The complexity of string partitioning

Anne Condon[1] Ján Maňuch[1,2] Chris Thachuk[1]

[1]Department of Computer Science
University of British Columbia

[2]Department of Mathematics
Simon Fraser University

CPM 2012

Outline

Synthesizing Novel Genes

design goal

synthesize a long duplex of DNA/RNA

5' - TGGCTATCTTAAAAAGAGGAGCTAGAAAAAAGGTACATCAGGAGCCAGCTAAACGCTCTTCTAAAATCTCCGACGAAGCCA - 3'

3' - ACCGATAGAATTTTTCTCCTCGATCTTTTTTCCATGTAGTCCTCGGTCGATTTGCGAGAAGATTTTAGAGGCTGCTTCGGT - 5'

Synthesizing Novel Genes

design goal

synthesize a long duplex of DNA/RNA

(some) challenges

- only short fragments can be synthesized reliably

5' - **TGGCTATCTTAAAAAGAGGAGCTAGAAAAAAGGTACATCAGGAGCCAGCTAAACGCTCTTCTAAAATCTCCGACGAAGCCA** - 3'

3' - **ACCGATAGAATTTTTCTCCTCGATCTTTTTTCCATGTAGTCCTCGGTCGATTTGCGAGAAGATTTTAGAGGCTGCTTCGGT** - 5'

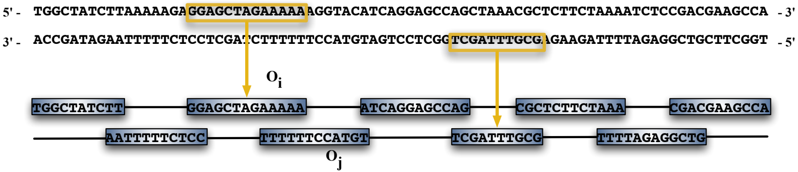
Synthesizing Novel Genes

design goal

synthesize a long duplex of DNA/RNA

(some) challenges

- only short fragments can be synthesized reliably
- fragments must *cover* duplex
- fragments must *self-assemble* correctly

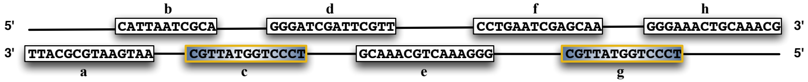


Collision Example

Example

Fragments c and g are:

- identical
- on the same strand

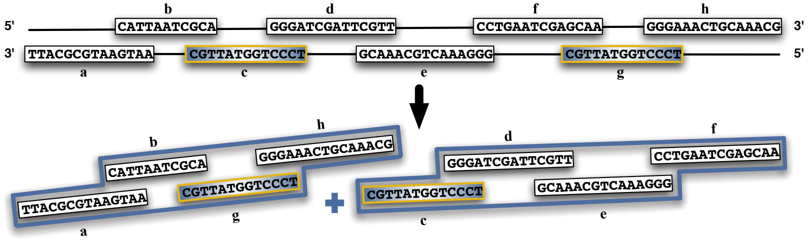


Collision Example

Example

Fragments c and g are:

- identical
- on the same strand



K-partition

Definition (K-partition of a string w)

A K-partition of a string w is a sequence $P = p_1, p_2, \ldots, p_\ell$ for some ℓ such that $w = p_1 p_2 \ldots p_\ell$ and for each $i = 1, \ldots \ell$, $|p_i| \leq K$.

Example

$$\underbrace{\text{m i}}_{p_1}\underbrace{\text{s s}}_{p_2}\underbrace{\text{i s}}_{p_3}\underbrace{\text{s i}}_{p_4}\underbrace{\text{p p}}_{p_5}\underbrace{\text{i}}_{p_6}$$

K-partition

Definition (K-partition of a string w)

A K-partition of a string w is a sequence $P = p_1, p_2, \ldots, p_\ell$ for some ℓ such that $w = p_1 p_2 \ldots p_\ell$ and for each $i = 1, \ldots \ell$, $|p_i| \leq K$.
We say: $p_1, \ldots, p_\ell$ are selected and for any $i \leq j$, string $p_i p_{i+1} \ldots p_j$ is super-selected.

Example

m i s s i s s i p p i

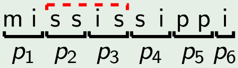
p_1 p_2 p_3 p_4 p_5 p_6

K-partition

Definition (K-partition of a string w)

A K-partition of a string w is a sequence $P = p_1, p_2, \ldots, p_\ell$ for some ℓ such that $w = p_1 p_2 \ldots p_\ell$ and for each $i = 1, \ldots \ell$, $|p_i| \leq K$.
We say: $p_1, \ldots, p_\ell$ are selected and for any $i \leq j$, string $p_i p_{i+1} \ldots p_j$ is super-selected.

