Sorting by Prefix Block-Interchanges

Anthony Labarre

The 31st International Symposium on Algorithms and Computation (ISAAC)

December 14th, 2020

Université Gustave Eiffel

Motivations (sorting problems)

General problem

Transform X into Y using as few operations as possible from S;
the length of an optimal sequence is the S-*distance* between X and Y.

Motivations (sorting problems)

General problem

Transform X into Y using as few operations as possible from S;
the length of an optimal sequence is the S-*distance* between X and Y.

Applications arise in:

1. computational biology:
2. interconnection networks:

Motivations (sorting problems)

General problem

Transform X into Y using as few operations as possible from S;
the length of an optimal sequence is the S-*distance* between X and Y.

Applications arise in:

1. computational biology:
    - X and Y are genomes, $S =$ mutations;
    - solution $=$ evolutionary scenario between X and Y;
2. interconnection networks:

Motivations (sorting problems)

General problem

Transform X into Y using as few operations as possible from S;
the length of an optimal sequence is the S-*distance* between X and Y.

Applications arise in:

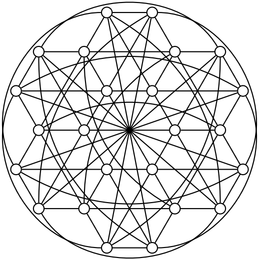
1. computational biology:
   - X and Y are genomes, S = mutations;
   - solution = evolutionary scenario between X and Y;
2. interconnection networks:
   - Cayley graph generated by S = network N;
   - X and Y are nodes in N;
   - solution = shortest routing path between X and Y;

Motivations (prefix constraints)

Additional restrictions are sometimes placed on operations to simplify the underlying problems or to obtain a "better" structure.

Example (S_4: exchanges $\mapsto$ prefix exchanges)

$$V = \text{permutations of } \{1, 2, \ldots, n\}$$
$$E = \{\{\pi, \sigma\} \ s.t. \ \exists(i,j) : \pi(i,j) = \sigma\}$$



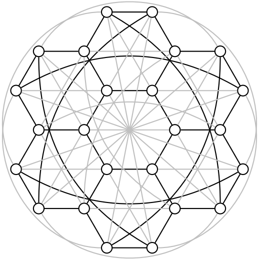
variant	$\|V\|$	degree	$\|E\|$	diameter
unrestricted	$n!$	$\binom{n}{2}$	$n!\binom{n}{2}/2$	$n-1$

Motivations (prefix constraints)

Additional restrictions are sometimes placed on operations to simplify the underlying problems or to obtain a “better” structure.

Example (S_4: exchanges $\mapsto$ prefix exchanges)

$$V = \text{permutations of } \{1, 2, \ldots, n\}$$
$$E = \{\{\pi, \sigma\} \ s.t.\ \exists (1, j) : \pi(1, j) = \sigma\}$$



variant	$\lvert V\rvert$	degree	$\lvert E\rvert$	diameter
unrestricted	$n!$	$\binom{n}{2}$	$n!\binom{n}{2}/2$	$n-1$
prefix	$n!$	$n-1$	$n!(n-1)/2$	$\lfloor 3(n-1)/2 \rfloor$

Motivations (block-interchanges)

Block-interchanges swap any two nonintersecting intervals:

Example (sorting by block-interchanges)

$$\pi = 7\ 1\ \boxed{4\ 5}\ 3\ \boxed{2}\ 6 \rightarrow \boxed{7}\ \boxed{1\ 2\ 3\ 4\ 5\ 6} \rightarrow 1\ 2\ 3\ 4\ 5\ 6\ 7$$

"unrestricted" "prefix"

- Sorting by (unrestricted) block-interchanges is easy;
- Sorting by **prefix** block-interchanges is:
  - NP-hard for strings [Cho+14];
  - open for permutations;

- Block-interchanges generalise a few other operations;

Current state of knowledge and context

The complexity of sorting problems on permutations is well-understood . . . .

Operation	Unrestricted	Prefix-constrained
signed reversal	in P	
reversal	NP-hard	
double cut-and-join	NP-hard	
signed double cut-and-join	in P	
exchange	in P	
block-transposition	NP-hard	
block-interchange	in P	

(see paper for references)

(You might know sorting by (signed) prefix reversals as *(burnt) pancake flipping*.)

