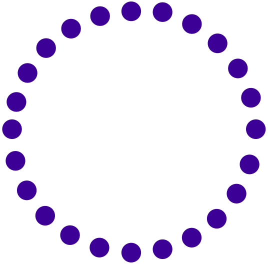
The Short Block–Move Closest Permutation Problem is NP-complete

Luís Felipe I. Cunha – UFRJ
Vinícius F. dos Santos – UFMG
Luis Antonio B. Kowada – UFF
Celina M. H. de Figueiredo – UFRJ

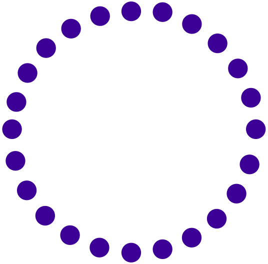
VII LAWCG, 2016
VII Latin American Workshop on Cliques in Graphs

November 9, 2016

Closest problems

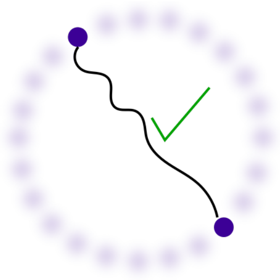


Closest problems



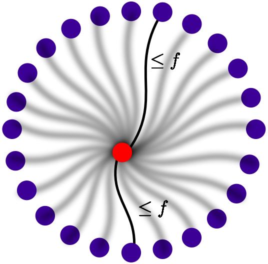
1. Given: Set of objects and an integer f.
2. It is known the "distance" between any pair of objects.
3. Is there an object for which the distance to each input is $\leq f$?

Closest problems



1. Given: Set of objects and an integer f.
2. It is known the “distance” between any pair of objects.
3. Is there an object for which the distance to each input is $\leq f$?

Closest problems



1. Given: Set of objects and an integer f.
2. It is known the “distance” between any pair of objects.
3. Is there an object for which the distance to each input is $\leq f$?

Hamming closest string problem

HAMMING CLOSEST STRING PROBLEM (H–CSP)
INPUT: Set of strings $\{s_1, s_2, \ldots, s_\ell\}$ over alphabet Σ of length m each, and a non-negative integer f.
QUESTION: Is there a string σ of length m such that $\max_{i=1,\ldots,\ell} d_H(s_i, \sigma) \leq f$?

- Hamming distance: Number of mismatched positions between two strings.

$$d_H(0\,1\,0\,1\,1\,1, \underline{1}\,1\,0\,\underline{0}\,1\,1) = 2$$

- H–CSP is NP-complete for a binary alphabet (Lanctot *et al.*, 03).

Hamming closest string problem

HAMMING CLOSEST STRING PROBLEM (H–CSP)
INPUT: Set of strings $\{s_1, s_2, \ldots, s_\ell\}$ over alphabet Σ of length m each, and a non-negative integer f.
QUESTION: Is there a string σ of length m such that $\max_{i=1,\ldots,\ell} d_H(s_i, \sigma) \leq f$?

- Hamming distance: Number of mismatched positions between two strings.

$$d_H(0\,1\,0\,1\,1\,1, \underline{1}\,1\,0\,\underline{0}\,1\,1) = 2$$

- H–CSP is NP-complete for a binary alphabet (Lanctot *et al.*, 03).

Hamming closest string problem

HAMMING CLOSEST STRING PROBLEM (H–CSP)
INPUT: Set of strings $\{s_1, s_2, \ldots, s_\ell\}$ over alphabet Σ of length m each, and a non-negative integer f.
QUESTION: Is there a string σ of length m such that $\max\limits_{i=1,\ldots,\ell} d_H(s_i, \sigma) \leq f$?

- Hamming distance: Number of mismatched positions between two strings.

$$d_H(0\,1\,0\,1\,1\,1,\ \underline{1}\,1\,0\,\underline{0}\,1\,1) = 2$$

- H–CSP is NP-complete for a binary alphabet (Lanctot *et al.*, 03).

