# Heurísticas para Rearranjo de Genomas com Genes Duplicados

Gabriel Henriques Siqueira

Orientador: Prof. Dr. Zanoni Dias

Universidade Estadual de Campinas

07 de outubro de 2019

# Roteiro

Motivação

Conceitos

Objetivos

Metodologia

Cronograma

Resultados Preliminares

# Motivação

- Determinar distância evolutiva entre genomas.

# Motivação

- Determinar distância evolutiva entre genomas.
- Geralmente se assume que não existem repetições de genes.

# Motivação

- Determinar distância evolutiva entre genomas.
- Geralmente se assume que não existem repetições de genes.
- Iremos propor heurísticas para o caso em que os genes podem estar duplicados.

# Motivação

- Determinar distância evolutiva entre genomas.
- Geralmente se assume que não existem repetições de genes.
- Iremos propor heurísticas para o caso em que os genes podem estar duplicados.
- Primeiro passo para entender o caso com genes multiplicados.

# Representação por Strings

- Representamos um genoma por uma string $S$.

# Representação por Strings

- Representamos um genoma por uma string $S$.
- Cada caractere corresponde a um gene (ou bloco conservado de genes).

# Representação por Strings

- Representamos um genoma por uma string $S$.
- Cada caractere corresponde a um gene (ou bloco conservado de genes).
- A orientação dos genes é representada por um sinal $+$ ou $-$.

# Representação por Strings

- Representamos um genoma por uma string $S$.
- Cada caractere corresponde a um gene (ou bloco conservado de genes).
- A orientação dos genes é representada por um sinal $+$ ou $-$.
- Algumas notações:

# Representação por Strings

- Representamos um genoma por uma string $S$.
- Cada caractere corresponde a um gene (ou bloco conservado de genes).
- A orientação dos genes é representada por um sinal $+$ ou $-$.
- Algumas notações:
  - $|S|$ = número de caracteres em $S$.

# Representação por Strings

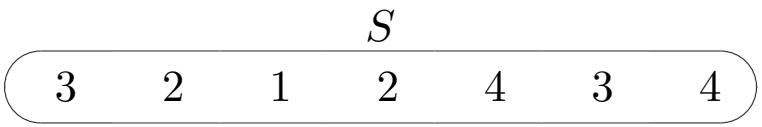
- Representamos um genoma por uma string $S$.
- Cada caractere corresponde a um gene (ou bloco conservado de genes).
- A orientação dos genes é representada por um sinal $+$ ou $-$.
- Algumas notações:
  - $|S|$ = número de caracteres em $S$.
  - $\Sigma_S$ = conjunto de caracteres de S.

# Representação por Strings

- Representamos um genoma por uma string $S$.
- Cada caractere corresponde a um gene (ou bloco conservado de genes).
- A orientação dos genes é representada por um sinal $+$ ou $-$.
- Algumas notações:
  - $|S|$ = número de caracteres em $S$.
  - $\Sigma_S$ = conjunto de caracteres de S.
  - $dup(S)$ = conjunto de caracteres duplicados de S.

# Representação por Strings

- Representamos um genoma por uma string $S$.
- Cada caractere corresponde a um gene (ou bloco conservado de genes).
- A orientação dos genes é representada por um sinal $+$ ou $-$.
- Algumas notações:
  - $|S|$ = número de caracteres em $S$.
  - $\Sigma_S$ = conjunto de caracteres de S.
  - $dup(S)$ = conjunto de caracteres duplicados de S.




$|S| = 7 \quad \Sigma_S = \{1, 2, 3, 4\}$
$dup(S) = \{2, 3, 4\}$

# Representação por Strings

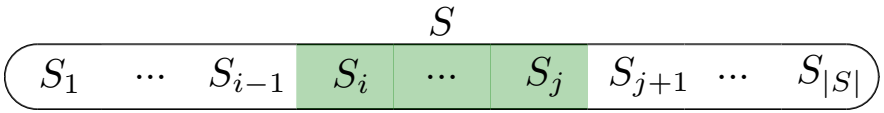
- Representamos um genoma por uma string $S$.
- Cada caractere corresponde a um gene (ou bloco conservado de genes).
- A orientação dos genes é representada por um sinal $+$ ou $-$.
- Algumas notações:
  - $|S|$ = número de caracteres em $S$.
  - $\Sigma_S$ = conjunto de caracteres de S.
  - $dup(S)$ = conjunto de caracteres duplicados de S.

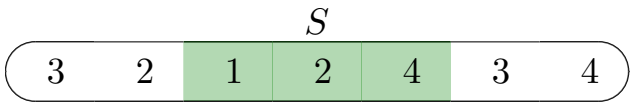
$$S$$
$$+3 \quad +2 \quad +1 \quad -2 \quad +4 \quad +3 \quad -4$$

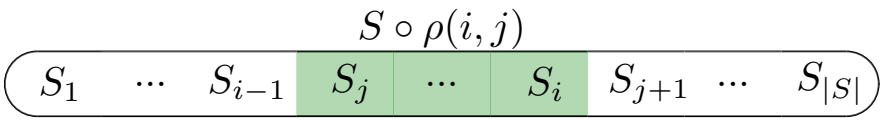
$$|S| = 7 \quad \Sigma_S = \{1, 2, 3, 4\}$$
$$dup(S) = \{2, 3, 4\}$$

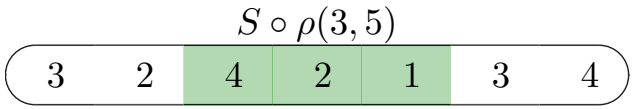
# Eventos de Rearranjo

- Reversão:



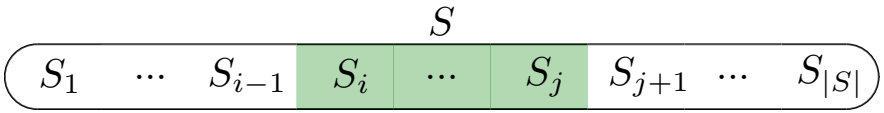



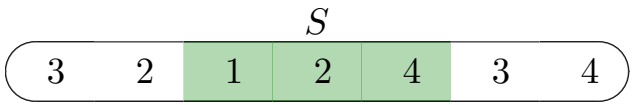



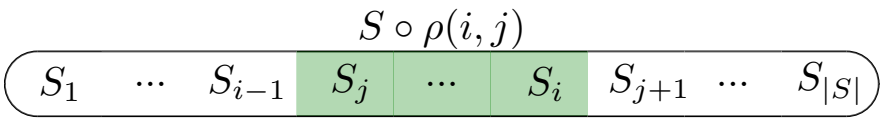



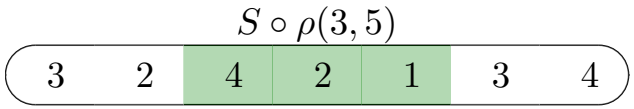
# Eventos de Rearranjo

- Reversão:

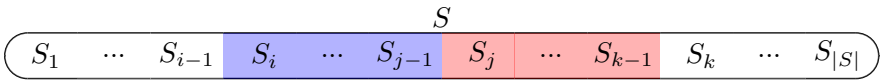


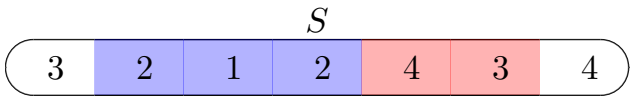



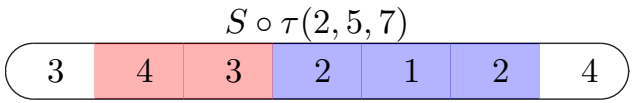




- Transposição:



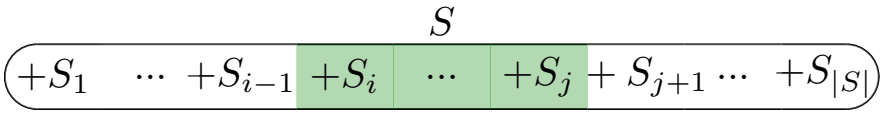



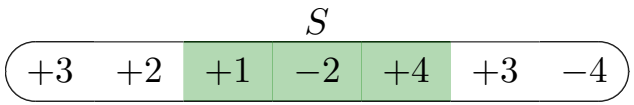


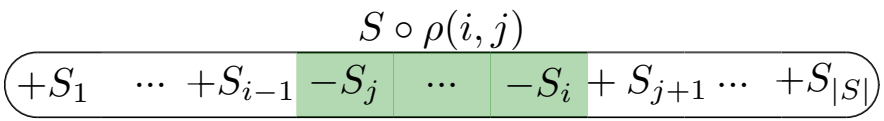


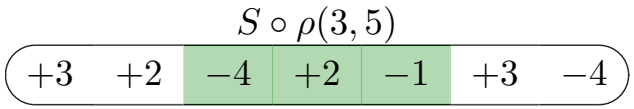
# Eventos de Rearranjo

- Reversão:

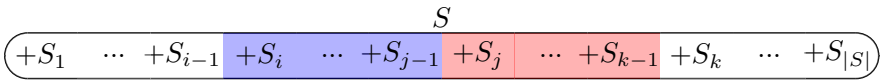


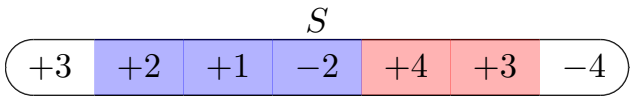



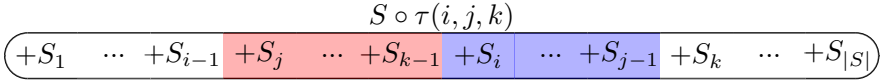



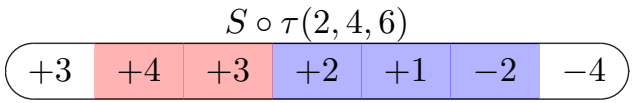




- Transposição:



# Eventos de Rearranjo

- Modelo de rearranjo: conjunto de eventos de rearranjo permitidos.

# Eventos de Rearranjo

- Modelo de rearranjo: conjunto de eventos de rearranjo permitidos.
- Distância de rearranjo ($d_{\mathcal{M}}(S, P)$): menor número de operações correspondentes a rearranjos do modelo $\mathcal{M}$ necessárias para transformar $S$ em $P$.

# Resultados Conhecidos

- Distância de Reversão sem sinal:

# Resultados Conhecidos

- Distância de Reversão sem sinal:
  - NP-Difícil [3];

# Resultados Conhecidos

- Distância de Reversão sem sinal:
  - NP-Difícil [3];
  - Aproximação de 1.375 sem caracteres repetidos [1];

# Resultados Conhecidos

- Distância de Reversão sem sinal:
  - NP-Difícil [3];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 4.4148 com caracteres duplicados [**?**];

# Resultados Conhecidos

- Distância de Reversão sem sinal:
  - NP-Difícil [3];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 4.4148 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].

# Resultados Conhecidos

- Distância de Reversão sem sinal:
  - NP-Difícil [3];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 4.4148 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].
- Distância de Reversão com sinal:

# Resultados Conhecidos

- Distância de Reversão sem sinal:
  - NP-Difícil [3];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 4.4148 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].
- Distância de Reversão com sinal:
  - Polinomial sem caracteres repetidos [5];

# Resultados Conhecidos

- Distância de Reversão sem sinal:
  - NP-Difícil [3];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 4.4148 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].
- Distância de Reversão com sinal:
  - Polinomial sem caracteres repetidos [5];
  - NP-Difícil com caracteres repetidos [8] e com caracteres duplicados [4];

# Resultados Conhecidos

- Distância de Reversão sem sinal:
  - NP-Difícil [3];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 4.4148 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].
- Distância de Reversão com sinal:
  - Polinomial sem caracteres repetidos [5];
  - NP-Difícil com caracteres repetidos [8] e com caracteres duplicados [4];
  - Aproximação de 2.2074 com caracteres duplicados [**?**];

# Resultados Conhecidos

- Distância de Reversão sem sinal:
  - NP-Difícil [3];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 4.4148 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].
- Distância de Reversão com sinal:
  - Polinomial sem caracteres repetidos [5];
  - NP-Difícil com caracteres repetidos [8] e com caracteres duplicados [4];
  - Aproximação de 2.2074 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].

# Resultados Conhecidos

- Distância de Transposição:

# Resultados Conhecidos

- Distância de Transposição:
  - NP-Difícil [2];

# Resultados Conhecidos

- Distância de Transposição:
  - NP-Difícil [2];
  - Aproximação de 1.375 sem caracteres repetidos [1];

# Resultados Conhecidos

- Distância de Transposição:
  - NP-Difícil [2];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 3.311 com caracteres duplicados [**?**];

# Resultados Conhecidos

- Distância de Transposição:
  - NP-Difícil [2];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 3.311 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].