Current state of knowledge and context

The complexity of sorting problems on permutations is well-understood . . . except in the prefix setting.

Operation	Unrestricted	Prefix-constrained
signed reversal	in P	open
reversal	NP-hard	NP-hard
double cut-and-join	NP-hard	open
signed double cut-and-join	in P	open
exchange	in P	in P
block-transposition	NP-hard	open
block-interchange	in P	open

(see paper for references)

(You might know sorting by (signed) prefix reversals as *(burnt) pancake flipping*.)

Results

1. We give a 2-approximation algorithm for sorting by prefix block-interchanges;
2. We show how to obtain tighter lower and upper bounds;
3. We prove that the diameter (i.e. the maximum value the distance can reach) is $\lfloor 2n/3 \rfloor$); (see paper)

The breakpoint graph $G(\pi)$ [HP99]

Given a permutation π in S_n:

Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)

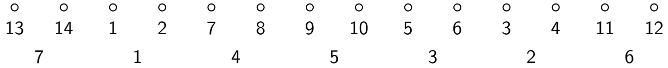
7 1 4 5 3 2 6

The breakpoint graph $G(\pi)$ [HP99]

Given a permutation π in S_n:

1. $\pi_i \mapsto (\pi'_{2i-1}, \pi'_{2i}) = (2\pi_i - 1, 2\pi_i)$;

Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)

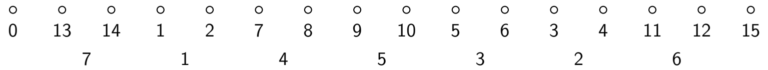


The breakpoint graph $G(\pi)$ [HP99]

Given a permutation π in S_n:

1. $\pi_i \mapsto (\pi'_{2i-1}, \pi'_{2i}) = (2\pi_i - 1, 2\pi_i)$;
2. add $\pi'_0 = 0$ and $\pi'_{2n+1} = 2n + 1$;

Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)

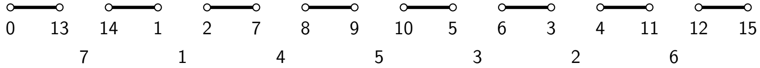


The breakpoint graph $G(\pi)$ [HP99]

Given a permutation π in S_n:

1. $\pi_i \mapsto (\pi'_{2i-1}, \pi'_{2i}) = (2\pi_i - 1, 2\pi_i)$;
2. add $\pi'_0 = 0$ and $\pi'_{2n+1} = 2n + 1$;
3. **black** edges: $\{\pi'_{2i}, \pi'_{2i+1}\}$; (consecutive positions)

Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)

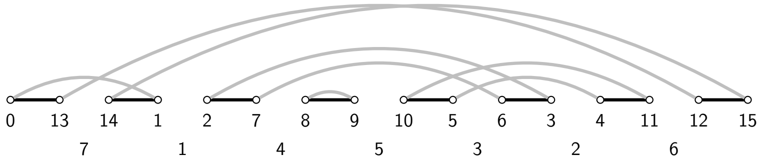


The breakpoint graph $G(\pi)$ [HP99]

Given a permutation π in S_n:

1. $\pi_i \mapsto (\pi'_{2i-1}, \pi'_{2i}) = (2\pi_i - 1, 2\pi_i)$;
2. add $\pi'_0 = 0$ and $\pi'_{2n+1} = 2n + 1$;
3. **black** edges: $\{\pi'_{2i}, \pi'_{2i+1}\}$; (consecutive positions)
4. grey edges: $\{2i, 2i + 1\}$; (consecutive values)

Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)

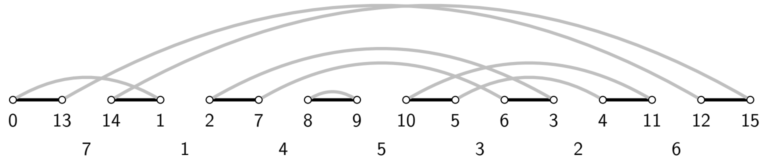


The breakpoint graph $G(\pi)$ [HP99]

Given a permutation π in S_n:

