R_q be the root of the component containing $S_{\pi'_1}$. Then $S_{\pi'_1} = S_{R_q}$ since all other sets in that component are subsets of S_{R_q}. We can make this sequence respect $\prec$ trivially by using identical sets in increasing order of index. □

Lemma 7.10 *Consider any nested linear exact instance $\mathcal{S} = S_1, S_2, \ldots, S_k$ with $n \ge k$ elements and p components with roots $R_1, R_2, \ldots, R_p$, and any permutation π of $(1, 2, \ldots, k)$ such that $\sigma^{\pi,\mathcal{S}}$ is optimal, $\pi_1 = R_q$ for some $1 \le q \le p$, and π respects $\prec$. Then for all $0 \le i \le k$, we have*

$$V^q_\pi(i) = \left|\bigcup\nolimits_{j \le i} S_{\pi_j} - \bigcup\nolimits_{j > i} S_{\pi_j}\right| - \left|\chi^{\mathcal{S}}_\pi(i-1)\right|.$$

Proof. We proceed by induction on i.

- $[i = 0]$. Then we have $V^q_\pi(i) = \sum_{j=1}^{0} v^q(T_{\pi_j}) = 0$ and $\bigcup_{j \le 0} S_{\pi_j} = \emptyset = \chi^{\mathcal{S}}_\pi(-1)$.
- $[i = 1]$. Then, by Proposition 7.8 and the fact that $\chi^{\mathcal{S}}_\pi(0) = \emptyset$, we have

$$\begin{aligned} V^q_\pi(1) &= \sum\nolimits_{j=1}^{1} v^q(T_{\pi_j}) \\ &= v^q(T_{\pi_1}) \\ &= \left|S_{R_q} - \bigcup\nolimits_{j: T_{\pi_1} \prec T_j} S_j\right| \\ &= \left|\bigcup\nolimits_{j \le 1} S_{\pi_j} - \bigcup\nolimits_{j > 1} S_{\pi_j}\right| \\ &= \left|\bigcup\nolimits_{j \le 1} S_{\pi_j} - \bigcup\nolimits_{j > 1} S_{\pi_j}\right| - |\chi^{\mathcal{S}}_\pi(0)|. \end{aligned}$$

- $[i \ge 2]$. Then we have

$$V^q_\pi(i) = \sum_{j=1}^{i} v^q(T_{\pi_j}) = v^q(T_{\pi_i}) + \sum_{j=1}^{i-1} v^q(T_{\pi_j}) = v^q(T_{\pi_i}) + V^q_\pi(i-1) \tag{1}$$

by the definition of V^q. Applying the induction hypothesis and the definition of v^q (since $\pi_i \ne R_q$), we have

$$V^q_\pi(i) = \left|S_{\pi_i} - \bigcup\nolimits_{j: T_{\pi_i} \prec T_{\pi_j}} S_{\pi_j}\right| - 1 + \left|\bigcup\nolimits_{j \le i-1} S_{\pi_j} - \bigcup\nolimits_{j > i-1} S_{\pi_j}\right| - \left|\chi^{\mathcal{S}}_\pi(i-2)\right|. \tag{2}$$

Since $\sigma^{\pi,\mathcal{S}}$ is optimal, and $\mathcal{S}$ is linear exact, and $n \ge k$, the move $\sigma^{\pi,\mathcal{S}}_{i-1}$ must be a translocation emitting a new element, so $|\chi^{\mathcal{S}}_\pi(i-1)| = 1 + |\chi^{\mathcal{S}}_\pi(i-2)|$. Therefore,

$$V^q_\pi(i) = \left|S_{\pi_i} - \bigcup\nolimits_{j: T_{\pi_i} \prec T_{\pi_j}} S_{\pi_j}\right| + \left|\bigcup\nolimits_{j \le i-1} S_{\pi_j} - \bigcup\nolimits_{j > i-1} S_{\pi_j}\right| - \left|\chi^{\mathcal{S}}_\pi(i-1)\right|. \tag{3}$$

