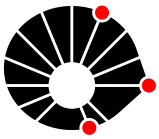

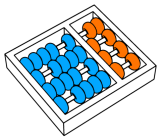

Formulações de PLI para Variantes do Problema de Partição de Strings

F. Romeiro *G. Siqueira* *Z. Dias*

Relatório Técnico - IC-PFG-24-14
Projeto Final de Graduação
2024 - Junho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Formulações de Programação Linear Inteira para Variantes do Problema de Partição de Strings

Felipe Romeiro* Gabriel Siqueira* Zanoni Dias*

Resumo

Este relatório analisa adaptações de dois modelos da literatura para variantes do problema de Partição Comum Mínima de Strings Balanceadas. Como esse problema se origina da Biologia Computacional, onde as strings representam genomas, propomos evoluções para os modelos que atendam variantes do problema. Nessas variantes consideramos o caso em que as strings não são balanceadas, representando genomas com conjuntos diferentes de genes, e o caso em que os caracteres possuem sinais positivo ou negativo, representando a orientação dos genes. Também levamos em conta variações considerando o número de nucleotídeos entre os genes. Ao final, os diferentes modelos foram testados e comparados em termos de tempo de execução e qualidade da solução.

1 Introdução

Problemas de distância de rearranjos de genomas e partição de strings são estudados na Biologia Computacional pois permitem estimar a distância evolutiva entre dois indivíduos. No problema de distância de rearranjos de genomas são dados dois genomas e um conjunto de operações permitidas, e busca-se a menor sequência de operações que transforme a primeira sequência de genes na segunda. Vários algoritmos de distância de rearranjos de genomas se baseiam no conceito de partição de strings e motivam as variantes do problema estudadas neste trabalho.

Uma forma de representar os genomas é por meio de strings, representando cada gene por um caractere. Adicionalmente, podemos adicionar outras informações biologicamente relevantes a essa representação, como a orientação dos genes e a quantidade de nucleotídeos entre os genes. Neste relatório usamos o termo genomas para nos referir a representação com informações adicionais sobre as regiões intergênicas. No problema de partição de strings são dadas duas strings ou genomas e o objetivo é achar o menor conjunto de substrings que podem ser concatenadas e resultarem nas strings de entrada. Sua forma mais básica é a Partição Comum Mínima de Strings Balanceadas (ou *MCBSP*, do inglês *Minimal Common Balanced String Partition*), em que uma string é anagrama da outra. Relaxando essa restrição, obtém-se a Partição Comum Mínima de Strings (*MCSP*, do inglês *Minimal Common String Partition*). Em alguns casos é interessante anotar a orientação dos genes, o que usualmente é realizado assinalando um sinal ($+$ ou $-$) para cada caractere das strings,

*Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP

resultando na Partição Comum Mínima de Strings Balanceadas com Sinais (*SMCBSP*, do inglês *Signed Minimal Common Balanced String Partition*) e na Partição Comum Mínima de Strings com Sinais (*SMCSP*, do inglês *Signed Minimal Common String Partition*).

Chen et al. (2005) mostraram que uma aproximação com fator ℓ para o problema SMCBSP pode ser usada para construir uma aproximação com fator 2ℓ para rearranjo de genomas considerando a operação de reversão. De forma similar uma aproximação com fator ℓ para o problema MCBSP pode ser usada para construir uma aproximação com fator 3ℓ para a distância de rearranjos de genomas considerando a operação de transposição (SHAPIRA; STORER, 2007).

Em trabalhos mais recentes, os problemas de distância de rearranjos de genomas têm considerado as quantidades de nucleotídeos entre genes, as quais nos referimos como tamanhos das regiões intergênicas (BULTEAU; FERTIN; TANNIER, 2016; BRITO; JEAN et al., 2020; OLIVEIRA et al., 2021; ALEXANDRINO; OLIVEIRA; DIAS et al., 2021; BRITO; OLIVEIRA et al., 2021; ALEXANDRINO; BRITO et al., 2022; ALEXANDRINO; OLIVEIRA; JEAN et al., 2022; SIQUEIRA; ALEXANDRINO; DIAS, 2022; ALEXANDRINO; OLIVEIRA; JEAN et al., 2023; SIQUEIRA; OLIVEIRA; DIAS, 2023). Utilizando essa informação temos também o problema de Partição Comum Mínima Intergênica de String com Sinais (*SMCISP*, do inglês *Signed Minimal Common Intergenic String Partition*). Também existe uma representação onde as regiões intergênicas possuem um intervalo de tolerância (BRITO; ALEXANDRINO et al., 2023), o que resulta no problema de Partição Comum Mínima Intergênica Flexível de Strings com Sinais (*SMCFISP*, do inglês *Signed Minimal Common Flexible Intergenic String Partition*).

Todas estas variantes são difíceis, o que motiva a exploração de algoritmos de Programação Linear Inteira. Goldstein, Kolman e Zheng (2004) demonstraram que MCBSP, MCSP e SMCSP são NP-difíceis. Siqueira, Oliveira e Dias (2023) demonstraram que SMCISP é NP-difícil. Siqueira, Alexandrino, Oliveira et al. (2023) demonstraram que SMCFISP é NP-difícil.

2 Definições

Nesta seção serão introduzidos conceitos utilizados ao longo do relatório. Os conceitos foram organizados pelos problemas que os motivam. Sendo assim, a Seção 2.1 explora os conceitos relacionados ao MCBSP. A Seção 2.2 aborda o MCSP. A Seção 2.3 aborda o SMCSP. A Seção 2.4 explora os conceitos do SMCISP. Por fim, a Seção 2.5 aborda o SMCFISP.

2.1 Partição Comum Mínima de Strings Balanceadas (*MCBSP*)

Uma string S é uma sequência de caracteres de tamanho $|S|$, onde o *i-ésimo* caractere é dado por S_i. Em alguns casos pode ser interessante identificar a orientação dos caracteres, o que é feito associando um sinal ($+$ ou $-$) a cada um deles. Nesse caso, a string é denominada *com sinais* e, caso contrário, *sem sinais*.

O conjunto de caracteres distintos de S é denominado *alfabeto* de S e é representado por Σ_S. Para evitar ambiguidade, os elementos de Σ_S são denominados *rótulos*, enquanto que

os elementos de S são chamados de caracteres. No caso de strings com sinais, os caracteres incluem os sinais e os rótulos não, ou seja, os caracteres $+\alpha$ e $-\alpha$ se referem ao rótulo α.

A quantidade de vezes que um rótulo $\alpha \in \Sigma_S$ ocorre em uma string S é denominada ocorrência de α e denotada por $occ(\alpha, S)$. Um rótulo $\beta \notin \Sigma_S$ tem sua ocorrência definida como 0 (isto é, $occ(\beta, S) = 0$).

Duas strings $S1$ e $S2$ são ditas balanceadas se possuem o mesmo alfabeto (isto é, $\Sigma_{S1} = \Sigma_{S2} = \Sigma$) e cada rótulo tem a mesma ocorrência nas duas strings, ou seja, $occ(\alpha, S1) = occ(\alpha, S2)$ para todo $\alpha \in \Sigma$. Isto implica que as duas strings têm o mesmo tamanho.

Dadas duas string $S1$ e $S2$ não balanceadas, se um rótulo $\alpha \in \Sigma$ aparece mais vezes em $S1$ (respectivamente $S2$) ele é dito *abundante em* $S1$ (respectivamente *abundante em* $S2$). Caso contrário, ele é denominado *raro*.

