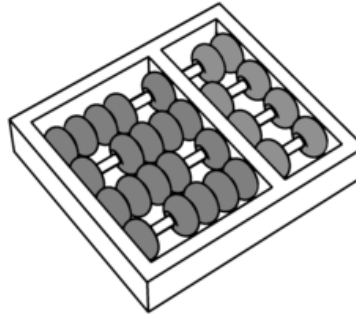


Universidade Estadual de Campinas

Instituto de Computação



Proposta de Dissertação de Mestrado

---

**O Problema da Ordenação de Permutações por Operações de Tamanho Limitado**

---

*Candidato:* Guilherme Henrique Santos Miranda

*Orientador:* Prof. Dr. Zanoni Dias

*Coorientadora:* Dra. Carla Negri Lintzmayer

## Resumo

Estimar a distância evolucionária entre genomas de dois organismos é uma tarefa desafiadora para Biologia Computacional. Uma das formas mais bem aceitas de estimar essa distância é considerar o tamanho da menor sequência de eventos de rearranjos necessários para transformar um genoma em outro, o que caracteriza o problema de distância de rearranjo. Computacionalmente, genomas podem ser representados como permutações de inteiros e, com isso, o problema pode ser reduzido ao problema de calcular o número mínimo de operações necessárias para transformar uma permutação na outra. As operações são dadas pelos modelos de rearranjo e afetam os segmentos do genoma de diferentes formas. Dentre os modelos mais comuns estão aqueles que permitem apenas reversões ou apenas transposições e já possuem ampla bibliografia. Modelos que permitem ambas as operações, entretanto, ainda são um grande desafio para a Teoria da Computação. Adicionalmente ao modelo de rearranjo, diferentes restrições podem ser impostas para o problema. Para esta dissertação de mestrado, propomos a realização de estudos sobre o problema quando uma restrição importante do ponto de vista biológico é adicionada: o tamanho das operações deve ser limitado a um valor dado. São conhecidos resultados apenas para quando as operações possuem tamanho máximo 2 ou 3, motivando ainda mais o estudo para quando as operações são limitadas a um tamanho arbitrário.

## 1 Introdução

Entender o processo evolucionário dos organismos é algo que sempre despertou curiosidade na humanidade e é um desafio importante para Biologia Computacional. A distância evolucionária entre genomas de dois organismos pode ser estimada com o tamanho da menor sequência de eventos de rearranjo necessários para transformar um genoma em outro, que representa o cenário mais provável de evolução, com base no *Princípio da Máxima Parcimônia*. Um rearranjo de genoma é um evento que ocorre com relativa raridade e modifica grandes trechos de um genoma, diferentemente das mutações pontuais, que afetam apenas moléculas individuais. Por possuir tais características, a estimativa da distância evolucionária com base nos eventos de rearranjos tende a ser mais precisa.

Computacionalmente, um genoma pode ser representado como uma permutação de inteiros, assumindo que não existam genes duplicados e que ele seja composto por um único cromossomo linear. Uma permutação é dita com sinais (sem sinais) caso a orientação dos genes seja conhecida (desconhecida). Devido à tal representação do genoma, o problema de estimar a distância evolucionária com a quantidade mínima de eventos de rearranjos necessários para transformar um genoma em outro pode ser reduzido ao problema de calcular o número mínimo de operações necessárias para transformar uma permutação em outra. Além disso, podemos representar um dos genomas como a permutação identidade e, dessa forma, este problema é equivalente ao problema de encontrar a quantidade mínima de rearranjos necessários para ordenar uma permutação.

Na literatura foram consideradas diferentes operações de rearranjo, como por exemplo operações de reversões, que invertem um determinado segmento do genoma, e operações de transposições, que trocam de posição dois segmentos adjacentes na permutação. Estas operações podem possuir restrições extras, como sobre em quais partes do genoma serão aplicadas ou sobre a quantidade máxima de elementos do segmento afetado, sendo tal quantidade definida como tamanho da operação. O conjunto de eventos permitidos para transformar um genoma é dado por um *modelo de rearranjo*. Inicialmente, foram estudados modelos que permitiam apenas um tipo de evento de rearranjo [2, 4]. Entretanto, modelos que permitem mais do que um único tipo de evento foram estudados recentemente [23, 37, 43]. Esta proposta tem ênfase na realização de um estudo sobre o problema de ordenação de permutações com operações de rearranjo com tamanhos limitados, utilizando modelos de rearranjo que permitem apenas operações de reversão e/ou apenas operações de transposição.

A relevância biológica para a restrição no tamanho do genoma parte da observação de que eventos de rearranjo que afetam grandes trechos do genoma ocorrem com pouca frequência, prevalecendo rearranjos que envolvem um baixo número de genes [12, 33]. Para modelos que permitem apenas reversões ou apenas transposições de tamanho no máximo 2, o problema foi provado possuir solução em tempo polinomial [28]. Quando o tamanho máximo permitido é 3, os melhores resultados conhecidos são algoritmos de 2-aproximação [27] e  $\frac{4}{3}$ -aproximação [25, 26], para modelos considerando apenas reversões e apenas transposições,

respectivamente. Não se tem conhecimento de resultados quando a restrição é dada como um tamanho arbitrário, o que é mais uma motivação para estudar tal problema.

O restante desta proposta é apresentado da seguinte forma. Na Seção 2 são apresentadas as principais definições e conceitos necessários para o entendimento da proposta. A revisão bibliográfica de diferentes variações do problema de ordenação de permutações é apresentada na Seção 3. Na Seção 4 são apresentados os objetivos almejados pelo trabalho proposto, em ordem de prioridade. A Seção 5 apresenta o plano de trabalho e, finalmente, a Seção 6 apresenta os materiais e métodos a serem utilizados no trabalho.

## 2 Fundamentação Teórica

Computacionalmente, um genoma é representado como uma  $n$ -upla que, ao assumirmos que não existe repetições de genes, pode ser dado como uma permutação  $\pi = (\pi_1 \pi_2 \dots \pi_n)$  onde  $\pi_i \in \{-n, -(n-1), \dots, -2, -1, +1, +2, \dots, +(n-1), +n\}$  e  $|\pi_i| \neq |\pi_j| \iff i \neq j$ . Um elemento  $\pi_i$  possui sinal quando a orientação do gene representado é conhecida e, nesse caso, a permutação é chamada de *permutação com sinais*. Quando não se tem conhecimento sobre a orientação dos genes, os sinais são omitidos e a permutação é denominada de *permutação sem sinais*. Uma *permutação estendida* pode ser obtida ao se adicionar os elementos  $\pi_0 = 0$  e  $\pi_{n+1} = +(n+1)$  em  $\pi$ . Por convenção, nesta proposta sempre iremos nos referir a permutações estendidas, entretanto o termo *estendida* e os elementos  $\pi_0$  e  $\pi_{n+1}$  serão omitidos. Também sempre estaremos nos referindo apenas a genomas lineares.