Example



K-partition

Definition (K-partition of a string w)

A K-partition of a string w is a sequence $P = p_1, p_2, \ldots, p_\ell$ for some ℓ such that $w = p_1 p_2 \ldots p_\ell$ and for each $i = 1, \ldots \ell$, $|p_i| \leq K$.
We say: $p_1, \ldots, p_\ell$ are selected and for any $i \leq j$, string $p_i p_{i+1} \ldots p_j$ is super-selected.

Definition (K-partition of a set of strings $\mathcal{W}$)

A K-partition of a set of strings $\mathcal{W}$ is a set of K-partitions of all strings in $\mathcal{W}$.

Example

m i s s i s s i p p i
p_1 p_2 p_3 p_4 p_5 p_6

a c g g g a t
c c t a g c g g a
c a g g g c t a

Problem definition

$\mathcal{X}$-Free String Partition ($\mathcal{X}$-SP) Problem

Instance: Finite alphabet Σ, a positive integer K, and a string w from Σ^*.

Problem definition

$\mathcal{X}$-Free String Partition ($\mathcal{X}$-SP) Problem

Instance Finite alphabet Σ, a positive integer K, and a string w from Σ^*.

Question Is there an $\mathcal{X}$-free, K-partition P of w?

Problem definition

$\mathcal{X}$-Free String Partition ($\mathcal{X}$-SP) Problem

Instance: Finite alphabet Σ, a positive integer K, and a string w from Σ^*.

Question: Is there an $\mathcal{X}$-free, K-partition P of w?

$\mathcal{X}$-Free Multiple String Partition ($\mathcal{X}$-SP) Problem

Instance: Finite alphabet Σ, a positive integer K, and a set of strings $\mathcal{W}$ from Σ^*.

Question: Is there an $\mathcal{X}$-free, K-partition P of $\mathcal{W}$?

Types of collisions

Types of collisions we consider

$\mathcal{X}$

equality (E)

✖ m i s s i s s i p p i

✔ m i s s i s s i p p i

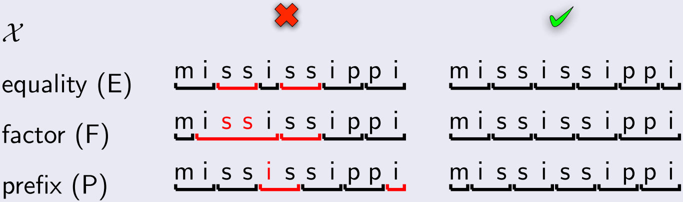
Types of collisions

Types of collisions we consider

$\mathcal{X}$	✗	✓
equality (E)	m i s s i s s i p p i	m i s s i s s i p p i
factor (F)	m i s s i s s i p p i	m i s s i s s i p p i

Types of collisions

Types of collisions we consider



Results

$\mathcal{X}$	constant partition size K	constant alphabet size L
equality (E)		
factor (F)		
prefix (P)		

Note.

If both K and L are constant, then any string longer than $L + L^2 + \cdots + L^K$ does not have an $\mathcal{X}$-free K-partition. Since the number of strings shorter or equal to this number is constant, the problem can be solved in constant time.
Similarly, the problem is easily solvable for the unary alphabet ($L = 1$) or the partition size $K = 1$.

Results

Previous results

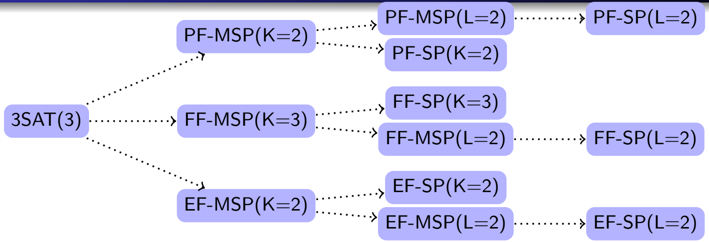
$\mathcal{X}$	constant partition size K	constant alphabet size L
equality (E)	**NP-c** for $K = 2$	**NP-c** for $L = 4$
factor (F)		
prefix (P)		

Results

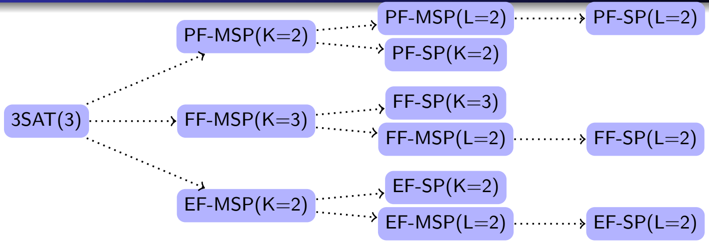
New results

$\mathcal{X}$	constant partition size K	constant alphabet size L
equality (E)	**NP-c** for $K = 2$	**NP-c** for $L = 2$
factor (F)	**NP-c** for $K = 3$	**NP-c** for $L = 2$
prefix (P)	**NP-c** for $K = 2$	**NP-c** for $L = 2$

