String Consensus Problems with Swaps and Substitutions

SPIRE 2025

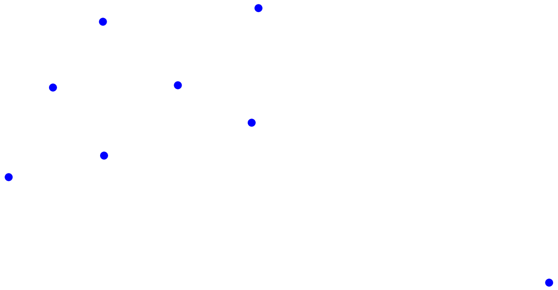
Estéban Gabory[1], Laurent Bulteau[2], Gabriele Fici[1], Hilde Verbeek[3]

[1]Università di Palermo, Italy [2]LIGM, CNRS, Université Gustave Eiffel, France
[3]CWI, Amsterdam, The Netherlands

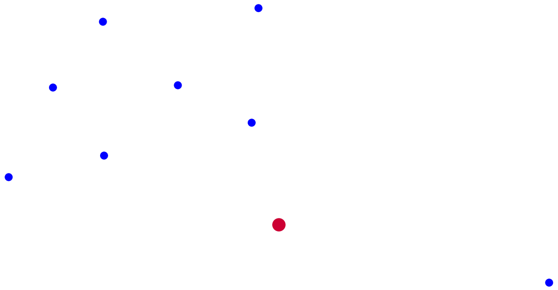
September 8th, 2025

Outline

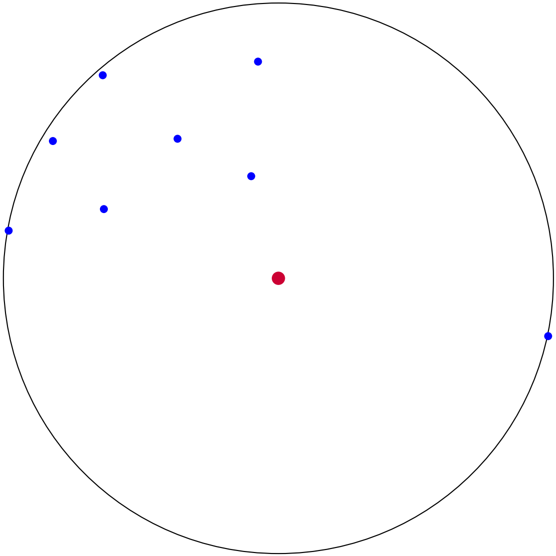
Smallest enclosing circle, Geometric median



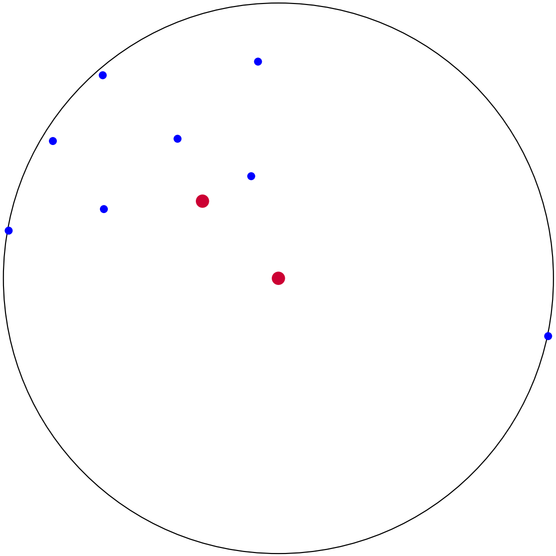
Smallest enclosing circle, Geometric median



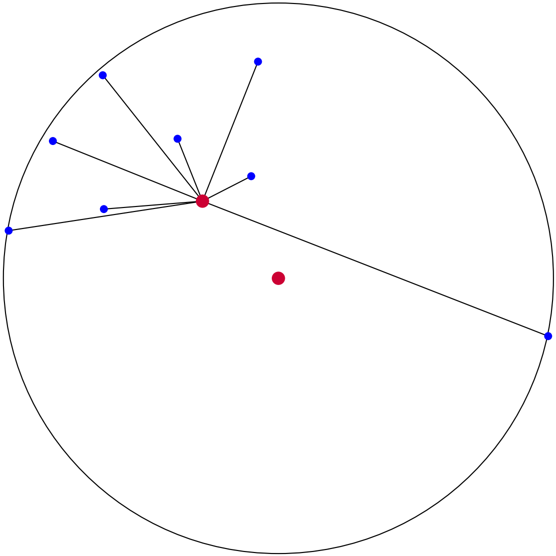
Smallest enclosing circle, Geometric median



Smallest enclosing circle, Geometric median



Smallest enclosing circle, Geometric median



Consensus String: Problem definition

Let us fix a distance ∂ on strings.

Consensus String: Problem definition

Let us fix a distance ∂ on strings.

$(\mathtt{r},\partial)$-CONSENSUS
Input: A set $\mathcal{S}$ of fixed length strings, integer d.
Output:

Consensus String: Problem definition

Let us fix a distance ∂ on strings.

$(\mathtt{r},\partial)$-Consensus
Input: A set $\mathcal{S}$ of fixed length strings, integer d.
Output: Is there a string s^* such that $\max\limits_{s\in\mathcal{S}} \partial(s, s^*) \leq d$?

Consensus String: Problem definition

Let us fix a distance ∂ on strings.

$(\mathtt{r},\partial)$-Consensus

Input: A set $\mathcal{S}$ of fixed length strings, integer d.

Output: Is there a string s^* such that $\max_{s\in\mathcal{S}} \partial(s, s^*) \leq d$?

$(\mathtt{s},\partial)$-Consensus

Input: A set $\mathcal{S}$ of fixed length strings, integer D.

Output: Is there a string s^* such that $\sum_{s\in\mathcal{S}} \partial(s, s^*) \leq D$?

Hamming distance

Definition (Hamming distance)

We write $\partial_{\mathtt{Ham}}$ for the Hamming distance, counting mismatches.