# Resultados Conhecidos

- Distância de Transposição:
  - NP-Difícil [2];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 3.311 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].
- Distância de Reversão e Transposição:

# Resultados Conhecidos

- Distância de Transposição:
  - NP-Difícil [2];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 3.311 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].
- Distância de Reversão e Transposição:
  - NP-Difícil com e sem sinais [7];

# Resultados Conhecidos

- Distância de Transposição:
  - NP-Difícil [2];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 3.311 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].
- Distância de Reversão e Transposição:
  - NP-Difícil com e sem sinais [7];
  - Aproximação de $2k$ ($k$ é a aproximação do algoritmo usado para decomposição em ciclos) sem caracteres repetidos e sem sinais [9];

# Resultados Conhecidos

- Distância de Transposição:
  - NP-Difícil [2];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 3.311 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].
- Distância de Reversão e Transposição:
  - NP-Difícil com e sem sinais [7];
  - Aproximação de $2k$ ($k$ é a aproximação do algoritmo usado para decomposição em ciclos) sem caracteres repetidos e sem sinais [9];
  - Aproximação de 2 sem caracteres repetidos e com sinais [10].

# Objetivos

O objetivo é estudar os seguintes problemas:

# Objetivos

O objetivo é estudar os seguintes problemas:

- Distância de Reversão em Strings sem Sinais e com Genes Duplicados ($DR$);

# Objetivos

O objetivo é estudar os seguintes problemas:

- Distância de Reversão em Strings sem Sinais e com Genes Duplicados ($DR$);
- Distância de Reversão em Strings com Sinais e com Genes Duplicados ($D\bar{R}$);

# Objetivos

O objetivo é estudar os seguintes problemas:

- Distância de Reversão em Strings sem Sinais e com Genes Duplicados ($DR$);
- Distância de Reversão em Strings com Sinais e com Genes Duplicados ($D\bar{R}$);
- Distância de Transposição em Strings sem Sinais e com Genes Duplicados ($DT$);

# Objetivos

O objetivo é estudar os seguintes problemas:

- Distância de Reversão em Strings sem Sinais e com Genes Duplicados ($DR$);
- Distância de Reversão em Strings com Sinais e com Genes Duplicados ($D\bar{R}$);
- Distância de Transposição em Strings sem Sinais e com Genes Duplicados ($DT$);
- Distância de Reversão e Transposição em Strings sem Sinais e com Genes Duplicados ($DRT$).

# Objetivos

O objetivo é estudar os seguintes problemas:

- Distância de Reversão em Strings sem Sinais e com Genes Duplicados ($DR$);
- Distância de Reversão em Strings com Sinais e com Genes Duplicados ($D\bar{R}$);
- Distância de Transposição em Strings sem Sinais e com Genes Duplicados ($DT$);
- Distância de Reversão e Transposição em Strings sem Sinais e com Genes Duplicados ($DRT$).
- Distância de Reversão e Transposição em Strings com Sinais e com Genes Duplicados ($D\bar{R}T$);

# Metodologia

- Desenvolvimento de Heurísticas para cada problema.

# Metodologia

- Desenvolvimento de Heurísticas para cada problema.
- Heurísticas baseadas em características comuns a todos os problemas.

# Metodologia

- Desenvolvimento de Heurísticas para cada problema.
- Heurísticas baseadas em características comuns a todos os problemas.
- Heurísticas baseadas em características especificas de cada problema.

# Metodologia

- Desenvolvimento de Heurísticas para cada problema.
- Heurísticas baseadas em características comuns a todos os problemas.
- Heurísticas baseadas em características especificas de cada problema.
- Criação de uma base de dados.

# Metodologia

- Desenvolvimento de Heurísticas para cada problema.
- Heurísticas baseadas em características comuns a todos os problemas.
- Heurísticas baseadas em características especificas de cada problema.
- Criação de uma base de dados.
- Comparação entre as heurísticas e com resultados da literatura.

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

1. Revisão da literatura;

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

1. Revisão da literatura;
2. Escrita da proposta de mestrado;

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

1. Revisão da literatura;
2. Escrita da proposta de mestrado;
3. Exame de Qualificação de Mestrado (EQM);

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

1. Revisão da literatura;
2. Escrita da proposta de mestrado;
3. Exame de Qualificação de Mestrado (EQM);
4. Participação no Programa de Estágio Docente (PED);

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

5. Investigação do problema *DR*;

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

5. Investigação do problema $DR$;
6. Investigação do problema $D\bar{R}$;

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

5. Investigação do problema $DR$;
6. Investigação do problema $D\bar{R}$;
7. Investigação do problema $DT$;

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

5. Investigação do problema $DR$;
6. Investigação do problema $D\bar{R}$;
7. Investigação do problema $DT$;
8. Investigação do problema $DRT$;

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

5. Investigação do problema $DR$;
6. Investigação do problema $D\bar{R}$;
7. Investigação do problema $DT$;
8. Investigação do problema $DRT$;
9. Investigação do problema $D\bar{R}T$;

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

10. Execução dos experimentos e comparação dos resultados;

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

10. Execução dos experimentos e comparação dos resultados;
11. Escrita da dissertação;

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

10. Execução dos experimentos e comparação dos resultados;
11. Escrita da dissertação;
12. Revisão da dissertação;

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

10. Execução dos experimentos e comparação dos resultados;
11. Escrita da dissertação;
12. Revisão da dissertação;
13. Defesa da dissertação.

# Mapeamentos em Permutações

- Mapeamos as strings em permutações.

# Mapeamentos em Permutações

- Mapeamos as strings em permutações.
- Assim, podemos usar resultados já conhecidos para permutações.

# Mapeamentos em Permutações

- Mapeamos as strings em permutações.
- Assim, podemos usar resultados já conhecidos para permutações.

$$S$$

| 3 | 2 | 1 | 2 | 4 | 3 | 4 |
|---|---|---|---|---|---|---|

$$S^{\mathbf{x}}$$

| $3''$ | $2'$ | 1 | $2''$ | $4''$ | $3'$ | $4'$ |
|---|---|---|---|---|---|---|

$$\mathbf{x} = \begin{array}{|c|c|c|} \hline 2 & 3 & 4 \\ \hline 0 & 1 & 1 \\ \hline \end{array}$$

# Mapeamentos em Permutações

- Mapeamos as strings em permutações.
- Assim, podemos usar resultados já conhecidos para permutações.

$S$

| 3 | 2 | 1 | 2 | 4 | 3 | 4 |
|---|---|---|---|---|---|---|

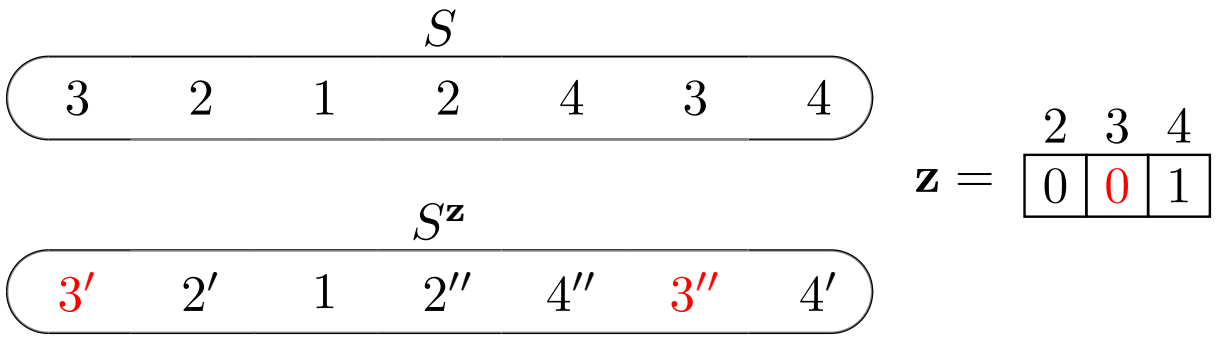
$S^{\mathbf{x}}$

| $3''$ | $2'$ | 1 | $2''$ | $4''$ | $3'$ | $4'$ |
|---|---|---|---|---|---|---|

$$\mathbf{x} = \begin{array}{|c|c|c|} \hline 2 & 3 & 4 \\ \hline 0 & 1 & 1 \\ \hline \end{array}$$

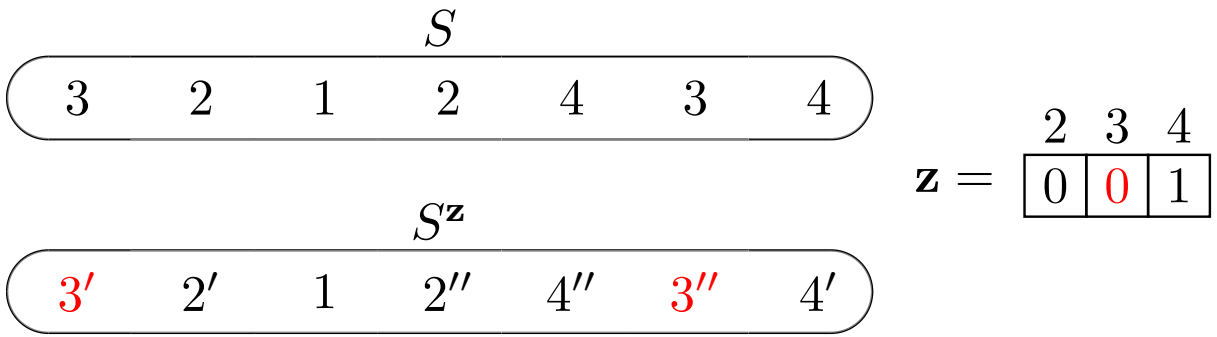
# Mapeamentos em Permutações

- Mapeamos as strings em permutações.
- Assim, podemos usar resultados já conhecidos para permutações.



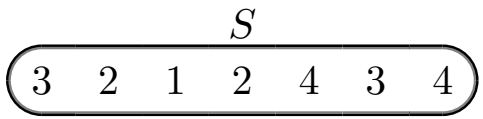
# Mapeamentos em Permutações

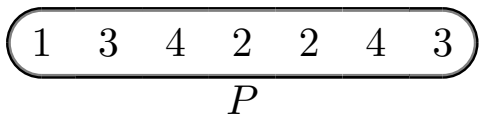
- Mapeamos as strings em permutações.
- Assim, podemos usar resultados já conhecidos para permutações.



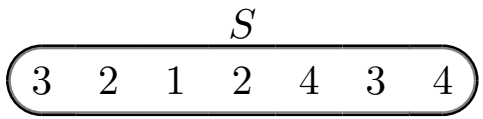

- Mapeamentos que só diferem por um bit são vizinhos.