Chain of reductions



Chain of reductions



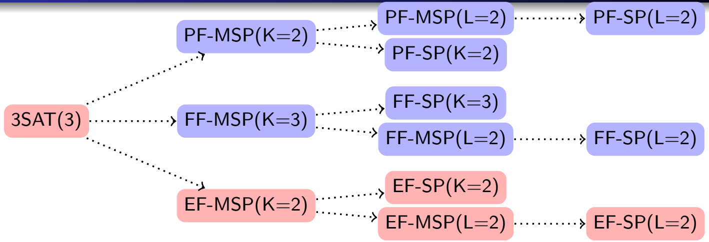
3SAT(3) Problem

Instance A formula ϕ with a set C of clauses over a set X of variables in conjunctive normal form such that:

- every clause contains 2 or 3 literals,
- each variable occurs in exactly 3 clauses, once negated and twice positive

Question Is ϕ satisfiable?

Chain of reductions



3SAT(3) Problem

Instance A formula ϕ with a set C of clauses over a set X of variables in conjunctive normal form such that:

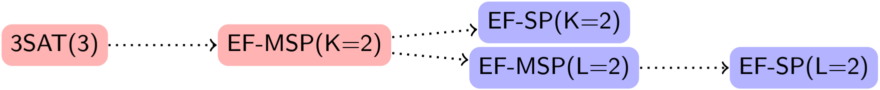
- every clause contains 2 or 3 literals,
- each variable occurs in exactly 3 clauses, once negated and twice positive

Question Is ϕ satisfiable?

Complexity of EF-MSP ($K = 2$)

Theorem

Equality-free Multiple String Partition Problem (EF-MSP) for $K = 2$ is NP-complete.



Reduction Overview

For an arbitrary 3SAT(3) instance, ϕ, with clause set C over variable set X, construct an EF-MSP ($K = 2$) instance as follows:

Reduction Overview

For an arbitrary 3SAT(3) instance, ϕ, with clause set C over variable set X, construct an EF-MSP ($K = 2$) instance as follows:

1. set
$$\Sigma = \{\hat{x}_i;\ x_i \in X\} \cup \{\hat{c}_i^j;\ c_i \in C \wedge j \leq |c_i|\} \cup \{\boxminus, \boxplus\}$$

   - symbol for each variable
   - symbol for each literal
   - forbidden symbols

Reduction Overview

For an arbitrary 3SAT(3) instance, ϕ, with clause set C over variable set X, construct an EF-MSP ($K = 2$) instance as follows:

1. set
   $$\Sigma = \{\hat{x}_i;\ x_i \in X\} \cup \{\hat{c}_i^j;\ c_i \in C \wedge j \leq |c_i|\} \cup \{\boxminus, \boxplus\}$$
   - ☐ symbol for each variable
   - ☐ symbol for each literal
   - ☐ forbidden symbols
2. set of strings $\mathcal{W} = \mathcal{C} \cup \mathcal{E} \cup \mathcal{F}$

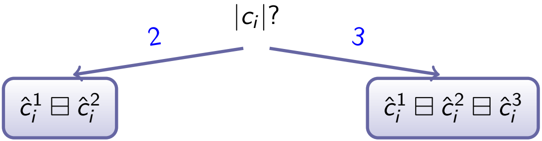
   $\mathcal{C}$ encodes the clauses of ϕ

   $\mathcal{E}$ ensures selected literals in $\mathcal{C}$ are *consistent*

$\mathcal{F} = \{\boxminus, \boxplus\}$ ensures *forbidden* substrings are not selected in $\mathcal{C} \cup \mathcal{E}$

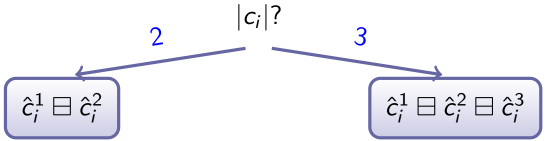
Constructing $\mathcal{C}$

1. For each clause $c_i \in C$, create the i^{th} *clause string*:



Constructing $\mathcal{C}$

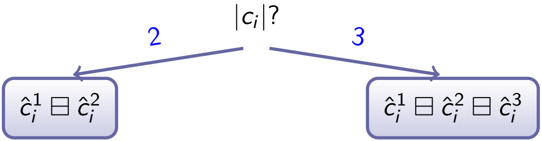
1. For each clause $c_i \in C$, create the i^{th} *clause string*:



2. Since $\boxminus$ is selected in $\mathcal{F}$, we have only the following equality-free 2-partitions of clause strings:

Constructing $\mathcal{C}$

1. For each clause $c_i \in C$, create the i^{th} *clause string*:



2. Since $\boxminus$ is selected in $\mathcal{F}$, we have only the following equality-free 2-partitions of clause strings:

Lemma

If no string from $\mathcal{F}$ is selected, exactly one literal symbol is selected for each clause string in any equality-free 2-partition of $\mathcal{C}$.

Constructing $\mathcal{E}$

1. For each variable $x_v \in X$, create the v^{th} *enforcer string*:

$$\hat{c}_i^p \boxplus \hat{c}_k^r \hat{x}_v \hat{c}_k^r \hat{x}_v \hat{c}_k^r \boxplus \hat{c}_j^q$$

- where $\hat{c}_i^p$ and $\hat{c}_j^q$ are the two positive occurrences,
- and $\hat{c}_k^r$ is the negated occurrence

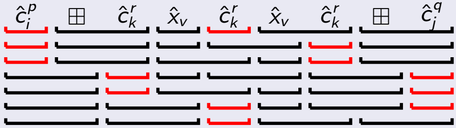
Lemma

Given that no string from $\mathcal{F}$ is selected, any equality-free 2-partition of $\mathcal{C} \cup \mathcal{E}$ must be "consistent".

Proof.

In all possible 2-partitions (9), there are two cases:

(i) $\hat{c}_k^r$ is selected: then $\hat{c}_k^r$ cannot be selected in $\mathcal{C}$
($\Rightarrow$ all selected literals of x_v in $\mathcal{C}$ are positive)

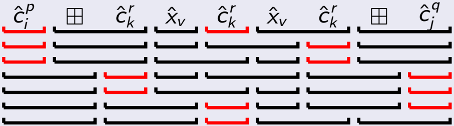


□

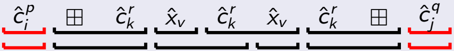
Proof.

In all possible 2-partitions (9), there are two cases:

(i) $\hat{c}_k^r$ is selected: then $\hat{c}_k^r$ cannot be selected in $\mathcal{C}$
($\Rightarrow$ all selected literals of x_v in $\mathcal{C}$ are positive)



(ii) $\hat{c}_i^p$ and $\hat{c}_j^q$ are selected: neither can be selected in $\mathcal{C}$
($\Rightarrow$ all selected literals of x_v in $\mathcal{C}$ are negated)

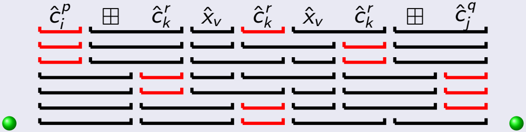


□

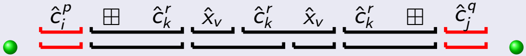
Proof.

In all possible 2-partitions (9), there are two cases:

(i) $\hat{c}_k^r$ is selected: then $\hat{c}_k^r$ cannot be selected in $\mathcal{C}$
($\Rightarrow$ all selected literals of x_v in $\mathcal{C}$ are positive)



(ii) $\hat{c}_i^p$ and $\hat{c}_j^q$ are selected: neither can be selected in $\mathcal{C}$
($\Rightarrow$ all selected literals of x_v in $\mathcal{C}$ are negated)



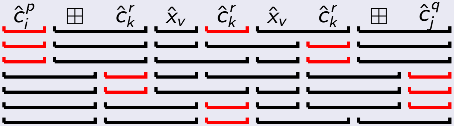
□

- There are equality-free 2-partitions of the enforcer string for the 4 consistent assignments to the 3 literals of variable x_v.

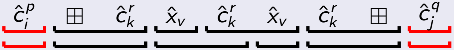
Proof.

In all possible 2-partitions (9), there are two cases:

(i) $\hat{c}_k^r$ is selected: then $\hat{c}_k^r$ cannot be selected in $\mathcal{C}$
($\Rightarrow$ all selected literals of x_v in $\mathcal{C}$ are positive)



(ii) $\hat{c}_i^p$ and $\hat{c}_j^q$ are selected: neither can be selected in $\mathcal{C}$
($\Rightarrow$ all selected literals of x_v in $\mathcal{C}$ are negated)



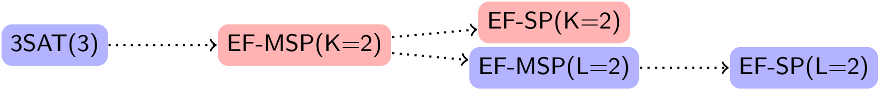
□

- There are equality-free 2-partitions of the enforcer string for the 4 consistent assignments to the 3 literals of variable x_v.
- reduction is in poly time/space (the total length of strings in $\mathcal{C} \cup \mathcal{E} \cup \mathcal{F}$ is at most $5|C| + 9|X| + 2$)

Complexity of EF-SP ($K = 2$)

Theorem

Equality-free String Partition Problem (EF-SP) for $K = 2$ is NP-complete.