Uma vez que os genomas possuem tal representação, podemos modelar o problema de calcular a distância de rearranjo, dada pelo tamanho da menor sequência de operações necessárias para transformar um genoma em outro, como o problema de encontrar a quantidade mínima de rearranjos necessários para ordenar uma permutação, ao representarmos um dos genomas como a permutação identidade  $\iota = (1 \ 2 \ \dots \ n)$ . Por esse motivo, tal problema é chamado de Problema de Ordenação de Permutações por Rearranjos.

Assim, dado um modelo de rearranjo  $M$  que determina as operações de rearranjo permitidas, o problema de ordenação de permutações por rearranjos consiste em encontrar a menor sequência de operações  $\rho_i$  pertencentes ao modelo tais que  $\pi \cdot \rho_1 \cdot \rho_2 \cdot \dots \cdot \rho_t = \iota$ ,

onde  $\pi \cdot \rho_i$  é uma operação de composição, semelhante a uma composição de funções [16]. O tamanho dessa sequência é definido como a distância de ordenação  $d_M(\pi, \iota) = d_M(\pi) = t$ .

## 2.1 Reversões

Uma operação ou evento de reversão afeta um segmento do genoma, revertendo seus elementos. Formalmente, uma reversão  $\rho(i, j)$  em uma permutação sem sinais, onde  $1 \leq i < j \leq n$ , é um evento tal que:

$$\pi \cdot \rho(i, j) = (\pi_1 \pi_2 \dots \pi_{i-1} \underline{\pi_j \pi_{j-1} \dots \pi_{i+1} \pi_i} \pi_{j+1} \dots \pi_{n-1} \pi_n).$$

Por exemplo, dada uma permutação sem sinais  $\pi = (1 \ 2 \ 3 \ 4 \ 5)$ , então  $\pi \cdot \rho(2, 4) = (1 \ \underline{4} \ \underline{3} \ 2 \ 5)$ . No caso de permutações com sinais, a operação de reversão  $\bar{\rho}(i, j)$ , com  $1 \leq i \leq j \leq n$ , além de inverter a ordem dos elementos, também inverte os sinais dos elementos. Por exemplo, dada uma permutação com sinais  $\pi = (+1 \ +2 \ -3 \ +4 \ +5)$ , então  $\pi \cdot \bar{\rho}(2, 4) = (+1 \ \underline{-4} \ \underline{+3} \ -2 \ +5)$ . As distâncias de ordenação por reversões são denotadas por  $d_r(\pi)$  e  $d_{\bar{r}}(\pi)$ , respectivamente para permutações sem sinais e com sinais. A distância de ordenação máxima por reversões entre todas as permutações de tamanho  $n$  é definida como o diâmetro  $D_r(n)$ , para permutações sem sinais, e  $D_{\bar{r}}(n)$ , para permutações com sinais. O tamanho de uma reversão (com ou sem sinais) é dado por  $j - i + 1$ . Além disso, chamamos de reversão de prefixo uma operação  $\rho_p(i) = \rho(1, i)$  e de reversão de sufixo uma operação  $\rho_s(i) = \rho(i, n)$ . Quando o modelo é restrito a apenas operações de reversões, definimos o problema de encontrar a distância de rearranjo como o Problema de Ordenação de Permutações por Reversões.

## 2.2 Transposições

Uma operação ou evento de transposição afeta dois segmentos adjacentes, trocando-os de lugar na permutação. Formalmente, uma transposição  $\tau(i, j, k)$ , para  $1 \leq i < j < k \leq n + 1$ , é um evento tal que:

$$\pi \cdot \tau(i, j, k) = (\pi_1 \pi_2 \dots \pi_{i-1} \underline{\pi_j \dots \pi_{k-1}} \underline{\pi_i \dots \pi_{j-1}} \pi_k \dots \pi_n).$$

Por exemplo, se  $\pi = (1\ 2\ 3\ 4\ 5)$ , então  $\pi \cdot \tau(1, 3, 5) = (\underline{3}\ \underline{4}\ 1\ 2\ 5)$ . Uma vez que as operações de transposições não alteram os sinais dos elementos, não há diferença quando aplicadas em permutações com sinais. A distância de ordenação por transposições é denotada por  $d_t(\pi)$  e o tamanho de uma transposição é dado por  $k - i$ . O diâmetro  $D_t(n)$  é definido como a distância de ordenação máxima por transposições para todas as permutações de tamanho  $n$ . Além disso, definimos uma transposição de prefixo como uma operação  $\tau_p(i, j) = \tau(1, i, j)$  e uma transposição de sufixos como  $\tau_s(i, j) = \tau(i, j, n+1)$ . Quando o modelo é restrito a apenas operações de transposições, definimos o problema de encontrar a distância de rearranjo como o Problema de Ordenação de Permutações por Transposições.

### 2.3 Transposições e Reversões

Quando o modelo de rearranjo permite ambas operações de reversão e transposição, o problema é definido como o Problema de Ordenação de Permutações por Reversões e Transposições. As distâncias de ordenação por reversões e transposições são denotadas por  $d_{rt}(\pi)$  e  $d_{\bar{r}t}(\pi)$ , respectivamente para permutações sem sinais e com sinais. A distância máxima de ordenação por reversões entre todas as permutações de tamanho  $n$  é definida como o diâmetro  $D_{rt}(n)$  e  $D_{\bar{r}t}(n)$ , para permutações sem sinais e com sinais, respectivamente. É importante ressaltar que quando as duas operações são permitidas, temos  $d_{rt}(\pi) \leq d_r$ ,  $d_{\bar{r}t}(\pi) \leq d_{\bar{r}}$  e  $d_{rt}(\pi) \leq d_t$ . Por exemplo, para  $\pi = (1\ 3\ 4\ 2\ 7\ 6\ 5)$  temos: (i)  $\pi \cdot \tau(2, 4, 5) \cdot \tau(5, 7, 8) \cdot \tau(6, 7, 8) = \iota$ , com  $d_t(\pi) = 3$ ; (ii)  $\pi \cdot \rho(2, 4) \cdot \rho(5, 7) \cdot \rho(3, 4) = \iota$ , com  $d_r(\pi) = 3$  e (iii)  $\pi \cdot \tau(2, 4, 5) \cdot \rho(5, 7) = \iota$ , com  $d_{rt}(\pi) = 2$ .

### 2.4 Breakpoints

Para o problema de ordenação por reversões sem sinais, um *breakpoint* é definido como um par de elementos  $(\pi_i, \pi_{i+1})$ , onde  $i \in [0, n]$ , tal que  $|\pi_{i+1} - \pi_i| \neq 1$  e denotamos a quantidade de *breakpoints* em  $\pi$  por  $b_r(\pi)$ . A mesma definição é utilizada quando o modelo de rearranjo permite tanto operações de reversão, quanto de transposição e, neste caso, a quantidade de *breakpoints* em  $\pi$  é denotada por  $b_{rt}(\pi)$ . Por exemplo, para  $\pi = (1\ 3\ 4\ 2\ 7\ 6\ 5)$  temos os *breakpoints*  $(1, 3)$ ,  $(4, 2)$ ,  $(2, 7)$  e  $(5, 8)$ , logo  $b_r(\pi) = b_{rt}(\pi) = 4$ .