By Proposition 7.8,

$$V_\pi^q(i) = \left|S_{\pi_i} - \bigcup_{j>i} S_{\pi_j}\right| + \left|\bigcup_{j\leq i-1} S_{\pi_j} - \bigcup_{j>i-1} S_{\pi_j}\right| - \left|\chi_\pi^{\mathcal{S}}(i-1)\right|. \tag{4}$$

The first term of this expression is simply the number of elements that appear in S_{π_i} and never in $S_{\pi_{i+1}}, S_{\pi_{i+2}}, \ldots, S_{\pi_k}$. The second is the number of elements that appear in $S_{\pi_1}, S_{\pi_2}, \ldots, S_{\pi_{i-1}}$ and never in $S_{\pi_i}, S_{\pi_{i+1}}, \ldots, S_{\pi_k}$. We can simply combine these terms since these two sets are disjoint:

$$V_\pi^q(i) = \left|\bigcup_{j\leq i} S_{\pi_j} - \bigcup_{j>i} S_{\pi_j}\right| - \left|\chi_\pi^{\mathcal{S}}(i-1)\right|. \tag{5}$$

This proves the lemma. □

Lemma 7.11 *For all nested linear exact instances $\mathcal{S}$ with n elements, $k \leq n$ sets, and p components, there exists some $q \in \{1, 2, \ldots, p\}$ such that $\mathsf{opt}(T, \prec, v^q) = 0$.*

Proof. Let $\sigma^{\pi,\mathcal{S}}$ be an optimal move sequence solving $\mathcal{S} = S_1, S_2, \ldots, S_k$ such that $\pi_1 = R_q$ and π respects $\prec$, for some $q \in \{1, 2, \ldots p\}$. (One exists by Proposition 7.9.) For all $1 \leq i \leq k-1$, we have that $\chi_\pi^{\mathcal{S}}(i-1)$ is a subset of $\bigcup_{j\leq i} S_{\pi_j} - \bigcup_{j>i} S_{\pi_j}$, since only elements that do not appear in the remainder of the genome can be emitted. Thus $V_\pi^q(i) = |\bigcup_{j\leq i} S_{\pi_j} - \bigcup_{j>i} S_{\pi_j}| - |\chi_\pi^{\mathcal{S}}(i-1)| \geq 0$ for all $0 \leq i \leq k-1$, and $V_\pi^q(0) = 0$. Therefore $\mathsf{opt}(T, \prec, v^q) = 0$, which is optimal. □

7.2 From Minimum Loans to Linear Synteny

We now establish the converse: if one of our Min Loan instances has $\mathsf{opt}(T, \prec, v^q) = 0$, then our nested synteny instance $\mathcal{S}$ is linear exact.

Proposition 7.12 *Consider any permutation π of $(1, 2, \ldots, k)$ where $\pi_x = R_q$, and let $\pi' = (R_q, \pi_1, \pi_2, \ldots, \pi_{x-1}, \pi_{x+1}, \ldots, \pi_k)$ be π with R_q moved to the front. If $\min_i V_\pi^q(i) = 0$ and π respects $\prec$, then $\min_i V_{\pi'}^q(i) = 0$ and π' respects $\prec$.*

Proof. From the fact that $v^q(T_{R_q}) \geq 0$, the modification of π' can only increase $\min_i V_{\pi'}^q(i)$ with respect to $\min_i V_\pi^q(i)$. Furthermore, we have not violated any constraints, since $T_j \not\prec T_{R_q}$ for all j, so π' also respects $\prec$. □

Proposition 7.13 *For any instance $\mathcal{S} = S_1, S_2, \ldots, S_k$, for every π a permutation of $(1, 2, \ldots, k)$, and for all $1 \leq i \leq k-1$, we have*

$$x_\pi^{\mathcal{S}}(i) = 1 \quad \iff \quad \left|\bigcup_{j\leq i+1} S_{\pi_j} - \bigcup_{j>i+1} S_{\pi_j}\right| - \left|\chi_\pi^{\mathcal{S}}(i-1)\right| \quad \geq \quad 1.$$

