



# Aplicação de Metaheurísticas ao Problema da Dupla Digestão em Cadeias de DNA

**Celio Henrique Nogueira Larcher Junior**

JUIZ DE FORA

DEZEMBRO, 2014

# Aplicação de Metaheurísticas ao Problema da Dupla Digestão em Cadeias de DNA

Celio Henrique Nogueira Larcher Junior

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Stênio Sã Rosário Furtado Soares

JUIZ DE FORA

DEZEMBRO, 2014

# Aplicação de Metaheurísticas ao Problema da Dupla Digestão em Cadeias de DNA

Celio Henrique Nogueira Larcher Junior

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

---

Stênio Sã Rosário Furtado Soares
D.Sc. em Ciência da Computação / UFF

---

Lorenza Leão Oliveira Moreno
D.Sc. em Informática / PUC-Rio

---

Luciana Brugiolo Gonçalves
D.Sc. em Ciência da Computação / UFF

JUIZ DE FORA

11 DE DEZEMBRO, 2014

*Aos meus amigos e irmãos.*

*Aos pais, pelo apoio e sustento.*

# Resumo

O Problema da Dupla Digestão em cadeias de DNA é um dos mais clássicos problemas envolvendo a prática de mapeamento genético em cadeias de DNA. Devido a sua complexidade (NP-Difícil), tentativas de resolução exata não são, em geral, boas abordagens e a utilização de metaheurísticas é fortemente indicada.

Para a utilização destas técnicas, uma questão a se considerar é seu caráter multiresultado, que apresenta desafios diferentes aos vistos em problemas tradicionais nos quais metaheurísticas geralmente são aplicadas. Neste sentido, o trabalho aqui proposto tem como objetivo aplicar duas metaheurísticas comuns à resolução de problemas relacionados à bioinformática, Recozimento Simulado e Algoritmo Genético, e analisar comparativamente o comportamento das mesmas na resolução de um problema desta natureza, conseguindo boas diretivas para a utilização em problemas com esta característica.

**Palavras-chave:** Problema da Dupla Digestão, Recozimento Simulado, Algoritmo Genético, Metaheurística.

# Abstract

The Double Digest Problem in DNA chains is one of the most classical problems involving the genetic mapping practices. Due your complexity (NP-Hard), attempts at exact solution aren't a good way for proceed, in general, and the use of metaheuristics is a strongly suitable.

In the application of these techniques, an issue to consider is its multiresult feature, bringing different challenges compared for the traditional problems wherein the metaheuristics usually are applied. In this way, our work proposed here aims at applying two common metaheuristics for the resolution of problems in bioinformatics field, Simulated Annealing and Genetic Algorithm, and analyzes comparatively the behavior of these techniques in resolution of one problem on this characteristic, getting good directives for use in problems with this issue.



**Keywords:** Double Digest Problem, Simulated Annealing, Genetic Algorithm, Metaheuristic.

# Agradecimentos

A meus pais e irmãos, pelo encorajamento e apoio.

Ao professor Stênio Soares pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos, aos funcionários do curso, que durante esses anos contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

E, por fim, aos bons amigos que fiz e que mantive durante esta caminhada, sem os quais se perderia, em boa parte, o brilho da chegada.

*"É que da bem-aventurança e da alegria na vida há pouco a ser dito enquanto duram; assim como as obras belas e maravilhosas, enquanto perduram para que os olhos as contemplem, são registros de si mesmas; e somente quando correm perigo ou são destruídas é que se transformam em poesia."*

*J. R. R. Tolkien (O Silmarillion: Dos Sindar)*

# Sumário

# Lista de Figuras

# Lista de Tabelas

# Lista de Abreviações

| | |
|---|---|
| DNA | Deoxyribonucleic Acid |
| DDP | Double Digest Problem |
| PDP | Partial Digest Problem |
| HGP | Human Genome Project |
| bp | Pares de Bases Nitrogenadas |

# 1 Introdução

## 1.1 Biologia e Ciência da Computação

Desde que o modelo da dupla hélice foi apresentado à comunidade científica por WATSON AND CRICK (1953), a biologia molecular fez enormes progressos na compreensão da estrutura básica de informação da vida, o DNA. Processos como clonagem de DNA e sequenciação de genes se tornaram técnicas padrões e os genomas de diversos seres vivos, incluindo o genoma humano, foram mapeados e estão disponíveis. Tais progressos, entretanto, não seriam possíveis sem a integração da biologia com outras áreas da ciência, como física, química e ciência da computação.

Se tratando desta última, o rápido desenvolvimento da biologia molecular está extremamente vinculado ao desenvolvimento de técnicas de automação eficientes para análise de dados. De fato, tópicos como técnicas eficientes de busca em grandes bases de dados, alinhamentos de sequências e previsão de estruturas tridimensionais são de estudo da ciência da computação, mas de grande interesse para a biologia molecular (CIELIEBAK, 2003).

É da necessidade de integrar duas ciências tão distantes, em uma primeira análise, que se fundamenta a criação e desenvolvimento da biologia molecular computacional ou bioinformática.

Um dos primeiros trabalhos a, de fato, integrar de forma relevante a capacidade de processamento computacional com a biologia ocorreu em 1984, quando um grupo de biologistas utilizou uma simples técnica combinatória para comparar um novo gene causador de câncer a todos os genes conhecidos até aquele momento. Com o método, foi descoberto que o tipo de câncer em questão é causado por um gene do crescimento normal que sofre uma falha no momento de sua replicação (PEVZNER AND WATERMAN, 1995).

Entre diversos outros trabalhos e pesquisas, talvez a contribuição mais notória tenha ocorrido durante a intensa utilização da bioinformática no Projeto Genoma Humano (HGP), onde sua presença, muito mais do que apenas para catalogar e armazenar dados,

foi fundamental para a descoberta de novos conhecimentos.

Interessante ainda citar, tal é o grau de sucesso na integração entre estas duas ciências, que, em iniciativas posteriores, foi possível observar pesquisas percorrendo o caminho inverso ao comentado até aqui. Baseando-se em fenômenos estudados pela biologia molecular, procurou-se desenvolver algoritmos e heurísticas para a resolução de problemas não necessariamente ligados a bioinformática. Um exemplo é o trabalho de ADLEMAN (1994), que nos apresenta um algoritmo para a resolução do caminho Hamiltoniano onde o problema é modelado tendo como base a estrutura das moléculas de DNA, sendo utilizadas operações análogas às práticas laboratoriais para descoberta de informação.

