Journal talk - Annelyse Thevenin

Sorting by transposition is Difficult

Laurent Bulteau    Guillaume Fertin    Irena Rusu

LINA, University of Nantes, France

2010
Introduction

In comparative genomics, there exist several distances such that the transposition distance. A transposition consists in swapping two consecutive sequences.

\[
G_1 : 0 4 2 1 3 5 \\
0 1 3 4 2 5 \\
G_2 : 0 1 2 3 4 5 \text{ (Id)}
\]

SORTING BY TRANSPOSITION (SBT) [Bafna and Pevzner, 1995]
Find the minimum number of transposition needed to transform a genome into another.
Aim

Prove that SBT is NP-hard

**Tool:** Polynomial reductions

SBT \[\rightarrow\] 3DT-collapsibility \[\rightarrow\] SAT
I - Three problems

1. Sorting by Transpositions
SBT - Context

- **Introduce** by Bafna and Pevzner - 1995
- There exist lot of **approximation** and **heuristics**
- The best known fixed-ratio algorithm being a **1.375-approximation** [Elias and Hartman - 2006]
- **Variants** of this problem: prefix transposition or distance between strings, etc.

Sorting a permutation by block-interchanges (i.e. exchanges of non-necessarily consecutive sequences) is a polynomial problem [Christie, 1996].
Study problems
3DT-collapsibility is NP-hard to decide
SBT problem is NP-hard to decide
Conclusion

Sorting by Transpositions Problem
3DT- collapsibility problem
SAT Problem

SBT - Definition

Given three integers $0 < i < j < k \leq n$, the **transposition** $\tau_{i,j,k}$ over $[0, n]$ is the following permutation:

$$
\begin{pmatrix}
0 \ldots i-1 & i \ i+1 \ldots \ j-1 & j \ j+1 \ldots & \ldots & k-1 & k \ k+1 \ldots \ n \\
0 \ldots i-1 & j \ j+1 \ldots & \ldots & k-1 & i \ i+1 \ldots \ j-1 & k \ k+1 \ldots \ n
\end{pmatrix}
$$

Let $\pi$ be a **permutation** of $[0; n]$. The **transposition distance** $d_t(\pi)$ from $\pi$ to $\text{Id}_n$ is the minimum value $k$ for which there exist $k$ transpositions $\tau_1, \tau_2, \ldots, \tau_k$ such that $\pi \circ \tau_k \circ \cdots \circ \tau_2 \circ \tau_1 = \text{Id}_n$.

**Sorting by Transpositions problem**

**INPUT:** A permutation $\pi$, an integer $k$

**QUESTION:** Is $d_t(\pi) \leq k$?
I - Three problems

2. 3DT-collapsibility
3DT-instance - definition

Let $\Sigma$ an alphabet of at most $n$ elements.

A 3DT-instance $I$ of span $n$ is composed of:
- A word composed by $\bullet$ and distinct letters from $\Sigma$, and
- a set of ordered triples of elements of $\Sigma$, partitioning $\Sigma$:
  \[ T_I = \{(a_i, b_i, c_i) \mid 1 \leq i \leq |T_I|\}. \]

Two examples with $n=6$:

\[
I = a_1 \ c_2 \ b_1 \ b_2 \ c_1 \ a_2 \quad \text{with} \quad T_I = \{(a_1, b_1, c_1), (a_2, b_2, c_2)\}
\]
\[
I' = \bullet \ b_2 \ \bullet \ c_2 \ \bullet \ a_2 \quad \text{with} \quad T_{I'} = \{(a_2, b_2, c_2)\}
\]
The function $\Psi: \Sigma \rightarrow [1; n]$ is an injection. $\Psi(\sigma)$ is the position of $\sigma$ in the word of $I$.