Para os problemas de ordenação de permutações com sinais por reversões e ordenação de permutações sem sinais por transposições, um *breakpoint* é definido como um par de elementos  $(\pi_i, \pi_{i+1})$  tal que  $\pi_{i+1} - \pi_i \neq 1$  e a quantidade de *breakpoints* de reversões com sinais e *breakpoints* de transposições é denotada por  $b_{\bar{r}}(\pi)$  e  $b_t(\pi)$ , respectivamente. Esta definição também é válida para o problema de ordenação de permutações com sinais quando ambas as operações são permitidas e, neste caso, a quantidade de *breakpoints* em  $\pi$  é denotada por  $b_{\bar{r}t}(\pi)$ . Por exemplo, considerando a permutação  $\pi = (+2 -1 +3 +4 +6 -5)$ , temos os *breakpoints*  $(0, +2), (+2, -1), (-1, +3), (+4, +6), (+6, -5)$  e  $(-5, +7)$ , logo  $b_{\bar{r}}(\pi) = b_{\bar{r}t}(\pi) = 6$ .

Observe que somente a permutação identidade não possui nenhum dos tipos de *breakpoints*.

## 2.5 Strips

Uma *strip* é definida como uma sequência maximal de elementos da permutação  $(\pi_i, \dots, \pi_j)$  tal que não existe *breakpoints* entre quaisquer pares  $(\pi_k, \pi_{k+1})$ , onde  $k \in [i, j)$ . Por exemplo, as permutações sem sinais  $\pi = (\underline{1} \ 3 \ 4 \ \underline{2} \ \underline{7} \ 6 \ 5)$  e com sinais  $\pi = (\underline{+1} \ \underline{+2} \ \underline{+5} \ \underline{+4} \ \underline{-3} \ \underline{+6})$  possuem, respectivamente, 4 e 5 *strips*, destacadas na permutação.

## 2.6 Inversões

Para uma permutação  $\pi$  com ou sem sinais, uma inversão é definida como um par de elementos  $(\pi_i, \pi_j)$  tal que  $\pi_i < \pi_j$  e  $1 \leq j < i \leq n$  e o número de inversões é denotado por  $inv(\pi)$ . Por exemplo, para  $\pi = (3 \ 1 \ 4 \ 2)$ , os pares  $(\pi_1, \pi_2), (\pi_1, \pi_4)$  e  $(\pi_3, \pi_4)$  são as inversões de  $\pi$ , logo  $inv(\pi) = 3$ .

## 2.7 Operações de Tamanho Limitado

Chamaremos as operações de reversão e transposição de  $\lambda$ -reversão e  $\lambda$ -transposição, respectivamente, quando tais operações possuírem restrição de tamanho máximo  $\lambda$ . Para valores de  $\lambda = 2$  e  $\lambda = 3$ , tais operações são chamadas de super curtas e curtas, respectivamente. Além disso, as distâncias de ordenação de permutações sem sinais por  $\lambda$ -reversões, por  $\lambda$ -transposições e por ambas as operações serão denotadas por  $d_r^\lambda(\pi)$ ,  $d_t^\lambda(\pi)$  e  $d_{rt}^\lambda(\pi)$ ,

respectivamente. Para permutações com sinais, a distância de ordenação por  $\lambda$ -reversões é denotada por  $d_{\tilde{r}}^{\lambda}(\pi)$  e, quando ambas operações são permitidas, a distância de ordenação é denotada por  $d_{\tilde{r}t}^{\lambda}(\pi)$ . Definiremos também os diâmetros  $D_r^{\lambda}(n)$ ,  $D_t^{\lambda}(n)$ ,  $D_{rt}^{\lambda}(n)$ ,  $D_{\tilde{r}}^{\lambda}(n)$  e  $D_{\tilde{r}t}^{\lambda}(n)$  como o maior valor de  $d_r^{\lambda}(\pi)$ ,  $d_t^{\lambda}(\pi)$ ,  $d_{rt}^{\lambda}(\pi)$ ,  $d_{\tilde{r}}^{\lambda}(\pi)$  e  $d_{\tilde{r}t}^{\lambda}(\pi)$ , respectivamente, para todas as permutações  $\pi$  de tamanho  $n$ . Por exemplo,  $d_r^2(\pi)$  é a distância de ordenação por reversões super curtas para uma permutação sem sinais  $\pi$ , enquanto  $D_{\tilde{r}t}^3(n)$  é o diâmetro de ordenação por reversões e transposições curtas para todas as permutações com sinais de tamanho  $n$ .

### 3 Revisão Bibliográfica

Nessa seção é apresentado o estado da arte para os problemas de ordenação de permutações por reversões, por transposições e por transposições e reversões, com diferentes restrições.

#### 3.1 Ordenação por Reversões

O Problema de Ordenação de Permutações sem Sinais por Reversões foi provado estar na classe de problemas NP-Difíceis [9] e diversos pesquisadores apresentaram algoritmos de aproximação para o problema. Um dos primeiros resultados foi obtido por Bafna e Pevzner [2] com um algoritmo de 1.75-aproximação. Posteriormente, foi proposto por Christie [10] um algoritmo com fator de aproximação 1.5. Berman e coautores [6] apresentaram um algoritmo de 1.375-aproximação, que é o melhor conhecido para o problema. Além disso, um limitante inferior para a distância,  $d_r(\pi) \geq \frac{b_r(\pi)}{2}$ , foi proposto por Bafna e Pevzner [3] e um limite superior,  $d_r(\pi) \leq b_r(\pi) - 1$ , foi proposto por Kececioğlu e Sankoff [31].

Curiosamente, o Problema de Ordenação de Permutações com Sinais por Reversões possui algoritmo exato em tempo polinomial. A primeira versão foi proposta por Hannenhalli e Pevzner [24], que posteriormente foi simplificada por Bergeron [5]. O melhor algoritmo proposto na literatura foi dado por Tannier e coautores [40], com complexidade de tempo sub-quadrática. Além disso, um algoritmo linear foi proposto por Bader e coautores [1] para quando se deseja apenas a distância de rearranjo.



### 3.2 Ordenação por Transposições

O Problema de Ordenação de Permutações por Transposições foi introduzido por Bafna e Pevzner [4] em 1995. A complexidade computacional desse problema permaneceu em aberto por mais de 15 anos, até que Bulteau e coautores [7] provaram que ele pertence a classe de problemas NP-Difíceis.

