On the Complexity of the Swap Common Superstring Problem

Paola Bonizzoni [1] Riccardo Dondi [2] Giancarlo Mauri [1]
Italo Zoppis [1]

[1]Università di Milano-Bicocca, [2]Università di Bergamo

ICTCS 2012, September 19, 2012

Outline

Outline

Motivations

In **computational biology**:

- given a set of (possible discordant) fragments (strings)
- compute a superstring that contains the maximum number of given fragments

In **AI planning**

- given a set of tasks (strings) to be accomplished
- compute a scheduling (superstring) of the tasks that satisfies the maximum number of given tasks (strings)

Motivations

To deal with these problems $\rightarrow$ variant of **shortest common superstring** problem:

- SWAP COMMON SUPERSTRING (SWCS) [Gotthilf et al, SPIRE 2010]

Swap Ordering

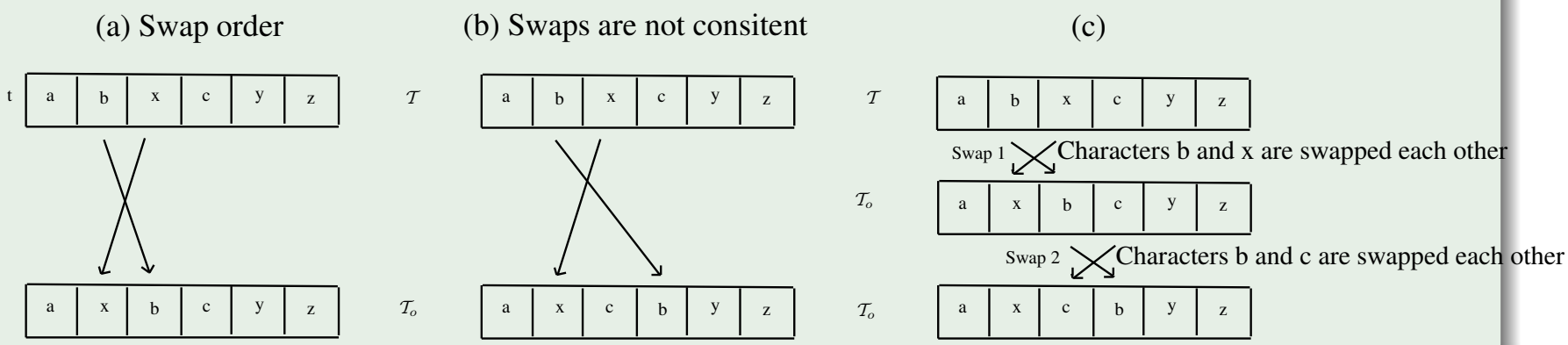
Definition

Given a text $\mathcal{T}$, a text $\mathcal{T}_o$ is called a *swap ordering of* $\mathcal{T}$ if is obtained by swapping only some pairs of adjacent distinct characters of $\mathcal{T}$.

Swaps must be **consistent**: a character in position i in $\mathcal{T} \rightarrow$ in position $j \in \{i-1, i, i+1\}$ in $\mathcal{T}_o$.

Swap Ordering

Example



Swap Common Superstring (SWCS)

Problem (SWCS)

Input: *a set S of input strings, a text $\mathcal{T}$ over an alphabet Σ.*
Output: *a swap ordering $\mathcal{T}_o$ of $\mathcal{T}$ such that the maximum number of strings in S are contained in $\mathcal{T}_o$*

Swap Common Superstring (SWCS)

Problem (SWCS)

Input: *a set S of input strings, a text $\mathcal{T}$ over an alphabet Σ.*
Output: *a swap ordering $\mathcal{T}_o$ of $\mathcal{T}$ such that the maximum number of strings in S are contained in $\mathcal{T}_o$*

Example

$\mathcal{T} = abxcyz$
$S = \{s_1 = abx, s_2 = xyz, s_3 = xyc\}$

$$\mathcal{T}_o = abxycz$$

contains s_1 and s_3

Previous Results

SWCS:

- **NP-hard** [Gotthilf et al, SPIRE 2010]
- **fixed-parameter tractable** when parameterized by the number of contained substrings [Bonizzoni, Dondi, Mauri, Zoppis, IPEC 2012]
- a variant of SWCS in which each occurrence of an input string in a swap ordering $\mathcal{T}_o$ is counted admits a poly-time algorithm [Gotthilf et al, SPIRE 2010]

Outline

Complexity of SWCS

We investigate the complexity of two restrictions of SWCS

- SWCS for Strings of **Bounded Length** ($l-$SWCS): SWCS restricted to input strings of lenght bounded by l
- **BINARY** SWCS (SWCS(2)): SWCS restricted to strings over binary alphabet ($\Sigma = \{0, 1\}$)

Complexity of SWCS

The results are proved given two L-reductions from MAXIMUM INDEPENDENT SET on Cubic Graphs (MISC)

Problem (MISC)

Input: *a cubic graph* $G = (V, E)$.
Output: *a maximum cardinality set* $V' \subseteq V$, *such that for each* $v_i, v_j \in V'$, *edge* $\{v_i, v_j\} \notin E$.