Reduction

EF-MSP instance I

- alphabet Σ
- $K = 2$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$

Reduction

EF-MSP instance I

- alphabet Σ
- $K = 2$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$

EF-SP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \Sigma \cup \{\boxdot, d_1, \ldots, d_{m-1}\}$
- $\bar{K} = K = 2$
- $\bar{w} = \boxdot \boxdot \boxdot \boxdot \boxminus w_1 d_1 \boxdot \boxdot d_1 w_2 d_2 \boxdot \boxdot d_2 \ldots d_{m-1} \boxdot \boxdot d_{m-1} w_m$

Reduction

EF-MSP instance I

- alphabet Σ
- $K = 2$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$

EF-SP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \Sigma \cup \{\boxdot, d_1, \ldots, d_{m-1}\}$
- $\bar{K} = K = 2$
- $\bar{w} = \boxdot\boxdot\boxdot\boxdot\boxminus w_1 d_1 \boxdot\boxdot d_1 w_2 d_2 \boxdot\boxdot d_2 \ldots d_{m-1} \boxdot\boxdot d_{m-1} w_m$

The following substrings have to be selected in a valid 2-partition $\bar{w}$:

$$\underbrace{\boxdot}\ \underbrace{\boxdot\ \boxdot}\ \boxdot\ \boxminus\ w_1\ d_1\ \boxdot\ \boxdot\ d_1\ w_2\ d_2\ \boxdot\ \boxdot\ d_2\ \ldots\ d_{m-1} \boxdot\ \boxdot d_{m-1} w_m$$

Reduction

EF-MSP instance I

- alphabet Σ
- $K = 2$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$

EF-SP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \Sigma \cup \{\boxdot, d_1, \ldots, d_{m-1}\}$
- $\bar{K} = K = 2$
- $\bar{w} = \boxdot\boxdot\boxdot\boxdot\boxminus w_1 d_1 \boxdot\boxdot d_1 w_2 d_2 \boxdot\boxdot d_2 \ldots d_{m-1} \boxdot\boxdot d_{m-1} w_m$

The following substrings have to be selected in a valid 2-partition $\bar{w}$:

$$\underbrace{\boxdot\ \boxdot}\ \underbrace{\boxdot\ \boxdot\ \boxminus}\ w_1\ \underbrace{d_1\ \boxdot}\ \underbrace{\boxdot\ d_1}\ w_2\ \underbrace{d_2\ \boxdot}\ \underbrace{\boxdot\ d_2}\ \ldots\ \underbrace{d_{m-1}\boxdot}\ \underbrace{\boxdot d_{m-1}}\ w_m$$

Reduction

EF-MSP instance I

- alphabet Σ
- $K = 2$
- $\mathcal{W} = \{w_1, \dots, w_m\}$

EF-SP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \Sigma \cup \{\boxdot, d_1, \dots, d_{m-1}\}$
- $\bar{K} = K = 2$
- $\bar{w} = \boxdot\boxdot\boxdot\boxdot\boxminus w_1 d_1 \boxdot\boxdot d_1 w_2 d_2 \boxdot\boxdot d_2 \dots d_{m-1} \boxdot\boxdot d_{m-1} w_m$

The following substrings have to be selected in a valid 2-partition $\bar{w}$:

$\boxdot\ \boxdot\ \boxdot\ \boxdot\ \boxminus\ w_1\ d_1\ \boxdot\ \boxdot\ d_1\ w_2\ d_2\ \boxdot\ \boxdot\ d_2\ \dots\ d_{m-1}\boxdot\ \boxdot d_{m-1} w_m$

- All strings $w_1, w_2, \dots, w_m$ are super-selected.