1. $\pi_i \mapsto (\pi'_{2i-1}, \pi'_{2i}) = (2\pi_i - 1, 2\pi_i)$;
2. add $\pi'_0 = 0$ and $\pi'_{2n+1} = 2n + 1$;
3. **black** edges: $\{\pi'_{2i}, \pi'_{2i+1}\}$; (consecutive positions)
4. grey edges: $\{2i, 2i + 1\}$; (consecutive values)

Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)



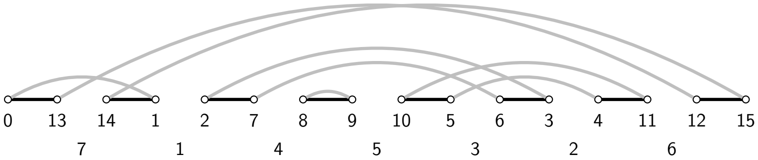
$G(\pi)$ is a collection of **alternating cycles**;

- *length of cycle* = number of black edges;
- goal: obtain only *trivial* (= length 1) cycles (see next slide);

(Prefix) Block-interchanges

- The *block-interchange* $\beta(i, j, k, \ell)$ swaps intervals $[i \cdots j-1]$ and $[k \cdots \ell - 1]$ in permutation π $\quad (1 \leq i < j \leq k < \ell \leq n+1)$;
- If $i = 1$, then β is a *prefix block-interchange* (pbi for short);
- Goal: sort π using as few pbis as possible;

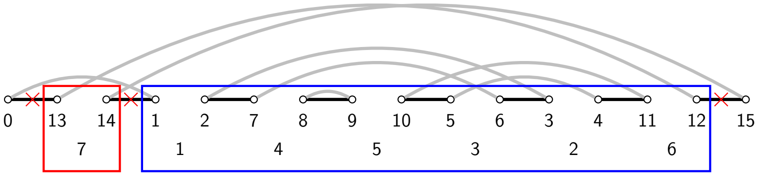
Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)



(Prefix) Block-interchanges

- The *block-interchange* $\beta(i, j, k, \ell)$ swaps intervals $[i \cdots j-1]$ and $[k \cdots \ell - 1]$ in permutation π $(1 \leq i < j \leq k < \ell \leq n+1)$;
- If $i = 1$, then β is a *prefix block-interchange* (pbi for short);
- Goal: sort π using as few pbis as possible;

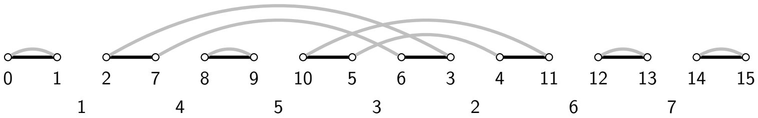
Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)



(Prefix) Block-interchanges

- The *block-interchange* $\beta(i, j, k, \ell)$ swaps intervals $[i \cdots j-1]$ and $[k \cdots \ell - 1]$ in permutation π $\quad (1 \leq i < j \leq k < \ell \leq n+1)$;
- If $i = 1$, then β is a *prefix block-interchange* (pbi for short);
- Goal: sort π using as few pbis as possible;

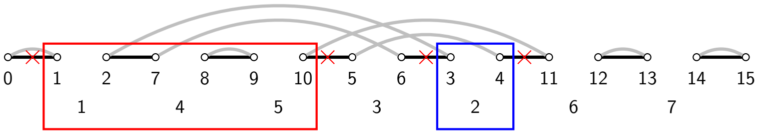
Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)



(Prefix) Block-interchanges

- The *block-interchange* $\beta(i, j, k, \ell)$ swaps intervals $[i \cdots j-1]$ and $[k \cdots \ell - 1]$ in permutation π $\quad (1 \le i < j \le k < \ell \le n+1)$;
- If $i = 1$, then β is a *prefix block-interchange* (pbi for short);
- Goal: sort π using as few pbis as possible;

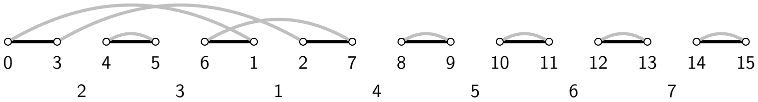
Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)