Proof. Notice that move $\sigma_i^{\pi,\mathcal{S}}$ takes as input the set $S_{\pi_{i+1}}$ and the previous merging set $\bigcup_{j<i+1} S_{\pi_j} - \chi_\pi^{\mathcal{S}}(i-1)$, all previously merged elements less those that have already been emitted by translocation. By definition, we have

$$\begin{aligned}
x_\pi^{\mathcal{S}}(i) = 1 \quad &\iff \quad \exists b\left[b \in S_{\pi_{i+1}} \cup \left(\bigcup\nolimits_{j\le i} S_{\pi_j} - \chi_\pi^{\mathcal{S}}(i-1)\right) \text{ and } b \notin \bigcup\nolimits_{j>i+1} S_{\pi_j}\right] \\
&\iff \quad \exists b\left[b \in \left(\bigcup\nolimits_{j\le i+1} S_{\pi_j} - \chi_\pi^{\mathcal{S}}(i-1)\right) - \bigcup\nolimits_{j>i+1} S_{\pi_j}\right] \\
&\iff \quad \left|\bigcup\nolimits_{j\le i+1} S_{\pi_j} - \chi_\pi^{\mathcal{S}}(i-1) - \bigcup\nolimits_{j>i+1} S_{\pi_j}\right| \quad \ge \quad 1 \\
&\iff \quad \left|\bigcup\nolimits_{j\le i+1} S_{\pi_j} - \bigcup\nolimits_{j>i+1} S_{\pi_j}\right| - |\chi_\pi^{\mathcal{S}}(i-1)| \quad \ge \quad 1
\end{aligned}$$

since $\chi_\pi^{\mathcal{S}}(i-1)$ and $\bigcup_{j>i+1} S_{\pi_j}$ are disjoint and $\chi_\pi^{\mathcal{S}}(i-1) \subseteq \bigcup_{j\le i+1} S_{\pi_j}$. □

Lemma 7.14 *For all nested instances $\mathcal{S}$ with n elements, $k \le n$ sets, and p components with roots $R_1, R_2, \dots, R_p$, and for any $q \in \{1, 2, \dots, p\}$, if $\mathsf{opt}(T, \prec, v^q) = 0$ then $\mathcal{S}$ is linear exact. Furthermore, if π is a permutation of the tasks that achieves $\mathsf{opt}(T, \prec, v^q) = 0$ then $|\sigma^{\pi,\mathcal{S}}| = n - 1$.*

Proof. Let $\mathcal{S} = S_1, S_2, \dots, S_k$, and let π be a permutation of $(1, 2, \dots, k)$ with $\pi_1 = R_q$ that respects $\prec$ such that $\min_i V_\pi^q(i) = 0$. (One exists by Proposition 7.12.) We will show that $x_\pi^{\mathcal{S}}(i) = 1$ for arbitrary $1 \le i \le k-1$, which proves the theorem.

By the optimality of π, we know that $0 \le V_\pi^q(i+1) = \sum_{\ell=1}^{i+1} v^q(T_{\pi_\ell})$. By the definition of v^q and the fact that $\pi_1 = R_q$, we know that

$$P_\pi^q(i+1) \quad = \quad \left|S_{R_q} - \bigcup\nolimits_{j:T_{R_q}\prec T_{\pi_j}} S_{\pi_j}\right| + \sum_{\ell=2}^{i+1}\left(\left|S_{\pi_\ell} - \bigcup\nolimits_{j:T_{\pi_\ell}\prec T_{\pi_j}} S_{\pi_j}\right| - 1\right) \quad \ge \quad 0. \tag{6}$$

Rearranging, we have

$$-i + \sum_{\ell=1}^{i+1}\left|S_{\pi_\ell} - \bigcup\nolimits_{j:T_{\pi_\ell}\prec T_{\pi_j}} S_{\pi_j}\right| \quad \ge \quad 0. \tag{7}$$