Definição 1. *Dadas duas strings balanceadas sem sinais $S1$ e $S2$, uma partição direta $(\mathbb{S}1, \mathbb{S}2, \phi)$ de $S1$ e $S2$ é composta por duas sequências $\mathbb{S}1$ e $\mathbb{S}2$ de strings e uma permutação ϕ tais que:*

- *$\mathbb{S}1$ e $\mathbb{S}2$ têm o mesmo tamanho ($|\mathbb{S}1| = |\mathbb{S}2|$);*
- *a concatenação das strings de $\mathbb{S}1$ é igual a $S1$;*
- *a concatenação das strings de $\mathbb{S}2$ é igual a $S2$;*
- $\mathbb{S}2_i = \mathbb{S}1_{\phi_i}$, $\forall 1 \leq i \leq |\mathbb{S}1|$

Exemplo 1. *Uma partição direta entre duas strings balanceadas $S1$ e $S2$ sem sinais.*

$$\begin{aligned}
S1 &= (\ E\ \ B\ \ A\ \ C\ \ D\ \ E\ \ D\ ) \\
S2 &= (\ D\ \ C\ \ A\ \ E\ \ B\ \ D\ \ E\ ) \\
\mathbb{S}1 &= \langle (E\ \ B), (A), (C), (D\ \ E), (D) \rangle \\
\mathbb{S}2 &= \langle (D), (C), (A), (E\ \ B), (D\ \ E) \rangle \\
\phi &= (\ 5\ \ 3\ \ 2\ \ 1\ \ 4\ )
\end{aligned}$$

Partição Comum Mínima de Strings Balanceadas (*MCBSP*)	
Entrada:	Duas strings balanceadas sem sinais $S1$ e $S2$.
Objetivo:	Encontrar uma partição direta $(\mathbb{S}1, \mathbb{S}2, \phi)$ de $S1$ e $S2$ tal que $\mathbb{S}1$ tenha o menor tamanho possível.

2.2 Partição Comum Mínima de Strings (*MCSP*)

Definição 2. *Dadas duas strings sem sinais $S1$ e $S2$, uma partição direta $(\mathbb{S}1, \mathbb{S}2, \sigma)$ de $S1$ e $S2$ é composta por duas sequências $\mathbb{S}1$ e $\mathbb{S}2$ de strings e uma bijeção σ entre os elementos de uma subsequência $\mathbb{S}1^\sigma$ de $\mathbb{S}1$ e os elementos de uma subsequência $\mathbb{S}2^\sigma$ de $\mathbb{S}2$ tais que:*

- *a concatenação das strings de* $\mathbb{S}1$ *é igual a* $S1$*;*
- *a concatenação das strings de* $\mathbb{S}2$ *é igual a* $S2$*;*
- σ *é tal que* $\sigma(\mathbb{S}1_i) = \mathbb{S}2_j$ *para* $\mathbb{S}1_i \in \mathbb{S}1$ *e* $\mathbb{S}2_j \in \mathbb{S}2$*, somente se* $\mathbb{S}1_i = \mathbb{S}2_j$*;*
- *sendo* X *o conjunto de caracteres em* $\mathbb{S}1$ *que não estão em* $\mathbb{S}1^\sigma$ *e* Y *o conjunto de caracteres em* $\mathbb{S}2$ *que não estão em* $\mathbb{S}2^\sigma$*, então* $X \cap Y = \emptyset$*.*

Exemplo 2. *Uma partição direta entre duas strings* $S1$ *e* $S2$ *sem sinais.*

$$
\begin{array}{rcl}
S1 & = & (\quad E \quad B \quad A \quad C \quad D \quad E \quad D \quad) \\
S2 & = & (\quad B \quad D \quad C \quad D \quad E \quad B \quad E \quad D \quad D \quad) \\
\mathbb{S}1 & = & \langle (E \quad B), (A), (C \quad D), (E \quad D) \rangle \\
\mathbb{S}2 & = & \langle (B \quad D), (C \quad D), (E \quad B), (E \quad D), (D) \rangle \\
 & & \sigma(\mathbb{S}1_1) = \mathbb{S}2_3, \sigma(\mathbb{S}1_3) = \mathbb{S}2_2, \sigma(\mathbb{S}1_4) = \mathbb{S}2_4 \\
X & = & \{A\} \\
Y & = & \{C \quad D\}
\end{array}
$$

Partição Comum Mínima de Strings (*MCSP*)	
Entrada:	Duas strings sem sinais $S1$ e $S2$.
Objetivo:	Encontrar uma partição direta $(\mathbb{S}1, \mathbb{S}2, \sigma)$ entre $S1$ e $S2$ tal que $\lvert\mathbb{S}1\rvert + \lvert\mathbb{S}2 \setminus \mathbb{S}2^\sigma\rvert$ seja mínimo.

2.3 Partição Comum Mínima de Strings com Sinais (*SMCSP*)

Dada uma string com sinais S, diz-se que uma string $Q = inv(S)$ é *inversa* de S se ela consiste dos caracteres de S na ordem inversa e com os sinais invertidos, isto é, $Q_i = -S_{|S|-i+1}$, $\forall 1 \leq i \leq |S|$.

Definição 3. *Dadas duas strings com sinais* $S1$ *e* $S2$*, uma partição com sinais* $(\mathbb{S}1, \mathbb{S}2, \sigma)$ *de* $S1$ *e* $S2$ *é composta por duas sequências* $\mathbb{S}1$ *e* $\mathbb{S}2$ *de strings e uma bijeção* σ *entre os elementos de uma subsequência* $\mathbb{S}1^\sigma$ *de* $\mathbb{S}1$ *e os elementos de uma subsequência* $\mathbb{S}2^\sigma$ *de* $\mathbb{S}2$ *tais que:*

- *a concatenação das strings de* $\mathbb{S}1$ *é igual a* $S1$*;*
- *a concatenação das strings de* $\mathbb{S}2$ *é igual a* $S2$*;*
- σ *é tal que* $\sigma(\mathbb{S}1_i) = \mathbb{S}2_j$ *para* $\mathbb{S}1_i \in \mathbb{S}1$ *e* $\mathbb{S}2_j \in \mathbb{S}2$*, somente se* $S1_i = S2_j$ *ou* $S1_i = inv(S2_j)$*;*
- *sendo* X *o conjunto de rótulos dos caracteres em* $\mathbb{S}1$ *que não estão em* $\mathbb{S}1^\sigma$ *e* Y *o conjunto de rótulos dos caracteres em* $\mathbb{S}2$ *que não estão em* $\mathbb{S}2^\sigma$*, então* $X \cap Y = \emptyset$*;*

Exemplo 3. *Uma partição com sinais entre duas strings $S1$ e $S2$ com sinais.*

$$\begin{aligned}
S1 &= (\ -A\ \ -E\ \ +B\ \ -A\ \ -C\ \ +D\ \ +E\ \ -D\ ) \\
S2 &= (\ -D\ \ +C\ \ +A\ \ -E\ \ +B\ \ +D\ \ -E\ ) \\
\mathbb{S}1 &= \langle(-A),(-E\ \ +B),(-A\ \ -C\ \ +D),(+E\ \ -D)\rangle \\
\mathbb{S}2 &= \langle(-D\ \ +C\ \ +A),(-E\ \ +B),(+D\ \ -E)\rangle \\
&\quad \sigma(\mathbb{S}1_2) = \mathbb{S}2_2, \sigma(\mathbb{S}1_3) = \mathbb{S}2_1, \sigma(\mathbb{S}1_4) = \mathbb{S}2_3 \\
X &= \{A\} \\
Y &= \emptyset
\end{aligned}$$

Partição Comum Mínima de Strings com Sinais (*SMCSP*)	
Entrada:	Duas strings com sinais $S1$ e $S2$.
Objetivo:	Encontrar uma partição com sinais $(\mathbb{S}1, \mathbb{S}2, \sigma)$ de $S1$ e $S2$ tal que $\lvert\mathbb{S}1\rvert + \lvert\mathbb{S}2 \setminus \mathbb{S}2^{\sigma}\rvert$ seja mínimo.