Hamming distance

Definition (Hamming distance)

We write $\partial_{\mathtt{Ham}}$ for the Hamming distance, counting mismatches.

Example

$\mathtt{s} = "\mathtt{Natural}"$

$\mathtt{t} = "\mathtt{Neutral}"$

Hamming distance

Definition (Hamming distance)

We write $\partial_{\mathtt{Ham}}$ for the Hamming distance, counting mismatches.

Example

$\mathtt{s} = "\mathtt{Natural}"$

$\mathtt{t} = "\mathtt{Neutral}"$

$\mathtt{d_H(s,t) = 3}$

Example

$$s_1 = \text{"}\texttt{adcda}\text{"}$$
$$s_2 = \text{"}\texttt{abcda}\text{"}$$
$$s_3 = \text{"}\texttt{adabb}\text{"}$$
$$s_4 = \text{"}\texttt{cdcda}\text{"}$$
$$s_5 = \text{"}\texttt{fffff}\text{"}$$

Example

$$s_1 = \texttt{"adcda"}$$

$$s_2 = \texttt{"abcda"}$$

$$s_3 = \texttt{"adabb"}$$

$$s_4 = \texttt{"cdcda"}$$

$$s_5 = \texttt{"fffff"}$$

$$s^* = \texttt{"adcda"} \quad d = 5 \; D = 11$$

Example

$$s_1 = \text{"adcda"}$$
$$s_2 = \text{"abcda"}$$
$$s_3 = \text{"adabb"}$$
$$s_4 = \text{"cdcda"}$$
$$s_5 = \text{"fffff"}$$

$$s^* = \text{"adcda"} \quad d = 5 \; D = 11$$
$$\hat{s} = \text{"adcff"} \quad d = 3 \; D = 14$$

Example

$$s_1 = \text{"adcda"}$$
$$s_2 = \text{"abcda"}$$
$$s_3 = \text{"adabb"}$$
$$s_4 = \text{"cdcda"}$$
$$s_5 = \text{"fffff"}$$

$$s^* = \text{"adcda"} \quad d = 5 \; D = 11$$
$$\hat{s} = \text{"adcff"} \quad d = 3 \; D = 14$$

In genreal, $(\mathtt{s},\partial)$-CONSENSUS is *easier* than $(\mathtt{r},\partial)$-CONSENSUS. For example, $(\mathtt{r},\partial_{\mathtt{Ham}}) -$ CONSENSUS is NP-hard, while $(\mathtt{s},\partial_{\mathtt{Ham}}) -$ CONSENSUS can be solved in linear time (majority string).

Outline

NP-hardness

Theorem (Frances and Litman, 1997)

The problem $(\mathtt{r}, \partial_{\mathtt{Ham}}) - \textsc{Consensus}$ *is* NP*-hard under the Hamming distance*

NP-hardness

Theorem (Frances and Litman, 1997)

The problem $(\mathtt{r},\partial_{\mathtt{Ham}}) - \textsc{Consensus}$ *is* NP*-hard under the Hamming distance*

Definition

We say that a problem is *FPT* for a parameter k if an instance of size n can be solved in $O(f(k) \cdot p(n,k))$ time, where f is some computable function and p is a polynomial.

NP-hardness

Theorem (Frances and Litman, 1997)

The problem $(\mathtt{r},\partial_{\mathtt{Ham}}) - \textsc{Consensus}$ *is* NP*-hard under the Hamming distance*

Definition

We say that a problem is *FPT* for a parameter k if an instance of size n can be solved in $O(f(k) \cdot p(n,k))$ time, where f is some computable function and p is a polynomial.

Theorem (Gramm et al., 2003)

$(\mathtt{r},\partial_{\mathtt{Ham}})$-$\textsc{Consensus}$ *is* FPT *with respect to the maximal distance*

Outline

Swap distance: definition

The *swap distance* $\partial_{\mathrm{S}}(s, t)$ between two strings s, t having the same length is the number of adjacent letter exchanges to obtain t from s. Each position can be swapped at most once.

Swap distance: definition

The *swap distance* $\partial_{\mathrm{S}}(s, t)$ between two strings s, t having the same length is the number of adjacent letter exchanges to obtain t from s. Each position can be swapped at most once.
Note that not every pair of strings is comparable. If two strings are comparable, we say that they are *matching*.

Swap distance: definition

The *swap distance* $\partial_{\mathrm{S}}(s,t)$ between two strings s,t having the same length is the number of adjacent letter exchanges to obtain t from s. Each position can be swapped at most once.
Note that not every pair of strings is comparable. If two strings are comparable, we say that they are *matching*.

Example

$\mathrm{s} = \mathrm{"ababc"}$

$\mathrm{t} = \mathrm{"babac"}$

Swap distance: definition

The *swap distance* $\partial_{\mathrm{S}}(s,t)$ between two strings s, t having the same length is the number of adjacent letter exchanges to obtain t from s. Each position can be swapped at most once.
Note that not every pair of strings is comparable. If two strings are comparable, we say that they are *matching*.

Example

$\mathtt{s} = "\mathtt{ababc}"$

$\mathtt{t} = "\mathtt{babac}"$

Swap distance: definition

The *swap distance* $\partial_{\mathrm{S}}(s, t)$ between two strings s, t having the same length is the number of adjacent letter exchanges to obtain t from s. Each position can be swapped at most once.
Note that not every pair of strings is comparable. If two strings are comparable, we say that they are *matching*.

Example

$\mathtt{s} = "\mathtt{ababc}"$

$\mathtt{t} = "\mathtt{babac}"$

Swap distance: definition

The *swap distance* $\partial_{\mathrm{S}}(s,t)$ between two strings s,t having the same length is the number of adjacent letter exchanges to obtain t from s. Each position can be swapped at most once.
Note that not every pair of strings is comparable. If two strings are comparable, we say that they are *matching*.