- $\sigma_1 \prec \sigma_2$ if $\Psi(\sigma_1) < \Psi(\sigma_2)$
- $\sigma_1 \prec \sigma_2$ if $\sigma_1 \prec \sigma_2$ and $\not\exists x \in \Sigma$, $\sigma_1 \prec x \prec \sigma_2$.
- The function $\text{succ}_I$: for all $(a, b, c) \in T_I$, $\Psi(a) \mapsto \Psi(b)$, $\Psi(b) \mapsto \Psi(c)$, and $\Psi(c) \mapsto \Psi(a)$.

\[
I = a_1 \ c_2 \ b_1 \ b_2 \ c_1 \ a_2 \
I' = \bullet \ b_2 \ \bullet \ c_2 \ \bullet \ a_2 \\
\text{with } T_I = \{(a_1, b_1, c_1), (a_2, b_2, c_2)\}
\text{with } T_{I'} = \{(a_2, b_2, c_2)\}\]
Let $I$ be a 3DT-instance, and $(a, b, c)$ be a triple of $T_I$. Write $i = \min\{\psi(a), \psi(b), \psi(c)\}$, $j = \text{succ}_I(i)$, and $k = \text{succ}_I(j)$.

The triplet $(a, b, c) \in T_I$ is well-ordered if we have $i < j < k$. In such case, we write $\tau[a, b, c, \psi]$ the transposition $\tau_{i,j,k}$.

\[ I = a_1 \ c_2 \ b_1 \ b_2 \ c_1 \ a_2 \quad \text{with} \quad T_I = \{(a_1, b_1, c_1), (a_2, b_2, c_2)\} \]
\[ I' = \bullet \ b_2 \ \bullet \ c_2 \ \bullet \ a_2 \quad \text{with} \quad T_{I'} = \{(a_2, b_2, c_2)\} \]
3DT-step

Definition: 3DT-step

Let \( I \) be a 3DT-instance with \((a, b, c) \in T_I\) a well-ordered triple. The \textbf{3DT-step} of parameter \((a, b, c)\) is the operation written \(\stackrel{(a,b,c)}{\longrightarrow}\), transforming \( I \) into the 3DT-instance \( I' \) such that

\[
T_{I'} = T_I - (a, b, c) \quad \text{and} \quad \Psi(\sigma) = \tau^{-1}(\Psi(\sigma)).
\]

\[
I = a_1 \ c_2 \ b_1 \ b_2 \ c_1 \ a_2 \quad \text{with} \quad T_I = \{(a_1, b_1, c_1), (a_2, b_2, c_2)\}
\]

\[
I' = \bullet \ b_2 \ \bullet \ c_2 \ \bullet \ a_2 \quad \text{with} \quad T_{I'} = \{(a_2, b_2, c_2)\}\]
3DT-collapsibility

**Definition: 3DT-collapsible**

A 3DT-instance $l$ is **3DT-collapsible** if there exists a sequence of 3DT-instances $l_k, l_{k-1}, \ldots, l_0$ such that $l_k = l$, $l_0 = \epsilon$, and

$$\forall i \in [1; k], \exists (a, b, c) \in T_l, l_i \xrightarrow{(a,b,c)} l_{i-1}.$$ 

$I$ and $I'$ are 3DT-collapsible, since we have

$$I \xrightarrow{(a_1,b_1,c_1)} I' \xrightarrow{(a_2,b_2,c_2)} \epsilon.$$

$I = a_1 \ c_2 \ b_1 \ b_2 \ c_1 \ a_2$ with $T_l = \{(a_1, b_1, c_1), (a_2, b_2, c_2)\}$

$I' = \bullet \ b_2 \ \bullet \ c_2 \ \bullet \ a_2$ with $T_{l'} = \{(a_2, b_2, c_2)\}$
3DT-collapsibility

**Definition: 3DT-collapsible**

A 3DT-instance $I$ is **3DT-collapsible** if there exists a sequence of 3DT-instances $I_k, I_{k-1}, \ldots, I_0$ such that $I_k = I$, $I_0 = \epsilon$, and

$$\forall i \in [1; k], \exists (a, b, c) \in T_I, I_i \xrightarrow{(a,b,c)} I_{i-1}.$$  

**3DT-Collapsibility problem**

**INPUT:** A 3DT-instance $I$

**QUESTION:** Is $I$ 3DT-collapsible?
I - Three problems

3. SAT
Definition - SAT

SAT problem

INPUT: Formula in conjunctive normal form $\phi$

QUESTION: Is $\phi$ satisfiable?

$$\phi = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x}_2 \lor x_4)$$

$\phi$ has two clauses $C_1$ and $C_2$ (denoted by parentheses), four boolean variables $(x_1, x_2, x_3, x_4)$, and three literals per clause.