2.4 Partição Comum Mínima Intergênica de String com Sinais (*SMCISP*)

Neste problema, um genoma $G = (S, \breve{S})$ é composto por uma string S e por uma sequência de regiões intergênicas $\breve{S}$ de tamanho $|S|-1$. O genoma é dito rígido se $\breve{S}$ for uma sequência de números naturais.

Dados dois genomas $G1 = (S1, \breve{S1})$ e $G2 = (S2, \breve{S2})$, diz-se que $G2 = inv(G1)$ é inversa de $G1$ se $S2 = inv(S1)$ e se $\breve{S2}$ é a sequência dos elementos de $\breve{S1}$ na ordem contrária, ou seja, $\breve{S2}_i = \breve{S1}_{|S1|-i}$, $\forall 1 \leq i \leq |S1| - 1$.

Uma *quebra* de um genoma $G1 = (S1, \breve{S1})$ é uma operação que separa $G1$ em dois genomas $G1' = (S1', \breve{S1'})$ e $G1'' = (S1'', \breve{S1''})$ em uma região intergênica $\breve{S1}_i$ tal que

- $|S1'| = i$;
- $S1'_j = S1_j$, $\forall 1 \leq j \leq i$;
- $|S1''| = |S1| - i$;
- $S1''_{j-i} = S1_j$, $\forall i < j \leq |S1|$;
- $|\breve{S1'}| = i$;
- $\breve{S1'}_j = \breve{S1}_j$, $\forall 1 \leq j < i$;
- $|\breve{S1''}| = |\breve{S1}| - i$;
- $\breve{S1''}_{j-i} = \breve{S1}_j$, $\forall i < j \leq |S1| - 1$;

Definição 4. *Dados dois genomas rígidos* $G1 = (S1, \breve{S}1)$ *e* $G2 = (S2, \breve{S}2)$*, uma partição com sinais* $(\mathbb{S}1, \mathbb{S}2, \sigma)$ *é composta por duas sequências de genomas* $\mathbb{S}1$ *e* $\mathbb{S}2$ *e uma bijeção* σ *entre os elementos de uma subsequência* $\mathbb{S}1^\sigma$ *de* $\mathbb{S}1$ *e os elementos de uma subsequência* $\mathbb{S}2^\sigma$ *de* $\mathbb{S}2$ *tais que:*

- *o genoma* $G1$ *possa ser quebrado nos genomas de* $\mathbb{S}1$*;*
- *o genoma* $G2$ *possa ser quebrado nos genomas de* $\mathbb{S}2$*;*
- σ *é tal que* $\sigma(\mathbb{S}1_i) = \mathbb{S}2_j$ *para* $\mathbb{S}1_i \in \mathbb{S}1$ *e* $\mathbb{S}2_j \in \mathbb{S}2$ *somente se* $\mathbb{S}1_i = \mathbb{S}2_j$ *ou* $\mathbb{S}1_i = inv(\mathbb{S}2_j)$*;*
- *sendo* X *o conjunto de rótulos dos caracteres em* $\mathbb{S}1$ *que não estão em* $\mathbb{S}1^\sigma$ *e* Y *o conjunto de rótulos dos caracteres em* $\mathbb{S}2$ *que não estão em* $\mathbb{S}2^\sigma$*, então* $X \cap Y = \emptyset$*;*

Exemplo 4. *Uma partição com sinais entre dois genomas rígidos* $(S1, \breve{S}1)$ *e* $(S2, \breve{S}2)$*.*

$$
\begin{array}{rcl}
S1 & = & (\ +A\ \ -E\ \ +B\ \ -A\ \ -C\ \ +D\ \ +E\ \ -D\ ) \\
S2 & = & (\ -D\ \ +C\ \ +A\ \ -E\ \ -B\ \ +D\ \ -E\ ) \\
\breve{S}1 & = & (\ 63\ \ 44\ \ 23\ \ 52\ \ 32\ \ 38\ \ 4\ ) \\
\breve{S}2 & = & (\ 32\ \ 0\ \ 63\ \ 42\ \ 34\ \ 4\ ) \\
\mathbb{S}1 & = & \langle((+A\ \ -E), (63)), ((+B), ()), ((-A), ()), ((-C\ \ +D), (32)), \\
 & & (+E\ \ -D, (4))\rangle \\
\mathbb{S}2 & = & \langle((-D\ \ +C, (32)), ((+A\ \ -E), (63)), ((-B), ()), ((+D\ \ -E), (4))\rangle \\
 & & \sigma(\mathbb{S}1_1) = \mathbb{S}2_2, \sigma(\mathbb{S}1_2) = \mathbb{S}2_3, \sigma(\mathbb{S}1_4) = \mathbb{S}2_1, \sigma(\mathbb{S}1_5) = \mathbb{S}2_4 \\
X & = & \{A\} \\
Y & = & \emptyset
\end{array}
$$

Partição Comum Mínima Intergênica de Strings com Sinais (*SMCFSP*)	
Entrada:	Dois genomas rígidos $G1$ e $G2$.
Objetivo:	Encontrar uma partição com sinais $(\mathbb{S}1, \mathbb{S}2, \sigma)$ de $G1$ e $G2$ tal que $\lvert\mathbb{S}1\rvert + \lvert\mathbb{S}2 \setminus \mathbb{S}2^\sigma\rvert$ seja mínimo.

2.5 Partição Comum Mínima Intergênica Flexível de Strings com Sinais (*SMCFISP*)

Um genoma flexível $G = (S, \breve{S})$ é composto por uma string S e uma sequência de pares de números naturais $\breve{S}$ de tamanho $|S| - 1$ tais que $\breve{S}_i = (\breve{S}_i^{min}, \breve{S}_i^{max})$ e $\breve{S}_i^{min} \leq \breve{S}_i^{max}$.

Dados um genoma rígido $G1 = (S1, \breve{S}1)$ e um genoma flexível $G2 = (S2, \breve{S}2)$, diz-se que $G1$ e $G2$ são compatíveis se $S1 = S2$ e $\breve{S}2_i^{min} \leq \breve{S}1_i \leq \breve{S}2_i^{max}$, $\forall 1 \leq i \leq |S1| - 1$.