By Proposition 7.7, we have $|\chi_\pi^{\mathcal{S}}(i-1)| \le i-1$. Applying Proposition 7.8 and this fact, we have

$$-1 - |\chi_\pi^{\mathcal{S}}(i-1)| + \sum_{\ell=1}^{i+1}\left|S_{\pi_\ell} - \bigcup\nolimits_{j>\ell} S_{\pi_j}\right| \quad \ge \quad 0. \tag{8}$$

The sets in the sum are simply the sets of all elements that appear for the last time in S_{π_ℓ}. These sets are disjoint, and can be rewritten as simply

$$\left|\bigcup\nolimits_{j\le i+1} S_{\pi_j} - \bigcup\nolimits_{j>i+1} S_{\pi_j}\right| - |\chi_\pi^{\mathcal{S}}(i-1)| - 1 \quad \ge \quad 0 \tag{9}$$

which by Lemma 7.13 gives us that $x_\pi^{\mathcal{S}}(i) = 1$. □

nested-linear-syntenic-distance($\mathcal{S}$)
// compute $\widetilde{D}(\mathcal{S})$ for any nested instance $\mathcal{S}$.

1. Remove any lonely singletons S_i from $\mathcal{S}$, and let $\mathcal{S}$ be the resulting instance.

   Let n, k, and p be the number of elements, sets, and components, respectively, in $\mathcal{S}$, and let the roots of the components be $R_1, R_2, \ldots, R_p$.

2. Let $T = \{T_1, T_2, \ldots, T_k, T_{\mathit{first}}, T_{\mathit{last}}\}$ be a set of tasks.

3. Let $\prec$ be the relation so that $T_{\mathit{first}} \prec T_{\mathit{last}}$, and, for all $1 \leq j \leq k$ and $1 \leq \ell \leq k$:

   - $T_j \prec T_{\mathit{last}}$;
   - $T_{\mathit{first}} \prec T_j$; and
   - $T_j \prec T_\ell$ if and only if $S_j \supset S_\ell$, or $S_j = S_\ell$ and $j \leq \ell$.

4. For each $1 \leq i \leq k$, let $v(T_i) := \left|\bigcup_{j:T_j \prec T_i} S_j - \bigcup_{j:T_i \prec T_j} S_j\right|$.

5. Let $\delta = \max(k - n, 0)$.

   For each $1 \leq q \leq p$:

   Binary search for the minimum x_q such that the instance $\mathcal{S}^{R_q \uparrow (\delta + x_q)}$ is linear exact, using the following decision procedure:

   (a) Let $v^q_{\delta+x_q}(T_j) := v(T_j) - 1$ for $j \neq R_q$, and let $v^q_{\delta+x_q}(T_{R_q}) := v(T_{R_q}) + \delta + x_q$. Let $v^q_{\delta+x_q}(T_{\mathit{first}}) := 0$ and $v^q_{\delta+x_q}(T_{\mathit{last}}) := 0$.

   (b) Return true if opt$(T, \prec, v^q_{\delta+x_q}) = 0$, computed using the series-parallel algorithm of either Abdel-Wahab and Kameda [1] or Monma and Sidney [19].

6. Return $n + \delta + \min_q(x_q) - 1$.

Figure 2: An algorithm for computing the linear syntenic distance of a nested instance.

7.3 An Algorithm for Nested Linear Synteny

We will make use of the Min Loan algorithms of Abdel-Wahab and Kameda [1] and Monma and Sidney [19] to determine $\widetilde{D}(\mathcal{S})$. Our algorithm is shown in Figure 2.

Theorem 7.15 *For all nested instances* $\mathcal{S}$*, we have* $\mathsf{nested\text{-}linear\text{-}syntenic\text{-}distance}(\mathcal{S}) = \widetilde{D}(\mathcal{S})$.

Proof. By definition, the deletion of lonely singletons does not affect the linear syntenic distance.

Note that for all $1 \le q \le p$ and for all x_q, the instance $\mathcal{S}^{R_q \uparrow (\delta + x_q)}$ is nested since we are adding completely fresh elements to the root of a component. Also observe that the precedence relation $\prec$ and the profit function $v^q_{\delta+x_q}$ meet the conditions of Definitions 7.2 and 7.4 for $\mathcal{S}^{R_q \uparrow (\delta + x_q)}$ as well as for $\mathcal{S}$. Note further that, by our choice of δ, the number of elements in $\mathcal{S}^{R_q \uparrow (\delta + x_q)}$ is no smaller than the number of sets.