O primeiro algoritmo para o problema foi proposto pelos mesmos autores que o introduziram, Bafna e Pevzner [4], com fator de aproximação 1.5 e complexidade de tempo  $O(n^2)$ . Foi também proposto pelos autores limitantes inferior  $d_t(\pi) \geq \frac{b_t(\pi)}{3}$  e superior  $d_t(\pi) \leq b_t(\pi)$  para as distâncias de rearranjo.

Christie [11] propôs posteriormente um algoritmo com complexidade de tempo  $O(n^4)$  mais simples de ser implementado, com o mesmo fator de aproximação. Um outro algoritmo com tempo quadrático e implementação mais simples foi proposto por Walter e coautores [44], entretanto com fator de aproximação 2.25. O melhor algoritmo conhecido para o problema possui fator de aproximação 1.375, complexidade de tempo  $O(n^2)$  e foi proposto em 2006 por Elias e Hartman [15].

Rusu [38] apresentou uma maneira de implementar vários algoritmos da literatura para os problema de ordenação de permutações por transposições e/ou reversões em tempo  $O(n \log n)$ , com o uso de *log-lists*.

### 3.3 Ordenação por Reversões e Transposições

Quando ambas as operações são permitidas no modelo de rearranjo, Walter e coautores [43] apresentaram limites inferiores para a distância e diâmetro, além de algoritmos de 3-aproximação e 2-aproximação, para as versões sem sinais e com sinais, respectivamente.

Posteriormente, Rahman e coautores [37] apresentaram, em 2008, um algoritmo de  $2k$ -aproximação para permutações sem sinais, sendo  $k$  o fator de aproximação do algoritmo para decomposição de ciclos. Dado o melhor valor de  $k$  conhecido [34], o fator de aproximação deste algoritmo é de  $2.8386 + \epsilon$ , para  $\epsilon > 0$ .

### 3.4 Ordenação por Operações de Prefixos e Sufixos

A versão do problema de ordenação de permutações por reversões com a restrição de que as operações podem ser aplicadas somente nos prefixos da permutação foi introduzida por Dweighthier [14] com o nome de Problema da Ordenação de Panquecas, em 1975. A descrição desse problema consistia, resumidamente, em ordenar uma pilha de panquecas de acordo com seus respectivos tamanhos, de forma que a menor panqueca ficasse no topo, a segunda menor panqueca logo abaixo dela, e assim sucessivamente, até a base da pilha, que conteria a maior panqueca. A reinterpretção deste problema para um problema de Rearranjo de Genomas foi dada por Pevzner e Waterman [36], em 1995 e o problema foi provado ser NP-Difícil em 2011 [8]. Gates e Papadimitriou [22] estudaram o Problema da Ordenação de Panquecas em 1979, a fim de encontrar a maior quantidade de movimentos necessários para ordenar qualquer pilha de panquecas e mostraram que ela está entre  $\frac{17n}{16}$  e  $\frac{5n+5}{3}$ . Quando o objetivo é encontrar a menor quantidade de movimentos para ordenar uma dada pilha de panquecas, o melhor resultado conhecido é um algoritmo de 2-aproximação [17].

O Problema de Ordenação por Transposições de Prefixo foi introduzido por Dias e Meidanis [13]. O melhor resultado conhecido para este problema é um algoritmo de 2-aproximação, dado pelos mesmos autores.

Sharmin e coautores [39] propuseram duas novas versões para o problema da ordenação de panquecas: (i) permitindo utilizar reversões e/ou transposições de prefixo e (ii) permitindo utilizar transposições, reversões e/ou transreversões de prefixo, onde uma transreversão é uma operação dada pela combinação de uma transposição e uma reversão. Estes mesmos autores apresentaram algoritmos de 3-aproximação e 2-aproximação para os problemas (i) e (ii), respectivamente.

Lintzmayer e coautores [35] introduziram problemas de ordenação de permutações (com e sem sinais) considerando operações tanto de sufixo quanto de prefixo, para reversões e/ou transposições. Os autores apresentaram algoritmos com fatores de aproximação  $2 + \epsilon$  para tais problemas, onde  $\epsilon$  é uma constante dividida pelo número de *breakpoints* na permutação de entrada.

### 3.5 Ordenação por Operações Curtas e Super Curtas

Quando o tamanho da operação é limitado a no máximo 2, Jerrum [28] mostrou que o problema de ordenação de permutação sem sinais por reversões (ou transposições) super curtas possui solução em tempo polinomial, com a prova dada por Knuth [32].

Quando o tamanho é limitado a no máximo 3, Heath e Vergara apresentaram um algoritmo de  $\frac{4}{3}$ -aproximação para o problema de ordenação de permutação por transposições curtas [26] e posteriormente apresentaram um algoritmo com fator de aproximação 2 para o problema de ordenação de permutação sem sinais por reversões curtas [27], o melhor resultado conhecido. Além disso, Vergara [42] também mostrou que o algoritmo de  $\frac{4}{3}$ -aproximação para o problema de ordenação por transposições curtas é também um algoritmo de 2-aproximação para o problema de ordenação por reversões e transposições curtas. Em 2012, um algoritmo com fator de aproximação  $(1 + \epsilon)$ , onde  $\epsilon = \frac{1}{\text{inv}(\pi)}$ , para o problema de ordenação de permutações por transposições curtas foi dado por Jiang e coautores [29]. Dois anos depois, um outro algoritmo com fator de aproximação  $\frac{5}{4}$  foi proposto por Jiang e coautores [30] também para o problema de ordenação por transposições curtas.

Três algoritmos de aproximação para o problema de ordenação de permutações com sinais por reversões curtas foram propostos por Galvão e Dias [18] em 2014, sendo o melhor deles com fator de aproximação 9. No ano seguinte, Galvão e coautores [20] apresentaram algoritmos exatos e com complexidade polinomial para os problemas de ordenação de permutações com sinais por reversões super curtas e ordenação de permutações com sinais por reversões e transposições super curtas, além de algoritmos com fator de aproximação 5 e 3 para os problemas de ordenação de permutações com sinais por reversões curtas e ordenação de permutações com sinais por reversões e transposições curtas, respectivamente. Além disso, também foi mostrado pelos autores que o fator de aproximação esperado para o algoritmo de ordenação de permutações por reversões curtas é 3, quando a permutação possui 12 ou mais elementos.