Closest permutation problem (CPP)

- And for other objects?
  - Permutations are special strings. Each letter appears exactly once.

    [2 4 1 5 3 6] is a permutation;

    [2 2 1 5 3 6] is not a permutation.
- And for other distances?
  - CPP only makes sense if the distance problem is polynomial or it is still open.

Closest permutation problem (CPP)

- And for other objects?
  - Permutations are special strings. Each letter appears exactly once.

[2 4 1 5 3 6] is a permutation;

[2 2 1 5 3 6] is not a permutation.

- And for other distances?
  - CPP only makes sense if the distance problem is polynomial or it is still open.

Closest permutation problem (CPP)

- And for other objects?
  - Permutations are special strings. Each letter appears exactly once.

    [2 4 1 5 3 6] is a permutation;

    [2 2 1 5 3 6] is not a permutation.
- And for other distances?
  - CPP only makes sense if the distance problem is polynomial or it is still open.

Closest permutation problem (CPP)

- And for other objects?
  - Permutations are special strings. Each letter appears exactly once.

    [2 4 1 5 3 6] is a permutation;

    [2 2 1 5 3 6] is not a permutation.
- And for other distances?
  - CPP only makes sense if the distance problem is polynomial or it is still open.

- Cayley–CPP is NP-complete (Popov, 07).

In previous works we proved that:

- Block-interchange–CPP is NP-complete;
- Breakpoint–CPP is NP-complete.

Closest permutation problem (CPP)

- And for other objects?
  - Permutations are special strings. Each letter appears exactly once.

    [2 4 1 5 3 6] is a permutation;

    [2 2 1 5 3 6] is not a permutation.
- And for other distances?
  - CPP only makes sense if the distance problem is polynomial or it is still open.

- Cayley–CPP is NP-complete (Popov, 07).

In previous works we proved that:

- Block-interchange–CPP is NP-complete;
- Breakpoint–CPP is NP-complete.

Closest permutation problem (CPP)

- And for other objects?
  - Permutations are special strings. Each letter appears exactly once.

    [2 4 1 5 3 6] is a permutation;

    [2 2 1 5 3 6] is not a permutation.
- And for other distances?
  - CPP only makes sense if the distance problem is polynomial or it is still open.
- In the present work we prove that:
  - Sufficient condition for optimal sorting by Short block-moves;
  - Short block-move–CPP is NP-complete.

Block-moves and Short block-moves

- Block-moves vs. Short block-moves :

Swap pair of contiguous blocks.

13 8 10 7 11 9 12 5 4 6 2 3 1

13 8 10 7 12 5 4 6 2 3 11 9 1

Block-moves and Short block-moves

- Block-moves vs. Short block-moves :

Swap pair of contiguous blocks.

13 8 10 7 11 9 12 5 4 6 2 3 1

13 8 10 7 12 5 4 6 2 3 11 9 1

Block-moves and Short block-moves

- Block-moves vs. Short block-moves :

Swap pair of contiguous blocks.

13 8 10 7 11 9 12 5 4 6 2 3 1

13 8 10 7 12 5 4 6 2 3 11 9 1

7 12 5 13 8 10 4 6 2 3 11 9 1

Block-moves and Short block-moves

- Block-moves vs. Short block-moves :

Swap pair of contiguous blocks.

13 8 10 7 11 9 12 5 4 6 2 3 1

13 8 10 7 12 5 4 6 2 3 11 9 1

7 12 5 13 8 10 4 6 2 3 11 9 1

Swap pair of contiguous blocks with total length ≤ 3.

13 8 10 7 11 9 12 5 4 6 2 3 1

13 8 10 7 12 11 9 5 4 6 2 3 1

Block-moves and Short block-moves

- Block-moves vs. Short block-moves:

Swap pair of contiguous blocks.