Example

$\mathtt{s} = "\mathtt{ababc}"$

$\mathtt{t} = "\mathtt{babac}"$

$\partial_{\mathrm{S}}(\mathtt{s},\mathtt{t}) = 2$

Swap distance: strings not matching

Even with the same letter frequencies, some pairs of strings might not match.

Swap distance: strings not matching

Even with the same letter frequencies, some pairs of strings might not match.

Example

$s = "abc"$

$t = "bca"$

Known results on $(\mathtt{r},\partial_{\mathrm{S}}) - \textsc{Consensus}$

Theorem (Amir et al., 2013)

$(\mathtt{r},\partial_{\mathrm{S}})$-$\textsc{Consensus}$ *is* NP *hard.*

Known results on $(\mathtt{r},\partial_{\mathrm{S}})-$ Consensus

Theorem (Amir et al., 2013)

$(\mathtt{r},\partial_{\mathrm{S}})$-Consensus *is* NP *hard.*

Problem

1. *Is* $(\mathtt{r},\partial_{\mathrm{S}})$-Consensus FPT *for d ?*
2. *Is* $(\mathtt{s},\partial_{\mathrm{S}})$-Consensus *linear? or at least polynomial ?*

First case: strings are pairwise matching

$$s_1 = \mathtt{abgacbahidabedfeda}$$
$$s_2 = \mathtt{bagacbaihdabedfdea}$$
$$s_3 = \mathtt{bagacbaihdbaedfeda}$$

First case: strings are pairwise matching

$$s_1 = \texttt{abgacbahidabedfeda}$$

$$s_2 = \texttt{bagacbaihdabedfdea}$$

$$s_3 = \texttt{bagacbaihdbaedfeda}$$

First case: strings are pairwise matching

$$s_1 = \mathtt{abgacbahidabedfeda}$$

$$s_2 = \mathtt{bagacbaihdabedfdea}$$

$$s_3 = \mathtt{bagacbaihdbaedfeda}$$

First case: strings are pairwise matching

$$h_1 = 00000000000000000.$$

$$s_2 = \mathtt{bagacbaihdabedfdea}$$

$$s_3 = \mathtt{bagacbaihdbaedfeda}$$

First case: strings are pairwise matching

$$\mathtt{h}_1 = 00000000000000000.$$

$$\mathtt{h}_2 = 10000001000000010.$$

$$\mathtt{s}_3 = \mathtt{bagacbaihdbaedfeda}$$

First case: strings are pairwise matching

$$\mathtt{h}_1 = \mathtt{00000000000000000}.$$

$$\mathtt{h}_2 = \mathtt{10000001000000010}.$$

$$\mathtt{h}_3 = \mathtt{10000001001000000}.$$

First case: strings are pairwise matching

$$\mathtt{h}_1 = \mathtt{00000000000000000}.$$
$$\mathtt{h}_2 = \mathtt{10000001000000010}.$$
$$\mathtt{h}_3 = \mathtt{10000001001000000}.$$
$$\mathtt{h}^* = \mathtt{10000001000000000}.$$

First case: strings are pairwise matching

$$\mathtt{s}_1 = \mathtt{abgacbahidabedfeda}$$
$$\mathtt{s}_2 = \mathtt{bagacbaihdabedfdea}$$
$$\mathtt{s}_3 = \mathtt{bagacbaihdbaedfeda}$$
$$\mathtt{h}^* = 100000010000000000.$$

First case: strings are pairwise matching

$$s_1 = \mathtt{abgacbahidabedfeda}$$

$$s_2 = \mathtt{bagacbaihdabedfdea}$$

$$s_3 = \mathtt{bagacbaihdbaedfeda}$$

$$s^* = \mathtt{bagacbaihdabedfeda}$$

Second case: Strings are not pairwise matching

Strings do not need to be pairwise matching to admit a consensus

Second case: Strings are not pairwise matching

Strings do not need to be pairwise matching to admit a consensus

Example

$\mathtt{s}_1 = \mathtt{bac}$

$\mathtt{s}_2 = \mathtt{acb}$

$\mathtt{s}^* = \mathtt{abc}$

Second case: Strings are not pairwise matching

Strings do not need to be pairwise matching to admit a consensus

Example

$s_1 = \mathtt{bac}$

$s_2 = \mathtt{acb}$

$s^* = \mathtt{abc}$

However, if the strings are not pairwise matching, the consensus must match each of them. This provides us with some information about a potential solution.

Second case: Strings are not pairwise matching (2)

We write $\mathcal{C}_{\mathcal{S}}$ for the strings matching with every string in $\mathcal{S}$.

Second case: Strings are not pairwise matching (2)

We write $\mathcal{C}_{\mathcal{S}}$ for the strings matching with every string in $\mathcal{S}$. We say that a swap at position i is *necessary* for $s \in \mathcal{S}$ if there is a swap at position i between s and s^* for *every* $s^* \in \mathcal{S}$.

Second case: Strings are not pairwise matching (2)

We write $\mathcal{C}_{\mathcal{S}}$ for the strings matching with every string in $\mathcal{S}$. We say that a swap at position i is *necessary* for $s \in \mathcal{S}$ if there is a swap at position i between s and s^* for *every* $s^* \in \mathcal{S}$.
Main idea: Necessary swaps can be deduced from the structure of $\mathcal{S}$. After doing all those necessary swaps, we obtain a set $\mathcal{S}'$ whose strings are pairwise compatible.

Second case: Strings are not pairwise matching (2)

We write $\mathcal{C}_{\mathcal{S}}$ for the strings matching with every string in $\mathcal{S}$. We say that a swap at position i is *necessary* for $s \in \mathcal{S}$ if there is a swap at position i between s and s^* for *every* $s^* \in \mathcal{S}$.
Main idea: Necessary swaps can be deduced from the structure of $\mathcal{S}$. After doing all those necessary swaps, we obtain a set $\mathcal{S}'$ whose strings are pairwise compatible. We then use the previous reduction to the Hamming case.