Definição 5. *Dados um genoma rígido* $G1 = (S1, \breve{S}1)$ *e um genoma flexível* $G2 = (S2, \breve{S}2)$*, uma partição com sinais* $(\mathbb{S}1, \mathbb{S}2, \sigma)$ *é composta por uma sequência de genomas rígidos* $\mathbb{S}1$*, uma sequência de genomas flexíveis* $\mathbb{S}2$ *e uma bijeção* σ *entre os elementos de uma subsequência* $\mathbb{S}1^\sigma$ *de* $\mathbb{S}1$ *e os elementos de uma subsequência* $\mathbb{S}2^\sigma$ *de* $\mathbb{S}2$ *tais que:*

- *o genoma* $G1$ *possa ser quebrado nos genomas de* $\mathbb{S}1$*;*
- *o genoma* $G2$ *possa ser quebrado nos genomas de* $\mathbb{S}2$*;*
- σ *é tal que* $\sigma(\mathbb{S}1_i) = \mathbb{S}2_j$ *para* $\mathbb{S}1_i \in \mathbb{S}1$ *e* $\mathbb{S}2_j \in \mathbb{S}2$ *somente se* $\mathbb{S}1_i$ *for compatível com* $\mathbb{S}2_j$ *ou com* $inv(\mathbb{S}2_j)$*;*
- *sendo* X *o conjunto de rótulos dos caracteres em* $\mathbb{S}1$ *que não estão em* $\mathbb{S}1^\sigma$ *e* Y *o conjunto de rótulos dos caracteres em* $\mathbb{S}2$ *que não estão em* $\mathbb{S}2^\sigma$*, então* $X \cap Y = \emptyset$*;*

Exemplo 5. *Uma partição com sinais entre um genoma rígido* $(S1, \breve{S}1)$ *e um genoma flexível* $(S2, \breve{S}2)$*.*

$$
\begin{aligned}
S1 &= (\ +A\ \ -E\ \ +B\ \ -A\ \ -C\ \ +D\ \ +E\ \ -D\ ) \\
S2 &= (\ -D\ \ +C\ \ +A\ \ -E\ \ +B\ \ +D\ \ -E\ ) \\
\breve{S}1 &= (\ 63\ \ 44\ \ 23\ \ 52\ \ 32\ \ 38\ \ 4\ ) \\
\breve{S}2 &= (\ (25,47)\ \ (0,54)\ \ (63,64)\ \ (42,45)\ \ (34,68)\ \ (0,10)\ ) \\
\mathbb{S}1 &= \langle((+A),()),((-E\ \ +B),(44)),((-A\ \ -C\ \ +D),(52\ \ 32)), \\
&\quad (+E\ \ -D,(4))\rangle \\
\mathbb{S}2 &= \langle((-D\ \ +C\ \ +A,((24,47)\ \ (0,54))),((-E\ \ +B),((42,45)), \\
&\quad ((+D\ \ -E),((0,10))))\rangle \\
&\quad \sigma(\mathbb{S}1_2) = \mathbb{S}2_2, \sigma(\mathbb{S}1_3) = \mathbb{S}2_1, \sigma(\mathbb{S}1_4) = \mathbb{S}2_3 \\
X &= \{A\} \\
Y &= \emptyset
\end{aligned}
$$

Partição Comum Mínima Intergênica Flexível de Strings com Sinais (*SMC-FISP*)	
Entrada:	Um genoma rígido $G1$ e um genoma flexível $G2$.
Objetivo:	Encontrar uma partição com sinais $(\mathbb{S}1, \mathbb{S}2, \sigma)$ de $G1$ e $G2$ tal que $\lvert\mathbb{S}1\rvert + \lvert\mathbb{S}2 \setminus \mathbb{S}2^\sigma\rvert$ seja mínimo.

3 Formulações de Programação Linear Inteira

Esta seção é dividida em duas partes. Na primeira são apresentadas duas formulações de programação linear inteira para MCBSP da literatura, enquanto que na segunda são discutidas as alterações necessárias sobre essas formulações para abordar cada um dos problemas introduzidos na seção anterior.

3.1 Formulações da Literatura

Blum, Lozano e Davidson (2015) introduziram a formulação de Blocos Comuns, daqui em diante identificada por **CB**. Dadas duas strings balanceadas de entrada $S1$ e $S2$, cada bloco comum b_i é representado por uma tripla $(t_i, k1_i, k2_i)$ composta por uma string t_i e dois índices $1 \leq k1_i, k2_i \leq |S1|$, indicando que a substring t_i inicia-se na posição $k1_i$ em $S1$ e na posição $k2_i$ em $S2$. Seja B o conjunto de todos os blocos comuns possíveis entre $S1$ e $S2$. Uma solução para o MCBSP pode ser representada por um subconjunto $B' \subseteq B$ tal que:

- $\sum_{b_i \in B'} |t_i| = |S1|$, isto é, a soma dos comprimentos das substrings dos blocos selecionados é igual ao comprimento das entradas e;
- para quaisquer dois blocos $(t_i, k1_i, k2_i) \neq (t_j, k1_j, k2_j) \in B'$, temos que não existe interseção entre os intervalos $[k1_i, k1_i + |t_i| - 1]$ e $[k1_j, k1_j + |t_j| - 1]$ e não existe interseção entre os intervalos $[k2_i, k2_i + |t_i| - 1]$ e $[k2_j, k2_j + |t_j| - 1]$.

O modelo **CB** usa, para cada bloco $b_i \in B$, uma variável binária x_i que indica se esse bloco faz parte ou não da solução S. Desta forma, pode-se escrever o modelo como a seguir, usando a notação de Blum (2020):

$$\min \quad \sum_{b_i \in B} x_i \tag{1}$$

$$\text{sujeito a} \quad \sum_{b_i \in B \text{ t.q. } k1_i \leq j < k1_i + |t_i|} x_i = 1 \qquad \forall 1 \leq j \leq |S1| \tag{2}$$

$$\sum_{b_i \in B \text{ t.q. } k2_i \leq j < k2_i + |t_i|} x_i = 1 \qquad \forall 1 \leq j \leq |S2| \tag{3}$$

$$x_i \in \{0, 1\} \qquad \forall b_i \in B$$

A função objetivo (1) garante que a solução encontrada será mínima. A restrição (2) garante que cada caractere de $S1$ é representado por um e somente um bloco de S, e portanto as duas condições da solução são válidas para $S1$. A restrição (3) é análoga para $S2$.

Um grande problema desta formulação é que substrings repetidas geram muito blocos, aumentando o tamanho da instância. Blum, Lozano e Davidson (2015) mostraram que grande parte dos blocos correspondem a substrings de tamanho 1.

Com este problema em mente, Blum e Raidl (2015) introduziram a formulação de Substrings Comuns, identificada por **CS**. Dadas duas strings balanceadas de entrada $S1$ e $S2$, seja T o conjunto de todas as substring comuns a $S1$ e $S2$. Para cada $t \in T$, seja $Q1_t$ (respectivamente $Q2_t$) o conjunto das posições em que t se inicia em $S1$ (respectivamente $S2$).