(Prefix) Block-interchanges

- The *block-interchange* $\beta(i, j, k, \ell)$ swaps intervals $[i \cdots j-1]$ and $[k \cdots \ell-1]$ in permutation π $\quad (1 \leq i < j \leq k < \ell \leq n+1)$;
- If $i = 1$, then β is a *prefix block-interchange* (pbi for short);
- Goal: sort π using as few pbis as possible;

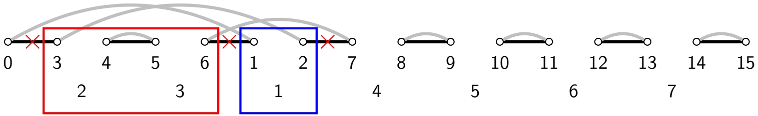
Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)



(Prefix) Block-interchanges

- The *block-interchange* $\beta(i, j, k, \ell)$ swaps intervals $[i \cdots j-1]$ and $[k \cdots \ell - 1]$ in permutation π $(1 \leq i < j \leq k < \ell \leq n+1)$;
- If $i = 1$, then β is a *prefix block-interchange* (pbi for short);
- Goal: sort π using as few pbis as possible;

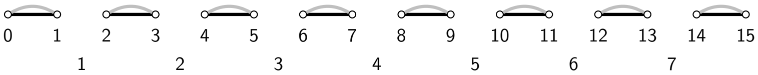
Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)



(Prefix) Block-interchanges

- The *block-interchange* $\beta(i, j, k, \ell)$ swaps intervals $[i \cdots j-1]$ and $[k \cdots \ell - 1]$ in permutation π $\quad (1 \leq i < j \leq k < \ell \leq n+1)$;
- If $i = 1$, then β is a *prefix block-interchange* (pbi for short);
- Goal: sort π using as few pbis as possible;

Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)



This shows that the *prefix block-interchange distance* of π ($pbid(\pi)$) is at most 3.

A first upper bound

We use the following function:

$$g(\pi) = \frac{n + 1 + c(G(\pi))}{2} - c_1(G(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1, \\ 1 & \text{otherwise.} \end{cases}$$

A first upper bound

We use the following function:

$$g(\pi) = \frac{n + 1 + c(G(\pi))}{2} - c_1(G(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1, \\ 1 & \text{otherwise.} \end{cases}$$

Our algorithm proves the following:

Theorem

For any π in S_n, we have $pbid(\pi) \leq g(\pi)$.

A first upper bound

We use the following function:

$$g(\pi) = \frac{n + 1 + c(G(\pi))}{2} - c_1(G(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1, \\ 1 & \text{otherwise.} \end{cases}$$

Our algorithm proves the following:

Theorem

For any π in S_n, we have $pbid(\pi) \leq g(\pi)$.

We assume $\pi_1 \neq 1$ (otherwise we simply move the longest sorted prefix $1\ 2\ \cdots\ k$ of π right before $k + 1$).

Remarks:

- Computations are easy but omitted lest the audience fall asleep;
- $g(\pi)$ is a *lower* bound on two other prefix distances;

The approximation algorithm: the case where $\pi_1 \neq 1$

Lemma ([HP99])

Every grey edge in a nontrivial cycle intersects another grey edge.

The approximation algorithm: the case where $\pi_1 \neq 1$

Lemma ([HP99])

Every grey edge in a nontrivial cycle intersects another grey edge.

We distinguish between "outer" grey edges and "inner" grey edges:

The approximation algorithm: the case where $\pi_1 \neq 1$

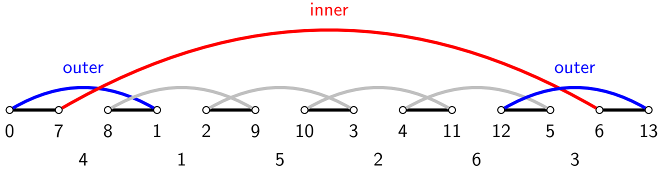
Lemma ([HP99])

Every grey edge in a nontrivial cycle intersects another grey edge.

We distinguish between "outer" grey edges and "inner" grey edges:

... because some grey edges are only intersected by outer grey edges.

Example



The approximation algorithm: the case where $\pi_1 \neq 1$

Proof ($pbid(\pi) \leq g(\pi)$).