Therefore, by Theorem 7.5, we have that the instance $\mathcal{S}^{R_q \uparrow (\delta + x_q)}$ is linear exact if and only if $\mathsf{opt}(T, \prec, v^q_{\delta+x_q}) = 0$. Since the precedence relation is series-parallel by Lemma 7.3, the series-parallel Min Loan algorithm correctly computes this value.

Thus, in Step 5, we find the smallest x_q so that $\mathcal{S}^{R_q \uparrow (\delta + x_q)}$ is linear exact for each $1 \le q \le p$. Let $x^* = \delta + \min_q x_q$. We know that

$$\exists q \in \{1, 2, \ldots, p\}\ \widetilde{D}(\mathcal{S}^{R_q \uparrow x^*}) = n + x^* - 1 \qquad \forall q \in \{1, 2, \ldots, p\}\ \widetilde{D}(\mathcal{S}^{R_q \uparrow x^*}) \ne n + x^* - 2 \tag{10}$$

By Proposition 4.3 and Corollary 5.4, then, we have $\widetilde{D}(\mathcal{S}) = n + x^* - 1 = n + \delta + \min_q x_q - 1 = \mathsf{nested\text{-}linear\text{-}syntenic\text{-}distance}(\mathcal{S})$. □

Theorem 7.16 *On a nested instance* $\mathcal{S}$ *with* n *elements,* k *sets, and* p *components, the algorithm* $\mathsf{nested\text{-}linear\text{-}syntenic\text{-}distance}(\mathcal{S})$ *requires* $O(pk^2 \log k + nk^2)$ *time.*

Proof. Removing any lonely singletons and computing $\prec$ and $v(T_j)$ in Steps 3 and 4 requires $O(nk^2)$ time since we must compute k^2 pairwise intersections of up to n elements.

In Step 5, the binary search requires at most $\log k$ iterations because we know that $x_q \le k$ for all $1 \le q \le p$; the Min Loan calls require $O(k^2)$ time each since there are $O(k)$ events in question, and we must run this search for each of the p components of the instance. □

By linear duality, we can also use the above algorithm to compute the linear syntenic distance of an instance whose dual is nested.

It is straightforward to modify $\mathsf{nested\text{-}linear\text{-}syntenic\text{-}distance}(\mathcal{S})$ to produce an optimal linear move sequence for $\mathcal{S}$, instead of just $\widetilde{D}(\mathcal{S})$. Using the Abdel-Waheb and Kameda [1] algorithm for Min Loan, we can acquire a permutation π of the tasks so that the optimum $\mathsf{opt}(T, \prec, v^q_{\delta+x_q})$ is achieved by completing the tasks in the order π. By Proposition 7.12, this optimum is also achieved by a permutation π'—computable from π in $O(k)$ time—in which $\pi'_1 = R_q$ for some root R_q. By Lemma 7.14, the instance $\mathcal{S}^{R_q \uparrow x^*}$ is linear exact, and we have

$$\widetilde{D}(\mathcal{S}^{R_q \uparrow x^*}) \;=\; |\sigma^{\pi', \mathcal{S}^{R_q \uparrow x^*}}| \;=\; n + x^* - 1 \;=\; \widetilde{D}(\mathcal{S}).$$

By Lemma 3.1 and the fact that $\mathcal{S}^{R_q \uparrow x^*}$ dominates $\mathcal{S}$, we have $\widetilde{D}(\mathcal{S}) = |\sigma^{\pi', \mathcal{S}^{R_q \uparrow x^*}}| \ge |\sigma^{\pi', \mathcal{S}}|$. Thus $\sigma^{\pi', \mathcal{S}}$ is optimal.

8 Acknowledgements

We would like to thank Anne Bergeron, Matt Lepinski, Chris Peikert, and Grant Wang for comments on previous drafts of this paper, and the anonymous referee for helpful comments and suggestions.