Para cada combinação de $t \in T$ e $k \in Q1_t$, este modelo usa uma variável binária $x^1_{t,k}$ para indicar que a substring t com início em k na string $S1$ faz parte da solução. As variáveis $x^2_{t,k}$ são análogas para $S2$. Desta forma, pode-se escrever o modelo como a seguir:

$$\min \quad \sum_{t \in T} \sum_{k \in Q1_t} x^1_{t,k} \tag{4}$$

$$\text{sujeito a} \quad \sum_{t \in T} \sum_{k \in Q1_t \text{ t.q. } k \le j < k+|t|} x^1_{t,k} = 1 \qquad \forall 1 \le j \le |S1| \tag{5}$$

$$\sum_{t \in T} \sum_{k \in Q2_t \text{ t.q. } k \le j < k+|t|} x^2_{t,k} = 1 \qquad \forall 1 \le j \le |S2| \tag{6}$$

$$\sum_{k \in Q1_t} x^1_{t,k} = \sum_{k \in Q2_t} x^2_{t,k} \qquad \forall t \in T \tag{7}$$

$$x^1_{t,k} \in \{0,1\} \qquad \forall t \in T, k \in Q1_t$$

$$x^2_{t,k} \in \{0,1\} \qquad \forall t \in T, k \in Q2_t$$

A função objetivo (4) indica a quantidade de substrings selecionadas e garante que a solução encontrada é mínima. A restrição (5) garante que cada caractere de $S1$ é englobado por uma e somente uma substring de $S1$ selecionada. A restrição (6) é análoga para $S2$. A restrição (7) garante que cada substring é selecionada uma mesma quantidade de vezes em cada uma das strings $S1$ e $S2$ e, portanto, é possível pareá-las e montar uma partição a partir das strings selecionadas.

3.2 Adaptações das Formulações

Nesta seção serão apresentadas as alterações necessárias para que as formulações introduzidas anteriormente tratem os problemas apresentados na Seção 2. A Seção 3.2.1 apresenta as mudanças necessárias para o MCSP. A Seção 3.2.2 apresenta as mudanças necessárias para o SMCSP. A Seção 3.2.3 apresenta as mudanças necessárias para o SMCISP e, a Seção 3.2.4, para o SMCFISP.

3.2.1 Partição Comum Mínima de Strings

No MCSP não se tem a garantia de que as strings são balanceadas, fazendo com que surjam partes exclusivas a uma das strings na partição. Nas definições da seção 2, estas partes são indicadas por $\mathbb{S}1 \setminus \mathbb{S}1^\sigma$ e $\mathbb{S}2 \setminus \mathbb{S}2^\sigma$. Para tratar estas partes, será introduzido o conceito de blocos exclusivos. Um bloco exclusivo é um bloco disjunto de todos os blocos comuns da partição. Porém, não é possível saber quais substrings serão parte de blocos exclusivos a priori, então usaremos um conjunto mais abrangente.

Teorema 1. *Dada uma partição com sinais* $(\mathbb{S}1, \mathbb{S}2, \sigma)$*, as partes exclusivas são formadas apenas por caracteres com rótulos abundantes.*

Demonstração. Por contradição e sem perda de generalidade, assuma que exista uma parte exclusiva $\mathbb{S}1_i$ que contém um caractere com um rótulo raro α. Pela definição de rótulo raro, α aparece ao menos $occ(\alpha, \mathbb{S}1)$ vezes em $\mathbb{S}2$. Porém, as partes comuns (não exclusivas) de $\mathbb{S}1$ e $\mathbb{S}2$ contém no máximo $occ(\alpha, \mathbb{S}1) - 1$ instâncias de α. Logo, as partes exclusivas de $\mathbb{S}2$

também têm ao menos uma instância de α. Isto implica uma contradição com a definição de partição que exige que os caracteres das partes exclusivas de $\mathbb{S}1$ não apareçam nas partes exclusivas de $\mathbb{S}2$. Portanto, as partes exclusivas são formadas apenas por caracteres com rótulos abundantes. □

Sabendo que os blocos exclusivos são formados apenas por caracteres com rótulos abundantes, considere o conjunto de todos os blocos exclusivos possíveis, ou seja, todas as substrings formadas somente por caracteres com rótulos abundantes. Denote um possível bloco exclusivo $e1_i$ de $S1$ pela tupla $(r1_i, l1_i)$, onde $r1_i$ é uma substring de caracteres com rótulos abundantes em $S1$ e $l1_i$ é a posição onde esta substring se inicia em $S1$ e denote um bloco exclusivo $e2_i = (r2_i, l2_i)$ de $S2$ de forma análoga. Sejam $E1$ e $E2$ o conjunto de todos os possíveis blocos exclusivos de $S1$ e $S2$ respectivamente.

Sendo $y1_i$ e $y2_i$ as variáveis binárias que indicam a inclusão das partes exclusivas dos conjuntos $E1$ e $E2$, respectivamente, na solução, considere a seguinte adaptação de **CB** para MCSP:

$$\min \quad \sum_{b_i \in B} x_i + \sum_{e1_i \in E1} y1_i + \sum_{e2_i \in E2} y2_i \tag{8}$$

$$\text{sujeito a} \quad \sum_{b_i \in B \text{ t.q. } k1_i \leq j < k1_i + |t_i|} x_i + \sum_{e1_i \in E1 \text{ t.q. } l1_i \leq j < l1_i + |r1_i|} y1_i = 1 \quad \forall 1 \leq j \leq |S1| \tag{9}$$

$$\sum_{b_i \in B \text{ t.q. } k2_i \leq j < k2_i + |t_i|} x_i + \sum_{e2_i \in E2 \text{ t.q. } l2_i \leq j < l2_i + |r2_i|} y2_i = 1 \quad \forall 1 \leq j \leq |S2| \tag{10}$$

$$x_i \in \{0,1\} \qquad \forall b_i \in B$$

$$y1_i \in \{0,1\} \qquad \forall e1_i \in E1$$

$$y2_i \in \{0,1\} \qquad \forall e2_i \in E2$$

A função objetivo (8) é a mesma da descrição do problema pois $\sum_{b_i \in B} x_i + \sum_{e1_i \in E1} y1_i$ contém todas as partes de $S1$ ($\mathbb{S}1$) e $\sum_{e2_i \in E2} y2_i$ contém as partes exclusivas de $S2$ ($\mathbb{S}2 \backslash \mathbb{S}2^{\sigma}$). A restrição (9) implica que todo caractere de $S1$ faz parte de um e somente um bloco comum ou bloco exclusivo. A restrição (10) é análoga para $S2$.

Note ainda que se o rótulo de um caractere é abundante em uma das strings, ele necessariamente é raro na outra. Isto implica que um rótulo só pode aparecer nos blocos exclusivos de uma das strings e, portanto, o conjunto de blocos encontrado por esta formulação satisfaz a definição de partição.

A alteração para a formulação **CS** é parecida. Considerando $E1$ e $E2$ como anteriormente e sendo $y1_i$ e $y2_i$ as variáveis que indicam a inclusão dos blocos exclusivo na solução:

$$\min \quad \sum_{t\in T}\sum_{k\in Q1_t} x^1_{t,k} + \sum_{e1_i\in E1} y1_i + \sum_{e2_i\in E2} y2_i \tag{11}$$

$$\text{sujeito a} \quad \sum_{t\in T}\sum_{k\in Q1_t \text{ t.q. } k\le j<k+|t|} x^1_{t,k} + \sum_{e1_i\in E1 \text{ t.q. } l1_i\le j<l1_i+|r1_i|} y1_j = 1 \quad \forall 1 \le j \le |S1| \tag{12}$$

$$\sum_{t\in T}\sum_{k\in Q2_t \text{ t.q. } k\le j<k+|t|} x^2_{t,k} + \sum_{e2_i\in E2 \text{ t.q. } l2_i\le j<l2_i+|r2_i|} y2_i = 1 \quad \forall 1 \le j \le |S2| \tag{13}$$

$$\sum_{k\in Q1_t} x^1_{t,k} = \sum_{k\in Q2_t} x^2_{t,k} \quad \forall t \in T \tag{14}$$

$$x^1_{t,k} \in \{0,1\} \quad \forall t \in T, k \in Q1_t$$

$$x^2_{t,k} \in \{0,1\} \quad \forall t \in T, k \in Q2_t$$

$$y1_i \in \{0,1\} \quad \forall e1_i \in E1$$

$$y2_i \in \{0,1\} \quad \forall e2_i \in E2$$

Novamente, a função objetivo (11) é a mesma do MCSP e as restrições (12) e (13) garantem que cada caractere é parte de um e somente um dos blocos selecionados. A restrição (14), assim como no MCBSP, garante que os blocos comuns são escolhidos na mesma quantidade em cada uma das strings.