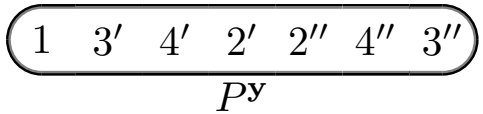
# Heurísticas Utilizando Mapeamentos

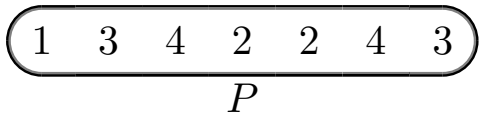


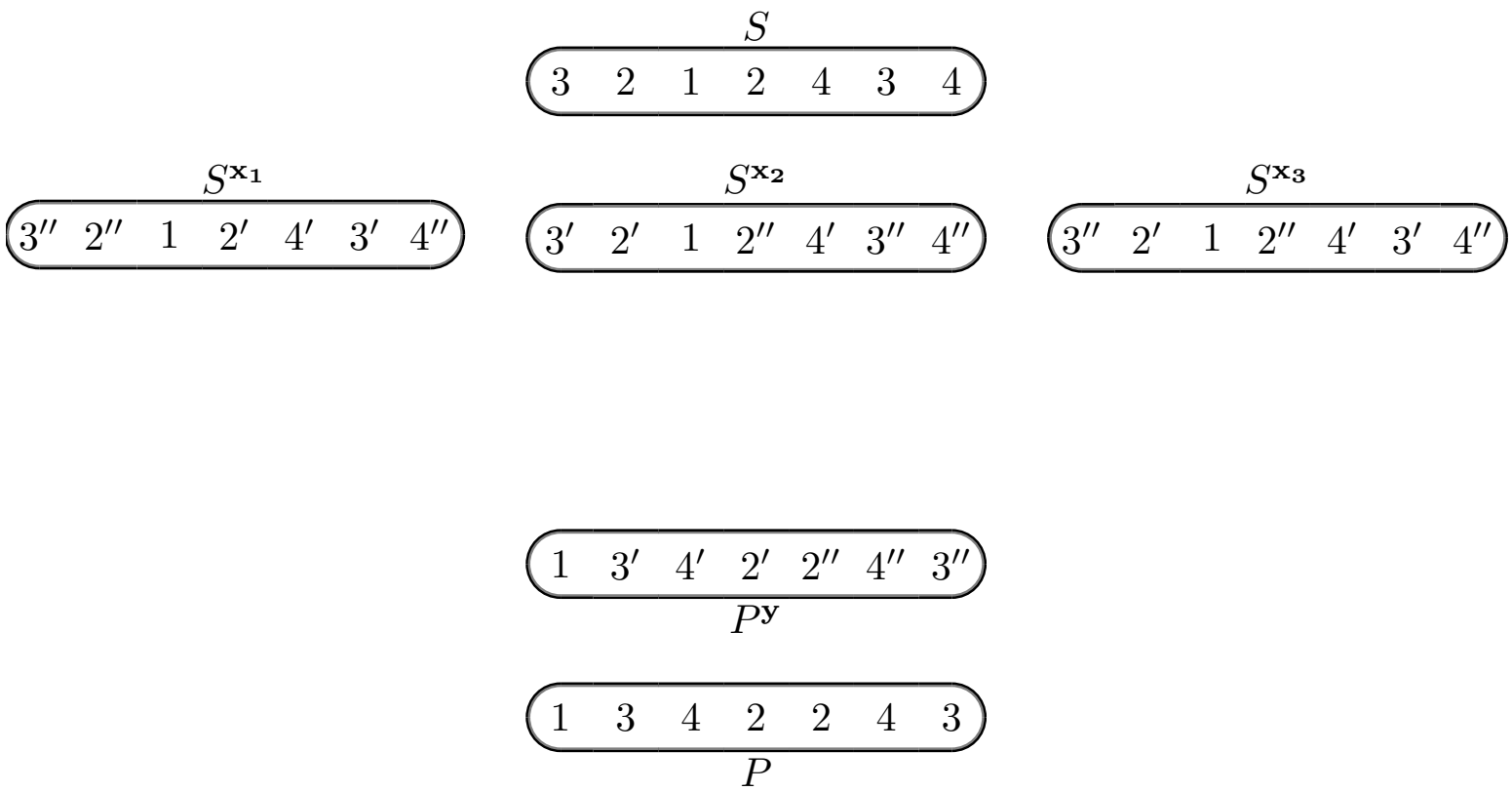
# Heurísticas Utilizando Mapeamentos



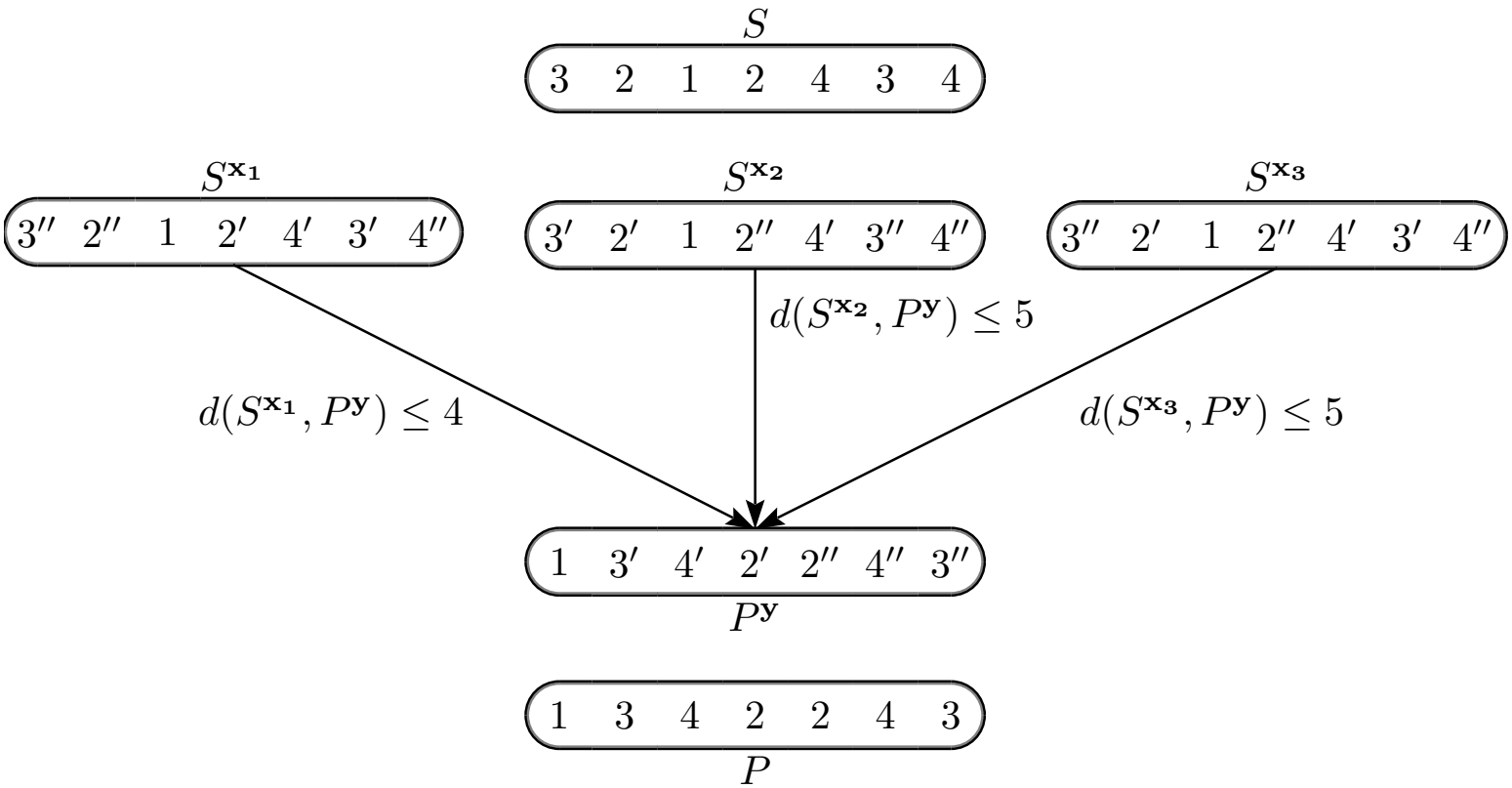



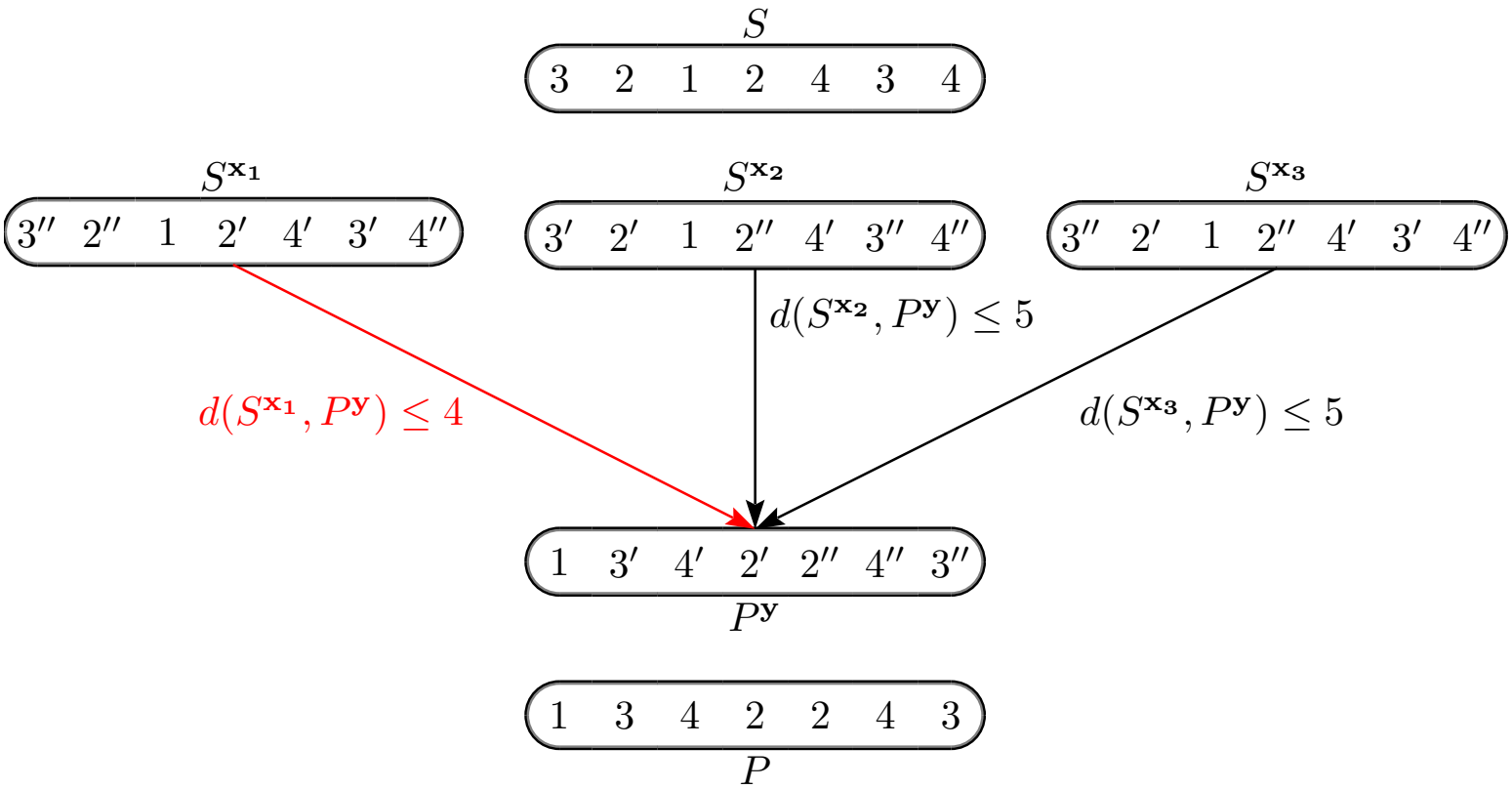
# Heurísticas Utilizando Mapeamentos



# Heurísticas Utilizando Mapeamentos



# Heurísticas Utilizando Mapeamentos



# Heurísticas Utilizando Mapeamentos

- Mapeamentos Aleatórios (MA).

# Heurísticas Utilizando Mapeamentos

- Mapeamentos Aleatórios (MA).
- Busca Local (BL):

# Heurísticas Utilizando Mapeamentos

- Mapeamentos Aleatórios (MA).
- Busca Local (BL):
  - Conjunto inicial de mapeamentos aleatórios.

# Heurísticas Utilizando Mapeamentos

- Mapeamentos Aleatórios (MA).
- Busca Local (BL):
  - Conjunto inicial de mapeamentos aleatórios.
  - Explora os vizinhos do melhor mapeamento conhecido até o momento.

# Heurísticas Utilizando Mapeamentos

- GRASP

# Heurísticas Utilizando Mapeamentos

- GRASP
  - Conjunto inicial de mapeamentos aleatórios.

# Heurísticas Utilizando Mapeamentos

- GRASP
  - Conjunto inicial de mapeamentos aleatórios.
  - Etapa de construção:

$$prob(\mathbf{RCL}, \alpha) = \frac{freq(\mathbf{RCL}, \alpha, 0)}{freq(\mathbf{RCL}, \alpha, 0) + freq(\mathbf{RCL}, \alpha, 1)}$$

| **RCL** | |
|---|---|
| $\mathbf{x_2} =$ <table><tr><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>0</td><td>1</td></tr></table> | $freq(\mathbf{RCL}, 2, 0) = 2, \quad freq(\mathbf{RCL}, 2, 1) = 0$ |
| $\mathbf{x_3} =$ <table><tr><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table> | $freq(\mathbf{RCL}, 3, 0) = 1, \quad freq(\mathbf{RCL}, 3, 1) = 1$ |
| | $freq(\mathbf{RCL}, 4, 0) = 1, \quad freq(\mathbf{RCL}, 4, 1) = 1$ |
| | $prob(\mathbf{RCL}, 2) = 100\%$ |
| | $prob(\mathbf{RCL}, 3) = 50\%$ |
| | $prob(\mathbf{RCL}, 4) = 50\%$ |

# Heurísticas Utilizando Mapeamentos

- GRASP
  - Conjunto inicial de mapeamentos aleatórios.
  - Etapa de construção:

$$prob(\mathbf{RCL}, \alpha) = \frac{freq(\mathbf{RCL}, \alpha, 0)}{freq(\mathbf{RCL}, \alpha, 0) + freq(\mathbf{RCL}, \alpha, 1)}$$

| **RCL** | |
|---|---|
| $\mathbf{x_2} =$ <table><tr><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>0</td><td>1</td></tr></table> | $freq(\mathbf{RCL}, 2, 0) = 2, \quad freq(\mathbf{RCL}, 2, 1) = 0$ |
| $\mathbf{x_3} =$ <table><tr><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table> | $freq(\mathbf{RCL}, 3, 0) = 1, \quad freq(\mathbf{RCL}, 3, 1) = 1$ |
| | $freq(\mathbf{RCL}, 4, 0) = 1, \quad freq(\mathbf{RCL}, 4, 1) = 1$ |
| | $prob(\mathbf{RCL}, 2) = 100\%$ |
| | $prob(\mathbf{RCL}, 3) = 50\%$ |
| | $prob(\mathbf{RCL}, 4) = 50\%$ |

  - Etapa de busca local: Adaptação da heurística anterior (BL).