SAT was the first known example of a NP-compleat problem.
II - 3DT-collapsibility is NP-hard to decide
1. **Define** for any boolean formula $\phi$, a corresponding 3DT-instance $I_\phi$.

2. **Prove** that $I_\phi$ is 3DT-collapsible iff $\phi$ is satisfiable.
II - 3DT-collapsibility is NP-hard to decide

1. Definitions
**Definition: $l$-block decomposition**

$/-\text{block-decomposition}$ $\mathcal{B}$ of a 3DT-instance $I$ of span $n$ is an $l$-tuple $(s_1, \ldots, s_l)$ such that $s_1 = 0$, for all $h \in [1; l - 1]$, $s_h < s_{h+1}$ and $s_l < n$.

Example of a 3-block-decomposition of $I$:

\[
\begin{array}{ccc|ccc|c}
& a_1 & c_2 & b_1 & b_2 & c_1 & a_2, \\
\hline
\end{array}
\]

$s_1 = a_1, s_2 = b_1, s_3 = a_2$. 
Definition: Variable - 1/2

A **variable** $A$ of a 3DT-instance $I$ is a pair of triples $A = [(a, b, c), (x, y, z)]$ of $T_I$. It is **valid** in an $l$-block-decomposition $B$ if:

(i) $\exists h_0 \in [1; l]$ such that $block_{I,B}(b) = block_{I,B}(x) = block_{I,B}(y) = h_0$

(ii) $\exists h_1 \in [1; l], h_1 \neq h_0$, such that $block_{I,B}(a) = block_{I,B}(c) = block_{I,B}(z) = h_1$

(iii) if $x < y$, then we have $x < b < y$

(iv) $a < z < c$
The 3DT-step $I \xrightarrow{(x,y,z)} I'$ is called the activation of A (it requires that $(x, y, z)$ is well-ordered).
The 3DT-step $I \xrightarrow{(x,y,z)} I'$ is called the **activation** of $A$ (it requires that $(x, y, z)$ is well-ordered).
Definition: Basic block

They define 4 basic blocks:

- **The basic block var**: \([A_1, A_2] = \text{var}(A)\)
  \[
  \begin{align*}
  [A_1, A_2] &= \text{var}(A) \\
  &= d_1 y_1 a d_2 y_2 e_1 a' e_2 x_1 b_1 f_1 c' z b' c x_2 b_2 f_2
  \end{align*}
  \]

- **The basic block copy**: \([A_1, A_2] = \text{copy}(A)\)
  \[
  \begin{align*}
  [A_1, A_2] &= \text{copy}(A) \\
  &= a y_1 e z d y_2 x_1 b_1 c x_2 b_2 f
  \end{align*}
  \]

- **The basic block or**: \(A = \text{or}(A_1, A_2)\)
  \[
  A = \text{or}(A_1, A_2) = a_1 b' z_1 a_2 d y a' x b f z_2 c_1 e c' c_2
  \]

- **The basic block and**: \(A = \text{and}(A_1, A_2)\)
  \[
  A = \text{and}(A_1, A_2) = a_1 e z_1 a_2 c_1 z_2 d y c_2 x b f
  \]
The basic block **copy**

The **input** variable: \( A = [(a, b, c), (x, y, z)] \)

The **output** variables: \( A_1 = [(a_1, b_1, c_1)], (x_1, y_1, z_1) \) and \( A_2 = [(a_2, b_2, c_2)], (x_2, y_2, z_2) \)

Any of the two output variables can only be activated after the input variable has been activated.
Behavior graph of four basic blocks

1. $[A_1, A_2] = \text{copy}(A)$
2. $A = \text{and}(A_1, A_2)$
3. $A = \text{or}(A_1, A_2)$
4. $[A_1, A_2] = \text{var}(A)$
II - 3DT-collapsibility is NP-hard to decide

2. Construction of a 3DT-instance
Construction - step 1

Let $\phi$ be a boolean formula, over the boolean variables $x_1, \ldots, x_m$, given in conjunctive normal form: $\phi = C_1 \land C_2 \ldots \land C_\gamma$.
The 3DT-instance $I_\phi$ is defined as an assembling of basic blocks.