By the previous lemma, the *first grey edge* $\{1, j\}$ intersects a grey edge $\{i, k\}$;

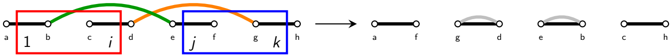
$\square$

The approximation algorithm: the case where $\pi_1 \neq 1$

Proof ($pbid(\pi) \leq g(\pi)$).

By the previous lemma, the *first grey edge* $\{1, j\}$ intersects a grey edge $\{i, k\}$;

1. if $\{i, k\}$ is inner, apply $\beta(1, i, j, k)$:



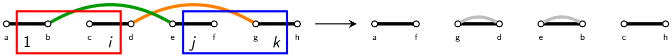
□

The approximation algorithm: the case where $\pi_1 \neq 1$

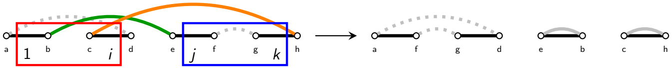
Proof ($pbid(\pi) \leq g(\pi)$).

By the previous lemma, the *first grey edge* $\{1, j\}$ intersects a grey edge $\{i, k\}$;

1. if $\{i, k\}$ is inner, apply $\beta(1, i, j, k)$:



2. if $\{i, k\}$ is outer with $j < k$, apply $\beta(1, i, j, k)$:



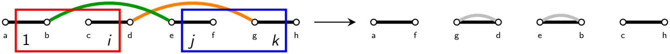
□

The approximation algorithm: the case where $\pi_1 \neq 1$

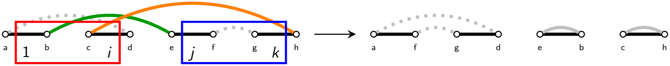
Proof ($pbid(\pi) \leq g(\pi)$).

By the previous lemma, the *first grey edge* $\{1, j\}$ intersects a grey edge $\{i, k\}$;

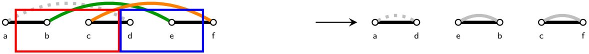
1. if $\{i, k\}$ is inner, apply $\beta(1, i, j, k)$:



2. if $\{i, k\}$ is outer with $j < k$, apply $\beta(1, i, j, k)$:



3. otherwise $\{i, k\}$ is outer with $j = k$; apply $\beta(1, i, i, k)$:



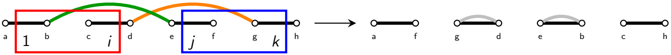
□

The approximation algorithm: the case where $\pi_1 \neq 1$

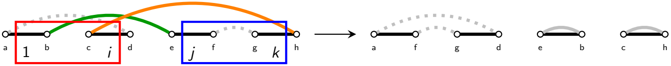
Proof ($pbid(\pi) \leq g(\pi)$).

By the previous lemma, the *first grey edge* $\{1, j\}$ intersects a grey edge $\{i, k\}$;

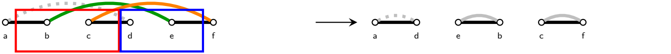
1. if $\{i, k\}$ is inner, apply $\beta(1, i, j, k)$:



2. if $\{i, k\}$ is outer with $j < k$, apply $\beta(1, i, j, k)$:



3. otherwise $\{i, k\}$ is outer with $j = k$; apply $\beta(1, i, i, k)$:



Each choice yields $g(\pi\beta) - g(\pi) \leq -1$ (details omitted), so $pbid(\pi) \leq g(\pi)$. □

Lower bounding *pbid* (outline)

We bound *pbid* using the following framework [Lab13]:

- $G(\pi)$ is itself a permutation (which we write $\overline{\pi}$);

Lower bounding *pbid* (outline)

We bound *pbid* using the following framework [Lab13]:

- $G(\pi)$ is itself a permutation (which we write $\overline{\pi}$);
- the mapping $\pi \mapsto \pi\beta$ translates to $\overline{\pi} \mapsto \overline{\pi}(\overline{\beta})^{\overline{\pi}}$;

Lower bounding *pbid* (outline)

We bound *pbid* using the following framework [Lab13]:

- $G(\pi)$ is itself a permutation (which we write $\overline{\pi}$);
- the mapping $\pi \mapsto \pi\beta$ translates to $\overline{\pi} \mapsto \overline{\pi}(\overline{\beta})^{\overline{\pi}}$;
- the image of a pbi $\overline{\beta}$ is $\overline{\beta(1, j, k, \ell)} = (j, \ell)(1, k)$;

Lower bounding *pbid* (outline)

We bound *pbid* using the following framework [Lab13]:

- $G(\pi)$ is itself a permutation (which we write $\overline{\pi}$);
- the mapping $\pi \mapsto \pi\beta$ translates to $\overline{\pi} \mapsto \overline{\pi}(\overline{\beta})^{\overline{\pi}}$;
- the image of a pbi $\overline{\beta}$ is $\overline{\beta(1, j, k, \ell)} = (j, \ell)(1, k)$;

Theorem

For all $\pi \in S_n$, we have pbid$(\pi) \geq g(\pi)/2$.

Proof idea.

□

... and therefore the approximation has ratio 2.

Lower bounding *pbid* (outline)

We bound *pbid* using the following framework [Lab13]:

- $G(\pi)$ is itself a permutation (which we write $\overline{\pi}$);
- the mapping $\pi \mapsto \pi\beta$ translates to $\overline{\pi} \mapsto \overline{\pi}(\overline{\beta})^{\overline{\pi}}$;
- the image of a pbi $\overline{\beta}$ is $\overline{\beta(1, j, k, \ell)} = (j, \ell)(1, k)$;

Theorem

For all $\pi \in S_n$*, we have* $pbid(\pi) \geq g(\pi)/2$.

Proof idea.

- Every sequence of pbis for π yields a sequence of special pairs of 2-cycles for $\overline{\pi} \Rightarrow pbid(\pi) \geq$ "special distance"$(\overline{\pi})$.

□

... and therefore the approximation has ratio 2.

Lower bounding *pbid* (outline)

We bound *pbid* using the following framework [Lab13]:

- $G(\pi)$ is itself a permutation (which we write $\overline{\pi}$);
- the mapping $\pi \mapsto \pi\beta$ translates to $\overline{\pi} \mapsto \overline{\pi}(\overline{\beta})^{\overline{\pi}}$;
- the image of a pbi $\overline{\beta}$ is $\overline{\beta(1, j, k, \ell)} = (j, \ell)(1, k)$;

Theorem

For all $\pi \in S_n$, *we have* $pbid(\pi) \geq g(\pi)/2$.

Proof idea.

- Every sequence of pbis for π yields a sequence of special pairs of 2-cycles for $\overline{\pi} \Rightarrow pbid(\pi) \geq$ "special distance"$(\overline{\pi})$.
- For every pbi β, we have $g(\overline{\pi\beta}) - g(\overline{\pi}) = g(\pi\beta) - g(\pi) \geq -2$;

□

... and therefore the approximation has ratio 2.

Lower bounding *pbid* (outline)

We bound *pbid* using the following framework [Lab13]:

- $G(\pi)$ is itself a permutation (which we write $\overline{\pi}$);
- the mapping $\pi \mapsto \pi\beta$ translates to $\overline{\pi} \mapsto \overline{\pi}(\overline{\beta})^{\overline{\pi}}$;
- the image of a pbi $\overline{\beta}$ is $\overline{\beta(1, j, k, \ell)} = (j, \ell)(1, k)$;

Theorem

For all $\pi \in S_n$, *we have* $pbid(\pi) \geq g(\pi)/2$.

Proof idea.

- Every sequence of pbis for π yields a sequence of special pairs of 2-cycles for $\overline{\pi} \Rightarrow pbid(\pi) \geq$ "special distance"$(\overline{\pi})$.
- For every pbi β, we have $g(\overline{\pi\beta}) - g(\overline{\pi}) = g(\pi\beta) - g(\pi) \geq -2$;
- Therefore: $pbid(\pi) \geq$ "special distance"$(\overline{\pi}) \geq g(\pi)/2$.

□

... and therefore the approximation has ratio 2.

A provably better upper bound using “short” cycles

Recall that we can always decrease $g(\cdot)$ by one. A more involved analysis of the proof (and additional ideas) yields:

Proposition

If $G(\pi)$ contains a “nonleftmost” 2-cycle, then there is a pbi that decreases $g(\pi)$ by 2.