Para o problema de ordenação de permutações circulares, onde se considera que o último e o primeiro elemento da permutação são adjacentes, também existe a opção de utilizar apenas operações super curtas. Para esse problema, Galvão e coautores [19] apresentaram,

Permutações	Modelo de Rearranjo	Complexidade	Melhor Resultado
Com Sinais	Reversões	Polinomial [24]	Tempo $O(n^{\frac{3}{2}})$ [40]
Sem Sinais		NP-Difícil [9]	1.375-aproximação [6]
Sem Sinais	Transposições	NP-Difícil [7]	1.375-aproximação [15]
Com Sinais	Reversões e Transposições	?	2-aproximação [43]
Sem Sinais		?	2k-aproximação [37]
Com Sinais	Reversões Super Curtas	Polinomial [20]	Tempo $O(n^3)$ [20]
Sem Sinais		Polinomial [32, p. 108]	Tempo $O(n^2)$ [28]
Sem Sinais	Transposições Super Curtas	Polinomial [32, p. 108]	Tempo $O(n^2)$ [28]
Com Sinais	Reversões e Transposições Super Curtas	Polinomial [20]	Tempo $O(n^3)$ [20]
Sem Sinais		Polinomial [32, p. 108]	Tempo $O(n^2)$ [28]
Com Sinais	Reversões Curtas	?	5-aproximação [20]
Sem Sinais		?	2-aproximação [27]
Sem Sinais	Transposições Curtas	?	$\frac{5}{4}$ -aproximação [30]
Com Sinais	Reversões e Transposições Curtas	?	3-aproximação [20]
Sem Sinais		?	2-aproximação [42]

Tabela 1: Estado da Arte para o Problema de Ordenação de Permutações.

em 2015, um algoritmo com tempo polinomial para permutações com sinais e, ainda, disponibilizaram no ano seguinte uma ferramenta *web* [21] para inferência filogenética baseada em rearranjos usando reversões super curtas.

### 3.6 Resumo: Estado da Arte

O estado da arte para as diferentes variações do Problema de Ordenação de Permutações por Rearranjo revisadas nesta proposta é apresentado, resumidamente, na Tabela 3.6.

## 4 Objetivos

O objetivo principal será obter algoritmos de aproximação para o Problema de Ordenação de Permutações com modelos de rearranjo nos quais as operações são limitadas a um tamanho máximo ( $\lambda$ ). Mais especificamente, estudaremos as seguintes versões do problema,

em ordem de prioridade:

1. Ordenação de permutações sem sinais por  $\lambda$ -reversões (objetivo primário);
2. Ordenação de permutações com sinais por  $\lambda$ -reversões (objetivo primário);
3. Ordenação de permutações sem sinais por  $\lambda$ -transposições (objetivo primário);
4. Ordenação de permutações sem sinais por  $\lambda$ -reversões e  $\lambda$ -transposições (objetivo secundário);
5. Ordenação de permutações com sinais por  $\lambda$ -reversões e  $\lambda$ -transposições (objetivo secundário).

## 5 Plano de Trabalho

O plano de trabalho é apresentado na Tabela 2. É válido ressaltar que o tempo despendido em cada atividade pode vir a sofrer mudanças, a medida que os resultados são obtidos, visto que alguns podem ser mais promissores do que outros.

	2017										2018										2019			
	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																								*

Tabela 2: Cronograma de atividades.

1. Obtenção dos créditos obrigatórios em disciplinas do programa de mestrado;
2. Revisão da literatura;
3. Escrita da proposta de mestrado;

4. Exame de Qualificação de Mestrado;
5. Participação no Programa de Estágio Docente (PED);
6. Investigação de algoritmos para o problema de ordenação de permutações sem sinais por  $\lambda$ -reversões;
7. Investigação de algoritmos para o problema de ordenação de permutações com sinais por  $\lambda$ -reversões;
8. Investigação de algoritmos para o problema de ordenação de permutações sem sinais por  $\lambda$ -transposições;
9. Investigação de algoritmos para o problema de ordenação de permutações com e sem sinais por  $\lambda$ -reversões e  $\lambda$ -transposições;
10. Escrita da dissertação;
11. Revisão da dissertação;
12. Defesa da dissertação.

## 6 Materiais e Métodos

O trabalho possui, em sua maioria, aspecto teórico. Assim, o estudo de teoremas e provas existentes na literatura proporcionarão a base para que novos conhecimentos possam ser obtidos para os problemas que iremos estudar. Além disso, programas serão implementados a fim de validar os resultados obtidos, quando necessário e sempre que possível.

## Referências

- [1] D. A. Bader, B. M. E. Moret, and M. Yan. A Linear-Time Algorithm for Computing Inversion Distance Between Signed Permutations with an Experimental Study. *Journal of Computational Biology*, 8:483–491, 2001.

- [2] V. Bafna and P. A. Pevzner. Sorting by Reversals: Genome Rearrangements in Plant Organelles and Evolutionary History of X Chromosome. *Molecular Biology and Evolution*, 12(2):239–246, 1995.
- [3] V. Bafna and P. A. Pevzner. Genome Rearrangements and Sorting by Reversals. *SIAM Journal on Computing*, 25(2):272–289, 1996.
- [4] V. Bafna and P. A. Pevzner. Sorting by Transpositions. *SIAM Journal on Discrete Mathematics*, 11(2):224–240, 1998.
- [5] A. Bergeron. A Very Elementary Presentation of the Hannenhalli-Pevzner Theory. *Discrete Applied Mathematics*, 146(2):134–145, 2005.
- [6] P. Berman, S. Hannenhalli, and M. Karpinski. 1.375-Approximation Algorithm for Sorting by Reversals. In R. Möhring and R. Raman, editors, *Proceedings of the 10th Annual European Symposium on Algorithms (ESA '2002)*, volume 2461 of *Lecture Notes in Computer Science*, pages 200–210. Springer-Verlag Berlin Heidelberg New York, Berlin/Heidelberg, Germany, 2002.
- [7] L. Bulteau, G. Fertin, and I. Rusu. Sorting by Transpositions is Difficult. *SIAM Journal on Computing*, 26(3):1148–1180, 2012.
- [8] L. Bulteau, G. Fertin, and I. Rusu. Pancake Flipping is Hard. *Journal of Computer and System Sciences*, 81(8):1556–1574, 2015.
- [9] A. Caprara. Sorting Permutations by Reversals and Eulerian Cycle Decompositions. *SIAM Journal on Discrete Mathematics*, 12(1):91–110, 1999.
- [10] D. A. Christie. A  $3/2$ -Approximation Algorithm for Sorting by Reversals. In H. Karloff, editor, *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '1998)*, pages 244–252, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.
- [11] D. A. Christie. *Genome Rearrangement Problems*. PhD thesis, Department of Computing Science, University of Glasgow, 1998.