1. Create a set of variables

- The variables $X_i, X_i^j, \bar{X}_i$ and $\bar{X}_i^j$ representing all occurrences of $x_i$ and of $\bar{x}_i$.
- The variable $\Gamma_C$ representing the clause $C_C$.
- The variables $A_\phi$ and $A_\phi^i$, representing the formula $\phi$.
- The intermediate variables $U, \bar{U}, V, W$ and $Y$. 

Construction - step 2

\[ \phi = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x}_2 \lor x_4) \]

2. Start with an empty 3DT-instance \( \epsilon \) and add blocks successively:

(*) Blocks \textbf{var} and \textbf{copy} defining the variables \( X_i, X_i^j, \overline{X}_i \) and \( \overline{X}_i^j \)

\[
\begin{align*}
[X_i, \overline{X}_i] &= \text{var}(A_{\phi}^i) \\
[X_i^1, U_i^2] &= \text{copy}(X_i) \\
[X_i^2, U_i^3] &= \text{copy}(U_i^2) \\
&\vdots \\
[X_i^{q_i-2}, U_i^{q_i-1}] &= \text{copy}(U_i^{q_i-2}) \\
[X_i^{q_i-1}, X_i^q] &= \text{copy}(U_i^{q_i-1}) \\
[\overline{X}_i^1, \overline{U}_i^2] &= \text{copy}(\overline{X}_i) \\
[\overline{X}_i^2, \overline{U}_i^3] &= \text{copy}(\overline{U}_i^2) \\
&\vdots \\
[\overline{X}_i^{q_i-2}, \overline{U}_i^{q_i-1}] &= \text{copy}(\overline{U}_i^{q_i-2}) \\
[\overline{X}_i^{q_i-1}, \overline{X}_i^q] &= \text{copy}(\overline{U}_i^{q_i-1})
\end{align*}
\]
Construction - step 2

\[ \phi = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_4) \]

2. Start with an empty 3DT-instance \( \epsilon \) and add blocks successively:

(**) Blocks or defining \( \Gamma_c \)

\[
\begin{align*}
V_c^2 &= \text{or}(L_1, L_2) \\
V_c^3 &= \text{or}(V_c^2, L_3) \\
& \vdots \\
V_c^{k-1} &= \text{or}(V_c^{k-2}, L_{k-1}) \\
\Gamma_c &= \text{or}(V_c^{k-1}, L_k)
\end{align*}
\]
Construction - step 2

\[ \phi = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_4) \]

2. Start with an empty 3DT-instance \( \epsilon \) and add blocks successively:

(*** Blocks and defining \( A_\phi \))

\[
\begin{align*}
W_2 &= \text{and}(\Gamma_1, \Gamma_2) \\
W_3 &= \text{and}(W_2, \Gamma_3) \\
&\vdots \\
W_{\gamma-1} &= \text{and}(W_{\gamma-2}, \Gamma_{\gamma-1}) \\
A_\phi &= \text{and}(W_{\gamma-1}, \Gamma_l)
\end{align*}
\]
Construction - step 2

\[ \phi = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x}_2 \lor x_4) \]

2. Start with an empty 3DT-instance \( \epsilon \) and add blocks successively:

(****) Blocks \textbf{copy} defining \( A^i_\phi \) and \( Y \).

\[
\begin{align*}
[A^1_\phi, Y_2] &= \text{copy}(A_\phi) \\
[A^2_\phi, Y_3] &= \text{copy}(Y_2) \\
&\vdots \\
[A^{m-2}_\phi, Y_{m-1}] &= \text{copy}(Y_{m-2}) \\
[A^{m-1}_\phi, A^m_\phi] &= \text{copy}(Y_{m-1})
\end{align*}
\]
II - 3DT-collapsibility is NP-hard to decide

2. Proof: Let \( \phi \) be satisfiable
Proof: Let $\phi$ be satisfiable - 1

Let $\phi$ be **satisfiable**. Let $P$ be the set of indices $i$ such that $x_i$ is assigned to true.