3.2.2 Partição Comum Mínima de Strings com Sinais

A formulação **CB** se mantém como no caso anterior, mas utilizando o conceito de *blocos comuns com sinais*. Dadas duas strings com sinais $S1$ e $S2$, um bloco comum com sinais b_i é representado pela tripla $(t_i, k1_i, k2_i)$ composta por uma string t_i e dois indíces $1 \le k1_i \le |S1|$ e $1 \le k2_i \le |S2|$, indicando que $S1$ possui ou a substring t_i ou $inv(t_i)$ na posição $k1_i$ (e o análogo para $S2$ e $k2_i$). Os demais conceitos se mantém.

A formulação **CS** passa por uma alteração parecida. Dadas duas strings $S1$ e $S2$, seja T o conjunto de todas as strings t tais que t ou $inv(t)$ é substring de $S1$ e de $S2$. Para cada $t \in T$, seja $Q1_t$ (respectivamente $Q2_t$) o conjunto das posições em que t se inicia em $S1$ (respectivamente $S2$). O resto da formulação se mantém igual ao último caso.

3.2.3 Partição Comum Mínima Intergênica de Strings com Sinais

Novamente as formulações são parecidas com o caso anterior, porém com alterações nos conceitos para englobar as regiões intergênicas.

Para o modelo **CB** será necessário introduzir o conceito de blocos comuns de regiões intergênicas. Dados dois genomas rígidos $G1 = (S1, \breve{S1})$ e $G2 = (S2, \breve{S2})$, um bloco comum é representado pela tripla $(t_i, k1_i, k2_i)$ composta por um genoma t_i e dois índices $1 \le k1_i \le |S1|$ e $1 \le k2_i \le |S2|$, indicando que $S1$ possui o genoma t_i ou $inv(t_i)$ na posição $k1_i$ (e o análogo para $S2$ e $k2_i$). O resto da formulação se mantém igual ao caso anterior.

Já para a formulação **CS**, o conjunto T passa a ser de genomas ao invés de strings. Dados dois genomas rígidos $G1 = (S1, \breve{S}1)$ e $G2 = (S2, \breve{S}2)$, seja T o conjunto de todos os genomas t tais que t ou $inv(t)$ aparece em $G1$ e $G2$. Para cada $t \in T$, seja $Q1_t$ (respectivamente $Q2_t$) o conjunto das posições em que t ou $inv(t)$ se inicia em $S1$ (respectivamente $S2$). O resto da formulação se mantém igual ao último caso.

3.2.4 Partição Comum Mínima Intergênica Flexível de Strings com Sinais

Como SMCIFSP possui um genoma rígido e um genoma flexível, será necessário alterar um pouco a estrutura dos conjuntos das formulações.

Para a formulação **CB**, dados um genoma rígido $G1 = (S1, \breve{S}1)$ e um genoma flexível $G2 = (S2, \breve{S}2)$, um bloco comum é representado pela tripla $(t_i, k1_i, k2_i)$, indicando que o genoma rígido t_i aparece em $G1$ na posição $k1_i$ e é compatível com o genoma flexível que se inicia na posição $k2_i$ em $G2$ ou com seu inverso. O resto da formulação se mantém.

A formulação **CS** precisa de maiores alterações. Dados um genoma rígido $G1 = (S1, \breve{S}1)$ e um genoma flexível $G2 = (S2, \breve{S}2)$, seja T o conjunto de todos os genomas contidos em $G1$ que são compatíveis com algum genoma contido em $G2$. Para $t \in T$, seja $Q1_t$ o conjunto das posições em que t ocorre em $G1$ e $Q2_t$ o conjunto das posições em que um genoma compatível com t ou com $inv(t)$ aparece em $G2$. O resto da formulação se mantém como antes.

4 Experimentos

Os modelos descritos anteriormente e suas adaptações foram desenvolvidos usando Python 3.12.2 e resolvidos utilizando Gurobi 11. Os testes a seguir foram executados em um computador com uma CPU "Intel(R) Xeon(R) CPU E5-2470 v2" de 40 núcleos de 2.40GHz, 32GB de memória RAM e 9GB de swap. Cada teste foi executado em uma única *thread*, sendo que a montagem dos modelos não teve limites de tempo e a execução dos modelos foi limitada em 1 hora.

Para a criação dos testes foram iterados dois parâmetros: quantidade de rótulos l entre $\{4, 20, 36\}$ e fração de operações conservativas c entre $\{0.6, 0.8, 1\}$. Para cada combinação de parâmetros foram criados 10 testes, sendo que para cada teste foi criado um genoma rígido com sinais de l rótulos e tamanho 1000 e foram executadas 1000 operações, sendo $1000c$ conservativas (todas reversões) e $1000(1 - c)$ indels (deleções seguidas de inserções), sobre esse para obter o segundo genoma. O genoma rígido original foi criado com a seleção aleatória e uniforme de genes de um alfabeto com l rótulos e de regiões intergenicas com valores no intervalo $[0, 100]$. As operações de reversão selecionam aleatoriamente de forma uniforme duas posições do genoma e invertem o segmento de genes e regiões intergênicas entre essas posições, invertendo os sinais dos genes afetados. Cada deleção remove um gene escolhido de forma aleatória e uniforme, além de remover aleatoriamente parte dos nucleotídeos presentes adjacentes a esse gene. Cada inserção insere um gene com um rótulo que não está presente no genoma em uma posição escolhida de forma aleatória e uniforme, além de inserir uma quantidade aleatória de nucleotídeos ao redor do gene. No máximo 100 nucleotídeos são inseridos em cada região.

$\|\Sigma\|$	Tolerância (%)	Tempo de execução do modelo (s)		Gap (%)		Tempo de execução total (s)	
		Média	DP	Média	DP	Média	DP
4	0	2103	284	0,0	0,0	3266	440
4	25	3540	204	1,6	0,8	4687	619
4	50	3381	480	0,6	0,4	4466	482
4	∞	3415	615	0,8	0,5	4503	619
20	0	3214	715	0,4	0,5	4230	718
20	25	2490	751	0,0	0,0	3518	735
20	50	2384	751	0,0	0,0	3397	750
20	∞	2824	892	0,2	0,4	3841	902
36	0	3603	1	0,5	0,2	4603	9
36	25	3471	416	0,3	0,2	4473	415
36	50	3472	412	0,3	0,3	4470	412
36	∞	3065	783	0,2	0,3	4071	785

Tabela 1: Resultados dos testes do modelo **CB** com 60% de operações conservativas. A tolerância 0 indica SMCISP e a tolerância ∞ indica SMCSP, ou seja, sem regiões intergênicas.