## 1.2 Análise Genômica

### 1.2.1 Introdução

Mesmo em uma rápida análise, seria fácil determinar proteínas e ácidos nucleicos como os principais componentes orgânicos formadores da vida na terra. Proteínas são a base para praticamente todas as funções de um organismo vivo, estando presentes na constituição de tecidos, na defesa do organismo, no transporte de substâncias e agindo como enzimas catalisadoras em reações químicas.

Cada proteína é formada por uma longa cadeia de aminoácidos posicionados em uma ordem específica e fundamental para as funções a que ela é designada. Existem 20 diferentes aminoácidos, sendo a montagem das proteínas feita pelo próprio organismo.

Mas onde são armazenadas as sequências de aminoácidos utilizadas na construção de cada proteína? A resposta é nos ácidos nucleicos. A informação para a montagem de todas as proteínas necessárias aos organismos vivos está em suas moléculas de DNA, longas cadeias constituídas em sua maior porção por moléculas de açúcares e fosfatos, onde estão ligadas 4 diferentes tipos de bases nitrogenadas: adenina, citosina, guanina e timina. Alguns organismos mais simples possuem, em lugar de DNA, moléculas de RNA tendo, como principal diferença, a existência de uracila no lugar de timina.

É a sequência destas bases nitrogenadas que armazena todas as informações necessárias para a construção das proteínas de cada organismo vivo conhecido no planeta

e, em última instância, para a definição dos próprios seres vivos. O conjunto contendo todas estas informações codificadas pelos ácidos nucleicos é denominado genoma.

De forma mais precisa, genoma é a coleção de todas as longas moléculas de DNA de um organismo vivo. Importante notar que, destas longas moléculas denominadas cromossomos, apenas uma parte é relevante para estudo, os genes, que são trechos que de fato codificam alguma proteína. Acredita-se atualmente que algo em torno de apenas 10% do genoma humano é composto por genes codificantes, tendo os 90% restante função apenas especulada.

É em encontrar os genes, associá-los as proteínas correspondentes e descobrir a função destas no funcionamento dos organismos que se tem por objetivo a analise genômica. Uma simples motivação para o estudo é o fato de que, com este conhecimento, surge a possibilidade de se analisar geneticamente indivíduos e, desta forma, encontrar falhas na constituição de seu DNA que poderiam originar doenças, podendo assim antecipar seu tratamento.

Outro fator a motivar o estudo é a possibilidade de manipulação genética. Alterando em nível genômico características de determinado organismo seria possível adequá-lo a desígnios humanos, sejam estes com fim econômico, científico, ou, até mesmo, social. Esta, porém, é uma prática ainda vista com certa restrição por vários estratos da sociedade.

### 1.2.2 Mapeamento Genético e Mapeamento Físico

Um fator inicial a ser considerado quando se tem em mente a análise genômica são as diferenças de magnitude que envolvem o problema. Unidades básicas de informação, as bases nitrogenadas são obtidas através de um processo denominado sequenciamento genético. Este processo é capaz de mapear sequências com tamanho em torno de 700 a 1000 pares de bases nitrogenadas (abreviado como bp a partir deste momento). Por outro lado, por exemplo, um cromossomo humano possui algo em torno de $10^8$ bp. Isto gera uma diferença de escala na ordem de $10^5$ entre as dimensões do que se é capaz de sequenciar e do que se deseja sequenciar. Esta desproporção é uma das principais fontes de problemas para a bioinformática.

Com tal diferença de magnitude, é possível perceber que uma abordagem de busca de genes em todas as bases nitrogenadas das cadeias de DNA seria extremamente complexa e, provavelmente, inviável. Mesmo se fosse possível o sequenciamento em comprimentos de cadeias maiores, a busca de genes seria, por si só, extremamente complexa.

A alternativa utilizada para se contornar estas limitações se passa pelo desenvolvimento de dois tipos de mapa em paralelo: mapas genéticos e mapas físicos.

Mapas genéticos contêm o lócus, ou localização, de cada gene em seu respectivo cromossomo. O processo utiliza o argumento de que genes próximos tem menor possibilidade de, durante a produção de gametas, se separarem. Desta forma, se for possível através da observação de um grande número de amostras de indivíduos e seus descentes, medir a probabilidade de determinadas características se manifestarem em conjunto em um mesmo indivíduo, é possível determinar a distância relativa dos genes que resultam nestas características. Caso uma determinada característica apareça atrelada a outra, os genes possuem alta probabilidade de estarem relativamente próximos compartilhando um mesmo cromossomo e a distância atribuída será menor. Por outro lado, se existe pouca ou quase nenhuma dependência entre estas características, há uma grande probabilidade destes genes estarem distantes no cromossomo ou em cromossomos distintos e a distância atribuída será maior. Existem, claro, outras dificuldades e pontos a serem considerados durante a montagem de um mapa genético, o que pode tornar o processo muito mais complexo do que foi apresentado, mas esses fatores fogem ao escopo deste trabalho.

Esta técnica, porém, possui duas grandes desvantagens: não informa quais são as distâncias reais, em pares de bases, ao longo do cromossomo e, caso os genes estejam muito próximos, é impossível determinar a ordem de ocorrência pois a probabilidade de separação tenderá a zero.

Mapas físicos não possuem estas duas limitações, uma vez que, devido a forma como são montados, refletem a distância real em pares de bases nitrogenadas. Por outro lado, para se construir um mapa físico é necessário se trabalhar em uma escala muito menor se comparado a dos cromossomos utilizados em mapas genéticos, apesar desta escala ainda ser muito grande para a aplicação direta do sequenciamento. Através de marcadores, um mapa físico é capaz de informar com precisão a localização de trechos da

cadeia com ordem de até $10^4$ bp, trabalhando com segmentos na faixa de $10^5$ bp a $10^6$ bp no processo. A Figura 1.1 mostra mais claramente a diferença de escala na aplicação de cada técnica.