Reduction

EF-MSP instance I

- alphabet Σ
- $K = 2$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$

EF-SP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \Sigma \cup \{\boxdot, d_1, \ldots, d_{m-1}\}$
- $\bar{K} = K = 2$
- $\bar{w} = \boxdot \boxdot \boxdot \boxdot \boxminus w_1 d_1 \boxdot \boxdot d_1 w_2 d_2 \boxdot \boxdot d_2 \ldots d_{m-1} \boxdot \boxdot d_{m-1} w_m$

The following substrings have to be selected in a valid 2-partition $\bar{w}$:

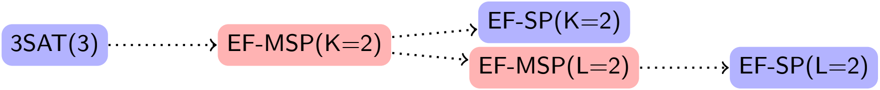
$\boxdot \; \boxdot \; \boxdot \; \boxdot \; \boxminus \; w_1 \; d_1 \; \boxdot \; \boxdot \; d_1 \; w_2 \; d_2 \; \boxdot \; \boxdot \; d_2 \; \ldots \; d_{m-1} \boxdot \; \boxdot d_{m-1} w_m$

- All strings $w_1, w_2, \ldots, w_m$ are super-selected.
- A valid 2-partition of $\bar{w}$ contains a valid 2-partition of $\mathcal{W}$, and a valid 2-partition of $\mathcal{W}$ can be extended to a valid 2-partition of $\bar{w}$.

Complexity of EF-MSP ($L = 2$)

Theorem

Equality-free Multiple String Partition Problem (EF-MSP) with binary alphabet ($L = 2$) is NP-complete.



Reduction

EF-MSP instance I

- alphabet $\Sigma = \{a_1, \ldots, a_L\}$
- $K = 2$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$

Reduction

EF-MSP instance I

- alphabet $\Sigma = \{a_1, \ldots, a_L\}$
- $K = 2$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$

EF-MSP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \{0, 1\}$
- $\bar{K} = 2\delta$, where $\delta \geq \log L$

Reduction

EF-MSP instance I

- alphabet $\Sigma = \{a_1, \dots, a_L\}$
- $K = 2$
- $\mathcal{W} = \{w_1, \dots, w_m\}$

EF-MSP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \{0, 1\}$
- $\bar{K} = 2\delta$, where $\delta \geq \log L$
- encode each letter a_i in Σ with a binary string $c_i \in C$ of length δ

Reduction

EF-MSP instance I

- alphabet $\Sigma = \{a_1, \ldots, a_L\}$
- $K = 2$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$

EF-MSP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \{0, 1\}$
- $\bar{K} = 2\delta$, where $\delta \geq \log L$
- encode each letter a_i in Σ with a binary string $c_i \in C$ of length δ
- let $h : \Sigma^* \to \bar{\Sigma}^*$ be homomorphism defined by $h(a_i) = c_i$, for every $i = 1, \ldots, L$

Reduction

EF-MSP instance I

- alphabet $\Sigma = \{a_1, \ldots, a_L\}$
- $K = 2$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$

EF-MSP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \{0, 1\}$
- $\bar{K} = 2\delta$, where $\delta \geq \log L$
- encode each letter a_i in Σ with a binary string $c_i \in C$ of length δ
- let $h : \Sigma^* \to \bar{\Sigma}^*$ be homomorphism defined by $h(a_i) = c_i$, for every $i = 1, \ldots, L$
- $$\begin{aligned}\bar{\mathcal{W}} = &\{h(w_1), \ldots, h(c_m)\} \cup \\ &\{\text{prefix}_i(c);\ c \in C, i = 1, \ldots, \delta - 1\} \cup \\ &\{\text{prefix}_i(cd);\ c, d \in C, i = \delta + 1, \ldots, 2\delta - 1\}\end{aligned}$$

Reduction Example

Example

EF-MSP instance I

- alphabet $\Sigma = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{e}\}$
- $K = 2$
- $\mathcal{W} = \{a, bc, de, ea\}$

Reduction Example

Example

EF-MSP instance I

- alphabet $\Sigma = \{a, b, c, d, e\}$
- $K = 2$
- $\mathcal{W} = \{a, bc, de, ea\}$

Example

EF-MSP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \{0, 1\}$
- $\delta = 3$, $\bar{K} = 6$
- $C = \{000, 001, 010, 011, 100\}$
- $\bar{\mathcal{W}} = \{000, 001010, 011100, 100000,$
  $0, 1, 00, 01, 10,$
  $0000, 0001, 00000, 00001, 00010,$
  $\vdots$
  $1000, 1001, 10000, 10001, 10010\}$

Reduction Example

Example

EF-MSP instance I

- alphabet $\Sigma = \{a, b, c, d, e\}$
- $K = 2$
- $\mathcal{W} = \{a, bc, de, ea\}$