Assim, esses testes refletem o SMCISP. Para o teste da solução para SMCSP, as regiões intergênicas foram descartadas. Para testar as alterações do SMCFISP, foram definidas duas tolerâncias $v \in \{0.25, 0.5\}$ e para cada entrada de teste $(G1, G2 = (S2, \breve{S}2))$ foram construídos dois genomas flexíveis $G2' = (S2', \breve{S}2')$ tal que para cada $G2_i \in G2$, $\breve{S}2'^{min}_i = (1-v)\breve{S}2_i$ e $\breve{S}2'^{max}_i = (1+v)\breve{S}2_i$ e $S2' = S2$.

As tabelas 1, 2 e 3 listam, respectivamente, os resultados dos testes do modelo **CB** com 60%, 80% e 100% de operações conservativas. As tabelas 4, 5 e 6 são análogas para o modelo **CS**.

Pode-se observar que os testes com mais caracteres tendem a executar mais rápido e a obterem um gap menor, o que é esperado pois existem menos substrings comuns entre as entradas, resultando em modelos menores e, portanto, mais fáceis de serem otimizados. Ainda, nota-se que em ambos os modelos que os testes com 80% e 100% de operações conservativas tendem a executar mais rápido que os testes com 60%, o que sugere que os blocos exclusivos são computacionalmente mais caros do que os blocos comuns. No caso do modelo **CB**, esse efeito é mais perceptível nos testes com 36 rótulos, enquanto que nos modelo **CS** esse efeito pode ser observado em quase todos os testes e, em ambos os casos, os tempos de execução com 100% de operações conservativas são quase negligenciáveis. Isso sugere que melhorias no tratamente de blocos exclusivos seriam vantajosas para os modelos estudados. Uma possível melhoria seria considerar somente blocos exclusivos que tivessem ambas as extremidades adjacentes ou a um bloco comum ou a uma extremidade da string.

Os resultados do modelo **CB** foram inconclusivos em relação ao impacto da tolerância das regiões intergênicas flexíveis na execução do modelo: há casos em que os testes com menor tolerância executaram melhor e outros em que os testes com mais tolerância foram melhores. Apesar disso, pode-se observar que tanto no caso de 60% de operações conser-

$\|\Sigma\|$	Tolerância (%)	Tempo de execução do modelo (s)		Gap (%)		Tempo de execução total (s)	
		Média	DP	Média	DP	Média	DP
4	0	1866	255	0,0	0,0	2982	257
4	25	3603	2	3,7	6,6	4719	26
4	50	3603	2	6,1	7,6	4746	94
4	∞	3607	14	12,6	12,2	4734	22
20	0	3253	396	0,1	0,5	4264	398
20	25	2618	707	0,1	0,2	3626	706
20	50	2169	445	0,0	0,0	3178	436
20	∞	3542	190	9,4	19,3	4549	189
36	0	3247	1125	0,3	0,2	4160	1437
36	25	2929	1129	0,2	0,2	3839	1408
36	50	2333	1127	0,1	0,1	3247	1366
36	∞	2431	1062	0,0	0,0	3349	1323

Tabela 2: Resultados dos testes do modelo **CB** com 80% de operações conservativas. A tolerância 0 indica SMCISP e a tolerância ∞ indica SMCSP, ou seja, sem regiões intergênicas.

$\|\Sigma\|$	Tolerância (%)	Tempo de execução do modelo (s)		Gap (%)		Tempo de execução total (s)	
		Média	DP	Média	DP	Média	DP
4	0	2	1	0,0	0,0	138	1
4	25	3600	0	1,1	0,3	3740	1
4	50	3600	0	2,1	0,5	3744	1
4	∞	3600	0	9,4	1,5	3757	1
20	0	0	0	0,0	0,0	29	0
20	25	0	0	0,0	0,0	29	0
20	50	0	0	0,0	0,0	29	0
20	∞	28	26	0,0	0,0	58	26
36	0	0	0	0,0	0,0	17	0
36	25	0	0	0,0	0,0	17	0
36	50	0	0	0,0	0,0	17	0
36	∞	0	0	0,0	0,0	17	0

Tabela 3: Resultados dos testes do modelo **CB** somente com operações conservativas. A tolerância 0 indica SMCISP e a tolerância ∞ indica SMCSP, ou seja, sem regiões intergênicas.

$\|\Sigma\|$	Tolerância (%)	Tempo de execução do modelo (s)		Gap (%)		Tempo de execução total (s)	
		Média	DP	Média	DP	Média	DP
4	0	906	246	0,0	0,0	1969	243
4	25	1825	414	0,0	0,0	2919	404
4	50	1317	355	0,0	0,0	2388	350
4	∞	1818	422	0,0	0,0	2881	426
20	0	1062	169	0,0	0,0	2127	157
20	25	1073	316	0,0	0,0	2207	419
20	50	1327	621	0,0	0,0	2420	647
20	∞	1649	773	0,0	0,0	2760	771
36	0	1089	184	0,0	0,0	2146	184
36	25	937	61	0,0	0,0	1972	57
36	50	1030	69	0,0	0,0	2090	96
36	∞	927	144	0,0	0,0	1966	150

Tabela 4: Resultados dos testes do modelo **CS** com 60% de operações conservativas. A tolerância 0 indica SMCISP e a tolerância ∞ indica SMCSP, ou seja, sem regiões intergênicas.

$\|\Sigma\|$	Tolerância (%)	Tempo de execução do modelo (s)		Gap (%)		Tempo de execução total (s)	
		Média	DP	Média	DP	Média	DP
4	0	1094	273	0,0	0,0	2232	269
4	25	3454	621	0,4	0,3	4617	654
4	50	3429	608	0,5	0,3	4624	668
4	∞	3715	112	1,4	0,3	4839	128
20	0	851	88	0,0	0,0	1864	93
20	25	832	67	0,0	0,0	1851	64
20	50	808	79	0,0	0,0	1822	78
20	∞	1163	155	0,0	0,0	2185	163
36	0	885	341	0,0	0,0	1837	676
36	25	852	303	0,0	0,0	1800	630
36	50	901	355	0,0	0,0	1832	655
36	∞	867	316	0,0	0,0	1808	639

Tabela 5: Resultados dos testes do modelo **CS** com 80% de operações conservativas. A tolerância 0 indica SMCISP e a tolerância ∞ indica SMCSP, ou seja, sem regiões intergênicas.

$\|\Sigma\|$	Tolerância (%)	Tempo de execução do modelo (s)		Gap (%)		Tempo de execução total (s)	
		Média	DP	Média	DP	Média	DP
4	0	0	0	0,0	0,0	5	0
4	25	3600	0	0,4	0,2	3614	0
4	50	3600	0	1,0	0,2	3618	1
4	∞	3600	0	4,1	0,5	3612	0
20	0	0	0	0,0	0,0	3	0
20	25	0	0	0,0	0,0	5	0
20	50	0	0	0,0	0,0	7	0
20	∞	2	1	0,0	0,0	10	1
36	0	0	0	0,0	0,0	3	0
36	25	0	0	0,0	0,0	4	0
36	50	0	0	0,0	0,0	5	0
36	∞	0	0	0,0	0,0	6	0

Tabela 6: Resultados dos testes do modelo **CS** somente com operações conservativas. A tolerância 0 indica SMCISP e a tolerância ∞ indica SMCSP, ou seja, sem regiões intergênicas.

vativas quanto 80%, os testes de SMCISP com 4 caracteres executaram mais rápido que os demais, o que é curioso pois, por um lado, espera-se que testes de SMCISP executem mais rápidos que os demais por terem menos blocos comuns, mas, por outro, os testes de SMCISP com mais caracteres executaram por mais tempo, o que é inesperado, já que nesses casos também há menos blocos comuns.