# Heurísticas Utilizando Mapeamentos

- Algoritmos Genéticos (AG):

# Heurísticas Utilizando Mapeamentos

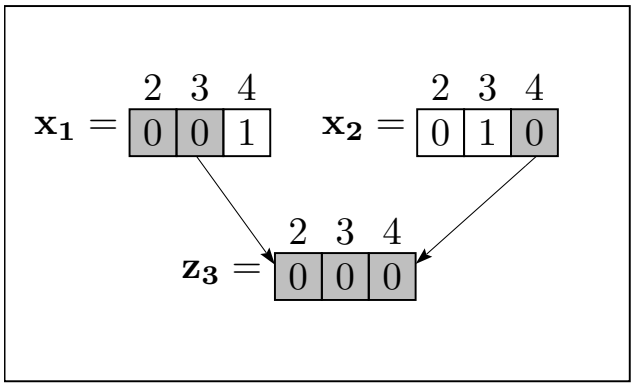
- Algoritmos Genéticos (AG):
  - População Inicial: Aleatória.

# Heurísticas Utilizando Mapeamentos

- Algoritmos Genéticos (AG):
  - População Inicial: Aleatória.
  - Seleção: $k$ melhores mapeamentos.

# Heurísticas Utilizando Mapeamentos

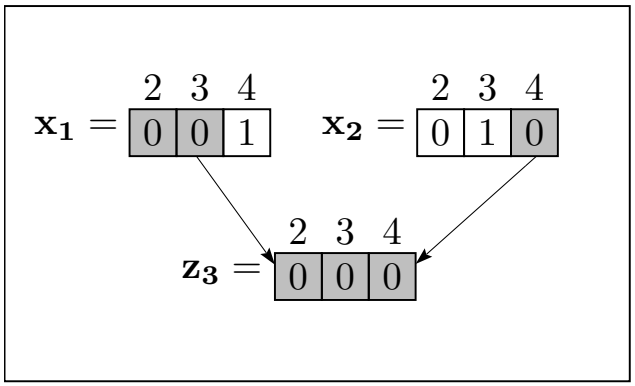
- Algoritmos Genéticos (AG):
  - População Inicial: Aleatória.
  - Seleção: $k$ melhores mapeamentos.
  - Cruzamento:



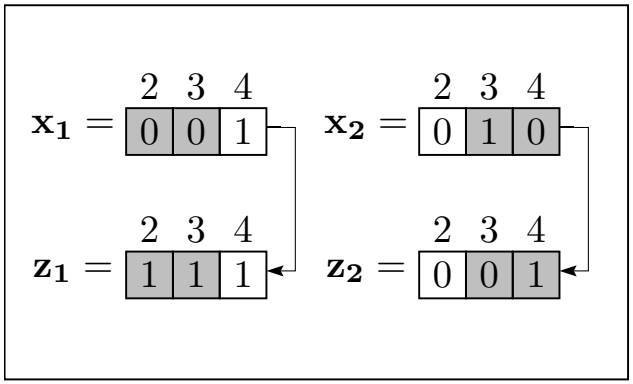
# Heurísticas Utilizando Mapeamentos

- Algoritmos Genéticos (AG):
  - População Inicial: Aleatória.
  - Seleção: $k$ melhores mapeamentos.

- Cruzamento:




- Mutação:



# Heurísticas Utilizando Mapeamentos

- Busca Tabu (BT):

# Heurísticas Utilizando Mapeamentos

- Busca Tabu (BT):
  - Parte de um mapeamento arbitrário.

# Heurísticas Utilizando Mapeamentos

- Busca Tabu (BT):
  - Parte de um mapeamento arbitrário.
  - Explora os vizinhos do mapeamento atual.

# Heurísticas Utilizando Mapeamentos

- Busca Tabu (BT):
  - Parte de um mapeamento arbitrário.
  - Explora os vizinhos do mapeamento atual.
  - Escolhe o melhor deles para ser o novo mapeamento.

# Heurísticas Utilizando Mapeamentos

- Busca Tabu (BT):
  - Parte de um mapeamento arbitrário.
  - Explora os vizinhos do mapeamento atual.
  - Escolhe o melhor deles para ser o novo mapeamento.
  - Lista tabu mantém os últimos bits trocados nos mapeamentos.

# Heurísticas Utilizando Mapeamentos

- Busca Tabu (BT):
  - Parte de um mapeamento arbitrário.
  - Explora os vizinhos do mapeamento atual.
  - Escolhe o melhor deles para ser o novo mapeamento.
  - Lista tabu mantém os últimos bits trocados nos mapeamentos.
  - Ao explorar os vizinhos, não olha para os que diferem do mapeamento atual por um bit da lista tabu.

# Heurísticas Utilizando Mapeamentos

- Simulated Annealing (SA):

# Heurísticas Utilizando Mapeamentos

- Simulated Annealing (SA):
  - Parte de um mapeamento arbitrário.

# Heurísticas Utilizando Mapeamentos

- Simulated Annealing (SA):
  - Parte de um mapeamento arbitrário.
  - Explora os vizinhos do mapeamento atual.

# Heurísticas Utilizando Mapeamentos

- Simulated Annealing (SA):
  - Parte de um mapeamento arbitrário.
  - Explora os vizinhos do mapeamento atual.
  - Para cada vizinho:

# Heurísticas Utilizando Mapeamentos

- Simulated Annealing (SA):
  - Parte de um mapeamento arbitrário.
  - Explora os vizinhos do mapeamento atual.
  - Para cada vizinho:
    - Se for o último vizinho deste mapeamento ou obtiver um resultado melhor que o do mapeamento atual, muda para este mapeamento.

# Heurísticas Utilizando Mapeamentos

- Simulated Annealing (SA):
  - Parte de um mapeamento arbitrário.
  - Explora os vizinhos do mapeamento atual.
  - Para cada vizinho:
    - Se for o último vizinho deste mapeamento ou obtiver um resultado melhor que o do mapeamento atual, muda para este mapeamento.
    - Caso contrário, muda para este mapeamento com probabilidade $exp\left(-\frac{\Delta_{score}}{k \times temp}\right)$. Onde: $\Delta_{score}$ é a diferença entre o resultado atual e o do novo mapeamento; $k \in \mathbb{R}$ é uma constante; $T \in \mathbb{R}$ é a temperatura atual.

# Heurísticas Utilizando Mapeamentos

- Simulated Annealing (SA):
  - Após um determinado número de iterações sem melhora no resultado, a temperatura é atualizada.

$$T \leftarrow \alpha T, 0 < \alpha < 1$$

# Heurística de Extremidades (Ext)

# Heurística de Extremidades (Ext)

- Ignoramos os elementos posicionados corretamente nas extremidades.

# Heurística de Extremidades (Ext)

- Ignoramos os elementos posicionados corretamente nas extremidades.
- Colocamos os elementos restantes nas extremidades com uma ou duas reversões. Utilizamos a função *rank*:

$$rank(R, S', P') = \frac{corretos(R, S', P')}{|R|}$$

# Heurística de Extremidades (Ext)

- Ignoramos os elementos posicionados corretamente nas extremidades.
- Colocamos os elementos restantes nas extremidades com uma ou duas reversões. Utilizamos a função *rank*:

$$rank(R, S', P') = \frac{corretos(R, S', P')}{|R|}$$

- Com strings com sinais uma terceira reversão pode ser usada para reverter a extremidade.

# Heurística de Extremidades (Ext)

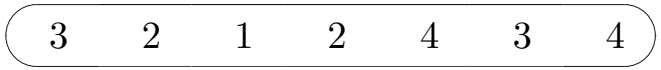
- Ignoramos os elementos posicionados corretamente nas extremidades.
- Colocamos os elementos restantes nas extremidades com uma ou duas reversões. Utilizamos a função *rank*:

$$rank(R, S', P') = \frac{corretos(R, S', P')}{|R|}$$

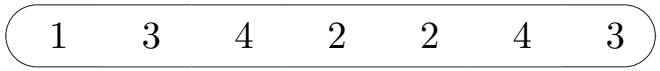
- Com strings com sinais uma terceira reversão pode ser usada para reverter a extremidade.
- Quando não temos mais nenhum caractere repetido aplicamos o algoritmo para distância entre permutações.