Starting from $I_\phi$, we can follow a path **3DT-steps** that activates all the variables of $I_\phi$ in the specific order.

We need six steps to activate all the variables of $I_\phi$. 
Proof: Let $\phi$ be satisfiable - 2

6 steps to activate all the variables of $I_\phi$

1. If $i \in P$, activate $X_i$ in block $\text{var}$ in (*). Then, we can activate some blocks $\text{copy}$ in (*).

Otherwise, activate $\bar{X}_i$ in block $\text{var}$ in (*). Then, we can activate some blocks $\text{copy}$ in (*).

\[
[X_i, \bar{X}_i] = \text{var}(A_i^\phi)
\]

\[
[X_i^1, U_i^2] = \text{copy}(X_i)
\]

\[
[X_i^2, U_i^3] = \text{copy}(U_i^2)
\]

\[
\vdots
\]

\[
[X_i^{q_i-2}, U_i^{q_i-1}] = \text{copy}(U_i^{q_i-2})
\]

\[
[X_i^{q_i-1}, X_i^{q_i}] = \text{copy}(U_i^{q_i-1})
\]

\[
[\bar{X}_i^{q_i-2}, \bar{U}_i^{q_i-1}] = \text{copy}(\bar{U}_i^{q_i-2})
\]

\[
[\bar{X}_i^{q_i-1}, \bar{X}_i^{q_i}] = \text{copy}(\bar{U}_i^{q_i-1})
\]

Proof: Let $\phi$ be satisfiable - 1

Proof: Let $I_\phi$ be 3DT-collapsible
Proof: Let $\phi$ be satisfiable - 2

6 steps to activate all the variables of $I_\phi$

2. For each $c$, since $C_c$ is true, at least one literal $\lambda_{p_0}$ is true. Using the block or in (**), we activate $V_c^p$ and finally $\Gamma_c$ ($L_{p_0} = X_j^i$ or $L_{p_0} = \overline{X}_j^i$).

\[
\begin{align*}
V_c^2 &= \text{or}(L_1, L_2) \\
V_c^3 &= \text{or}(V_c^2, L_3) \\
&\vdots \\
V_c^{k-1} &= \text{or}(V_c^{k-2}, L_{k-1}) \\
\Gamma_c &= \text{or}(V_c^{k-1}, L_k)
\end{align*}
\]
Proof: Let $\phi$ be satisfiable - 2

6 steps to activate all the variables of $I_\phi$

3. Since all variables $\Gamma_C$ have been activated, we can activate $W_C$ and $A_\phi$ using block and in (***)

\[
\begin{align*}
W_2 &= \text{and}(\Gamma_1, \Gamma_2) \\
W_3 &= \text{and}(W_2, \Gamma_3) \\
&\vdots \\
W_{\gamma-1} &= \text{and}(W_{\gamma-2}, \Gamma_{\gamma-1}) \\
A_\phi &= \text{and}(W_{\gamma-1}, \Gamma_l)
\end{align*}
\]
Proof: Let $\phi$ be satisfiable - 2

6 steps to activate all the variables of $I_\phi$

4. Using blocks **copy** in (****), we activate $Y_i$ and $A^1_\phi, \ldots, A^m_\phi$.

\[
\begin{align*}
[A^1_\phi, Y_2] &= \text{copy}(A_\phi) \\
[A^2_\phi, Y_3] &= \text{copy}(Y_2) \\
&\vdots \\
[A^{m-2}_\phi, Y_{m-1}] &= \text{copy}(Y_{m-2}) \\
[A^{m-1}_\phi, A^m_\phi] &= \text{copy}(Y_{m-1})
\end{align*}
\]
Proof: Let $\phi$ be satisfiable - 2