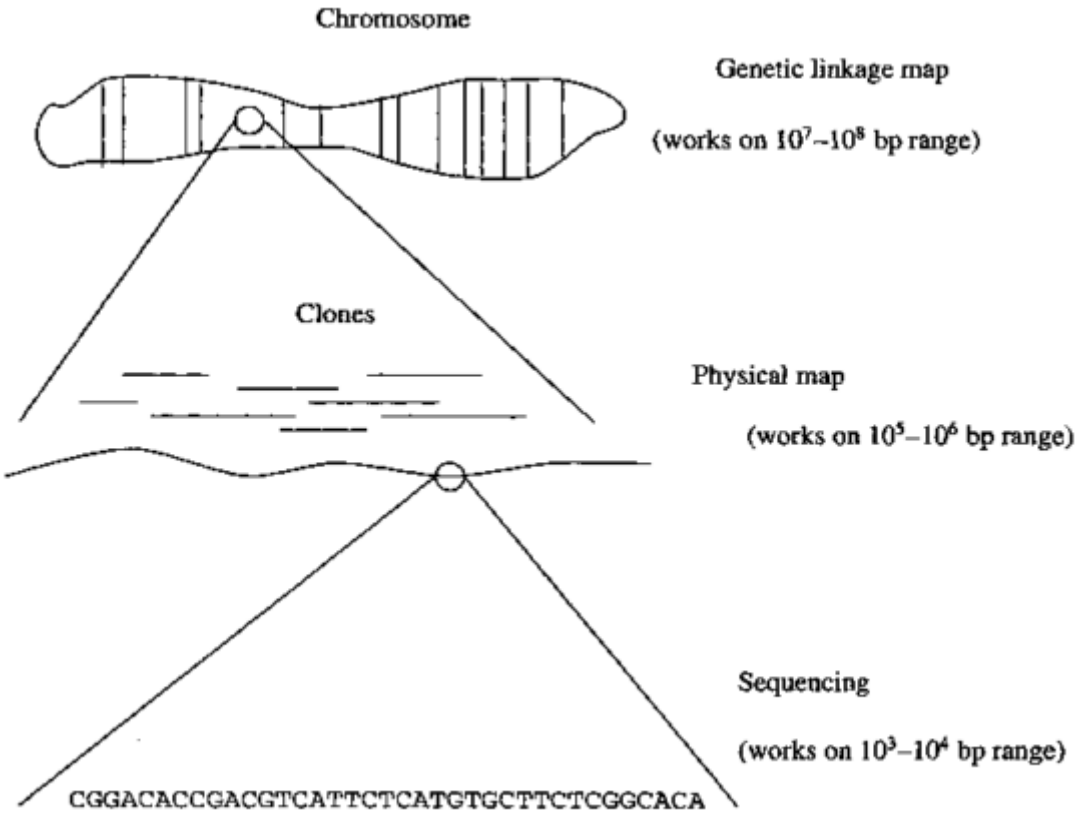


Figura 1.1: Diferenças de escala na análise genômica (SETUBAL AND MEIDANIS, 1997)

Ainda neste contexto, em sentido de esclarecimento, técnicas de sequenciamento podem aceitar cadeias com comprimento em torno a $10^3$ bp, mas o processo de sequenciamento em si das bases nitrogenadas possui capacidade bem menor, de 700 bp a 1000 bp, como mencionado anteriormente. As técnicas para obtenção de mapas físicos serão discutidas na próxima subseção.

### 1.2.3 Métodos de Mapeamento Físico

De maneira simplificada, mapas físicos consistem em uma lista com a localização de diversos marcadores ao longo da extensão do genoma. Estes marcadores situam segmentos de DNA já sequenciados, tendo como fim permitir a "navegação" pelo genoma de forma a

ser possível localizar estas sequências.

Inicialmente, a criação destes mapas se passa pela segmentação de trechos da cadeia de DNA em diversas partes menores. A segmentação é necessária tendo em vista que o processo laboratorial de se extrair informações em cadeias muito longas é inviável. Os "cortes" são feitos através da utilização de um tipo especial de enzima denominada nuclease ou enzima de restrição, enzima esta com propriedade de fragmentar cadeias apenas em regiões com sequência genética específica.

O desafio deste procedimento está na remontagem desta cadeia, de forma que ela apresente novamente seu aspecto original. Apenas o tamanho dos fragmentos resultantes do processo de digestão de uma enzima não fornece nenhuma informação sobre a ordem em que se organizavam e a dimensão ainda é muito elevada para se realizar o sequenciamento, sendo necessário, portanto, algum outro fator ou processo que possibilite a reconstrução. Neste sentido, dois tipos de abordagem são geralmente utilizadas: mapeamento por hibridação ou mapeamento por locais de restrição.

O processo de mapeamento por hibridação faz uso da elaboração de "uma biblioteca de *clones*" do segmento de DNA alvo. *Clones*, neste contexto, representam os fragmentos resultantes da aplicação de diversas enzimas sobre um conjunto de cópias da cadeia original. Cada um destes é avaliado por um conjunto de *probes* (pequenas amostras que reagem com cadeias de determinada sequência) e recebe um *fingerprinting* referente. É possível reparar que, como são criadas diversas cópias da cadeia de entrada, haverá sobreposição de trechos em elementos da biblioteca, apresentando, os *clones* em questão, *fingerprintings* semelhantes. Com base nas informações de sobreposição e no *fingerprinting* de cada *clone*, tenta-se reconstituir a ordem original da cadeia.

Em contrapartida ao *fingerprinting*, o mapeamento por restrição se utiliza apenas do comprimento dos fragmentos e das informações trazidas pelo procedimento de aplicação das enzimas. Em uma primeira etapa, cria-se um conjunto de clones do segmento de entrada, utilizando, em cada exemplar, uma determinada enzima de restrição. A seguir, é medido o comprimento de cada fragmento, sendo comum a utilização de gel eletroforese[1]

[1] Técnica de separação de moléculas baseada na migração de partículas em um determinado gel durante a aplicação de uma diferença de potencial. Na técnica, o tamanho das moléculas afeta seu deslocamento, permitindo a aferição do comprimento.

no processo. Por fim, com posse dos comprimentos dos fragmentos e tendo em mente o processo que os originou, procura-se reposicionar os cortes em sua ordem original.

### 1.2.4 Processos de Mapeamento por Restrição

Diversos procedimentos foram propostos para se realizar mapeamento por restrição, variando em número de enzimas utilizadas, métodos de aplicação ou possibilidade de marcação prévia de trechos da cadeia. Dentre estes, duas linhas se destacam e serão apresentadas aqui: mapeamento por digestões parciais e mapeamento por dupla digestão.

#### Mapeamento por Digestão Parcial

No mapeamento por digestão parcial uma enzima é aplicada sobre diversos clones de uma mesma cadeia. Estas aplicações devem ocorrer por diferentes tempos, resultando em conjuntos com estados de digestão distintos. Mede-se então o comprimento dos fragmentos e tenta-se reposicionar os cortes da cadeia em sua ordem original. O problema associado ao posicionamento dos fragmentos resultantes deste processo de digestão é conhecido como *Partial Digest Problem* (PDP). Um esquema do processo pode ser observado na Figura 1.2.