- [12] D. A. Dalevi, N. Eriksen, K. Eriksson, and S. G. E. Andersson. Measuring Genome Divergence in Bacteria: A Case Study Using Chlamyidian Data. *Journal of Molecular Evolution*, 55(1):24–36, 2002.
- [13] Z. Dias and J. Meidanis. Sorting by Prefix Transpositions. In A. H. F. Laender and A. L. Oliveira, editors, *Proceedings of the 9th International Symposium on String Processing and Information Retrieval (SPIRE'2002)*, volume 2476 of *Lecture Notes in Computer Science*, pages 65–76. Springer-Verlag Berlin Heidelberg New York, Berlin/Heidelberg, Germany, 2002.
- [14] H. Dweighter. Problem E2569. *American Mathematical Monthly*, 82:1010, 1975.
- [15] I. Elias and T. Hartman. A 1.375-Approximation Algorithm for Sorting by Transpositions. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(4): 369–379, 2006.
- [16] G. Fertin, A. Labarre, I. Rusu, É. Tannier, and S. Vialette. *Combinatorics of Genome Rearrangements*. Computational Molecular Biology. The MIT Press, London, England, 2009.
- [17] J. Fischer and S. W. Ginzinger. A 2-Approximation Algorithm for Sorting by Prefix Reversals. In G. S. Brodal and S. Leonardi, editors, *Proceedings of the 13th Annual European Conference on Algorithms (ESA'2005)*, volume 3669 of *Lecture Notes in Computer Science*, pages 415–425, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [18] G. R. Galvão and Z. Dias. Approximation Algorithms for Sorting by Signed Short Reversals. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB'2014)*, pages 360–369, New York, NY, USA, 2014. ACM.
- [19] G. R. Galvão, C. Baudet, and Z. Dias. Sorting Signed Circular Permutations by Super Short Reversals. In R. Harrison, Y. Li, and I. Măndoiu, editors, *Proceedings of the 11th International Symposium on Bioinformatics Research and Applications (ISBRA'2015)*,



- Lecture Notes in Computer Science, pages 272–283. Springer International Publishing, Switzerland, 2015.
- [20] G. R. Galvão, O. Lee, and Z. Dias. Sorting Signed Permutations by Short Operations. *Algorithms for Molecular Biology*, 10(1):1–17, 2015.
- [21] G. R. Galvão, C. Baudet, and Z. Dias. Sorting Circular Permutations by Super Short Reversals. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(3):620–633, 2017.
- [22] W. H. Gates and C. H. Papadimitriou. Bounds for Sorting by Prefix Reversal. *Discrete Mathematics*, 27(1):47–57, 1979.
- [23] Q.-P. Gu, S. Peng, and I. H. Sudborough. A 2-Approximation Algorithm for Genome Rearrangements by Reversals and Transpositions. *Theoretical Computer Science*, 210(2):327–339, 1999.
- [24] S. Hannenhalli and P. A. Pevzner. Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals. *Journal of the ACM*, 46(1):1–27, 1999.
- [25] L. S. Heath and J. P. C. Vergara. Sorting by Bounded Block-moves. *Discrete Applied Mathematics*, 88(1-3):181–206, 1998.
- [26] L. S. Heath and J. P. C. Vergara. Sorting by Short Block-Moves. *Algorithmica*, 28(3):323–352, 2000.
- [27] L. S. Heath and J. P. C. Vergara. Sorting by Short Swaps. *Journal of Computational Biology*, 10(5):775–789, 2003.
- [28] M. R. Jerrum. The Complexity of Finding Minimum-length Generator Sequences. *Theoretical Computer Science*, 36(2-3):265–289, 1985.
- [29] H. Jiang, D. Zhu, and B. Zhu. A  $(1+\epsilon)$ -Approximation Algorithm for Sorting by Short Block-Moves. *Theoretical Computer Science*, 437:1–8, 2012.

- [30] H. Jiang, H. Feng, and D. Zhu. An  $5/4$ -Approximation Algorithm for Sorting Permutations by Short Block Moves. In H. Ahn and C. Shin, editors, *Proceedings of the 25th International Symposium on Algorithms and Computation (ISAAC'2014)*, Lecture Notes in Computer Science, pages 491–503. Springer International Publishing, 2014.
- [31] J. D. Kececioglu and D. Sankoff. Exact and Approximation Algorithms for Sorting by Reversals, with Application to Genome Rearrangement. *Algorithmica*, 13:180–210, 1995.
- [32] D. E. Knuth. *Fundamental Algorithms: The art of Computer Programming*. 1973.
- [33] J.-F. Lefebvre, N. El-Mabrouk, E. R. M. Tillier, and D. Sankoff. Detection and validation of single gene inversions. *Bioinformatics*, 19(1):i190–i196, 2003.
- [34] G. Lin and T. Jiang. A Further Improved Approximation Algorithm for Breakpoint Graph Decomposition. *Journal of Combinatorial Optimization*, 8(2):183–194, 2004.
- [35] C. N. Lintzmayer, G. Fertin, and Z. Dias. Sorting Permutations by Prefix and Suffix Rearrangements. *Journal of Bioinformatics and Computational Biology*, 15(1):1750002, 2017.
- [36] P. A. Pevzner and M. S. Waterman. Open Combinatorial Problems in Computational Molecular Biology. In *Proceedings of the 3rd Israel Symposium on the Theory of Computing Systems (ISTCS'1995)*, pages 158–173, Washington, DC, USA, 1995. IEEE Computer Society.
- [37] 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.
- [38] I. Rusu. Log-Lists and Their Applications to Sorting by Transpositions, Reversals and Block-Interchanges. *Theoretical Computer Science*, 660:1–15, 2017.
- [39] M. Sharmin, R. Yeasmin, M. Hasan, A. Rahman, and M. S. Rahman. Pancake Flipping with Two Spatulas. *Electronic Notes in Discrete Mathematics*, 36:231–238, 2010.
- [40] E. Tannier, A. Bergeron, and M.-F. Sagot. Advances on Sorting by Reversals. *Discrete Applied Mathematics*, 155(6-7):881–888, 2007.

- [41] G. Tesler. GRIMM: Genome Rearrangements Web Server. *Bioinformatics*, 18(3):492–493, 2002.
- [42] J. P. C. Vergara. *Sorting by Bounded Permutations*. PhD thesis, Virginia Polytechnic Institute and State University, 1998.
- [43] 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. IEEE Computer Society.
- [44] M. E. M. T. Walter, Z. Dias, and J. Meidanis. A New Approach for Approximating The Transposition Distance. In F. M. Titsworth, editor, *Proceedings of the 7th String Processing and Information Retrieval (SPIRE'2000)*, pages 199–208, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

## A Resultados Computacionais

Neste apêndice são apresentados alguns resultados obtidos com experimentos que foram realizados a fim de entender melhor o problema na prática.

### A.1 Análise dos Tamanhos das Operações utilizando o GRIMM

A fim de averiguar com que frequência operações que modificam pequenos trechos de um genoma ocorrem no problema de ordenação por reversões em permutações com sinais, realizamos experimentos utilizando o *software* GRIMM (Genome Rearrangements In Man and Mouse) [41] para encontrar a sequência ótima de operações para transformar um genoma em outro. Esta ferramenta também permite priorizar o uso de reversões menores na geração da solução ótima. Construimos histogramas com as frequências dos tamanhos das reversões utilizadas nos testes, que contaram com um total de 10000 permutações com sinais de tamanho 1000. Os resultados podem ser visualizados nas Figuras 1 e 2. É fácil ver que operações pequenas ocorrem com maior frequência em ambos os gráficos. Além disso, também analisamos a porcentagem de reversões unitárias (que apenas invertem o sinal de um elemento) e as porcentagens de operações super curtas e curtas. Quando as operações foram escolhidas arbitrariamente, as porcentagens foram 0.51%, 0.50% e 0.49%, respectivamente, com tamanho médio igual a 282.96. Quando as operações foram escolhidas dando prioridade para reversões curtas, os percentuais obtidos foram 3.66%, 1.85% e 1.65%, respectivamente, enquanto o tamanho médio das operações foi de 59.10.