Disentanglement: example

$$s^* = ?????????$$
$$s_1 = \mathtt{abgabcahi}$$
$$s_2 = \mathtt{agbcaabih}$$
$$s_3 = \mathtt{abgcabaih}$$

Disentanglement: example

$$s^* = ?????????$$
$$s_1 = \boxed{a}bgabcahi$$
$$s_2 = \boxed{a}gbcaabih$$
$$s_3 = \boxed{a}bgcabaih$$

Disentanglement: example

$$s^* = \mathtt{a????????}$$
$$s_1 = \boxed{\mathtt{a}}\mathtt{bgabcahi}$$
$$s_2 = \boxed{\mathtt{a}}\mathtt{gbcaabih}$$
$$s_3 = \boxed{\mathtt{a}}\mathtt{bgcabaih}$$

Disentanglement: example

$$s^* = \mathtt{a????????}$$
$$s_1 = \mathtt{a}\boxed{\mathtt{b}}\mathtt{gabcahi}$$
$$s_2 = \mathtt{a}\boxed{\mathtt{g}}\mathtt{bcaabih}$$
$$s_3 = \mathtt{a}\boxed{\mathtt{b}}\mathtt{gcabaih}$$

Disentanglement: example

$$
\begin{aligned}
\mathtt{s}^* &= \mathtt{a????????} \\
\mathtt{s}_1 &= \mathtt{a}\boxed{\mathtt{bg}}\mathtt{abcahi} \\
\mathtt{s}_2 &= \mathtt{a}\boxed{\mathtt{gb}}\mathtt{caabih} \\
\mathtt{s}_3 &= \mathtt{a}\boxed{\mathtt{bg}}\mathtt{cabaih}
\end{aligned}
$$

Disentanglement: example

$$
\begin{aligned}
\mathtt{s}^* &= \mathtt{a????????}\\
\mathtt{s}_1 &= \mathtt{abg\boxed{ab}cahi}\\
\mathtt{s}_2 &= \mathtt{agb\boxed{ca}abih}\\
\mathtt{s}_3 &= \mathtt{abg\boxed{ca}baih}
\end{aligned}
$$

Disentanglement: example

$$
\begin{aligned}
s^* &= \mathtt{a????????} \\
s_1 &= \mathtt{abga}\boxed{\mathtt{ab}}\mathtt{cahi} \\
s_2 &= \mathtt{agb}\boxed{\mathtt{ca}}\mathtt{abih} \\
s_3 &= \mathtt{abg}\boxed{\mathtt{ca}}\mathtt{baih}
\end{aligned}
$$

Disentanglement: example

$$s^* = \mathtt{a??\boxed{???}???}$$

$$s_1 = \mathtt{abgabcahi}$$

$$s_2 = \mathtt{agbcaabih}$$

$$s_3 = \mathtt{abgcabaih}$$

Disentanglement: example

$$s^* = \mathtt{a??\boxed{???}???}$$

$$s_1 = \mathtt{abgab cahi}$$

$$s_2 = \mathtt{agbcaabih}$$

$$s_3 = \mathtt{abgcabaih}$$

Disentanglement: example

$$s^* = \texttt{a??}\boxed{\texttt{???}}\texttt{???}$$

$$s_1 = \texttt{abgabcahi}$$

$$s_2 = \texttt{agbcaabih}$$

$$s_3 = \texttt{abgcabaih}$$

Disentanglement: example

$$s^* = \texttt{a??}\boxed{\texttt{???}}\texttt{???}$$
$$s_1 = \texttt{abgabcahi}$$
$$s_2 = \texttt{agbcaabih}$$
$$s_3 = \texttt{abgcabaih}$$

Disentanglement: example

$$s^* = \mathtt{a???}\color{red}{\mathtt{cb}}\mathtt{???}$$

$$s'_1 = \mathtt{abga}\color{red}{\mathtt{cb}}\mathtt{ahi}$$

$$s_2 = \mathtt{agbcaabih}$$

$$s_3 = \mathtt{abgcabaih}$$

Disentanglement: example

$$s^* = \texttt{a???}\textcolor{red}{\texttt{c}}\texttt{b???}$$
$$s_1' = \texttt{abgacbahi}$$
$$s_2 = \texttt{agbc}\textcolor{red}{\texttt{a}}\texttt{abih}$$
$$s_3 = \texttt{abgc}\textcolor{red}{\texttt{a}}\texttt{baih}$$

Disentanglement: example

$$s^* = \texttt{a??}\texttt{ac}\texttt{b???}$$
$$s'_1 = \texttt{abgacbahi}$$
$$s'_2 = \texttt{agb}\texttt{ac}\texttt{abih}$$
$$s'_3 = \texttt{abg}\texttt{ac}\texttt{baih}$$

Disentanglement: example

$$s^* = \texttt{a??ac}\textcolor{red}{\texttt{b}}\texttt{???}$$
$$s'_1 = \texttt{abgacbahi}$$
$$s'_2 = \texttt{agbac}\textcolor{red}{\texttt{a}}\texttt{bih}$$
$$s'_3 = \texttt{abgacbaih}$$

Disentanglement: example

$$s^* = \texttt{a??ac}\textcolor{red}{\texttt{ba}}\texttt{??}$$
$$s'_1 = \texttt{abgacbahi}$$
$$s''_2 = \texttt{agbac}\textcolor{red}{\texttt{ba}}\texttt{ih}$$
$$s'_3 = \texttt{abgacbaih}$$

Disentanglement: example

$$s^* = \mathtt{a??acba??}$$
$$s'_1 = \mathtt{abgacba}\boxed{\mathtt{hi}}$$
$$s''_2 = \mathtt{agbacba}\boxed{\mathtt{ih}}$$
$$s'_3 = \mathtt{abgacba}\boxed{\mathtt{ih}}$$

Disentanglement: example

$$s'_1 = \texttt{abgacbahi} \qquad 1 \text{ swap}$$
$$s''_2 = \texttt{agbacbaih} \qquad 2 \text{ swap}$$
$$s'_3 = \texttt{abgacbaih} \qquad 1 \text{ swap}$$