Complexity of $10 - SWCS$

Theorem

$10 - SWCS$ *is APX-hard.*

Proof.

L-reduction from MAXIMUM INDEPENDENT SET on Cubic Graphs (MISC). □

Complexity of $10-$ SWCS

For each $v_i \in V$, three sets of input strings:

- $S_{i,1} = \{w_i a_{i,j} w_j, w_i a_{i,h} w_h, w_i a_{i,l} w_l\}$
- $S_{i,2} = \{w_i x_i a_{i,j}, w_i x_i a_{i,h}, w_i x_i a_{i,l}\}$
- $S_{i,3} = \{x_i w_i a_{i,j} w_j x_i w_i a_{i,h} w_h x_i w_i\}$.

The text $\mathcal{T}$

$$\mathcal{T} = \mathcal{T}_1 yyy \mathcal{T}_2 \ldots yyy \ldots yyy \mathcal{T}_n$$

where

$$\mathcal{T}_i = w_i x_i a_{i,j} w_j w_i x_i a_{i,h} w_h w_i x_i a_{i,l} w_l$$

Complexity of 10 – SWCS

In a swap ordering $\mathcal{T}'$ of $\mathcal{T}$, the substring $\mathcal{T}_i'$:

- *configuration a*:

$$\mathcal{T}_i' = x_i \underbrace{w_i a_{i,j} w_j}_{S_{i,1}} x_i \underbrace{w_i a_{i,h} w_h}_{S_{i,1}} x_i \underbrace{w_i a_{i,l} w_l}_{S_{i,1}}$$

  Covers 4 input strings ($S_{i,1} \cup S_{i,3}$)

- *configuration b* $\mathcal{T}_i'$ is identical to $\mathcal{T}_i$

$$\mathcal{T}_i = \underbrace{w_i x_i a_{i,j}}_{S_{i,2}} w_j \underbrace{w_i x_i a_{i,h}}_{S_{i,2}} \underbrace{w_h w_i x_i a_{i,l}}_{S_{i,2}} w_l$$

  Covers 3 input strings ($S_{i,2}$)

Complexity of $10-$ SWCS

Lemma

If $\mathcal{T}_i'$ and $\mathcal{T}_i'$ have a configuration a, then $\{v_i, v_j\} \notin E$

As a consequence:

- $\mathcal{T}_i'$ with *configuration a* $\Rightarrow$ vertex v_i in an **independent set** of G
- $\mathcal{T}_i'$ with *configuration b* $\Rightarrow$ vertex v_i in a **vertex cover** of G.

Lemma

There is an independent set of G of size p if and only if there is a swap ordering of $\mathcal{T}$ that covers $4p + 3(|V| - p)$ input strings.

Complexity of SWCS(2)

Theorem

SWCS(2) *is APX-hard.*

Proof.

L-reduction from MAXIMUM INDEPENDENT SET on Cubic Graphs (MISC). □

Complexity of SWCS(2)

The text $\mathcal{T}$:

$$\mathcal{T} = SE \cdot B(v_1) \cdot SE \cdot B(v_2) \cdot \ldots SE \cdot B(v_q) \cdot SE$$

where

- $SE = 1111100000 \rightarrow$ **separation block**
- $B(v_i)$ binary string associated with v_i

Complexity of SWCS(2)

Set S of input strings:

- for each $v_i \in V$, two input strings:

$$s'_i = 00000 \cdot B(v_i) \cdot SE$$

$$s''_i = SE \cdot B(v_i) \cdot 11111$$

- for each edge $\{v_i, v_j\} \in E$, an input string $s_{i,j}$ that can be covered (after some swaps) in $B(v_i)$ or $B(v_j)$

Complexity of SWCS(2)

For each $B(v_i)$:

- with appropriate swaps in $B(v_i) \Rightarrow s_{i,j}, s_{i,h}, s_{i,k}$ contained in $\mathcal{T}'$, with $\{v_i, v_j\}, \{v_i, v_h\}, \{v_i, v_k\} \in E \Rightarrow v_i$ in an **independent set** of G
- no swap in $B(v_i) \Rightarrow \mathcal{T}'$ covers the strings $s_i', s_i'' \Rightarrow v_i$ in a **vertex cover** of G

Complexity of SWCS(2)

Lemma

If $s_{i,j}$ is covered by $B(v_i)$, then $B(v_j)$ covers s'_j and s''_j.

As a consequence:

Lemma

There is an independent set of G of size p if and only if there is a swap ordering of $\mathcal{T}$ that covers $3p + 2(|V| - p)$ input strings.

Outline

Open Problems

Open Problems:

- **computational complexity** of $l-$SWCS, for $2 \leq l \leq 9$
- **approximation complexity** of SWCS

Thank you!