# Heurística de Extremidades (Ext)

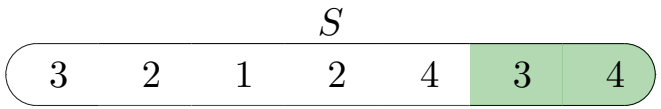
$S$

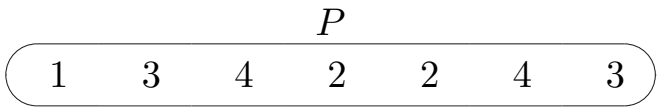



$P$

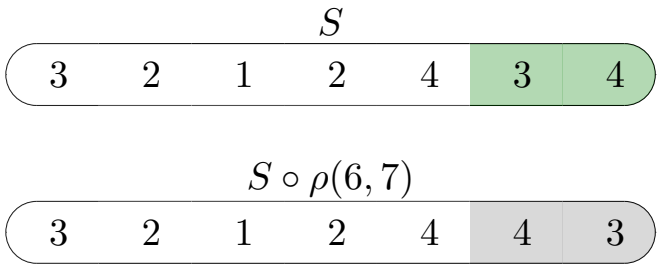


# Heurística de Extremidades (Ext)

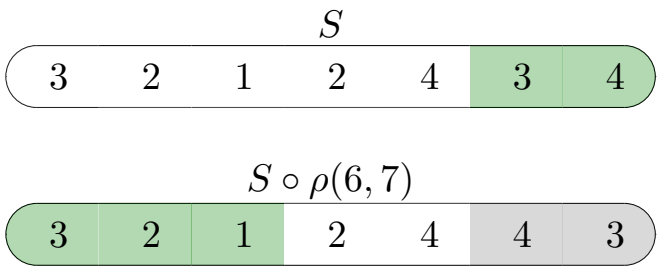


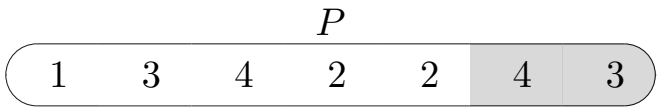



# Heurística de Extremidades (Ext)

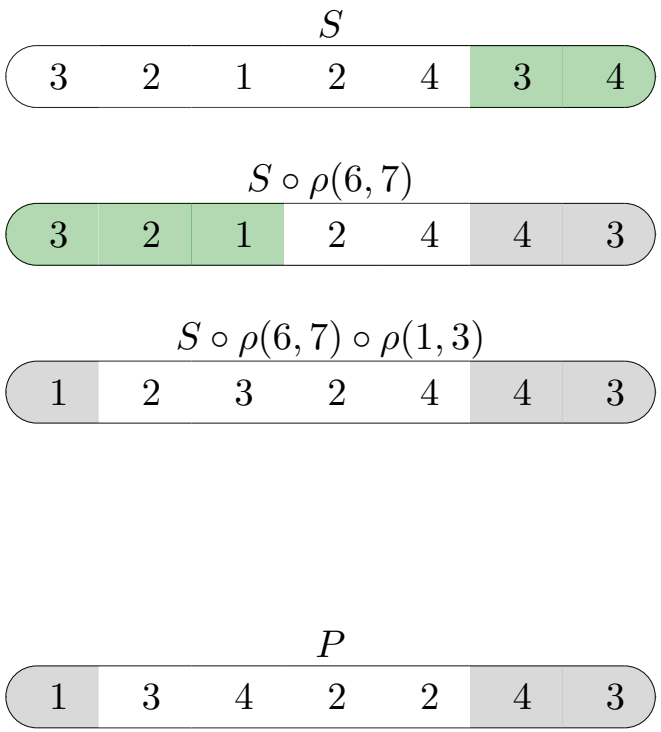


# Heurística de Extremidades (Ext)

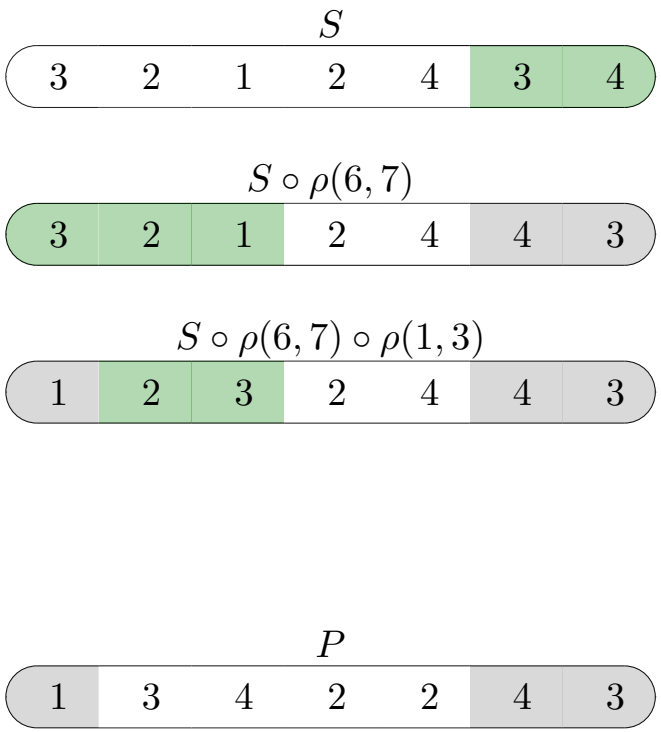


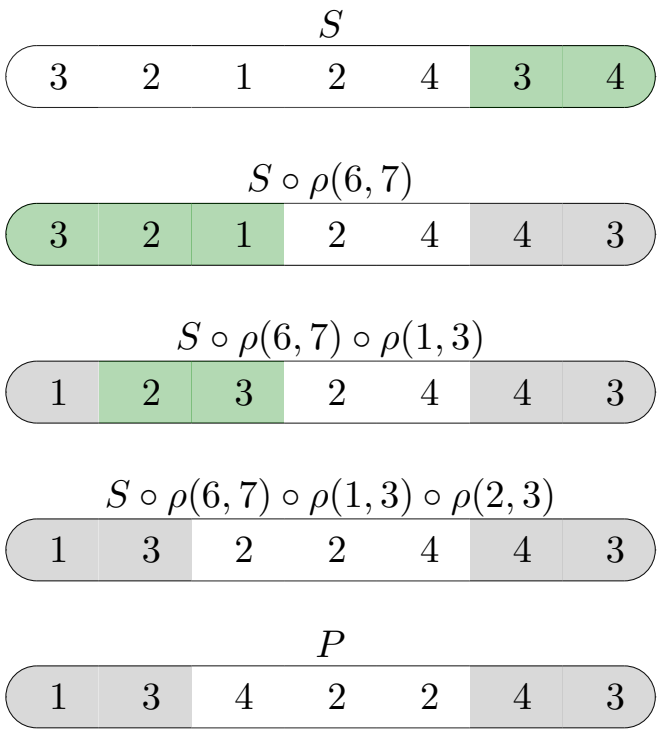
# Heurística de Extremidades (Ext)



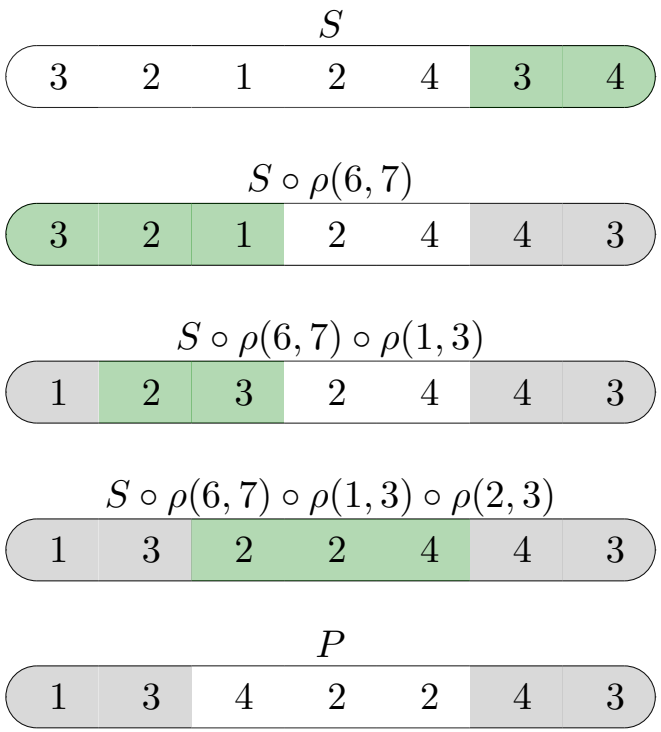
# Heurística de Extremidades (Ext)



# Heurística de Extremidades (Ext)



# Heurística de Extremidades (Ext)



# Heurística de Extremidades (Ext)

$S$

| 3 | 2 | 1 | 2 | 4 | 3 | 4 |
|---|---|---|---|---|---|---|

$S \circ \rho(6,7)$

| 3 | 2 | 1 | 2 | 4 | 4 | 3 |
|---|---|---|---|---|---|---|

$S \circ \rho(6,7) \circ \rho(1,3)$

| 1 | 2 | 3 | 2 | 4 | 4 | 3 |
|---|---|---|---|---|---|---|

$S \circ \rho(6,7) \circ \rho(1,3) \circ \rho(2,3)$

| 1 | 3 | 2 | 2 | 4 | 4 | 3 |
|---|---|---|---|---|---|---|

$P = S \circ \rho(6,7) \circ \rho(1,3) \circ \rho(2,3) \circ \rho(3,5)$

| 1 | 3 | 4 | 2 | 2 | 4 | 3 |
|---|---|---|---|---|---|---|

# Resultados

- Criamos uma base de dados para testar as heurísticas.

# Resultados

- Criamos uma base de dados para testar as heurísticas.
  - 20 grupos com 1000 pares de strings cada.

# Resultados

- Criamos uma base de dados para testar as heurísticas.
  - 20 grupos com 1000 pares de strings cada.
  - 10 grupos variando os tamanhos (100 até 1000 em intervalos de 100) e com 25% de caracteres duplicados.

# Resultados

- Criamos uma base de dados para testar as heurísticas.
  - 20 grupos com 1000 pares de strings cada.
  - 10 grupos variando os tamanhos (100 até 1000 em intervalos de 100) e com 25% de caracteres duplicados.
  - 10 grupos variando a porcentagem de caracteres duplicados (5% até 50% em intervalos de 5%) e com tamanho 500.

# Resultados

- Criamos uma base de dados para testar as heurísticas.
  - 20 grupos com 1000 pares de strings cada.
  - 10 grupos variando os tamanhos (100 até 1000 em intervalos de 100) e com 25% de caracteres duplicados.
  - 10 grupos variando a porcentagem de caracteres duplicados (5% até 50% em intervalos de 5%) e com tamanho 500.
  - Aplicamos na string origem 25% de seu tamanho em reversões para gerar a string destino.

# Resultados

- Comparamos com o algoritmo SOAR [4], que possui fator de aproximação 3 para o caso com sinais. Uma simples adaptação permite usar este algoritmo para o caso sem sinais, mas o fator de aproximação é 6.

# Resultados

- Comparamos com o algoritmo SOAR [4], que possui fator de aproximação 3 para o caso com sinais. Uma simples adaptação permite usar este algoritmo para o caso sem sinais, mas o fator de aproximação é 6.
- Calculamos uma estimativa para o erro das distâncias:

$$EED = \frac{|D_A - OP|}{OP}$$

Onde $OP$ é o número de operações aplicadas para gerar a string destino e $D_A$ é a distância encontrada pelo algoritmo (heurística ou SOAR).

# Resultados

Tabela: Média das distâncias obtidas pelas nossas heurísticas e pelo algoritmo SOAR para o problema *DR*, variando o tamanho das strings e fixando a porcentagem de caracteres duplicados em 25%.

| Tamanho da String | OP | MA | BL | GRASP | AG | BT | SA | Ext | SOAR |
|---|---|---|---|---|---|---|---|---|---|
| **100** | 25 | 32.622 | 24.221 | 24.221 | 24.330 | 25.040 | 24.210 | 40.263 | 29.649 |
| **200** | 50 | 75.310 | 49.609 | 49.609 | 49.754 | 50.746 | 49.703 | 88.928 | 61.423 |
| **300** | 75 | 120.937 | 75.388 | 75.370 | 75.602 | 76.989 | 75.453 | 137.788 | 94.158 |
| **400** | 100 | 167.302 | 100.989 | 100.902 | 101.300 | 104.097 | 101.125 | 187.646 | 126.339 |
| **500** | 125 | 214.613 | 126.789 | 126.564 | 127.132 | 131.977 | 126.941 | 237.501 | 158.971 |
| **600** | 150 | 262.344 | 152.743 | 152.050 | 152.907 | 162.439 | 153.111 | 288.311 | 191.212 |
| **700** | 175 | 310.991 | 179.640 | 177.967 | 179.171 | 195.072 | 180.650 | 338.460 | 224.106 |
| **800** | 200 | 359.602 | 207.730 | 203.833 | 205.327 | 234.159 | 210.680 | 388.726 | 256.414 |
| **900** | 225 | 408.307 | 238.634 | 229.581 | 231.809 | 276.216 | 246.230 | 438.124 | 289.041 |
| **1000** | 250 | 457.218 | 274.032 | 255.768 | 258.538 | 326.164 | 288.486 | 488.634 | 321.878 |
| **$EED_{med}$(%)** | | 67.812 | 3.872 | 2.402 | 2.878 | 11.329 | 5.052 | 87.043 | 26.195 |

# Resultados

Tabela: Média das distâncias obtidas pelas nossas heurísticas e pelo algoritmo SOAR para o problema $D\bar{R}$, variando o tamanho das strings e fixando a porcentagem de caracteres duplicados em 25%.

| Tamanho da String | OP | MA | BL | GRASP | AG | BT | SA | Ext | SOAR |
|---|---|---|---|---|---|---|---|---|---|
| **100** | 25 | 33.887 | 24.713 | 24.713 | 24.713 | 25.600 | 24.713 | 46.510 | 24.729 |
| **200** | 50 | 75.538 | 49.735 | 49.735 | 49.735 | 50.718 | 49.735 | 107.418 | 49.745 |
| **300** | 75 | 119.025 | 74.742 | 74.742 | 74.744 | 75.707 | 74.742 | 173.157 | 74.757 |
| **400** | 100 | 163.337 | 99.712 | 99.712 | 99.712 | 100.679 | 99.712 | 237.756 | 99.726 |
| **500** | 125 | 208.529 | 124.733 | 124.732 | 124.732 | 125.752 | 124.732 | 307.454 | 124.741 |
| **600** | 150 | 254.094 | 149.730 | 149.720 | 149.722 | 150.901 | 149.720 | 373.993 | 149.740 |
| **700** | 175 | 299.666 | 174.758 | 174.742 | 174.742 | 176.120 | 174.742 | 441.690 | 174.770 |
| **800** | 200 | 345.839 | 199.786 | 199.737 | 199.737 | 201.458 | 199.737 | 517.942 | 199.748 |
| **900** | 225 | 392.036 | 224.786 | 224.721 | 224.721 | 226.960 | 224.721 | 589.635 | 224.734 |
| **1000** | 250 | 438.758 | 249.841 | 249.731 | 249.731 | 252.324 | 249.733 | 655.677 | 249.740 |
| **$EED_{med}$(%)** | | 63.878 | 0.332 | 0.322 | 0.323 | 1.336 | 0.322 | 140.050 | 0.334 |

# Resultados

Tabela: Média das distâncias obtidas pelas nossas heurísticas e pelo algoritmo SOAR para o problema *DR*, variando a porcentagem de caracteres duplicados e fixando o tamanho das strings em 500.