Disentanglement: example

$$h_1 = 00000000 \qquad 1 \text{ swap}$$

$$h_2 = 01000001 \qquad 2 \text{ swap}$$

$$h_3 = 00000001 \qquad 1 \text{ swap}$$

Disentanglement: example

$$\mathtt{h}^* = \mathtt{01000001}$$

$$\mathtt{h}_1 = \mathtt{00000000} \qquad 3 \text{ swap}$$

$$\mathtt{h}_2 = \mathtt{01000001} \qquad 2 \text{ swap}$$

$$\mathtt{h}_3 = \mathtt{00000001} \qquad 2 \text{ swap}$$

Disentanglement: example

$$\mathtt{h}^* = \mathtt{01000001}$$

$$\mathtt{h}_1 = \mathtt{00000000}$$ 3 swap

$$\mathtt{h}_2 = \mathtt{01000001}$$ 2 swap

$$\mathtt{h}_3 = \mathtt{00000001}$$ 2 swap

Swap: Main theorem

Theorem

We have the following:

- $(\texttt{s}, \partial_{\texttt{S}})$-$\textsc{Consensus}$ *can be solved in* $\mathcal{O}(kn)$ *time.*
- $(\texttt{r}, \partial_{\texttt{S}}) - \textsc{Consensus}(d) \in \mathsf{FPT}$.

Outline

Problem definition

Closest string under Hamming distance

Algorithms for the swap distance

Algorithms for the SH distance

SH distance

Definition

Let s, t be two strings having the same length. The *SH* (Swap-Hamming) distance $\partial_{\mathtt{SH}}(s,t)$ between s and t is defined as following :

$$\partial_{\mathtt{SH}}(s,t) = \min_{\sigma\in\mathcal{S}}(\partial_{\mathtt{S}}(s,\sigma s) + \partial_{\mathtt{Ham}}(\sigma s,t))$$

Where $\mathcal{S}$ is the set of swap transformations starting from s.

Proposition

$\partial_{\mathtt{SH}}(s,t)$ can be computed greedily, from left to right.

SH distance

Proposition

$\partial_{\mathtt{SH}}(s, t)$ can be computed greedily, from left to right.

Example

$$\mathtt{s}_1 = \text{"}\mathtt{natural}\text{"}$$

$$\mathtt{s}_2 = \text{"}\mathtt{neutral}\text{"}$$

$\partial_{\mathtt{SH}}(\mathtt{s}_1, \mathtt{s}_2) =$

SH distance

Proposition

$\partial_{\mathrm{SH}}(s, t)$ can be computed greedily, from left to right.

Example

$$s_1 = \text{"}\mathtt{natural}\text{"}$$

$$s_2 = \text{"}\mathtt{neutral}\text{"}$$

$$\partial_{\mathrm{SH}}(s_1, s_2) =$$

SH distance

Proposition

$\partial_{\mathrm{SH}}(s, t)$ can be computed greedily, from left to right.

Example

$$\mathtt{s}_1 = \text{"}\mathtt{natural}\text{"}$$

$$\mathtt{s}_2 = \text{"}\mathtt{neutral}\text{"}$$

$\partial_{\mathrm{SH}}(\mathtt{s}_1, \mathtt{s}_2) = 1$

SH distance

Proposition

$\partial_{\mathrm{SH}}(s, t)$ can be computed greedily, from left to right.

Example

$$\mathtt{s}_1 = \text{"}\mathtt{natural}\text{"}$$

$$\mathtt{s}_2 = \text{"}\mathtt{neutral}\text{"}$$

$\partial_{\mathrm{SH}}(\mathtt{s}_1, \mathtt{s}_2) = 2$

SH distance

Proposition

$\partial_{\mathrm{SH}}(s, t)$ can be computed greedily, from left to right.

Example

$$\mathtt{s}_1 = \text{"}\mathtt{natural}\text{"}$$

$$\mathtt{s}_2 = \text{"}\mathtt{neutral}\text{"}$$

$\partial_{\mathrm{SH}}(\mathtt{s}_1, \mathtt{s}_2) = 2$

Problem

Is $(\mathtt{r}, \partial_{\mathrm{SH}})$-Consensus FPT *for d?*

Sketch of the proof from Gramm et al., 2003

$$s_1 =$$

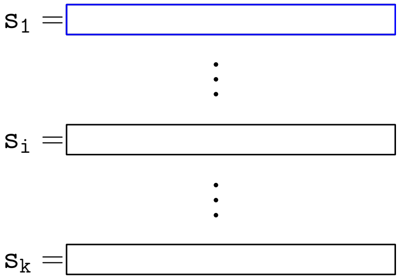
$$\vdots$$

$$s_i =$$

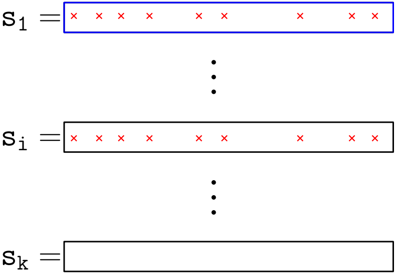
$$\vdots$$

$$s_k =$$

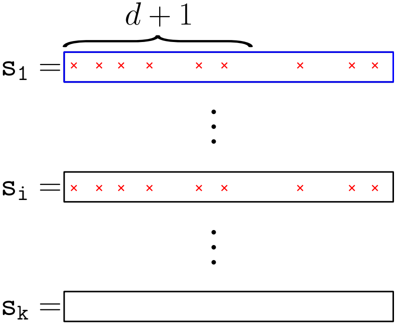
Sketch of the proof from Gramm et al., 2003



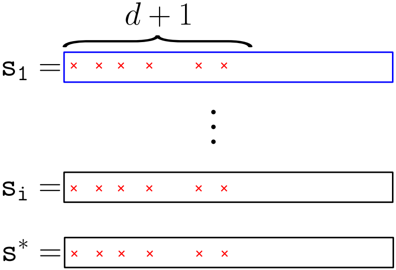
Sketch of the proof from Gramm et al., 2003