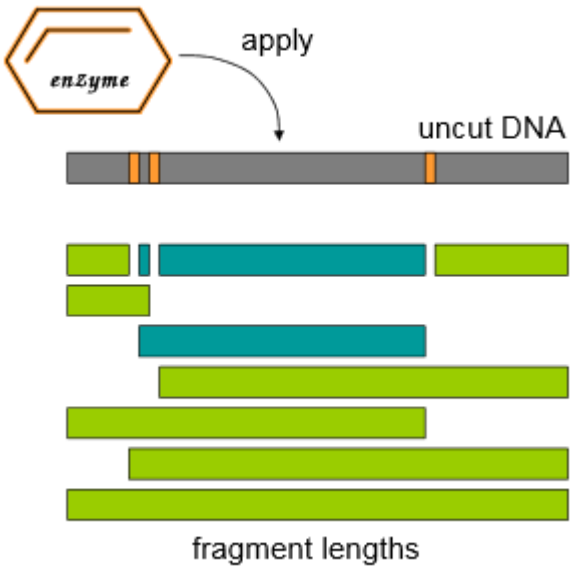


Figura 1.2: Esquema do processo de digestão do PDP (MNEIMNEH, 2008)

Similar ao PDP, o *Simplified Partial Digest Problem* (SPDP) aplica apenas uma enzima de restrição sobre o conjunto de cadeias de DNA. Sua diferenciação está em como estes cortes são realizados. O SPDP não se utiliza de múltiplos estados de digestão. Em seu lugar, permite que cada clone apresente apenas um corte, tendo em posse o conjunto de fragmentos com todos os únicos cortes possíveis. Um exemplar com digestão completa também é usado no processo de montagem.

Outra variação é denominada *Proped Partial Digest Problem* (PPDP). Possui processo de construção de suas instâncias também muito semelhante ao PDP, mas apresenta uma etapa adicional. Nesta etapa, uma *probe* é utilizada para se hibridizar os fragmentos de DNA oriundos do processo de digestão. O método se utiliza, então, apenas dos fragmentos que contém o ponto hibridizado pela *probe* para determinar a solução.

### Mapeamento por Dupla Digestão

Diferente do método de mapeamento por digestão parcial e suas variantes, no mapeamento por dupla digestão duas enzimas distintas são aplicadas sobre clones de uma mesma cadeia. No processo, cada enzima digere um exemplar da cadeia e, em um terceiro exemplar, as duas enzimas são aplicadas ao mesmo tempo. Com o conjunto dos valores associados ao tamanho dos fragmentos tenta-se reposicionar os cortes da cadeia em sua ordem original. O problema associado ao reposicionamento dos fragmentos neste processo é conhecido como *Double Digest Problem* (DDP). Um esquema do processo pode ser observado na Figura 1.3.

Uma última variante a ser comentada, semelhante ao DDP, é denominada *Enhanced Double Digest Problem* (EDDP). Nesta, após a aplicação de uma primeira enzima (A), a segunda enzima selecionada (B) é aplicada sobre os fragmentos resultantes da digestão de A. O mesmo processo é feito analogamente aos fragmentos de uma primeira digestão de B, tendo como dados resultantes os fragmentos da digestão de A, de B e das sucessivas digestões (AB e BA).

O foco deste trabalho está nos mapas de restrição, em especial no DDP que será apresentado com mais detalhes nos próximos capítulos. Sua escolha em detrimento a outros métodos de mapeamento por restrição é justificada por, apesar de ser compu-

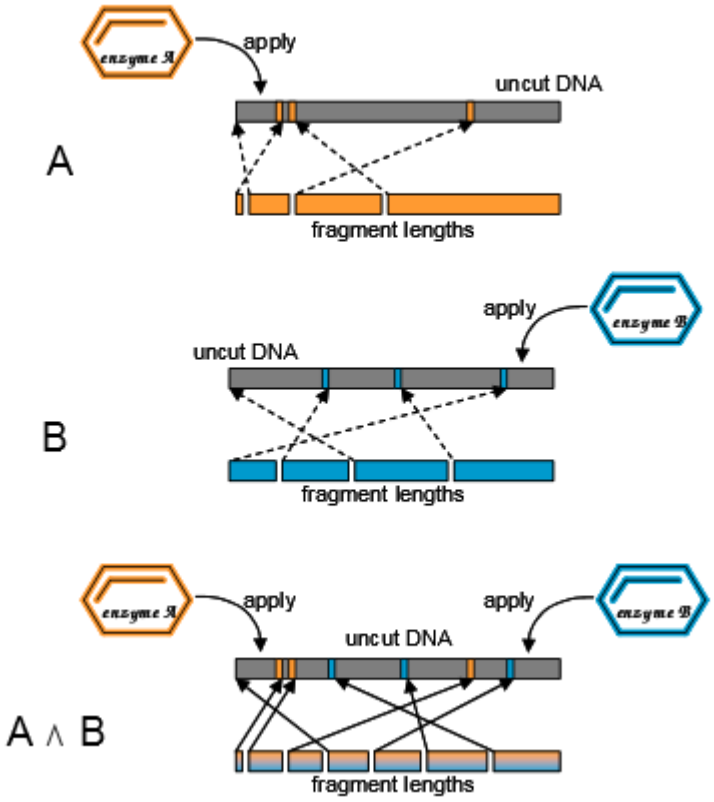


Figura 1.3: Esquema do processo de digestão do DDP (MNEIMNEH, 2008)

tacionalmente mais complexo se comparado, por exemplo, ao PDP, seu procedimento laboratorial é simples, o que faz com que seu uso em um contexto prático seja mais difundido (PEVZNER, 2000), mesmo que venha sendo, atualmente, suplantado por técnicas mais modernas baseadas em hibridação.

Outro fator interessante a motivar o estudo deste tema é seu caráter multiresultado, ausente nas formulações tradicionais de problemas de otimização. Este caráter traz um estudo diferenciado das metaheurísticas utilizadas, demandando uma adaptação específica à aplicação das mesmas e possibilitando, desta forma, além do fornecimento de contribuições a resolução do problema em si, apresentar mais informações para o comportamento das metaheurísticas utilizadas em problemas com esta característica.