| Porcentagem de Caracteres Duplicados | OP | MA | BL | GRASP | AG | BT | SA | Ext | SOAR |
|---|---|---|---|---|---|---|---|---|---|
| **5%** | 125 | 134.343 | 126.937 | 126.947 | 126.946 | 128.099 | 126.941 | 220.849 | 158.428 |
| **10%** | 125 | 152.601 | 126.722 | 126.719 | 126.795 | 128.014 | 126.880 | 226.814 | 159.023 |
| **15%** | 125 | 172.652 | 126.650 | 126.619 | 126.793 | 128.299 | 126.776 | 231.217 | 158.640 |
| **20%** | 125 | 193.402 | 126.610 | 126.486 | 126.784 | 129.582 | 126.652 | 235.371 | 159.100 |
| **25%** | 125 | 214.370 | 126.737 | 126.539 | 127.052 | 132.199 | 126.979 | 237.777 | 158.622 |
| **30%** | 125 | 235.559 | 127.136 | 126.546 | 128.112 | 136.713 | 127.332 | 240.985 | 158.621 |
| **35%** | 125 | 256.117 | 128.233 | 126.855 | 131.490 | 151.481 | 128.825 | 243.008 | 158.987 |
| **40%** | 125 | 275.992 | 130.624 | 127.535 | 138.476 | 176.440 | 134.311 | 247.298 | 159.089 |
| **45%** | 125 | 295.152 | 136.610 | 129.276 | 147.987 | 206.944 | 151.521 | 249.715 | 158.916 |
| **50%** | 125 | 313.842 | 149.244 | 133.305 | 158.778 | 238.118 | 187.979 | 252.684 | 159.083 |
| **$EED_{med}$(%)** | 88.358 | 5.307 | 2.858 | 8.215 | 27.299 | 10.579 | 100.952 | 47.968 | |

# Resultados

Tabela: Média das distâncias obtidas pelas nossas heurísticas e pelo algoritmo SOAR para o problema $D\bar{R}$, variando a porcentagem de caracteres duplicados e fixando o tamanho das strings em 500.

| Porcentagem de Caracteres Duplicados | OP | MA | BL | GRASP | AG | BT | SA | Ext | SOAR |
|---|---|---|---|---|---|---|---|---|---|
| **5%** | 125 | 132.174 | 124.738 | 124.738 | 124.738 | 125.780 | 124.738 | 309.498 | 124.738 |
| **10%** | 125 | 149.082 | 124.731 | 124.731 | 124.731 | 125.705 | 124.731 | 311.776 | 124.732 |
| **15%** | 125 | 168.042 | 124.714 | 124.714 | 124.714 | 125.721 | 124.714 | 311.453 | 124.718 |
| **20%** | 125 | 188.064 | 124.723 | 124.723 | 124.723 | 125.665 | 124.723 | 306.794 | 124.732 |
| **25%** | 125 | 208.490 | 124.772 | 124.770 | 124.770 | 125.787 | 124.770 | 306.136 | 124.777 |
| **30%** | 125 | 229.110 | 124.770 | 124.725 | 124.727 | 126.419 | 124.725 | 307.132 | 124.733 |
| **35%** | 125 | 250.342 | 124.927 | 124.740 | 124.739 | 133.672 | 124.735 | 306.883 | 124.753 |
| **40%** | 125 | 271.571 | 125.371 | 124.725 | 124.735 | 151.795 | 124.705 | 306.229 | 124.742 |
| **45%** | 125 | 293.122 | 126.253 | 124.753 | 124.810 | 174.074 | 124.730 | 307.330 | 124.765 |
| **50%** | 125 | 313.862 | 128.189 | 124.818 | 124.908 | 195.524 | 124.717 | 310.325 | 124.766 |
| **$EED_{med}$(%)** | 84.787 | 0.682 | 0.253 | 0.264 | 14.396 | 0.243 | 162.983 | 0.253 | |

# Referências I

[1] P. Berman, S. Hannenhalli, and M. Karpinski.
1.375-Approximation Algorithm for Sorting by Reversals.
In *Proceedings of the 10th Annual European Symposium on Algorithms (ESA'2002)*, ESA'2002, pages 200–210, London, UK, 2002. Springer-Verlag.

[2] L. Bulteau, G. Fertin, and I. Rusu.
Sorting by Transpositions is Difficult.
*SIAM Journal on Computing*, 26(3):1148–1180, 2012.

[3] A. Caprara.
Sorting Permutations by Reversals and Eulerian Cycle Decompositions.
*SIAM Journal on Discrete Mathematics*, 12(1):91–110, 1999.

# Referências II

[4] X. Chen, J. Zheng, Z. Fu, P. Nan, Y. Zhong, S. Lonardi, and T. Jiang.
Assignment of Orthologous Genes via Genome Rearrangement.
*IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(4):302–315, 2005.

[5] S. Hannenhalli and P. A. Pevzner.
Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals.
*Journal of ACM*, 46(1):1–27, 1999.

# Referências III

[6] P. Kolman and T. Waleń.
Reversal Distance for Strings with Duplicates: Linear Time Approximation Using Hitting Set.
In *Proceedings of the 4th International Workshop on Approximation and Online Algorithms (WAOA'2006)*, pages 279–289, Berlin, Heidelberg, 2007.

[7] A. R. Oliveira, K. L. Brito, U. Dias, and Z. Dias.
On the complexity of sorting by reversals and transpositions problems.
*Journal of Computational Biology*.
PMID: 31120331.

# Referências IV

[8] A. J. Radcliffe, A. D. Scott, and E. L. Wilmer.
Reversals and Transpositions Over Finite Alphabets.
*SIAM Journal on Discrete Mathematics*, 19(1):224–244, 2005.

[9] A. Rahman, S. Shatabda, and M. Hasan.
An Approximation Algorithm for Sorting by Reversals and Transpositions.
*Journal of Discrete Algorithms*, 6(3):449–457, 2008.

[10] M. E. M. T. Walter, Z. Dias, and J. Meidanis.
Reversal and Transposition Distance of Linear Chromosomes.
In *Proceedings of the 5th International Symposium on String Processing and Information Retrieval (SPIRE'1998)*, pages 96–102, Los Alamitos, CA, USA, 1998.

# Heurísticas para Rearranjo de Genomas com Genes Duplicados

Gabriel Henriques Siqueira

Orientador: Prof. Dr. Zanoni Dias

Universidade Estadual de Campinas

07 de outubro de 2019

# Roteiro

Motivação

Conceitos

Objetivos

Metodologia

Cronograma

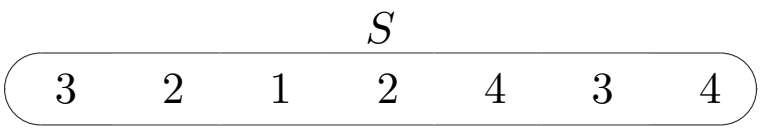
Resultados Preliminares

# Motivação

- Determinar distância evolutiva entre genomas.
- Geralmente se assume que não existem repetições de genes.
- Iremos propor heurísticas para o caso em que os genes podem estar duplicados.
- Primeiro passo para entender o caso com genes multiplicados.

# Representação por Strings

- Representamos um genoma por uma string $S$.
- Cada caractere corresponde a um gene (ou bloco conservado de genes).
- A orientação dos genes é representada por um sinal $+$ ou $-$.
- Algumas notações:
  - $|S|$ = número de caracteres em $S$.
  - $\Sigma_S$ = conjunto de caracteres de S.
  - $dup(S)$ = conjunto de caracteres duplicados de S.




$|S| = 7 \quad \Sigma_S = \{1, 2, 3, 4\}$
$dup(S) = \{2, 3, 4\}$

# Representação por Strings

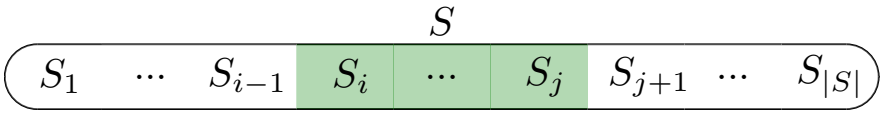
- Representamos um genoma por uma string $S$.
- Cada caractere corresponde a um gene (ou bloco conservado de genes).
- A orientação dos genes é representada por um sinal $+$ ou $-$.
- Algumas notações:
  - $|S|$ = número de caracteres em $S$.
  - $\Sigma_S$ = conjunto de caracteres de S.
  - $dup(S)$ = conjunto de caracteres duplicados de S.

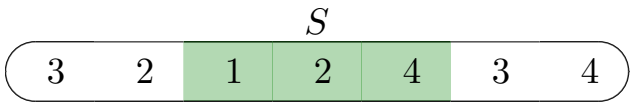
$$S$$
$$+3 \quad +2 \quad +1 \quad -2 \quad +4 \quad +3 \quad -4$$

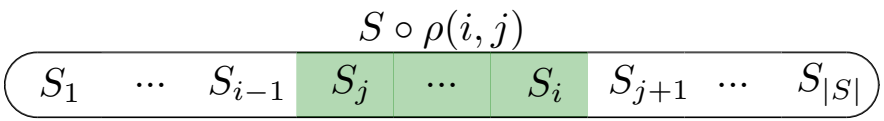
$$|S| = 7 \quad \Sigma_S = \{1, 2, 3, 4\}$$
$$dup(S) = \{2, 3, 4\}$$

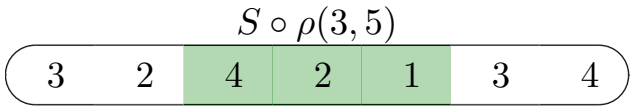
# Eventos de Rearranjo

- Reversão:

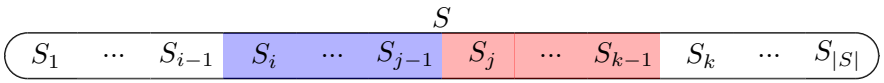


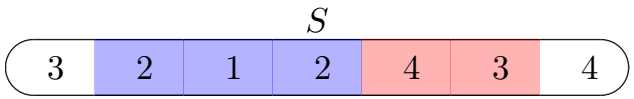



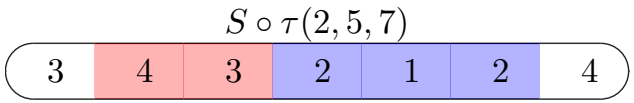




- Transposição:



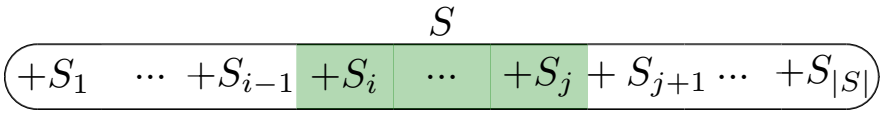



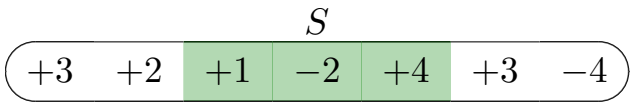


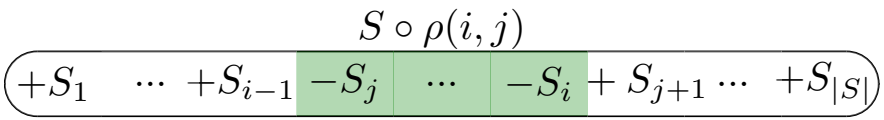


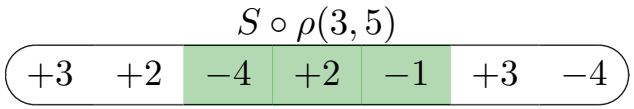
# Eventos de Rearranjo

- Reversão:

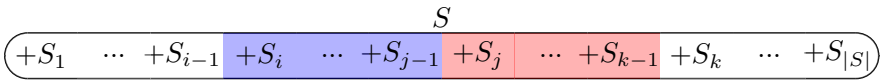


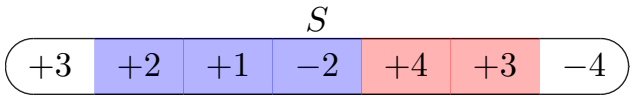



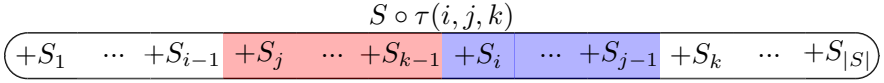



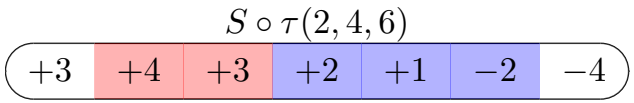




- Transposição:



# Eventos de Rearranjo

- Modelo de rearranjo: conjunto de eventos de rearranjo permitidos.
- Distância de rearranjo ($d_{\mathcal{M}}(S, P)$): menor número de operações correspondentes a rearranjos do modelo $\mathcal{M}$ necessárias para transformar $S$ em $P$.