13 8 10 7 11 9 12 5 4 6 2 3 1

13 8 10 7 12 5 4 6 2 3 11 9 1

7 12 5 13 8 10 4 6 2 3 11 9 1

Swap pair of contiguous blocks with total length ≤ 3.

13 8 10 7 11 9 12 5 4 6 2 3 1

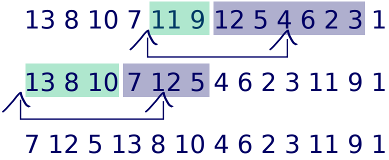
13 8 10 7 12 11 9 5 4 6 2 3 1

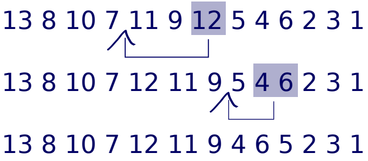
Block-moves and Short block-moves

- Block-moves vs. Short block-moves :

Swap pair of contiguous blocks.

Swap pair of contiguous blocks with total length ≤ 3.





Block-moves and Short block-moves

- Block-moves vs. Short block-moves :

Swap pair of contiguous blocks.

13 8 10 7 11 9 12 5 4 6 2 3 1

13 8 10 7 12 5 4 6 2 3 11 9 1

7 12 5 13 8 10 4 6 2 3 11 9 1

Swap pair of contiguous blocks with total length ≤ 3.

13 8 10 7 11 9 12 5 4 6 2 3 1

13 8 10 7 12 11 9 5 4 6 2 3 1

13 8 10 7 12 11 9 4 6 5 2 3 1

- Well-known in Genome Rearrangement field.

  Permutations represent genomes, Operations represent mutations.

Block-moves and Short block-moves

- Block-moves vs. Short block-moves:

Swap pair of contiguous blocks.

13 8 10 7 11 9 12 5 4 6 2 3 1

13 8 10 7 12 5 4 6 2 3 11 9 1

7 12 5 13 8 10 4 6 2 3 11 9 1

Swap pair of contiguous blocks with total length ≤ 3.

13 8 10 7 11 9 12 5 4 6 2 3 1

13 8 10 7 12 11 9 5 4 6 2 3 1

13 8 10 7 12 11 9 4 6 5 2 3 1

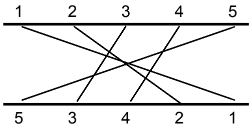
- Sorting by Block-moves (transpositions) is NP-complete (Bulteau et al., 12).
- Sorting by Short block-moves is an open problem.

Short block-move distance

- How to guarantee bounds on the Short block-move distances?
  - Permutation graph

Short block-move distance

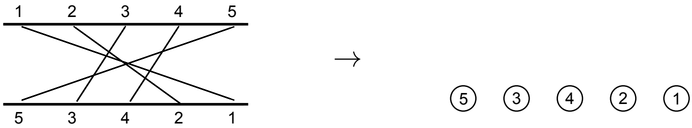
- How to guarantee bounds on the Short block-move distances?
  - Permutation graph



$PG([5\ 3\ 4\ 2\ 1])$.

Short block-move distance

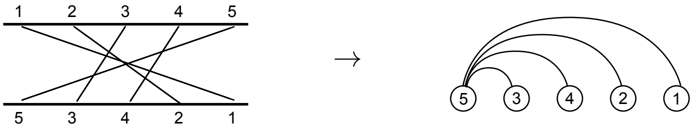
- How to guarantee bounds on the Short block-move distances?
  - Permutation graph



$PG([5\ 3\ 4\ 2\ 1])$.

Short block-move distance

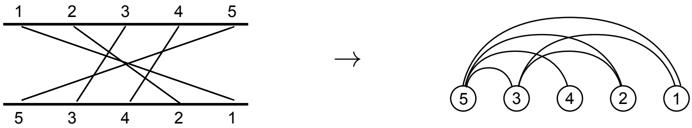
- How to guarantee bounds on the Short block-move distances?
  - Permutation graph



PG([5 3 4 2 1]).