### A.2 Distância e Diâmetro para Permutações Pequenas

Alguns resultados para permutações pequenas são apresentados nas Tabelas 3 a 12. No total, 10 tabelas são apresentadas. As 6 primeiras delas contêm as distâncias médias e os diâmetros para todas as permutações sem sinais (duas para cada problema), com  $n$  e  $\lambda$  variando de 2 até 10. As demais 4 tabelas apresentam as distâncias médias e os diâmetros para todas as permutações com sinais (também duas para cada problema), onde os valores de  $n$  variam de 2 até 9.

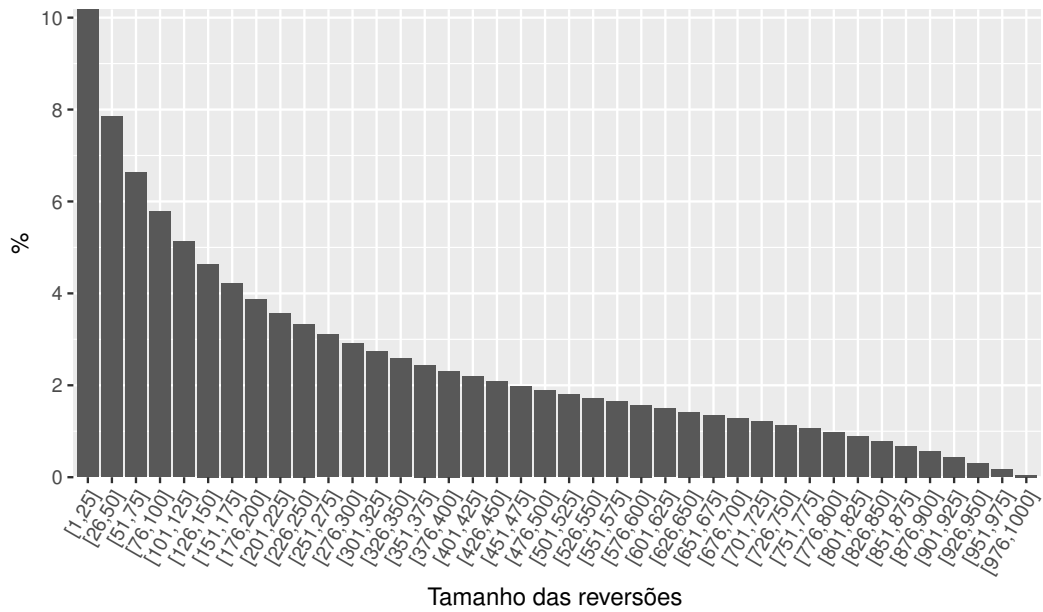


Figura 1: Histograma com as frequências dos tamanhos das reversões utilizadas pelo GRIMM para ordenar permutações de tamanho 1000, sem priorizar reversões menores.

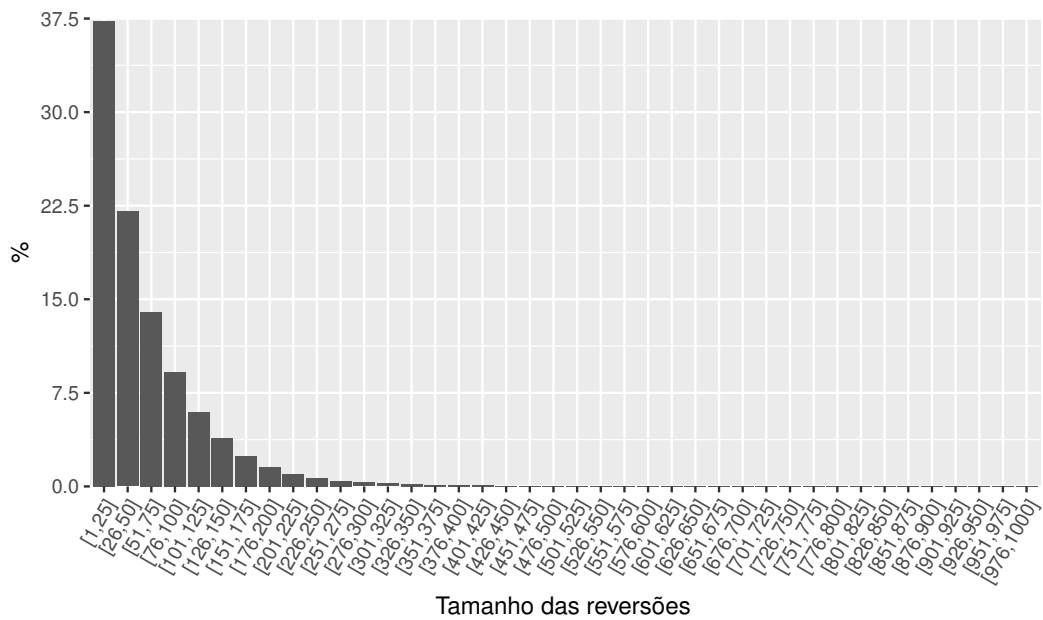


Figura 2: Histograma com as frequências dos tamanhos das reversões utilizadas pelo GRIMM para ordenar permutações de tamanho 1000, priorizando reversões menores.

### A.2.1 Problema de Ordenação por Reversões em Permutações sem Sinais

$n$	Tamanho máximo das operações ( $\lambda$ )								
-	2	3	4	5	6	7	8	9	10
2	0.50	-	-	-	-	-	-	-	-
3	1.50	1.17	-	-	-	-	-	-	-
4	3.00	2.08	1.75	-	-	-	-	-	-
5	5.00	3.17	2.51	2.39	-	-	-	-	-
6	7.50	4.45	3.33	3.11	3.04	-	-	-	-
7	10.50	5.90	4.22	3.84	3.74	3.70	-	-	-
8	14.00	7.54	5.22	4.61	4.45	4.40	4.38	-	-
9	18.00	9.35	6.29	5.42	5.16	5.10	5.08	5.06	-
10	22.50	11.33	7.45	6.30	5.90	5.81	5.78	5.77	5.76

Tabela 3: Distâncias Médias para o Problema de Ordenação por Reversões em Permutações sem Sinais.