A provably better upper bound using "short" cycles

Recall that we can always decrease $g(\cdot)$ by one. A more involved analysis of the proof (and additional ideas) yields:

Proposition

If $G(\pi)$ contains a "nonleftmost" 2-cycle, then there is a pbi that decreases $g(\pi)$ by 2.

Letting $c_2^{\emptyset}(G(\pi))$ denote the number of such 2-cycles, we get:

A provably better upper bound using "short" cycles

Recall that we can always decrease $g(\cdot)$ by one. A more involved analysis of the proof (and additional ideas) yields:

Proposition

If $G(\pi)$ contains a "nonleftmost" 2-cycle, then there is a pbi that decreases $g(\pi)$ by 2.

Letting $c_2^{\emptyset}(G(\pi))$ denote the number of such 2-cycles, we get:

Theorem

For any π in S_n, we have $pbid(\pi) \leq g(\pi) - \lceil c_2^{\emptyset}(G(\pi))/2 \rceil$.

A provably better upper bound using "short" cycles

Recall that we can always decrease $g(\cdot)$ by one. A more involved analysis of the proof (and additional ideas) yields:

Proposition

If $G(\pi)$ contains a "nonleftmost" 2-cycle, then there is a pbi that decreases $g(\pi)$ by 2.

Letting $c_2^{\emptyset}(G(\pi))$ denote the number of such 2-cycles, we get:

Theorem

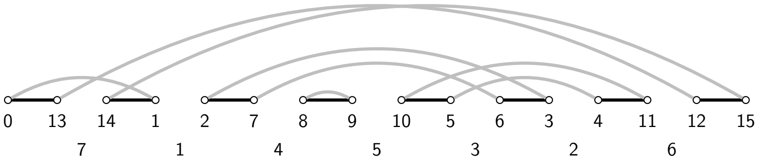
For any π in S_n, we have $pbid(\pi) \leq g(\pi) - \lceil c_2^{\emptyset}(G(\pi))/2 \rceil$.

The "/2" part stems from the fact that exploiting a 2-cycle sometimes leads to "destroying" another 2-cycle.

A probably better lower bound using "components"

A *component* of $G(\pi)$ is a connected component of the intersection graph of its nontrivial cycles.

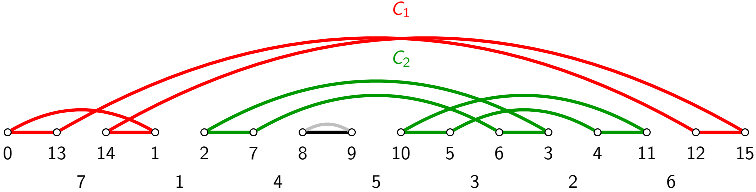
Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)



A probably better lower bound using "components"

A *component* of $G(\pi)$ is a connected component of the intersection graph of its nontrivial cycles.

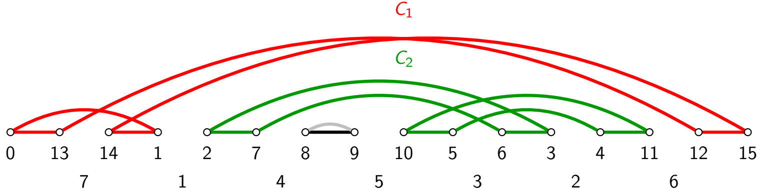
Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)



A probably better lower bound using "components"

A *component* of $G(\pi)$ is a connected component of the intersection graph of its nontrivial cycles.

Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)

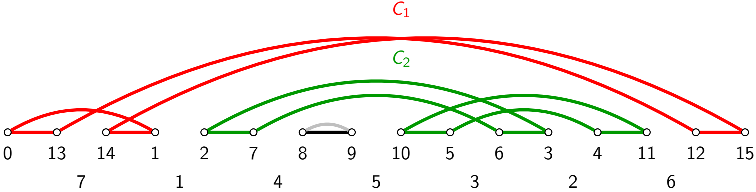


- Pbis are restricted block-interchanges, so $pbid(\pi) \geq bid(\pi)$;

A probably better lower bound using "components"

A *component* of $G(\pi)$ is a connected component of the intersection graph of its nontrivial cycles.