Short block-move distance

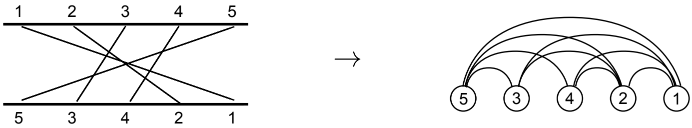
- How to guarantee bounds on the Short block-move distances?
  - Permutation graph



$PG([5\ 3\ 4\ 2\ 1])$.

Short block-move distance

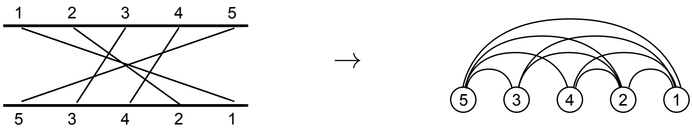
- How to guarantee bounds on the Short block-move distances?
  - Permutation graph



$PG([5\ 3\ 4\ 2\ 1])$.

Short block-move distance

- How to guarantee bounds on the Short block-move distances?
  - Permutation graph



$$5 = \lceil \frac{\#Edges\ of\ PG(\pi)}{2} \rceil \leq d_{sbm}(\pi) \leq \#Edges\ of\ PG(\pi) = 9.$$

Sufficient condition for an optimal sequence

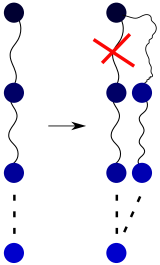
- Each connected component of π can be sorted independently.

Sufficient condition for an optimal sequence

- Each connected component of π can be sorted independently.

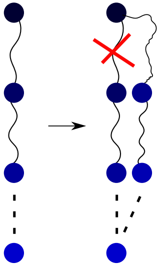
Sufficient condition for an optimal sequence

- Each connected component of π can be sorted independently.



Sufficient condition for an optimal sequence

- Each connected component of π can be sorted independently.



- This property does not hold for greater block-moves.

Short block-move–CPP is NP-complete

- Hamming–CSP $\propto$ Short block-move–CPP
  1. Transform any binary string s into a specific permutation π_s.
  2. Prove that $d_H(s) = d_{sbm}(\pi_s)$.
  3. Prove that a solution for the Hamming–CSP implies in a solution for the Short block-move–CPP, and vice versa.

Short block-move–CPP is NP-complete

- Hamming–CSP $\propto$ Short block-move–CPP
  1. Transform any binary string s into a specific permutation π_s.
  2. Prove that $d_H(s) = d_{sbm}(\pi_s)$.
  3. Prove that a solution for the Hamming–CSP implies in a solution for the Short block-move–CPP, and vice versa.

Short block-move–CPP is NP-complete

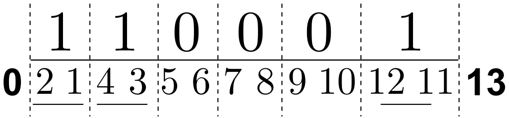
- Hamming–CSP $\propto$ Short block-move–CPP
  1. Transform any binary string s into a specific permutation π_s.
  2. Prove that $d_H(s) = d_{sbm}(\pi_s)$.
  3. Prove that a solution for the Hamming–CSP implies in a solution for the Short block-move–CPP, and vice versa.

Short block-move–CPP is NP-complete

- Hamming–CSP $\propto$ Short block-move–CPP
  1. Transform any binary string s into a specific permutation π_s.
  2. Prove that $d_H(s) = d_{sbm}(\pi_s)$.
  3. Prove that a solution for the Hamming–CSP implies in a solution for the Short block-move–CPP, and vice versa.