## 1.3 Objetivos

Esta pesquisa procura explorar nuances relacionadas à aplicação de metaheurísticas ao Problema da Dupla Digestão, analisando técnicas e métodos que mais se adequam ao

contexto.

Mais precisamente, espera-se poder responder questões relativas à própria resolução do problema, tais como, qual entre as metaheurísticas utilizadas é mais adequada para resolução do Problema da Dupla Digestão? Qual é mais ágil para a obtenção de um conjunto de soluções aceitável? Qual manteve desempenho mais confiável durante sua execução?

Além destas, espera-se que seja possível indicar um bom conjunto de parâmetros para a execução de cada heurística, bem como aconselhar uma boa implementação de cada operador necessário, podendo servir a futuras abordagens deste problema.

Ainda espera-se que seja possível, devido ao contexto, desenvolver boas soluções para problemas que apresentem a particularidade de existência de múltiplas soluções desejadas. Fato que, apesar de pouco comum em problemas gerais de otimização, é de grande importância em contextos específicos e pouco explorado na literatura.

A estrutura deste texto se divide em 5 capítulos. O primeiro, já apresentado, introduz o problema, apresentando seu contexto biológico e objetivos estipulados. O segundo capítulo apresenta uma descrição detalhada do problema, com sua modelagem matemática, subproblemas associados, estado da arte, análise de complexidade, além de algumas outras nuances e particularidades. O terceiro capítulo mostra as abordagens heurísticas utilizadas para sua resolução, com a explicação dos métodos e detalhamento da implementação. O quarto capítulo contém resultados das abordagens utilizadas, bem como comparações entre estas e outras presentes na literatura. Por fim, o capítulo final apresenta as conclusões obtidas com o trabalho e sugestões de trabalhos futuros.

# 2 Apresentação do problema

## 2.1 Definição Biológica

No contexto da biologia molecular, uma prática cotidiana e fundamental é o mapeamento físico de cadeias de DNA. Uma dificuldade inerente a este processo é o fato destas cadeias serem, frequentemente, muito longas para o procedimento laboratorial necessário, sendo prática comum a fragmentação da cadeia para análise de partes menores. Através destes fragmentos, é simples a tarefa de se obter os comprimentos e as sequências de bases correspondentes. Para se realizar estes cortes é feita a aplicação de enzimas especiais (*nucleases*), capazes de fragmentar as cadeias apenas em locais com determinada sequência genética.

O problema deste procedimento advém da montagem dos fragmentos em sua ordem original, de forma a se reagrupar a cadeia de DNA. O tamanho dos fragmentos resultantes do processo de digestão de uma enzima não fornece nenhuma informação sobre a ordem em que se organizavam, sendo necessário, portanto, algum outro fator ou processo que possibilite sua reconstrução.

No intuito de solucionar este problema de escalas, um dos métodos utilizados é denominado Dupla Digestão em cadeias de DNA. Na Dupla Digestão, duas enzimas distintas são aplicadas sobre clones de uma mesma cadeia. Cada enzima digere um exemplar da cadeia em questão e, em um terceiro exemplar, as duas enzimas são aplicadas em conjunto. É feita a aferição do comprimento dos fragmentos resultantes destas digestões através da utilização de um método como gel eletroforese e, com apenas estes valores, procura-se remontar a cadeia com os cortes em sua disposição original. O problema abordado aqui está na reconstrução destas cadeias e, em última instância, montagem do mapa físico, denominado *Double Digest Problem* (DDP).

## 2.2 Definição Matemática

Segue o modelo matemático do DDP, como visto em BLAZEWICZ ET AL (2005). Sejam $\mathscr{A} = \{a_1, a_2, \cdots, a_m\}$, $\mathscr{B} = \{b_1, b_2, \cdots, b_n\}$ e $\mathscr{C} = \{c_1, c_2, \cdots, c_k\}$ multiconjuntos de valores numéricos que indicam o comprimento dos fragmentos obtidos em cada etapa do processo de digestão, com $\mathscr{A}$ e $\mathscr{B}$ referentes as digestões individuais e $\mathscr{C}$ relacionado a digestão de ambas as enzimas (note que $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = \sum_{l=1}^{k} c_l$). Sejam $\psi = \{1, 2, \cdots, m\}$, $\omega = \{1, 2, \cdots, n\}$ permutações dos elementos dos multiconjuntos $\mathscr{A}$ e $\mathscr{B}$ respectivamente. Pela ordenação destes multiconjuntos conforme as permutações $\psi$ e $\omega$ é possível obter um conjunto $\mathscr{S}$ (considere $\sum_{i=1}^{0} a_{\psi(i)} = \sum_{j=1}^{0} b_{\omega(j)} = 0$), tal que:

$$\mathscr{S} = \left\{ s_k : s_k = \sum_{1 \leqslant i \leqslant p} a_{\psi(i)} \vee s = \sum_{1 \leqslant j \leqslant q} b_{\omega(j)} \mid 0 \leqslant p \leqslant m, 0 \leqslant q \leqslant n \right\}$$

Conjunto onde, para todo par $\{s_i, s_j\} \in \mathscr{S}$, tem-se que $s_i \leqslant s_j$, se e somente se $i \leqslant j$, para $0 \leqslant i$ e $j \leqslant k$. Este conjunto $\mathscr{S}$, para ser válido, deve ser compatível com o multiconjunto $\mathscr{C}$, ou seja:

$$\mathscr{C}(\psi, \omega) = \{c_i(\psi, \omega) : c_i(\psi, \omega) = s_i - s_{i-1}, 1 \leqslant i \leqslant k\}$$

O *Double Digest Problem* consiste em, de posse dos multiconjuntos $\mathscr{A}$, $\mathscr{B}$ e $\mathscr{C}$, encontrar o par de permutações ($\psi$,$\omega$) de forma que este par seja capaz de induzir um conjunto $\mathscr{S}$ válido. A solução consiste nos multiconjuntos $\mathscr{A}$ e $\mathscr{B}$ ordenados de acordo com o par de permutações ($\psi$,$\omega$).

## 2.3 Múltiplas Soluções

Nota-se que, entre o problema biológico e o problema matemático, existe certa disparidade no conceito de soluções válidas. Em um contexto biológico, para qualquer cadeia de entrada, existe apenas um mapa físico que é solução aceita para o problema, sendo este o mapa que posiciona os cortes em suas posições originais as do momento de aplicação das enzimas. Por outro lado, para o problema matemático, existem, em muitos casos,

inúmeras permutações válidas para solucionar o problema. Claro que permutações distintas podem representar o mesmo mapa físico desde que haja repetição de comprimento de algum fragmento, neste caso não havendo alteração prática. Mas é situação frequente haver permutações distintas gerando mapas físicos distintos. Um exemplo pode ser visto na Figura 2.1.