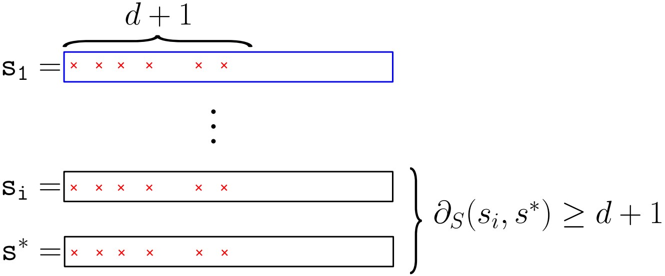
Sketch of the proof from Gramm et al., 2003



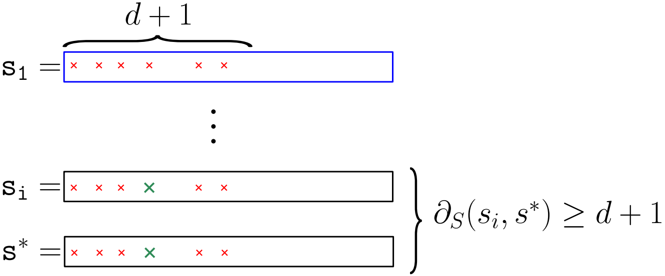
Sketch of the proof from Gramm et al., 2003



Sketch of the proof from Gramm et al., 2003



Sketch of the proof from Gramm et al., 2003



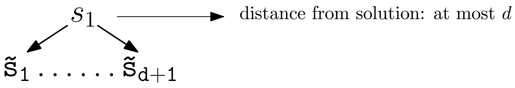
Sketch of the proof from Gramm et al., 2003

s_1

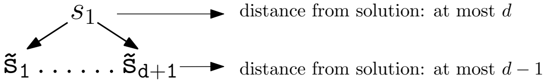
Sketch of the proof from Gramm et al., 2003

$$\begin{tikzcd}[column sep=small] & s_1 \arrow[dl] \arrow[dr] & \\ \tilde{\mathtt{s}}_1 & \cdots\cdots & \tilde{\mathtt{s}}_{\mathtt{d}+1} \end{tikzcd}$$

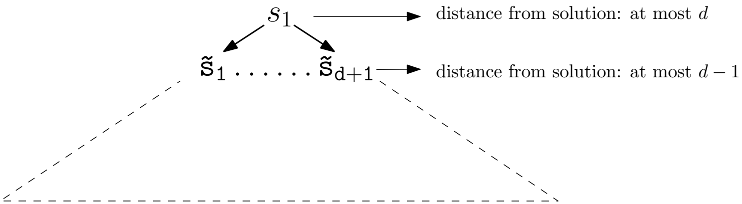
Sketch of the proof from Gramm et al., 2003



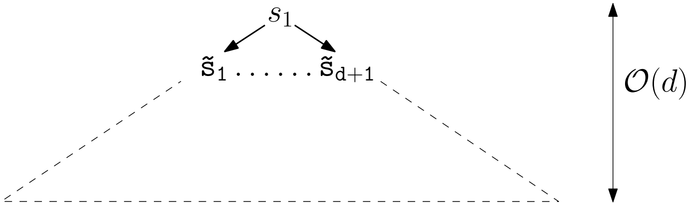
Sketch of the proof from Gramm et al., 2003



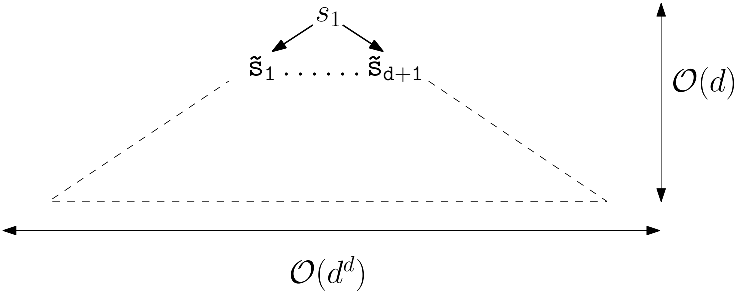
Sketch of the proof from Gramm et al., 2003



Sketch of the proof from Gramm et al., 2003



Sketch of the proof from Gramm et al., 2003



$(\mathtt{r},\partial_{\mathtt{SH}})$-Consensus

Theorem

$(\mathtt{r},\partial_{\mathtt{SH}}) - \textsc{Consensus}(d) \in \mathsf{FPT}$.

$(\mathtt{r},\partial_{\mathtt{SH}})$-Consensus

Theorem

$(\mathtt{r},\partial_{\mathtt{SH}}) - \textsc{Consensus}(d) \in \mathsf{FPT}$.

Proof.

Same as for Hamming in [Frances and Litman, 1997], but we also try some candidates obtained by swapping positions. □

$(\mathtt{s}\,,\partial_{\mathrm{SH}})$-Consensus

Problem

Is $(\mathtt{s}\,,\partial_{\mathrm{SH}})$-Consensus *linear? or at least polynomial ?*

$(\mathtt{s}, \partial_{\mathtt{SH}})$-CONSENSUS

$$s_1 = \mathtt{baba}$$
$$s_2 = \mathtt{cabc}$$
$$s_3 = \mathtt{abca}$$

$(\mathtt{s}, \partial_{\mathtt{SH}})$-Consensus

$$s_1 = \mathtt{baba}$$

$$s_2 = \mathtt{cabc}$$

$$s_3 = \mathtt{baca}$$

$$s^* = \mathtt{abab}$$

$(\mathtt{s}, \partial_{\mathtt{SH}})$-Consensus

$$s_1 = \mathtt{baba}$$

$$s_2 = \mathtt{cabc}$$

$$s_3 = \mathtt{baca}$$

$$s^* = \mathtt{abab}$$

$$\sum \partial_{SH}(s_i, s^*) = 6$$

$(\mathtt{s}, \partial_{\mathtt{SH}})$-Consensus

$$s_1 = \mathtt{ab}$$
$$s_2 = \mathtt{ab}$$
$$s_3 = \mathtt{ac}$$
$$s^* = \mathtt{ba}$$