Example

EF-MSP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \{0, 1\}$
- $\delta = 3$, $\bar{K} = 6$
- $C = \{000, 001, 010, 011, 100\}$
- $\bar{\mathcal{W}} = \{000, 001010, 011100, 100000,$

$$0, 1, 00, 01, 10,$$
$$0000, 0001, 00000, 00001, 00010,$$
$$\vdots$$
$$1000, 1001, 10000, 10001, 10010\}$$

Reduction Example

Example

EF-MSP instance I

- alphabet $\Sigma = \{a, b, c, d, e\}$
- $K = 2$
- $\mathcal{W} = \{a, bc, de, ea\}$

Example

EF-MSP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \{0, 1\}$
- $\delta = 3$, $\bar{K} = 6$
- $C = \{000, 001, 010, 011, 100\}$
- $\bar{\mathcal{W}} = \{000, 001010, 011100, 100000,$
  $0, 1, 00, 01, 10,$
  $0000, 0001, 00000, 00001, 00010,$
  $\vdots$
  $1000, 1001, 10000, 10001, 10010\}$

Complexity of EF-MSP ($L = 2$)

Theorem

Equality-free Multiple String Partition Problem (EF-MSP) with binary alphabet ($L = 2$) is NP-complete.

Proof.

- It can be shown by induction that each string in $\{\text{prefix}_i(c);\ c \in C, i = 1, \ldots, \delta - 1\} \cup \{\text{prefix}_i(cd);\ c, d \in C, i = \delta + 1, \ldots, 2\delta - 1\}$ has to be selected complete.

Complexity of EF-MSP ($L = 2$)

Theorem

Equality-free Multiple String Partition Problem (EF-MSP) with binary alphabet ($L = 2$) is NP-complete.

Proof.

- It can be shown by induction that each string in $\{\text{prefix}_i(c);\ c \in C, i = 1, \ldots, \delta - 1\} \cup \{\text{prefix}_i(cd);\ c, d \in C, i = \delta + 1, \ldots, 2\delta - 1\}$ has to be selected complete.
- It then follows that each string in $\{h(w_1), \ldots, h(w_m)\}$ has to be partitioned into substrings of length δ or 2δ.

Complexity of EF-MSP ($L = 2$)

Theorem

Equality-free Multiple String Partition Problem (EF-MSP) with binary alphabet ($L = 2$) is NP-complete.

Proof.

- It can be shown by induction that each string in $\{\text{prefix}_i(c);\ c \in C, i = 1, \ldots, \delta - 1\} \cup \{\text{prefix}_i(cd);\ c, d \in C, i = \delta + 1, \ldots, 2\delta - 1\}$ has to be selected complete.
- It then follows that each string in $\{h(w_1), \ldots, h(w_m)\}$ has to be partitioned into substrings of length δ or 2δ.
- Hence, a valid 2δ-partition of $\bar{\mathcal{W}}$ can be translated to a valid 2-partition of $\mathcal{W}$, and vice versa.

Complexity of EF-MSP ($L = 2$)

Theorem

Equality-free Multiple String Partition Problem (EF-MSP) with binary alphabet ($L = 2$) is NP-complete.

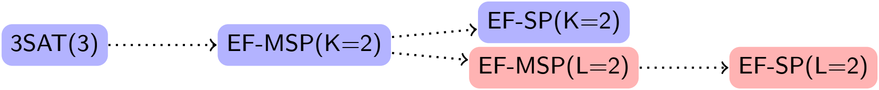
Proof.

- It can be shown by induction that each string in $\{\text{prefix}_i(c);\ c \in C, i = 1, \ldots, \delta - 1\} \cup \{\text{prefix}_i(cd);\ c, d \in C, i = \delta + 1, \ldots, 2\delta - 1\}$ has to be selected complete.
- It then follows that each string in $\{h(w_1), \ldots, h(w_m)\}$ has to be partitioned into substrings of length δ or 2δ.
- Hence, a valid 2δ-partition of $\bar{\mathcal{W}}$ can be translated to a valid 2-partition of $\mathcal{W}$, and vice versa.
- The total length of new strings $||\bar{\mathcal{W}}|| = \delta||\mathcal{W}|| + (3L^2 + L)(\delta - 1)\delta/2$.

Complexity of EF-SP ($L = 2$)

Theorem

Equality-free String Partition Problem (EF-SP) with binary alphabet ($L = 2$) is NP-complete.