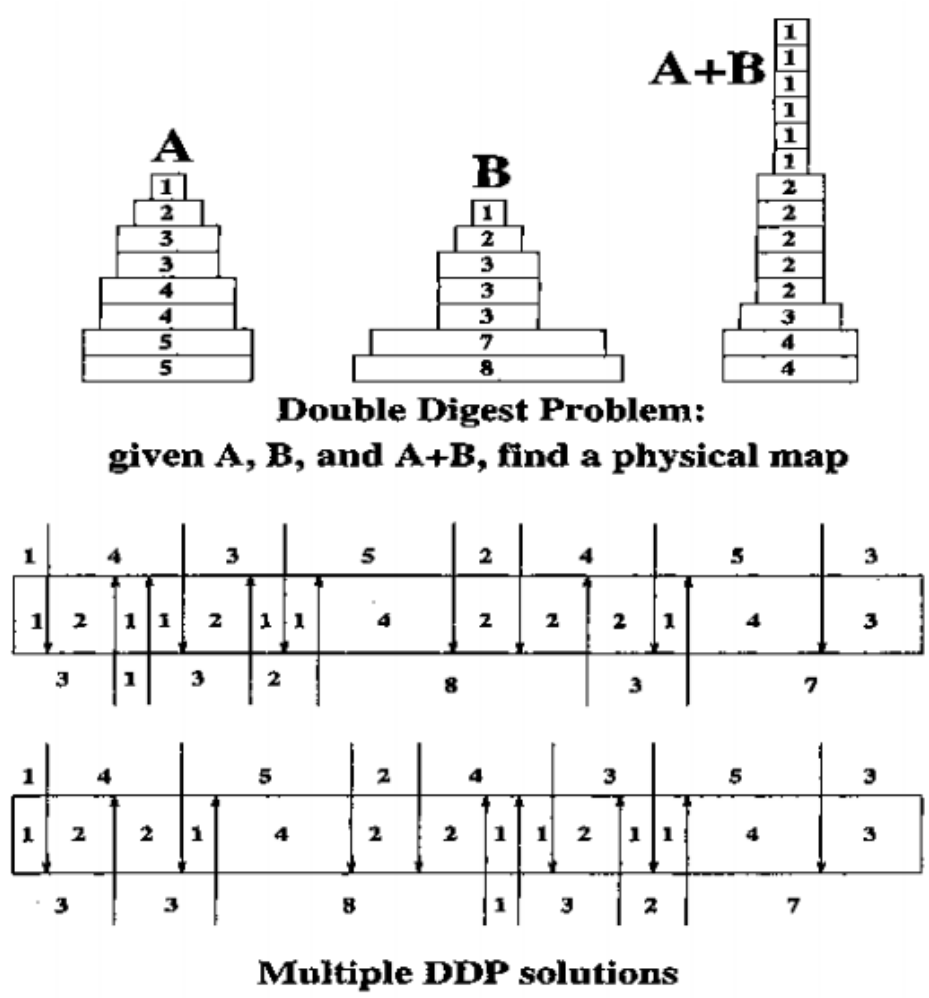


Figura 2.1: Exemplo de multiplas soluções do DDP (PEVZNER, 2000)

Neste exemplo, foram fornecidos os multiconjuntos $\mathscr{A} = \{1, 2, 3, 3, 4, 4, 5, 5\}$, $\mathscr{B} = \{1, 2, 3, 3, 3, 7, 8\}$ e $\mathscr{C} = \{1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 4, 4\}$. Um par de permutações válido que soluciona o problema matemático é $(\psi = \{1, 5, 3, 7, 2, 6, 8, 4\}, \omega = \{3, 1, 4, 2, 7, 5, 6\})$, induzindo a solução $\mathscr{A}_\psi = \{1, 4, 3, 5, 2, 4, 5, 3\}$, $\mathscr{B}_\omega = \{3, 1, 3, 2, 8, 3, 7\}$. Mas esta não é a única solução para a instância, outro par possível seria $(\psi' = \{1, 5, 7, 2, 6, 3, 8, 4\}$, $\omega' = \{3, 4, 7, 1, 5, 2, 6\})$ que gera $\mathscr{A}_{\psi'} = \{1, 4, 5, 2, 4, 3, 5, 3\}$, $\mathscr{B}_{\omega'} = \{3, 3, 8, 1, 3, 2, 7\}$, mapa físico distinto ao apresentado anteriormente.

Este fator torna o DDP ainda mais complexo. Não é suficiente encontrar apenas um par de permutações válido ($\psi$,$\omega$) que gere um mapa físico e não existe matematicamente nenhum indicador que seja capaz de avaliar se um par de permutações ou um mapa físico é mais adequado que outro. Isto obriga que, para solucionar adequadamente o problema biológico, se encontre todos os mapas físicos distintos para que estes possam ser passados a um especialista de domínio que faça a avaliação de cada mapa.

Devido ao esforço dispendido com a avaliação, este é um fator limitante da abordagem. De fato, GOLDSTEIN AND WATERMAN (1987) provaram que o número de soluções do problema cresce exponencialmente de acordo com o tamanho da cadeia considerando a disposição dos cortes feita de forma independente com probabilidades de ação de cada enzima em determinado ponto dadas por $p_1$ e $p_2$. Isto faz com que, para organismos de genoma extenso, este método, bem como outros baseados em mapeamento por restrição, seja preterido por outros baseados em mapeamento por hibridização. Mas, apesar desta contraindicação, para genomas mais simples este ainda é um método viável o que, somado a sua simplicidade, não descarta a validade de seu estudo.

## 2.4 Existência de Cortes Coincidentes

Locais com cortes coincidentes são regiões da cadeia onde ambas as enzimas, devido à sequência de bases, fragmentam a cadeia em um mesmo local, ou em locais tão próximos que as técnicas para medição dos fragmentos não são capazes de diferenciar os cortes. A tarefa de descobrir o número de cortes coincidentes na cadeia é simples, bastando apenas se utilizar da relação presente no número de fragmentos de cada processo de digestão.

$$|A| + |B| = |C| + |A \cap B|$$

No caso padrão de uma cadeia linear é considerada a existência de um corte coincidente no ponto zero. Em cadeias circulares não se considera a existência de um corte na origem.