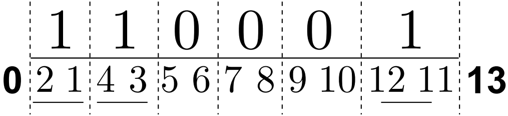
Short block-move–CPP is NP-complete

- Hamming–CSP $\propto$ Short block-move–CPP
  1. Transform any binary string s into a specific permutation π_s.
  2. Prove that $d_H(s) = d_{sbm}(\pi_s)$.
  3. Prove that a solution for the Hamming–CSP implies in a solution for the Short block-move–CPP, and vice versa.



Short block-move–CPP is NP-complete

- Hamming–CSP $\propto$ Short block-move–CPP
  1. Transform any binary string s into a specific permutation π_s.
  2. Prove that $d_H(s) = d_{sbm}(\pi_s)$.
  3. Prove that a solution for the Hamming–CSP implies in a solution for the Short block-move–CPP, and vice versa.



Sorting each connected component separately

Short block-move–CPP is NP-complete

- Hamming–CSP ∝ Short block-move–CPP
  1. Transform any binary string s into a specific permutation π_s.
  2. Prove that $d_H(s) = d_{sbm}(\pi_s)$.
  3. Prove that a solution for the Hamming–CSP implies in a solution for the Short block-move–CPP, and vice versa.

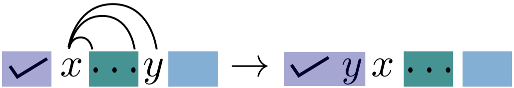
- ($\Rightarrow$) From string to permutation.

Given a solution string, construct a solution permutation by the strategy in 1.

Short block-move–CPP is NP-complete

- Hamming–CSP $\propto$ Short block-move–CPP
  1. Transform any binary string s into a specific permutation π_s.
  2. Prove that $d_H(s) = d_{sbm}(\pi_s)$.
  3. Prove that a solution for the Hamming–CSP implies in a solution for the Short block-move–CPP, and vice versa.

- ($\Leftarrow$) From permutation to string.
  - We transform any solution permutation into a new solution permutation from which we can obtain the corresponding binary string by the strategy in 1.



Further Work

- Is p-block-move–CPP NP-complete?

- Median problem: Given π_1, π_2, π_3, an integer k, and a metric M, is there σ s.t. $\sum_{i=1}^{3} d_M(\sigma, \pi_i) \leq k$?
  - Polynomial for binary strings on Hamming distance, NP-complete for permutations on Breakpoint distance.

  - But, for permutations regarding other operations?

Further Work

- Is *p*-block-move–CPP NP-complete?
- Median problem: Given π_1, π_2, π_3, an integer k, and a metric M, is there σ s.t. $\sum_{i=1}^{3} d_M(\sigma, \pi_i) \leq k$?
  - Polynomial for binary strings on Hamming distance, NP-complete for permutations on Breakpoint distance.
  - But, for permutations regarding other operations?

Further Work

- Is *p*-block-move–CPP NP-complete?
- Median problem: Given π_1, π_2, π_3, an integer k, and a metric M, is there σ s.t. $\sum_{i=1}^{3} d_M(\sigma, \pi_i) \leq k$?
  - Polynomial for binary strings on Hamming distance, NP-complete for permutations on Breakpoint distance.
  - But, for permutations regarding other operations?

Further Work

- Is *p*-block-move–CPP NP-complete?

- Median problem: Given π_1, π_2, π_3, an integer k, and a metric M, is there σ s.t. $\sum_{i=1}^{3} d_M(\sigma, \pi_i) \leq k$?
  - Polynomial for binary strings on Hamming distance, NP-complete for permutations on Breakpoint distance.

  - But, for permutations regarding other operations?

Further Work

- Is p-block-move–CPP NP-complete?
- Median problem: Given π_1, π_2, π_3, an integer k, and a metric M, is there σ s.t. $\sum_{i=1}^{3} d_M(\sigma, \pi_i) \leq k$?
  - Polynomial for binary strings on Hamming distance, NP-complete for permutations on Breakpoint distance.
  - But, for permutations regarding other operations?

Thanks for your attention!