$(\mathtt{s}\,,\partial_{\mathtt{SH}})$-Consensus

$$s_1 = \mathtt{bab}$$
$$s_2 = \mathtt{cab}$$
$$s_3 = \mathtt{ac}$$
$$s^* = \mathtt{ba}$$

$(\mathtt{s}, \partial_{\mathtt{SH}})$-Consensus

$$s_1 = \ldots\mathtt{cabababab}$$
$$s_2 = \ldots\mathtt{ccbababab}$$
$$s_3 = \ldots\mathtt{cccababab}$$

$(\mathtt{s},\partial_{\mathtt{SH}})$-Consensus

$$
\begin{aligned}
s_1 &= \ldots\mathtt{cabababab} \\
s_2 &= \ldots\mathtt{ccbababab} \\
s_3 &= \ldots\mathtt{cccababab} \\
&\ldots \\
s^* &= \ldots\mathtt{abababab a}
\end{aligned}
$$

$(\mathtt{s}, \partial_{\mathtt{SH}})$-Consensus

$$s_1 = \ldots\mathtt{cababab}\textcolor{blue}{\mathtt{ab}}$$
$$s_2 = \ldots\mathtt{ccbabab}\textcolor{blue}{\mathtt{ab}}$$
$$s_3 = \ldots\mathtt{cccabab}\textcolor{blue}{\mathtt{ab}}$$
$$\ldots$$
$$s^* = \ldots\mathtt{abababa}$$

Reachable sets

Definition

We say that a set of indices $W \subseteq \{1, \dots k\}$ is *reachable* at position i if there exists a string s^* such that $s_j \in \mathcal{S}$ and s^* have a swap at position i if and only $j \in W$.

Reachable sets

Definition

We say that a set of indices $W \subseteq \{1, \dots k\}$ is *reachable* at position i if there exists a string s^* such that $s_j \in \mathcal{S}$ and s^* have a swap at position i if and only $j \in W$.

Lemma

At each position i, there are at most ki reachable sets.

Reachable sets

Definition

We say that a set of indices $W \subseteq \{1, \ldots k\}$ is *reachable* at position i if there exists a string s^* such that $s_j \in \mathcal{S}$ and s^* have a swap at position i if and only $j \in W$.

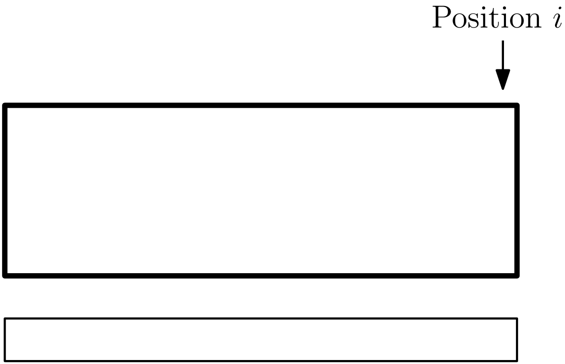
Lemma

At each position i, there are at most ki reachable sets.

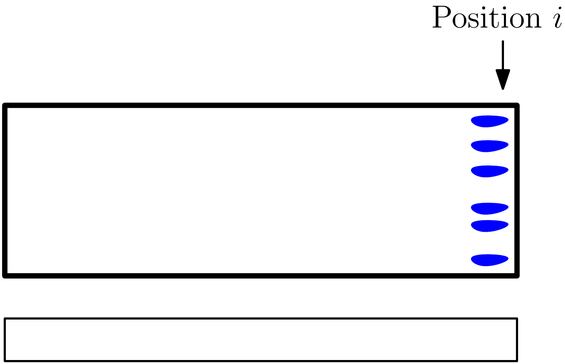
Hence, we can do dynamic programming, computing the best solution for each reachable set at each position.