Algumas abordagens, em efeito de simplificação, consideram o problema de uma cadeia linear sem cortes coincidentes (exceto na origem). Esta variação, denominada

*Disjoint Double Digest Problem*, é válida pois, em experimentos reais, usualmente são escolhidas enzimas que nunca cortam os mesmos locais na cadeia (CIELIEBAK, 2003). Ao se considerar a cadeia sem cortes coincidentes, o número de soluções diminui consideravelmente (para n cortes coincidentes, pode-se considerar o aumento de soluções, equivalentes ou não, em um fator de (n+1)!), além disto, é possível se usar o fato de cada fragmento possuir, ao menos, duas intersecções ou ser interno ao fragmento do multiconjunto análogo, fator muito explorado nestes casos.

## 2.5 Erros no Processo de Digestão

Processos de mapeamento em geral e, em particular, a dupla digestão possuem sempre, em um contexto real, a presença de erros associados aos dados. Estes erros podem tanto ter origem na própria ação da enzima, que pode falhar no processo de corte, como podem ocorrer devido a falhas no método de aferição dos dados resultantes do processo. De maneira geral, pode se classificar a existência de erro, devido a sua origem, em quatro tipos (CIELIEBAK, 2002):

- Cortes incompletos: Como mencionado, uma enzima pode falhar em sua ação em determinadas regiões. Isto origina uma quantia menor de fragmentos com comprimentos maiores, gerando, provavelmente, inconformidade com as informações presentes nos outros multiconjuntos.

- Aferição do comprimento: Medir com precisão absoluta o comprimento dos fragmentos é praticamente impossível, devendo-se sempre considerar a existência de erros neste processo. Tipicamente, erros de medição estão na faixa entre 2% a 7% em relação ao tamanho do fragmento.

- Perda de pequenos fragmentos: No processo de medição, se utilizando do gel eletroforese, é possível que pequenos fragmentos não sejam detectados por terem um percurso mais longo.

- Emparelhamento: Fragmentos com quase o mesmo comprimento podem, acidentalmente, ocupar o mesmo local na escala de medição se sobrepondo. Isto faz com que,

erroneamente, apenas um fragmento seja detectado.

## 2.6 Complexidade

Uma questão importante a ser analisada a cerca do DDP é o quão complexo exatamente é o problema. A intuição de que se trata de um problema extremamente difícil é grande e, de fato, o DDP é Fortemente NP-Completo, inclusive em sua versão simplificada, o *Disjoint Double Digest Problem*. Segue a prova, como feita em CIELIEBAK (2003), baseada na redução do *3-Partition Problem* ao *Disjoint Double Digest Problem*:

**Definição 3-Partition Problem:** Dados 3n inteiros positivos, $q_1, q_2, \cdots, q_{3n}$ e um inteiro h tal que $\sum_{i=1}^{3n} q_i = nh$ e $\frac{h}{4} < q_i < \frac{h}{2}$ para $i \in \{1, 2, 3, \cdots, 3n\}$, existem n triplas disjuntas de forma que suas somas resultem no valor de h?

Considerando os valores de $s = \sum_{i=1}^{3n} q_i$ e $t = (n+1)s$, constrói-se a instância do *Disjoint Double Digest Problem* como se segue:

| | |
|---|---|
| $a_i = q_i$ | para $1 \leqslant i \leqslant 3n$ |
| $\hat{a_j} = 2t$ | para $1 \leqslant j \leqslant n-1$ |
| $b_j = h + 2t$ | para $1 \leqslant j \leqslant n-2$ |
| $\hat{b_k} = h + t$ | para $1 \leqslant k \leqslant 2$ |
| $c_i = q_i$ | para $1 \leqslant i \leqslant 3n$ |
| $\hat{c_j} = t$ | para $1 \leqslant j \leqslant 2n-2$ |

Seja o multiconjunto $\mathscr{A}$ composto por $a_i$'s e $\hat{a_j}$'s, $\mathscr{B}$ composto por $b_j$'s e $\hat{b_k}$'s e $\mathscr{C}$ composto por $c_i$'s e $\hat{c_j}$'s, então Soma(A)=Soma(B)=Soma(C)=$s+(2n-2)$, seus cortes são disjuntos (cadeia linear: $(4n-1)+(n) = (5n-2)+1$) e, portanto, estes multiconjuntos são instâncias válidas para o *Disjoint Double Digest Problem*.

Se existir uma solução para a instância do *3-Partition Problem*, existiriam n triplas, cada uma somando h. Pode-se então formar uma solução, começando do ponto zero, em que, dos elementos do multiconjunto $\mathscr{A}$, cada $a_i$ é adjacente aos componentes da mesma tripla e intercalados por um elemento $\hat{a_j}$. Para os elementos de $\mathscr{B}$ é determinada

a ordem $\hat{b_1}, b_1, b_2, \cdots, b_{n-2}, \hat{b_2}$. Por fim, em $\mathscr{C}$ os elementos $c_i$ são dispostos com a mesma ordem dos elementos $a_i$ com cada tripla sendo separada por dois elementos $\hat{c_j}$. Com esta disposição, cada fim de fragmento presente tanto na ordenação do multiconjunto $\mathscr{A}$, quanto na ordenação do multiconjunto $\mathscr{B}$, possui um fim de fragmento correspondente na disposição em $\mathscr{C}$ e, desta forma, existe um mapa válido para a instância. Por outro lado, caso não existam n triplas que somem h, não existe nenhum par de permutações que gere uma disposição dos multiconjuntos capaz de apresentar um mapa válido, portanto a redução é correta. Um exemplo da redução pode ser visto na Figura 2.2.