The first author was supported in part by a Churchill Scholarship from the Winston Churchill Foundation, an NSF Graduate Research Fellowship, and the ONR Young Investigator Award of the second author. The majority of this work was performed while at Cornell University and the University of Cambridge.

The second author was supported in part by a David and Lucile Packard Foundation Fellowship, an Alfred P. Sloan Research Fellowship, an ONR Young Investigator Award, and NSF Faculty Early Career Development Award CCR-9701399.

References

[1] H. M. Abdel-Wahab and T. Kameda. Scheduling to minimize maximum cumulative costs subject to series-parallel precedence constraints. *Operations Research*, 26(1):141–158, January/February 1978.

[2] Vineet Bafna and Pavel A. Pevzner. Genome rearrangements and sorting by reversals. *SIAM J. Comput.*, 25(2):272–289, April 1996.

[3] Vineet Bafna and Pavel A. Pevzner. Sorting by transpositions. *SIAM J. Discrete Math.*, 11(2):224–240, May 1998.

[4] Piotr Berman and Sridhar Hannenhalli. Fast sorting by reversals. In *7th Annual Conference on Combinatorial Pattern Matching*, pages 168–185, 1996.

[5] Piotr Berman, Sridhar Hannenhalli, and Marek Karpinski. 1.375-approximation algorithm for sorting by reversals. *Electronic Colloquium on Computational Complexity (ECCC)*, 8(47), 2001.

[6] Piotr Berman and Marek Karpinski. On some tighter inapproximability results. *Electronic Colloquium on Computational Complexity*, Report No. 29, 1998.

[7] Alberto Caprara. Sorting permutations by reversals and Eulerian cycle decompositions. *SIAM J. Discrete Math.*, 12(1):91–110, February 1999.

[8] D. A. Christie. A 3/2-approximation algorithm for sorting by reversals. In *9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 244–252, January 1998.

[9] Bhaskar DasGupta, Tao Jiang, Sampath Kannan, Ming Li, and Elizabeth Sweedyk. On the complexity and approximation of syntenic distance. *Discrete Appl. Math.*, 88(1–3):59–82, November 1998.

[10] Jason Ehrlich, David Sankoff, and Joseph H. Nadeau. Synteny conservation and chromosome rearrangements during mammalian evolution. *Genetics*, 147(1):289–296, September 1997.

[11] Niklas Erikson. $(1+\varepsilon)$-approximation of sorting by reversals and transpositions. In *1st Annual Workshop on Algorithms in Bioinformatics*, pages 227–237, August 2001.

[12] Henrik Eriksson, Kimmo Eriksson, Johan Karlander, Lars Svensson, and Johan Wästlund. Sorting a bridge hand. *Discrete Math.*, 241(1–3):289–300, October 2001.

[13] Vincent Ferretti, Joseph H. Nadeau, and David Sankoff. Original synteny. In *7th Annual Symposium on Combinatorial Pattern Matching*, pages 159–167, June 1996.

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

[15] Qian-Ping Gu, Shietung Peng, and Ivan Hal Sudborough. A 2-approximation algorithm for genome rearrangements by reversals and transpositions. *Theoret. Comp. Sci.*, 210(2):327–339, January 1999.

[16] Sridhar Hannenhalli and Pavel A. Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM*, 46(1):1–27, January 1999.

[17] J. D. Kececioglu and D. Sankoff. Exact and approximation algorithms for sorting by reversals, with application to genome rearrangment. *Algorithmica*, 13(1/2):180–210, January/February 1995.

[18] David Liben-Nowell. On the structure of syntenic distance. *J. Comp. Bio.*, 8(1):53–67, February 2001.

[19] C. L. Monma and J. B. Sidney. A general algorithm for optimal job sequencing with series-parallel precedence constraints. Technical Report 347, School of Operations Research, Cornell University, 1977.

[20] David Sankoff and Joseph H. Nadeau. Conserved synteny as a measure of genomic distance. *Discrete Appl. Math.*, 71(1–3):247–257, December 1996.