Dynamic programming

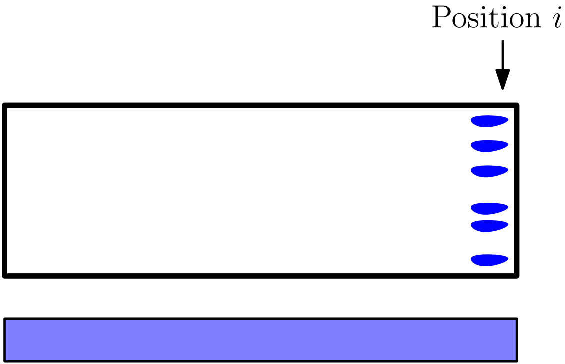
Dynamic programming



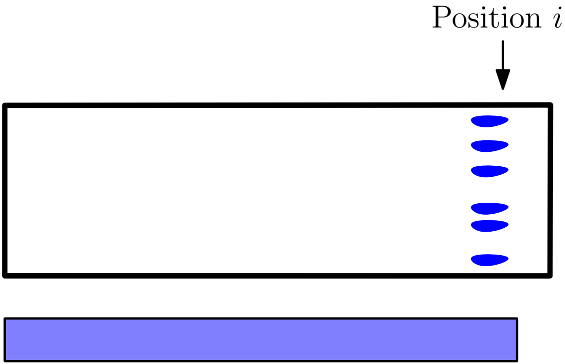
Dynamic programming



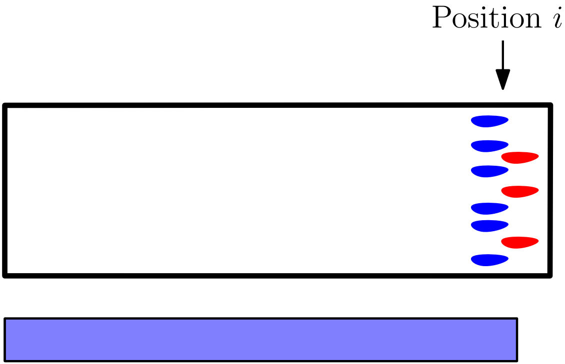
Dynamic programming



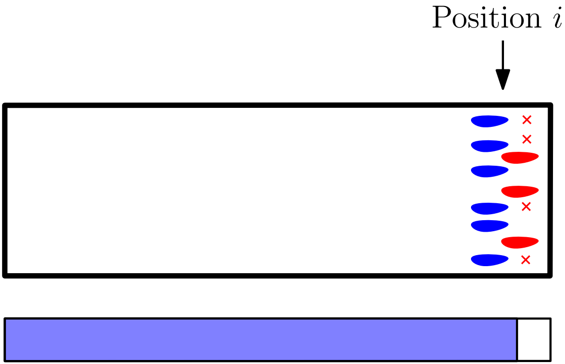
Dynamic programming



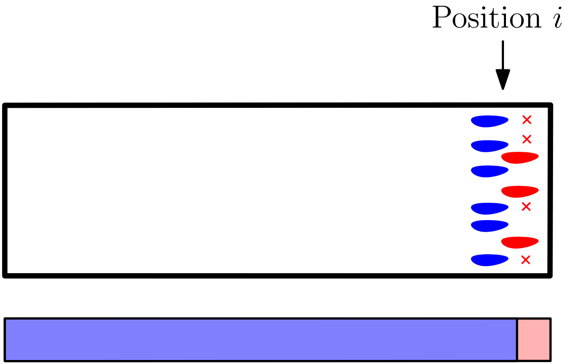
Dynamic programming



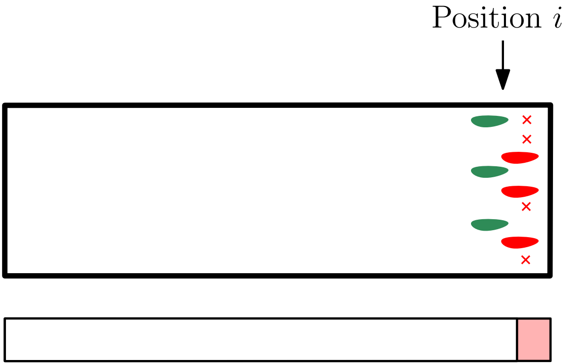
Dynamic programming



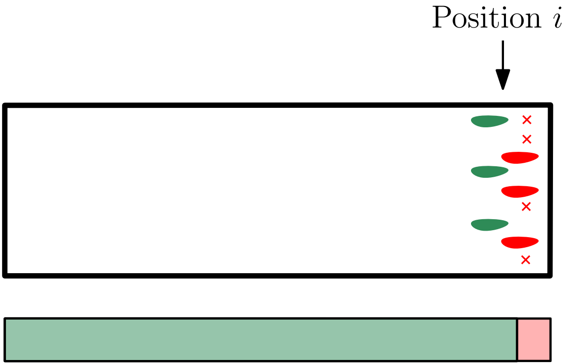
Dynamic programming



Dynamic programming



Dynamic programming



$(\mathtt{s}, \partial_{\mathtt{SH}})$-Consensus: Wrapping up

- At most *kn* reachable sets per position

$(\mathtt{s}, \partial_{\mathtt{SH}})$-Consensus: Wrapping up

- At most kn reachable sets per position
- Each of them can be extended in at most $\mathcal{O}(k)$ or $\mathcal{O}(|\Sigma|)$ ways

$(\mathtt{s}, \partial_{\mathtt{SH}})$-CONSENSUS: Wrapping up

- At most *kn* reachable sets per position
- Each of them can be extended in at most $\mathcal{O}(k)$ or $\mathcal{O}(|\Sigma|)$ ways
- Updating distances and computing swap sets both take $\mathcal{O}(k)$

$(\mathtt{s}, \partial_{\mathtt{SH}})$-CONSENSUS: Wrapping up

- At most *kn* reachable sets per position
- Each of them can be extended in at most $\mathcal{O}(k)$ or $\mathcal{O}(|\Sigma|)$ ways
- Updating distances and computing swap sets both take $\mathcal{O}(k)$

Theorem

One can solve $(\mathtt{s}, \partial_{\mathtt{SH}})$-CONSENSUS *in* $\mathcal{O}(k^3 n^2)$ *or* $\mathcal{O}(|\Sigma| k^2 n^2)$ *time.*

Results

We showed the following:

- The problems $(\mathtt{r}, \partial_{\mathrm{S}}) - \textsc{Consensus}$ and $(\mathtt{r}, \partial_{\mathrm{SH}}) - \textsc{Consensus}$ are FPT for d.
- The problem $(\mathtt{s}, \partial_{\mathrm{S}}) - \textsc{Consensus}$ can be solved in linear time.
- The problem $(\mathtt{s}, \partial_{\mathrm{SH}}) - \textsc{Consensus}$ can be solved in polynomial time.

Results

We showed the following:

- The problems $(\mathtt{r},\partial_{\mathtt{S}}) - \textsc{Consensus}$ and $(\mathtt{r},\partial_{\mathtt{SH}}) - \textsc{Consensus}$ are FPT for d.
- The problem $(\mathtt{s},\partial_{\mathtt{S}}) - \textsc{Consensus}$ can be solved in linear time.
- The problem $(\mathtt{s},\partial_{\mathtt{SH}}) - \textsc{Consensus}$ can be solved in polynomial time.

We also studied the bi-objective problem $(\mathtt{rs},\partial_{\mathtt{S}}) - \textsc{Consensus}$, and show that it can be reduced to $(\mathtt{rs},\partial_{\mathtt{Ham}}) - \textsc{Consensus}$.

Future directions

We leave the following problems open:

- Is $(\texttt{rs}, \partial_{\mathrm{SH}}) - \textsc{Consensus}(d) \in$ FPT?
- Can the algorithm for $(\texttt{s}, \partial_{\mathrm{SH}}) - \textsc{Consensus}$ be further optimized?
- Can our algorithms be generalized to allow deleting a certain number of outlier strings or columns in $\mathcal{S}$ such that the resulting substrings admit a solution (cf. Bulteau and Schmid, 2020)?
- Can those problems be studied in a learning-augmented perspective?

Thank you! Questions ?