Já no modelo **CS** o impacto da tolerância das regiões intergênicas é mais perceptível, o que é esperado, já que uma maior tolerância permite a construção de mais blocos comuns. Também é interessante notar que este modelo é melhor que o anterior em todos os cenários, executando mais rápido e obtendo gaps menores dentro do tempo proposto.

Ainda é interessante observar que nos casos de strings desbalanceadas, independente dos parâmetros, o tempo de montagem do modelo foi de cerca de 1000 segundos, o que, considerando o tempo de execução de 1 hora, é um tempo razoável. Além disso, em todos os cenários, o desvio padrão do tempo de execução do modelo frequentemente esteve próximo do desvio padrão do tempo total, indicando uma consistência que sugere que a etapa de modelagem sofre pouco impacto em razão das instâncias e que a maior parte da variação do tempo dos resultados vem da fase de execução do modelo.

5 Conclusão

Este trabalho demonstrou que as formulações **CB** e **CS** para MCBSP, introduzidas em Blum, Lozano e Davidson (2015) e Blum e Raidl (2015), podem ser adaptadas de maneira simples para resolver variantes comuns do problema de partição de strings. Os modelos considerando essas adaptações resolvem as instâncias testadas com respostas de qualidade, dado o tempo de execução escolhido para os testes. Em alguns casos as instâncias foram

resolvidas de forma ótima dentro do tempo limite selecionado.

Em trabalhos futuros, pode ser feita uma exploração de outros modelos para os problemas propostos. Também pode ser interessante o estudo de heurísticas para acelerar o tempo de solução dos modelos desenvolvidos. Outra possibilidade, dado o impacto dos blocos exclusivos nos tempos de execução dos modelos, seria tentar reduzir a quantidade destes blocos.

Referências

ALEXANDRINO, Alexsandro Oliveira; BRITO, Klairton Lima; OLIVEIRA, Andre Rodrigues; DIAS, Ulisses; DIAS, Zanoni. Reversal and Indel Distance with Intergenic Region Information. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, p. 1–13, 2022.

ALEXANDRINO, Alexsandro Oliveira; OLIVEIRA, Andre Rodrigues; DIAS, Ulisses; DIAS, Zanoni. Incorporating Intergenic Regions into Reversal and Transposition Distances with Indels. **Journal of Bioinformatics and Computational Biology**, v. 19, n. 06, 2021.

ALEXANDRINO, Alexsandro Oliveira; OLIVEIRA, Andre Rodrigues; JEAN, Géraldine; FERTIN, Guillaume; DIAS, Ulisses; DIAS, Zanoni. Reversal and Transposition Distance on Unbalanced Genomes Using Intergenic Information. **Journal of Computational Biology**, v. 30, n. 8, p. 861–876, 2023.

ALEXANDRINO, Alexsandro Oliveira; OLIVEIRA, Andre Rodrigues; JEAN, Géraldine; FERTIN, Guillaume; DIAS, Ulisses; DIAS, Zanoni. Transposition Distance Considering Intergenic Regions for Unbalanced Genomes. **International Symposium on Bioinformatics Research and Applications (ISBRA)**, v. 13760, p. 100–113, 2022.

BLUM, Christian. Minimum common string partition: on solving large-scale problem instances. **International Transactions in Operational Research**, v. 27, n. 1, p. 91–111, 2020.

BLUM, Christian; LOZANO, José A.; DAVIDSON, Pinacho. Mathematical programming strategies for solving the minimum common string partition problem. **European Journal of Operational Research**, v. 242, n. 3, p. 769–777, 2015.

BLUM, Christian; RAIDL, Günther R. Computational performance evaluation of two integer linear programming models for the minimum common string partition problem. **Optimization Letters**, v. 10, n. 1, p. 189–205, 2015.

BRITO, Klairton Lima; ALEXANDRINO, Alexsandro Oliveira; OLIVEIRA, Andre Rodrigues; DIAS, Ulisses; DIAS, Zanoni. Genome Rearrangement Distance with a Flexible Intergenic Regions Aspect. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, v. 20, n. 03, p. 1641–1653, 2023.

BRITO, Klairton Lima; JEAN, Géraldine; FERTIN, Guillaume; OLIVEIRA, Andre Rodrigues; DIAS, Ulisses; DIAS, Zanoni. Sorting by Genome Rearrangements on both Gene Order and Intergenic Sizes. **Journal of Computational Biology**, v. 27, n. 2, p. 156–174, 2020.

BRITO, Klairton Lima; OLIVEIRA, Andre Rodrigues; ALEXANDRINO, Alexsandro Oliveira; DIAS, Ulisses; DIAS, Zanoni. An Improved Approximation Algorithm for the Reversal and Transposition Distance Considering Gene Order and Intergenic Sizes. **Algorithms for Molecular Biology**, v. 16, n. 1, p. 1–21, 2021.

BULTEAU, Laurent; FERTIN, Guillaume; TANNIER, Eric. Genome Rearrangements with Indels in Intergenes Restrict the Scenario Space. **BMC Bioinformatics**, v. 17, n. 14, p. 426, 2016.

CHEN, Xin; ZHENG, Jie; FU, Zheng; NAN, Peng; ZHONG, Yang; LONARDI, Stefano; JIANG, Tao. Assignment of Orthologous Genes via Genome Rearrangement. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, v. 2, n. 4, p. 302–315, 2005.

GOLDSTEIN, Avraham; KOLMAN, Petr; ZHENG, Jie. Minimum Common String Partition Problem: Hardness and Approximations. Edição: R. Fleischer e G. Trippen. **International Symposium on Algorithms and Computation (ISAAC)**, p. 484–495, 2004.

OLIVEIRA, Andre Rodrigues; JEAN, Géraldine; FERTIN, Guillaume; BRITO, Klairton Lima; DIAS, Ulisses; DIAS, Zanoni. Sorting Permutations by Intergenic Operations. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, v. 18, n. 6, p. 2080–2093, 2021.

SHAPIRA, Dana; STORER, James A. Edit distance with move operations. **Journal of Discrete Algorithms**, v. 5, n. 2, p. 380–392, 2007.

SIQUEIRA, Gabriel; ALEXANDRINO, Alexsandro Oliveira; DIAS, Zanoni. Signed Rearrangement Distances Considering Repeated Genes and Intergenic Regions. **Proceedings of 14th International Conference on Bioinformatics and Computational Biology (BICoB'2022)**, v. 83, p. 31–42, 2022.

SIQUEIRA, Gabriel; ALEXANDRINO, Alexsandro Oliveira; OLIVEIRA, Andre Rodrigues; JEAN, Géraldine; FERTIN, Guillaume; DIAS, Zanoni. Approximating Rearrangement Distances with Replicas and Flexible Intergenic Regions. **International Symposium on Bioinformatics Research and Applications (ISBRA)**, v. 14248, p. 241–254, 2023.

SIQUEIRA, Gabriel; OLIVEIRA, Alexsandro Alexandrino; DIAS, Zanoni. Signed rearrangement distances considering repeated genes, intergenic regions, and indels. **Journal of Combinatorial Optimization**, v. 46, n. 2, p. 16, 2023.