6 steps to activate all the variables of $I_{\phi}$

5. Since the variables $A_i^\phi$ has been activated, we activate the remaining variable $X_i$ or $\bar{X}_i$ and $U_i^j$ or $\bar{U}_i^j$ in the block $\text{var}$ in (*)

\[
\begin{align*}
[X_i, \bar{X}_i] &= \text{var}(A_i^\phi) \\
[X_i^1, U_i^2] &= \text{copy}(X_i) \\
[X_i^2, U_i^3] &= \text{copy}(U_i^2) \\
&\vdots \\
[X_i^{q_i-2}, U_i^{q_i-1}] &= \text{copy}(U_i^{q_i-2}) \\
[X_i^{q_i-1}, X_i^q] &= \text{copy}(U_i^{q_i-1}) \\
[\bar{X}_i^1, \bar{U}_i^2] &= \text{copy}(\bar{X}_i) \\
[\bar{X}_i^2, \bar{U}_i^3] &= \text{copy}(\bar{U}_i^2) \\
&\vdots \\
[\bar{X}_i^{q_i-2}, \bar{U}_i^{q_i-1}] &= \text{copy}(\bar{U}_i^{q_i-2}) \\
[\bar{X}_i^{q_i-1}, \bar{X}_i^q] &= \text{copy}(\bar{U}_i^{q_i-1})
\end{align*}
\]
Proof: Let \( \phi \) be satisfiable - 2

6 steps to activate all the variables of \( I_\phi \)

6. In (**), since all variables \( L_p \) have been activated, we activate the remaining intermediate variables \( V_C^P \).

\[
\begin{align*}
V_C^2 & = \text{or}(L_1, L_2) \\
V_C^3 & = \text{or}(V_C^2, L_3) \\
& \quad \vdots \\
V_C^{k-1} & = \text{or}(V_C^{k-2}, L_{k-1}) \\
V_C & = \text{or}(V_C^{k-1}, L_k)
\end{align*}
\]
Proof: Let $\phi$ be satisfiable - 3

Every variable has been activated
$\Rightarrow$ the resulting instance is 3DT-collapsible.

If $\phi$ is satisfiable then $I_\phi$ is 3DT-collapsible.
II - 3DT-collapsibility is NP-hard to decide

3. Proof: Let $I_\phi$ is 3DT-collapsible
Let $I_\phi$ be **3DT-collapsible**. Let $Q$ be the set of **variables** activated before $A_\phi$ and $P$ the **set of indices** $i$ such that $X_i \in Q$.

3 steps to show that the true assignment defined by $(x_i = \text{true} \iff i \in P)$ satisfies the formula $\phi$.

1. $A^i_C$ cannot belong to $Q$ (**copy (****))). Hence

   \[ \bar{X}_i \in Q \Rightarrow X_i \not\in Q \text{ (**var in (**))} \]

   \[ X^i_j \in Q \Rightarrow X_i \in Q \text{ (**copy in (**))} \]

   \[ \bar{X}^i_j \in Q \Rightarrow \bar{X}_i \in Q \text{ (**copy in (**))} \]
Let $I_\phi$ be 3DT-collapsible. Let $Q$ be the set of variables activated before $A_\phi$ and $P$ the set of indices $i$ such that $X_i \in Q$.

3 steps to show that the true assignment defined by $(x_i = \text{true} \iff i \in P)$ satisfies the formula $\phi$.

2. Since $A_\phi$ is defined in a block $A_\phi = \text{and}(W_{\lambda-1}, \Gamma_\lambda)$ in (***) , we necessarily have: $W_{\lambda-1} \in Q$ and $\Gamma_\lambda \in Q$.

Since $W_{\lambda-1}$ is defined by $W_{\lambda-1} = \text{and}(W_{\lambda-2}, \Gamma_{\lambda-1})$, we also have $W_{\lambda-2} \in Q$ and $\Gamma_{\lambda-1} \in Q$.