Example (for $\pi = 7\ 1\ 4\ 5\ 3\ 2\ 6$)



- Pbis are restricted block-interchanges, so $pbid(\pi) \geq bid(\pi)$;
- The number of components ($=CC(G(\pi))$) will help improve on this trivial result;

A probably better lower bound

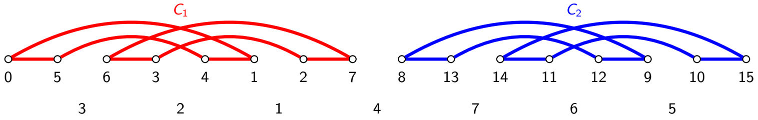
Theorem

For any π *in* S_n, *we have* $pbid(\pi) \geq bid(\pi) + CC(G(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1, \\ 1 & \text{otherwise.} \end{cases}$

Proof idea.

□

Example



A probably better lower bound

Theorem

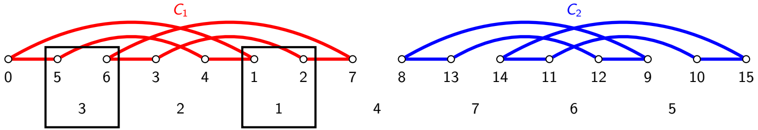
For any π in S_n, we have $pbid(\pi) \geq bid(\pi) + CC(G(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1, \\ 1 & \text{otherwise.} \end{cases}$

Proof idea.

- Merging components does not "help" $\Rightarrow$ sort each of them separately;

□

Example



A probably better lower bound

Theorem

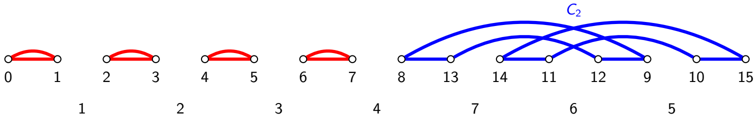
For any π in S_n, we have $pbid(\pi) \geq bid(\pi) + CC(G(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1, \\ 1 & \text{otherwise.} \end{cases}$

Proof idea.

- Merging components does not "help" $\Rightarrow$ sort each of them separately;
- Sorting each component separately cannot be achieved with less than $bid(\pi)$ operations;

□

Example



A probably better lower bound

Theorem

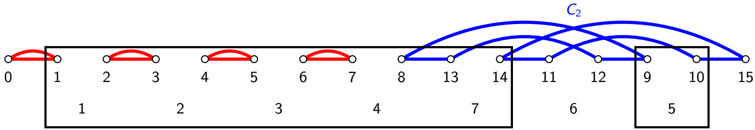
For any π *in* S_n, *we have* $pbid(\pi) \geq bid(\pi) + CC(G(\pi)) - \begin{cases} 0 & \text{if } \pi_1 = 1, \\ 1 & \text{otherwise.} \end{cases}$

Proof idea.

- Merging components does not "help" $\Rightarrow$ sort each of them separately;
- Sorting each component separately cannot be achieved with less than $bid(\pi)$ operations;
- "Accessing" each component except the leftmost one requires an additional operation.

□

Example



Future work

- Complexity?
- Can an approximation ratio lower than 2 be achieved?
- Can tighter bounds be obtained?
- Impacts of results on sorting **strings** by pbis?

Thanks!

Questions?

Selected references

[Cho+14] Shih-Wen Chou et al. "Prefix Block-Interchanges on Binary Strings". *Proceedings of the International Computer Symposium on Intelligent Systems and Applications*. Vol. 274. Frontiers in Artificial Intelligence and Applications. Taichung, Taiwan, 2014, pp. 1960–1969. DOI: 10.3233/978-1-61499-484-8-1960.

[HP99] Sridhar Hannenhalli and Pavel A. Pevzner. "Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals". *Journal of the ACM* 46.1 (1999), pp. 1–27. DOI: 10.1145/300515.300516.

[Lab13] Anthony Labarre. "Lower Bounding Edit Distances between Permutations". *SIAM Journal on Discrete Mathematics* 27.3 (2013), pp. 1410–1428. DOI: 10.1137/13090897X.