Reduction overview

EF-MSP instance I

- alphabet $\Sigma = \{0, 1\}$
- $K = 2\delta$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$, where each w_i starts with 0

Reduction overview

EF-MSP instance I

- alphabet $\Sigma = \{0, 1\}$
- $K = 2\delta$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$, where each w_i starts with 0

EF-SP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \{0, 1\}$
- $\bar{K} = K = 2\delta$

Reduction overview

EF-MSP instance I

- alphabet $\Sigma = \{0, 1\}$
- $K = 2\delta$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$, where each w_i starts with 0

EF-SP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \{0, 1\}$
- $\bar{K} = K = 2\delta$
- $\bar{w} = w_1 d_1 w_2 d_2 \ldots d_{m-1} w_m$, where delimiter strings $d_1, \ldots, d_{m-1}$ are binary strings design in the way that in any valid K-partition of $\bar{w}$, they are super-selected.

Reduction overview

EF-MSP instance I

- alphabet $\Sigma = \{0, 1\}$
- $K = 2\delta$
- $\mathcal{W} = \{w_1, \ldots, w_m\}$, where each w_i starts with 0

EF-SP instance $\bar{I}$

- alphabet $\bar{\Sigma} = \{0, 1\}$
- $\bar{K} = K = 2\delta$
- $\bar{w} = w_1 d_1 w_2 d_2 \ldots d_{m-1} w_m$, where delimiter strings $d_1, \ldots, d_{m-1}$ are binary strings design in the way that in any valid K-partition of $\bar{w}$, they are super-selected.

Note. The length of $\bar{w}$ is $||\mathcal{W}|| + mK^2$.

Delimiter strings design

$$
\begin{aligned}
d_1 &= 0(1)^{K-1}(1)^{K(K-1)/2}(1)^{K-1}0 \\
d_i &= \text{chain}(\text{bin}(j)) \quad \text{for } i > 1 \\
\text{bin}(j) &= \text{binary representation of } j \text{ without leading } 1 \\
\text{chain}(\underbrace{s'10^i}_{s}) &= \text{pad}^{\mathrm{R}}_{K-|s|}(s)\,\text{pad}_{K-|s|}(s'1)\,\text{pad}^{\mathrm{R}}_{K-|s|}(s'1)\,\text{pad}_{K-|s|}(s'10)\,\text{pad}^{\mathrm{R}}_{K-|s|}(s'10) \\
&\quad \ldots \text{pad}_{K-|s|}(s'1(0)^{i-1})\,\text{pad}^{\mathrm{R}}_{K-|s|}(s'1(0)^{i-1})\,\text{pad}_{K-|s|}(s) \\
\text{pad}_i(s) &= 1^{i-1}0s
\end{aligned}
$$

Delimiter strings design

$$\begin{aligned}
d_1 &= 0(1)^{K-1}(1)^{K(K-1)/2}(1)^{K-1}0 \\
d_i &= \text{chain}(\text{bin}(j)) \quad \text{for } i > 1 \\
\text{bin}(j) &= \text{binary representation of } j \text{ without leading } 1 \\
\text{chain}(\underbrace{s'10^i}_{s}) &= \text{pad}^{\text{R}}_{K-|s|}(s)\,\text{pad}_{K-|s|}(s'1)\,\text{pad}^{\text{R}}_{K-|s|}(s'1)\,\text{pad}_{K-|s|}(s'10)\,\text{pad}^{\text{R}}_{K-|s|}(s'10) \\
&\quad \ldots \text{pad}_{K-|s|}(s'1(0)^{i-1})\,\text{pad}^{\text{R}}_{K-|s|}(s'1(0)^{i-1})\,\text{pad}_{K-|s|}(s) \\
\text{pad}_i(s) &= 1^{i-1}0s
\end{aligned}$$

Example

$d_1 = 0(1)^{K-1}(1)^{K(K-1)/2}(1)^{K-1}0$

$d_2 = \text{chain}(0) = 00(1)^{K-2}(1)^{K-2}00(1)^{K-2}(1)^{K-2}00$

$d_3 = \text{chain}(1) = 10(1)^{K-2}(1)^{K-2}01$

$d_4 = \text{chain}(00) = 000(1)^{K-3}(1)^{K-3}00(1)^{K-3}(1)^{K-3}0000(1)^{K-3}(1)^{K-3}000$

$d_5 = \text{chain}(01) = 100(1)^{K-3}(1)^{K-3}001$

Results

$\mathcal{X}$-Free (Multiple) String Partition Problem

$\mathcal{X}$	constant partition size K	constant alphabet size L
equality (E)	**NP-c** for $K = 2$	**NP-c** for $L = 2$
factor (F)	**NP-c** for $K = 3$	**NP-c** for $L = 2$
prefix (P)	**NP-c** for $K = 2$	**NP-c** for $L = 2$

Open Questions

Open Questions

- Consider different types of collisions, for example, the selected strings have to form a code.
- Complexity of collision-free string partitioning for string duplexes.
- Complexity when considering collisions with inexact matching.

Questions?

NSERC CRSNG
MITACS