# Resultados Conhecidos

- Distância de Reversão sem sinal:
  - NP-Difícil [3];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 4.4148 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].
- Distância de Reversão com sinal:
  - Polinomial sem caracteres repetidos [5];
  - NP-Difícil com caracteres repetidos [8] e com caracteres duplicados [4];
  - Aproximação de 2.2074 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].

# Resultados Conhecidos

- Distância de Transposição:
  - NP-Difícil [2];
  - Aproximação de 1.375 sem caracteres repetidos [1];
  - Aproximação de 3.311 com caracteres duplicados [**?**];
  - Aproximação de $\theta(k)$, onde $k$ é o número máximo de cópias de um caractere [6].
- Distância de Reversão e Transposição:
  - NP-Difícil com e sem sinais [7];
  - Aproximação de $2k$ ($k$ é a aproximação do algoritmo usado para decomposição em ciclos) sem caracteres repetidos e sem sinais [9];
  - Aproximação de 2 sem caracteres repetidos e com sinais [10].

# Objetivos

O objetivo é estudar os seguintes problemas:

- Distância de Reversão em Strings sem Sinais e com Genes Duplicados ($DR$);
- Distância de Reversão em Strings com Sinais e com Genes Duplicados ($D\bar{R}$);
- Distância de Transposição em Strings sem Sinais e com Genes Duplicados ($DT$);
- Distância de Reversão e Transposição em Strings sem Sinais e com Genes Duplicados ($DRT$).
- Distância de Reversão e Transposição em Strings com Sinais e com Genes Duplicados ($D\bar{R}T$);

# Metodologia

- Desenvolvimento de Heurísticas para cada problema.
- Heurísticas baseadas em características comuns a todos os problemas.
- Heurísticas baseadas em características especificas de cada problema.
- Criação de uma base de dados.
- Comparação entre as heurísticas e com resultados da literatura.

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

1. Revisão da literatura;
2. Escrita da proposta de mestrado;
3. Exame de Qualificação de Mestrado (EQM);
4. Participação no Programa de Estágio Docente (PED);

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

5. Investigação do problema $DR$;
6. Investigação do problema $D\bar{R}$;
7. Investigação do problema $DT$;
8. Investigação do problema $DRT$;
9. Investigação do problema $D\bar{R}T$;

# Cronograma das Atividades

| | 2019 | | | | | | | | | | 2020 | | | | | | | | | | | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D | J | F |
| 1 | * | * | * | * | * | | | | * | | | | * | | | | * | | | | | | | |
| 2 | | | | | * | * | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | * | | | | | | | | | | | | | | | | |
| 4 | | | | | | * | * | * | * | * | | | | | | | | | | | | | | |
| 5 | * | * | * | | | * | * | * | * | | | | | | | | | | | | | | | |
| 6 | | | * | * | * | * | * | * | * | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | * | * | * | * | * | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | * | * | * | * | * | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | * | * | * | * | * | | | |
| 10 | | | | * | * | | | * | * | | | * | * | | | * | * | | | * | * | | | |
| 11 | | | | | * | | | | * | | | | * | | | | * | | | | * | * | | |
| 12 | | | | | | | | | | | | | | | | | | | | | | * | * | * |
| 13 | | | | | | | | | | | | | | | | | | | | | | | | * |

10. Execução dos experimentos e comparação dos resultados;
11. Escrita da dissertação;
12. Revisão da dissertação;
13. Defesa da dissertação.

# Mapeamentos em Permutações

- Mapeamos as strings em permutações.
- Assim, podemos usar resultados já conhecidos para permutações.

$$S$$

| 3 | 2 | 1 | 2 | 4 | 3 | 4 |
|---|---|---|---|---|---|---|

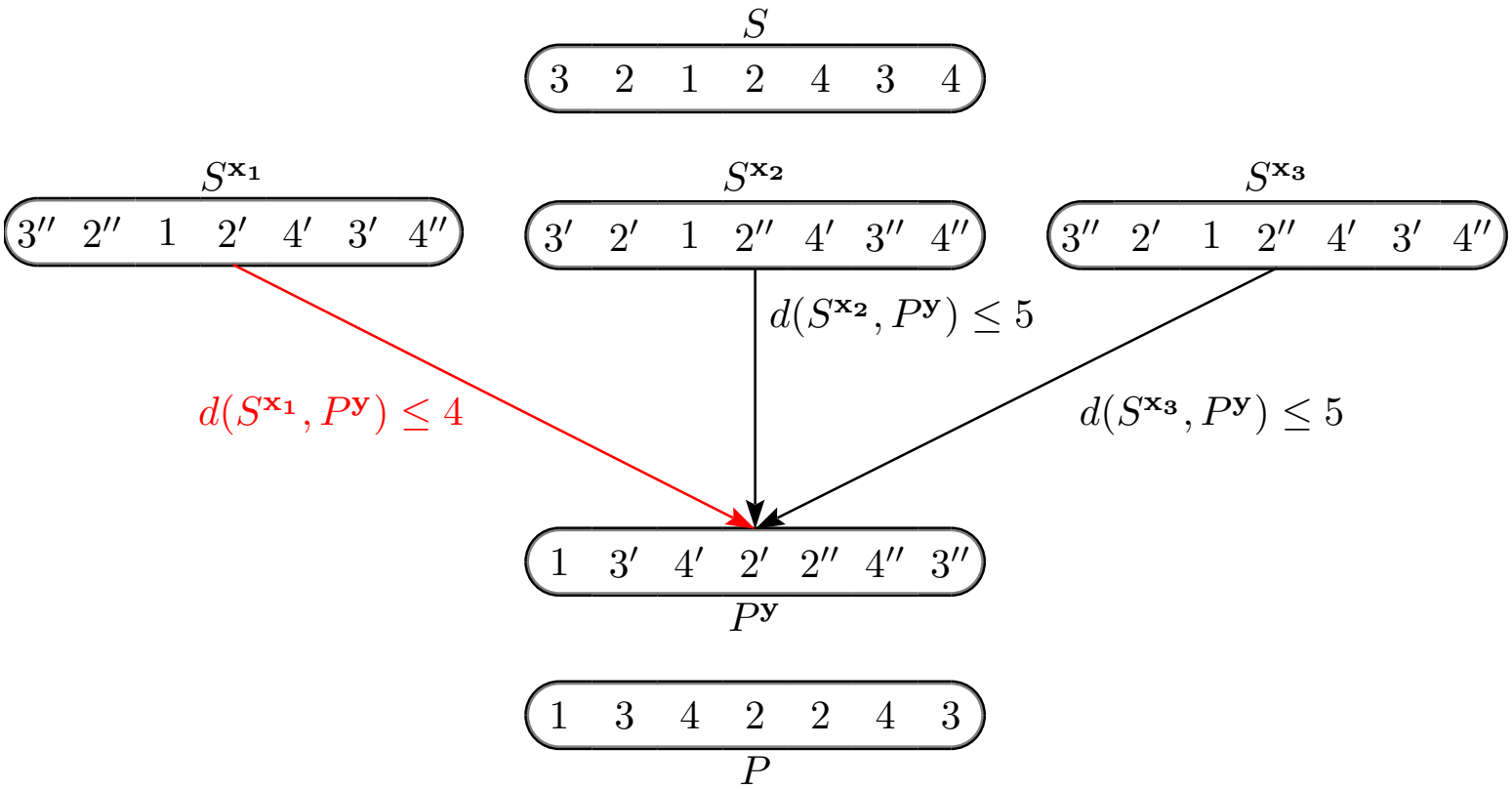
$$S^{\mathbf{x}}$$

| $3''$ | $2'$ | 1 | $2''$ | $4''$ | $3'$ | $4'$ |
|---|---|---|---|---|---|---|

$$\mathbf{x} = \begin{array}{|c|c|c|} \hline 2 & 3 & 4 \\ \hline 0 & 1 & 1 \\ \hline \end{array}$$

- Mapeamentos que só diferem por um bit são vizinhos.

# Heurísticas Utilizando Mapeamentos



# Heurísticas Utilizando Mapeamentos

- Mapeamentos Aleatórios (MA).
- Busca Local (BL):
  - Conjunto inicial de mapeamentos aleatórios.
  - Explora os vizinhos do melhor mapeamento conhecido até o momento.

# Heurísticas Utilizando Mapeamentos

- GRASP
  - Conjunto inicial de mapeamentos aleatórios.
  - Etapa de construção:

$$prob(\mathbf{RCL}, \alpha) = \frac{freq(\mathbf{RCL}, \alpha, 0)}{freq(\mathbf{RCL}, \alpha, 0) + freq(\mathbf{RCL}, \alpha, 1)}$$

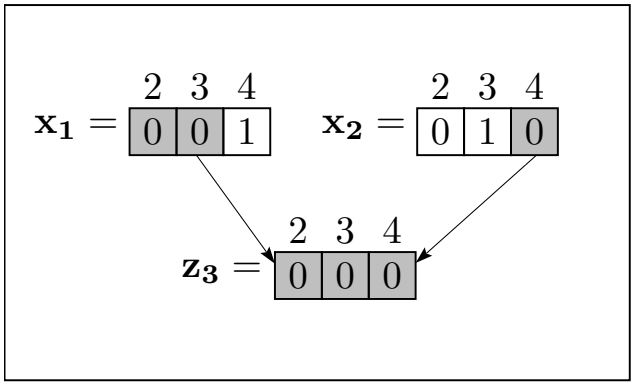
| **RCL** | |
|---|---|
| $\mathbf{x_2} =$ <table><tr><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>0</td><td>1</td></tr></table> | $freq(\mathbf{RCL}, 2, 0) = 2, \quad freq(\mathbf{RCL}, 2, 1) = 0$ |
| $\mathbf{x_3} =$ <table><tr><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table> | $freq(\mathbf{RCL}, 3, 0) = 1, \quad freq(\mathbf{RCL}, 3, 1) = 1$ |
| | $freq(\mathbf{RCL}, 4, 0) = 1, \quad freq(\mathbf{RCL}, 4, 1) = 1$ |
| | $prob(\mathbf{RCL}, 2) = 100\%$ |
| | $prob(\mathbf{RCL}, 3) = 50\%$ |
| | $prob(\mathbf{RCL}, 4) = 50\%$ |

  - Etapa de busca local: Adaptação da heurística anterior (BL).

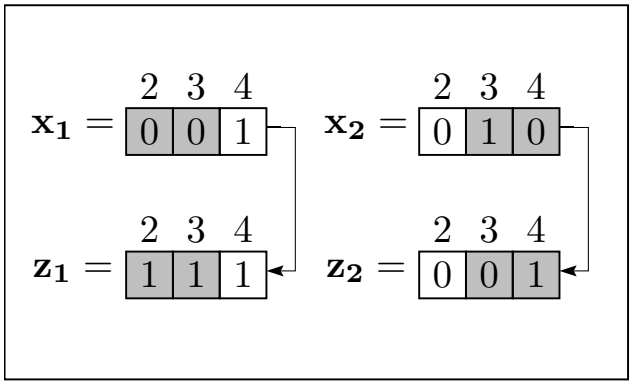
# Heurísticas Utilizando Mapeamentos

- Algoritmos Genéticos (AG):
  - População Inicial: Aleatória.
  - Seleção: $k$ melhores mapeamentos.

- Cruzamento:




- Mutação:



# Heurísticas Utilizando Mapeamentos

- Busca Tabu (BT):
  - Parte de um mapeamento arbitrário.
  - Explora os vizinhos do mapeamento atual.
  - Escolhe o melhor deles para ser o novo mapeamento.
  - Lista tabu mantém os últimos bits trocados nos mapeamentos.
  - Ao explorar os vizinhos, não olha para os que diferem do mapeamento atual por um bit da lista tabu.

# Heurísticas Utilizando Mapeamentos

- Simulated Annealing (SA):
  - Parte de um mapeamento arbitrário.
  - Explora os vizinhos do mapeamento atual.
  - Para cada vizinho:
    - Se for o último vizinho deste mapeamento ou obtiver um resultado melhor que o do mapeamento atual, muda para este mapeamento.
    - Caso contrário, muda para este mapeamento com probabilidade $exp\left(-\frac{\Delta_{score}}{k \times temp}\right)$. Onde: $\Delta_{score}$ é a diferença entre o resultado atual e o do novo mapeamento; $k \in \mathbb{R}$ é uma constante; $T \in \mathbb{R}$ é a temperatura atual.