$n$	Tamanho máximo das operações ( $\lambda$ )								
-	2	3	4	5	6	7	8	9	10
2	1	-	-	-	-	-	-	-	-
3	3	2	-	-	-	-	-	-	-
4	6	4	3	-	-	-	-	-	-
5	10	5	4	4	-	-	-	-	-
6	15	7	5	5	5	-	-	-	-
7	21	10	6	6	6	6	-	-	-
8	28	14	7	7	7	7	7	-	-
9	36	16	9	8	8	8	8	8	-
10	45	19	11	9	9	9	9	9	9

Tabela 4: Diâmetros para o Problema de Ordenação por Reversões em Permutações sem Sinais ( $D_r^\lambda(n)$ ).

### A.2.2 Problema de Ordenação por Transposições em Permutações sem Sinais

$n$	Tamanho máximo das operações ( $\lambda$ )								
-	2	3	4	5	6	7	8	9	10
2	0.50	-	-	-	-	-	-	-	-
3	1.50	1.00	-	-	-	-	-	-	-
4	3.00	1.79	1.54	-	-	-	-	-	-
5	5.00	2.79	2.19	2.08	-	-	-	-	-
6	7.50	4.04	2.91	2.67	2.60	-	-	-	-
7	10.50	5.54	3.76	3.29	3.17	3.12	-	-	-
8	14.00	7.28	4.74	3.97	3.73	3.67	3.63	-	-
9	18.00	9.27	5.82	4.71	4.32	4.21	4.17	4.15	-
10	22.50	11.52	7.04	5.54	4.95	4.74	4.68	4.66	4.64

Tabela 5: Distâncias Médias para o Problema de Ordenação por Transposições em Permutações sem Sinais.

$n$	Tamanho máximo das operações ( $\lambda$ )								
-	2	3	4	5	6	7	8	9	10
2	1	-	-	-	-	-	-	-	-
3	3	2	-	-	-	-	-	-	-
4	6	3	3	-	-	-	-	-	-
5	10	5	3	3	-	-	-	-	-
6	15	8	5	4	4	-	-	-	-
7	21	11	6	5	5	4	-	-	-
8	28	14	8	6	5	5	5	-	-
9	36	18	10	7	6	6	6	5	-
10	45	23	12	9	7	7	6	6	6

Tabela 6: Diâmetros para o Problema de Ordenação por Transposições ( $D_t^\lambda(n)$ ) em Permutações sem Sinais.

**A.2.3 Problema de Ordenação por Reversões e Transposições em Permutações sem Sinais**

$n$	Tamanho máximo das operações ( $\lambda$ )								
-	2	3	4	5	6	7	8	9	10
2	0.50	-	-	-	-	-	-	-	-
3	1.50	0.83	-	-	-	-	-	-	-
4	3.00	1.62	1.38	-	-	-	-	-	-
5	5.00	2.42	1.95	1.80	-	-	-	-	-
6	7.50	3.55	2.60	2.39	2.30	-	-	-	-
7	10.50	4.75	3.35	2.94	2.81	2.75	-	-	-
8	14.00	6.22	4.22	3.57	3.36	3.27	3.23	-	-
9	18.00	7.80	5.14	4.24	3.90	3.77	3.71	3.68	-
10	22.50	9.61	6.19	4.98	4.50	4.30	4.20	4.15	4.13

Tabela 7: Distâncias Médias para o Problema de Ordenação por Reversões e Transposições em Permutações sem Sinais.

$n$	Tamanho máximo das operações ( $\lambda$ )								
-	2	3	4	5	6	7	8	9	10
2	1	-	-	-	-	-	-	-	-
3	3	1	-	-	-	-	-	-	-
4	6	3	2	-	-	-	-	-	-
5	10	4	3	3	-	-	-	-	-
6	15	6	4	4	3	-	-	-	-
7	21	9	5	4	4	4	-	-	-
8	28	11	7	5	5	5	4	-	-
9	36	14	8	6	5	5	5	5	-
10	45	17	10	7	6	6	6	6	5

Tabela 8: Diâmetros para o Problema de Ordenação por Reversões e Transposições em Permutações sem Sinais ( $D_{rt}^\lambda(n)$ ).



### A.2.4 Problema de Ordenação por Reversões em Permutações com Sinais

$n$	Tamanho máximo das operações ( $\lambda$ )							
-	2	3	4	5	6	7	8	9
2	1.50	-	-	-	-	-	-	-
3	3.00	2.35	-	-	-	-	-	-
4	5.00	3.46	3.23	-	-	-	-	-
5	7.50	4.62	4.21	4.12	-	-	-	-
6	10.50	5.97	5.20	5.08	5.03	-	-	-
7	14.00	7.44	6.24	6.03	5.98	5.96	-	-
8	18.00	9.09	7.34	6.99	6.93	6.91	6.90	-
9	22.50	10.90	8.50	7.96	7.88	7.86	7.86	7.85

Tabela 9: Distâncias Médias para o Problema de Ordenação por Reversões em Permutações com Sinais.

$n$	Tamanho máximo das operações ( $\lambda$ )							
-	2	3	4	5	6	7	8	9
2	3	-	-	-	-	-	-	-
3	6	3	-	-	-	-	-	-
4	10	6	5	-	-	-	-	-
5	15	7	6	6	-	-	-	-
6	21	9	7	7	7	-	-	-
7	28	11	9	9	8	8	-	-
8	36	16	10	10	9	9	9	-
9	45	17	12	11	11	10	10	10

Tabela 10: Diâmetros para o Problema de Ordenação por Reversões em Permutações com Sinais ( $D_{\bar{r}}^{\lambda}(n)$ ).

**A.2.5 Problema de Ordenação por Reversões e Transposições em Permutações com Sinais**

$n$	Tamanho máximo das operações ( $\lambda$ )							
-	2	3	4	5	6	7	8	9
2	1.25	-	-	-	-	-	-	-
3	2.33	1.83	-	-	-	-	-	-
4	3.83	2.67	2.49	-	-	-	-	-
5	5.80	3.58	3.20	3.09	-	-	-	-
6	8.26	4.65	3.93	3.73	3.67	-	-	-
7	11.22	5.87	4.75	4.42	4.30	4.26	-	-
8	14.69	7.26	5.66	5.12	4.89	4.81	4.78	-
9	18.66	8.83	6.65	5.87	5.55	5.42	5.37	5.35

Tabela 11: Distâncias Médias para o Problema de Ordenação por Reversões e Transposições em Permutações com Sinais.

$n$	Tamanho máximo das operações ( $\lambda$ )							
-	2	3	4	5	6	7	8	9
2	2	-	-	-	-	-	-	-
3	4	3	-	-	-	-	-	-
4	7	4	4	-	-	-	-	-
5	11	5	5	4	-	-	-	-
6	16	7	6	5	5	-	-	-
7	22	9	7	6	6	6	-	-
8	29	12	8	7	7	6	6	-
9	37	15	10	8	7	7	7	7

Tabela 12: Diâmetros para o Problema de Ordenação por Reversões e Transposições em Permutações com Sinais ( $D_{rt}^\lambda(n)$ ).