Recursively: $\Gamma_c \in Q$ for each $c \in [1; \lambda]$. 
Let $I_\phi$ be \textbf{3DT-collapsible}. Let $Q$ be the set of variables activated before $A_\phi$ and $P$ the set of indices $i$ such that $X_i \in Q$.

3 steps to show that the true assignment defined by $(x_i = \text{true} \iff i \in P)$ satisfies the formula $\phi$.

3. For each clause $C_c$, there exists some $p_0$ such that the variable $L_{p_0}$ is activated before $\Gamma_c$: hence $l_{p_0} \in Q$.

If the corresponding literal $\lambda_{p_0}$ is the $j$-th occurrence of $x_i$ (resp. $\neg x_i$), then $L_{p_0} = X_i^j$ (resp. $X_i^j$) thus $X_i \in Q$ (resp $\bar{X}_i \in Q$) and $i \in P$ (resp. $i \notin P$).

The literal $\lambda_{p_0}$ is true in the truth assignment defined by $(x_i = \text{true} \iff i \in P)$. 
So, if \( I_\phi \) is 3DT-collapsible, they have found a truth assignment such that at least one literal is true in each clause of the formula \( \phi \), and thus \( \phi \) is satisfiable.
So, if $I_\phi$ is 3DT-collapsible, they have found a truth assignment such that at least one literal is true in each clause of the formula $\phi$, and thus $\phi$ is satisfiable.

**Theorem**

3DT-collapsibility problem is NP-hard.

**Proof:** Let $\phi$ be a boolean formula, and $I_\phi$ the 3DT-instance defined previously. The construction of $I_\phi$ is polynomial in the size of $\phi$, and $\phi$ is satisfiable iff $I_\phi$ is 3DT-collapsible. □
III - SBT is NP-hard to decide

1. Construction of a permutation $\pi_I$
Build $\pi_I$ from $I$

Aim: Build in polynomial time a permutation $\pi_I$ such that $I \sim \pi_I$.

**Theorem**

Let $I$ be a 3DT-instance of span $n$ with $\mathcal{B}$ an $l$-block-decomposition such that $(I, \mathcal{B})$ is an assembling of basic blocks.

Then there exists a permutation $\pi_I$, computable in polynomial time in $n$, such that $I \sim \pi_I$.

The permutation $\pi_I$ defined by this theorem is in fact a 3-permutation, i.e. a permutation whose cycle graph contains only 3-cycles.
III - SBT is NP-hard to decide

2. Proof
Proof

1. Given any instance $\phi$ of \textbf{SAT}, create a \textbf{3DT-instance} $I_\phi$, being an assembling of basic blocks, which is 3DT-collapsible iff $\phi$ is satisfiable.

2. Then create a \textbf{3-permutation} $\pi_{I_\phi}$ equivalent to $I_\phi$ (previous theorem).

The above two steps can be done in \textbf{polynomial} time.
Proof

Finally, set \( k = \frac{d_b(\pi I_\phi)}{3} = \frac{n}{3} \). We then have:

\[
\phi \text{ is satisfiable } \iff I_\phi \text{ is 3DT-collapsible} \\
\iff d_t(\pi I_\phi) = k \text{ (because } \pi I_\phi \sim I_\phi) \\
\iff d_t(\pi I_\phi) \leq k \text{ (because } d_t(\pi) \geq \frac{d_b(\pi)}{3})
\]

Theorem

**Sorting by Transpositions** problem is NP-hard.
Conclusions
Conclusion

Main theorem

**Sorting by Transpositions** problem is NP-hard.

Corollary

The following two decision problems are NP-hard:

- Given a permutation $\pi$ of $\{0; n\}$, is the equality $d_t(\pi) = \frac{d_b(\pi)}{3}$ satisfied?
- Given a 3-permutation $\pi$ of $\{0; n\}$, is the equality $d_t(\pi) = \frac{n}{3}$ satisfied?
Prospect

- A polynomial-time **approximation** scheme?

- Relevant **parameters** for which problem is fixed parameter tractable?