# Heurísticas Utilizando Mapeamentos

- Simulated Annealing (SA):
  - Após um determinado número de iterações sem melhora no resultado, a temperatura é atualizada.

$$T \leftarrow \alpha T, 0 < \alpha < 1$$

# Heurística de Extremidades (Ext)

- Ignoramos os elementos posicionados corretamente nas extremidades.
- Colocamos os elementos restantes nas extremidades com uma ou duas reversões. Utilizamos a função *rank*:

$$rank(R, S', P') = \frac{corretos(R, S', P')}{|R|}$$

- Com strings com sinais uma terceira reversão pode ser usada para reverter a extremidade.
- Quando não temos mais nenhum caractere repetido aplicamos o algoritmo para distância entre permutações.

# Heurística de Extremidades (Ext)

$S$

| 3 | 2 | 1 | 2 | 4 | 3 | 4 |
|---|---|---|---|---|---|---|

$S \circ \rho(6,7)$

| 3 | 2 | 1 | 2 | 4 | 4 | 3 |
|---|---|---|---|---|---|---|

$S \circ \rho(6,7) \circ \rho(1,3)$

| 1 | 2 | 3 | 2 | 4 | 4 | 3 |
|---|---|---|---|---|---|---|

$S \circ \rho(6,7) \circ \rho(1,3) \circ \rho(2,3)$

| 1 | 3 | 2 | 2 | 4 | 4 | 3 |
|---|---|---|---|---|---|---|

$P = S \circ \rho(6,7) \circ \rho(1,3) \circ \rho(2,3) \circ \rho(3,5)$

| 1 | 3 | 4 | 2 | 2 | 4 | 3 |
|---|---|---|---|---|---|---|

# Resultados

- Criamos uma base de dados para testar as heurísticas.
  - 20 grupos com 1000 pares de strings cada.
  - 10 grupos variando os tamanhos (100 até 1000 em intervalos de 100) e com 25% de caracteres duplicados.
  - 10 grupos variando a porcentagem de caracteres duplicados (5% até 50% em intervalos de 5%) e com tamanho 500.
  - Aplicamos na string origem 25% de seu tamanho em reversões para gerar a string destino.

# Resultados

- Comparamos com o algoritmo SOAR [4], que possui fator de aproximação 3 para o caso com sinais. Uma simples adaptação permite usar este algoritmo para o caso sem sinais, mas o fator de aproximação é 6.
- Calculamos uma estimativa para o erro das distâncias:

$$EED = \frac{|D_A - OP|}{OP}$$

Onde $OP$ é o número de operações aplicadas para gerar a string destino e $D_A$ é a distância encontrada pelo algoritmo (heurística ou SOAR).

# Resultados

Tabela: Média das distâncias obtidas pelas nossas heurísticas e pelo algoritmo SOAR para o problema *DR*, variando o tamanho das strings e fixando a porcentagem de caracteres duplicados em 25%.

| Tamanho da String | OP | MA | BL | GRASP | AG | BT | SA | Ext | SOAR |
|---|---|---|---|---|---|---|---|---|---|
| **100** | 25 | 32.622 | 24.221 | 24.221 | 24.330 | 25.040 | 24.210 | 40.263 | 29.649 |
| **200** | 50 | 75.310 | 49.609 | 49.609 | 49.754 | 50.746 | 49.703 | 88.928 | 61.423 |
| **300** | 75 | 120.937 | 75.388 | 75.370 | 75.602 | 76.989 | 75.453 | 137.788 | 94.158 |
| **400** | 100 | 167.302 | 100.989 | 100.902 | 101.300 | 104.097 | 101.125 | 187.646 | 126.339 |
| **500** | 125 | 214.613 | 126.789 | 126.564 | 127.132 | 131.977 | 126.941 | 237.501 | 158.971 |
| **600** | 150 | 262.344 | 152.743 | 152.050 | 152.907 | 162.439 | 153.111 | 288.311 | 191.212 |
| **700** | 175 | 310.991 | 179.640 | 177.967 | 179.171 | 195.072 | 180.650 | 338.460 | 224.106 |
| **800** | 200 | 359.602 | 207.730 | 203.833 | 205.327 | 234.159 | 210.680 | 388.726 | 256.414 |
| **900** | 225 | 408.307 | 238.634 | 229.581 | 231.809 | 276.216 | 246.230 | 438.124 | 289.041 |
| **1000** | 250 | 457.218 | 274.032 | 255.768 | 258.538 | 326.164 | 288.486 | 488.634 | 321.878 |
| **$EED_{med}$(%)** | | 67.812 | 3.872 | 2.402 | 2.878 | 11.329 | 5.052 | 87.043 | 26.195 |

# Resultados

Tabela: Média das distâncias obtidas pelas nossas heurísticas e pelo algoritmo SOAR para o problema $D\bar{R}$, variando o tamanho das strings e fixando a porcentagem de caracteres duplicados em 25%.

| Tamanho da String | OP | MA | BL | GRASP | AG | BT | SA | Ext | SOAR |
|---|---|---|---|---|---|---|---|---|---|
| **100** | 25 | 33.887 | 24.713 | 24.713 | 24.713 | 25.600 | 24.713 | 46.510 | 24.729 |
| **200** | 50 | 75.538 | 49.735 | 49.735 | 49.735 | 50.718 | 49.735 | 107.418 | 49.745 |
| **300** | 75 | 119.025 | 74.742 | 74.742 | 74.744 | 75.707 | 74.742 | 173.157 | 74.757 |
| **400** | 100 | 163.337 | 99.712 | 99.712 | 99.712 | 100.679 | 99.712 | 237.756 | 99.726 |
| **500** | 125 | 208.529 | 124.733 | 124.732 | 124.732 | 125.752 | 124.732 | 307.454 | 124.741 |
| **600** | 150 | 254.094 | 149.730 | 149.720 | 149.722 | 150.901 | 149.720 | 373.993 | 149.740 |
| **700** | 175 | 299.666 | 174.758 | 174.742 | 174.742 | 176.120 | 174.742 | 441.690 | 174.770 |
| **800** | 200 | 345.839 | 199.786 | 199.737 | 199.737 | 201.458 | 199.737 | 517.942 | 199.748 |
| **900** | 225 | 392.036 | 224.786 | 224.721 | 224.721 | 226.960 | 224.721 | 589.635 | 224.734 |
| **1000** | 250 | 438.758 | 249.841 | 249.731 | 249.731 | 252.324 | 249.733 | 655.677 | 249.740 |
| **$EED_{med}$(%)** | | 63.878 | 0.332 | 0.322 | 0.323 | 1.336 | 0.322 | 140.050 | 0.334 |

# Resultados

Tabela: Média das distâncias obtidas pelas nossas heurísticas e pelo algoritmo SOAR para o problema *DR*, variando a porcentagem de caracteres duplicados e fixando o tamanho das strings em 500.

| Porcentagem de Caracteres Duplicados | OP | MA | BL | GRASP | AG | BT | SA | Ext | SOAR |
|---|---|---|---|---|---|---|---|---|---|
| **5%** | 125 | 134.343 | 126.937 | 126.947 | 126.946 | 128.099 | 126.941 | 220.849 | 158.428 |
| **10%** | 125 | 152.601 | 126.722 | 126.719 | 126.795 | 128.014 | 126.880 | 226.814 | 159.023 |
| **15%** | 125 | 172.652 | 126.650 | 126.619 | 126.793 | 128.299 | 126.776 | 231.217 | 158.640 |
| **20%** | 125 | 193.402 | 126.610 | 126.486 | 126.784 | 129.582 | 126.652 | 235.371 | 159.100 |
| **25%** | 125 | 214.370 | 126.737 | 126.539 | 127.052 | 132.199 | 126.979 | 237.777 | 158.622 |
| **30%** | 125 | 235.559 | 127.136 | 126.546 | 128.112 | 136.713 | 127.332 | 240.985 | 158.621 |
| **35%** | 125 | 256.117 | 128.233 | 126.855 | 131.490 | 151.481 | 128.825 | 243.008 | 158.987 |
| **40%** | 125 | 275.992 | 130.624 | 127.535 | 138.476 | 176.440 | 134.311 | 247.298 | 159.089 |
| **45%** | 125 | 295.152 | 136.610 | 129.276 | 147.987 | 206.944 | 151.521 | 249.715 | 158.916 |
| **50%** | 125 | 313.842 | 149.244 | 133.305 | 158.778 | 238.118 | 187.979 | 252.684 | 159.083 |
| **$EED_{med}$(%)** | 88.358 | 5.307 | 2.858 | 8.215 | 27.299 | 10.579 | 100.952 | 47.968 | |

# Resultados

Tabela: Média das distâncias obtidas pelas nossas heurísticas e pelo algoritmo SOAR para o problema $D\bar{R}$, variando a porcentagem de caracteres duplicados e fixando o tamanho das strings em 500.

| Porcentagem de Caracteres Duplicados | OP | MA | BL | GRASP | AG | BT | SA | Ext | SOAR |
|---|---|---|---|---|---|---|---|---|---|
| **5%** | 125 | 132.174 | 124.738 | 124.738 | 124.738 | 125.780 | 124.738 | 309.498 | 124.738 |
| **10%** | 125 | 149.082 | 124.731 | 124.731 | 124.731 | 125.705 | 124.731 | 311.776 | 124.732 |
| **15%** | 125 | 168.042 | 124.714 | 124.714 | 124.714 | 125.721 | 124.714 | 311.453 | 124.718 |
| **20%** | 125 | 188.064 | 124.723 | 124.723 | 124.723 | 125.665 | 124.723 | 306.794 | 124.732 |
| **25%** | 125 | 208.490 | 124.772 | 124.770 | 124.770 | 125.787 | 124.770 | 306.136 | 124.777 |
| **30%** | 125 | 229.110 | 124.770 | 124.725 | 124.727 | 126.419 | 124.725 | 307.132 | 124.733 |
| **35%** | 125 | 250.342 | 124.927 | 124.740 | 124.739 | 133.672 | 124.735 | 306.883 | 124.753 |
| **40%** | 125 | 271.571 | 125.371 | 124.725 | 124.735 | 151.795 | 124.705 | 306.229 | 124.742 |
| **45%** | 125 | 293.122 | 126.253 | 124.753 | 124.810 | 174.074 | 124.730 | 307.330 | 124.765 |
| **50%** | 125 | 313.862 | 128.189 | 124.818 | 124.908 | 195.524 | 124.717 | 310.325 | 124.766 |
| **$EED_{med}$(%)** | 84.787 | 0.682 | 0.253 | 0.264 | 14.396 | 0.243 | 162.983 | 0.253 | |

# Referências I

[1] P. Berman, S. Hannenhalli, and M. Karpinski.
1.375-Approximation Algorithm for Sorting by Reversals.
In *Proceedings of the 10th Annual European Symposium on Algorithms (ESA'2002)*, ESA'2002, pages 200–210, London, UK, 2002. Springer-Verlag.

[2] L. Bulteau, G. Fertin, and I. Rusu.
Sorting by Transpositions is Difficult.
*SIAM Journal on Computing*, 26(3):1148–1180, 2012.

[3] A. Caprara.
Sorting Permutations by Reversals and Eulerian Cycle Decompositions.
*SIAM Journal on Discrete Mathematics*, 12(1):91–110, 1999.

# Referências II

[4] X. Chen, J. Zheng, Z. Fu, P. Nan, Y. Zhong, S. Lonardi, and T. Jiang.
Assignment of Orthologous Genes via Genome Rearrangement.
*IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(4):302–315, 2005.

[5] S. Hannenhalli and P. A. Pevzner.
Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals.
*Journal of ACM*, 46(1):1–27, 1999.

# Referências III

[6] P. Kolman and T. Waleń.
Reversal Distance for Strings with Duplicates: Linear Time Approximation Using Hitting Set.
In *Proceedings of the 4th International Workshop on Approximation and Online Algorithms (WAOA'2006)*, pages 279–289, Berlin, Heidelberg, 2007.

[7] A. R. Oliveira, K. L. Brito, U. Dias, and Z. Dias.
On the complexity of sorting by reversals and transpositions problems.
*Journal of Computational Biology*.
PMID: 31120331.

# Referências IV

[8] A. J. Radcliffe, A. D. Scott, and E. L. Wilmer.
Reversals and Transpositions Over Finite Alphabets.
*SIAM Journal on Discrete Mathematics*, 19(1):224–244, 2005.

[9] A. Rahman, S. Shatabda, and M. Hasan.
An Approximation Algorithm for Sorting by Reversals and Transpositions.
*Journal of Discrete Algorithms*, 6(3):449–457, 2008.

[10] M. E. M. T. Walter, Z. Dias, and J. Meidanis.
Reversal and Transposition Distance of Linear Chromosomes.
In *Proceedings of the 5th International Symposium on String Processing and Information Retrieval (SPIRE'1998)*, pages 96–102, Los Alamitos, CA, USA, 1998.