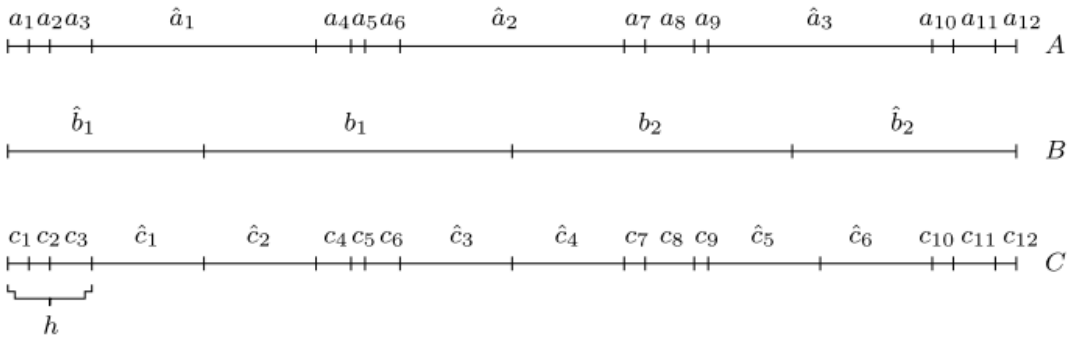


Figura 2.2: Exemplo de redução 3-Partition Problem para Disjoint Double Digest Problem (CIELIEBAK, 2003)

É evidente que, se provando a NP-Completude de sua versão simplificada, se prova a complexidade da versão completa do DDP. O caráter NP-Difícil do problema é um forte indicativo de que a busca por um algoritmo polinomial, ou mesmo pseudo-polinomial, não é uma boa abordagem.

## 2.7 Trabalhos Relacionados

Em 1970, foram descobertas as primeiras enzimas de restrição para o DNA e, deste então, biologistas as utilizam para a construção de mapas de restrição. Devido à importância desta etapa no processo de transcrição do DNA, diversos algoritmos foram propostos para automatizar o processo de mapeamento.

Aparentemente, o primeiro algoritmo para construção de um mapa de restrição foi apresentado por STEFIK (1978), que desenvolveu um método de busca exaustiva

aplicando alguns conceitos de inteligência artificial.

Também, utilizando-se de um método de busca exaustiva, PEARSON (1982) propôs um algoritmo específico para resolução do DDP, mais ágil por testar um menor número de permutações, realizando combinações entre os fragmentos de $\mathscr{A}$ e $\mathscr{B}$ e, após, verificando a validade da resposta no multiconjunto $\mathscr{C}$.

Devido à complexidade do problema, porém, abordagens simplistas como estas não se mostraram eficazes. Como comentado anteriormente, GOLDSTEIN AND WATERMAN (1987) provou que o DDP é NP-Completo e que o número de soluções em fragmentos produzidos por cortes aleatórios, cresce de forma exponencial de acordo com o comprimento da cadeia. Desta forma, muitos trabalhos se focaram em desenvolver outros métodos de representação do problema e aplicações de heurísticas e metaheurísticas.

No contexto de alternativas para representação do problema, WATERMAN AND GRIGGS (1986) propõe um método de representação denominado grafo de intervalo e demonstra as relações de equivalência entre grafos de intervalo e o DDP.

Uma heurística para resolução do problema pode ser vista em GOLDSTEIN AND WATERMAN (1987), onde é proposta uma implementação de Recozimento Simulado baseada em um método para resolução do problema do caixeiro viajante.

Relacionado a instâncias com erros, é apresentado em ALLISON AND YEE (1988) uma forma de representação, baseada na teoria da separação de Pratt, que possibilita uma análise mais adequada da qualidade das soluções, considerando estes erros advindos do processo de medição dos fragmentos.

Visto a enorme quantidade de soluções válidas para uma instância do problema e a semelhança entre muitas delas, SCHMITT AND WATERMAN (1991) propõe um método de classificação para as soluções de uma instância do DDP denominado Overlap Size Equivalence Class e operações para obtenção do conjunto de soluções dessas classes através de *cassette transformations*.

Como continuação, PEVZNER (1995) propõe que as *cassette transformations* de SCHMITT AND WATERMAN (1991) podem ser representadas como um grafo Euleriano, sendo o conjunto de soluções de cada *cassette* equivalente aos circuitos Eulerianos existentes neste grafo. Porém, este método só é válido para instâncias do problema com

cortes disjuntos.

Neste contexto ainda, o trabalho de MARTIN (1994) generaliza o método de PEVZNER (1995), passando a admitir também mapas de restrição com cortes coincidentes.

Uma variante do DDP é proposta em KAO ET AL (2003) denominada *Enhanced Double Digest Problem*. Neste mesmo artigo é proposto um algoritmo em tempo polinomial para resolução do problema quando não há existência de fragmentos de mesmo comprimento nos conjuntos de digestões sucessivas. Para casos, contudo, onde ocorrem essas repetições, a complexidade aumenta para O(k!n), sendo k o número de duplicatas em um dos conjuntos.

Mais recentemente WU AND ZHANG (2008) utilizou o MIP para resolução do DDP. Aplicando técnicas de programação linear inteira, como a aplicação de planos de corte, e executando um algoritmo de branch-and-bound, conseguiu resultados muito superiores aos atingidos anteriormente, chegando a solucionar instâncias da ordem de $|A|$=242 e $|B|$=250, mas se limitando a descoberta de um único mapa.

Outra abordagem já explorada na literatura é a aplicação de um Algoritmo Molecular para o DDP, que pode ser observada em GANJTABESH ET AL (2009). Neste artigo é utilizado um modelo similar ao modelo Adleman-Lipton para resolução do problema.

SUR-KOLAY ET AL (2009) propõe um Algoritmo Genético associado a um procedimento de descoberta de soluções envolvendo *cassettes*. Mas, apesar de se sair bem em pequenas instâncias e diminuir significativamente o espaço de busca, este procedimento exato é custoso (O($(\mathscr{A} + \mathscr{B})^3$)) para cada *cassette* distinto encontrado).

Por fim, em um contexto mais atual, GANJTABESH ET AL (2012) apresenta um Algoritmo Genético para o DDP que atinge bons resultados em um tempo aceitável, superando inclusive a abordagem anterior de SUR-KOLAY ET AL (2009), mas buscando novamente apenas uma solução.

O DDP é um problema antigo, porém pouco explorado pela comunidade acadêmica. Isso se deve, em maior parte, por sua complexidade, tanto em um ponto de vista computacional, quanto em um ponto de vista prático, pela necessidade de se analisar diversos mapas viáveis manualmente.

Ainda assim, é possível identificar tentativas para sua resolução na literatura. Em um primeiro momento, se utilizando de uma abordagem mais simplista, trabalhos como de STEFIK (1978) e PEARSON (1982) procuraram solução exata para o problema se utilizando de técnicas de busca exaustiva, apresentando um resultado apenas satisfatório e se tratando apenas de instâncias de pequena dimensão.

Abordagens mais promissoras seguiram a linha de classificação de soluções, diminuindo, desta forma, o espaço de busca em que algoritmos deveriam ser aplicados, tendo destaque a utilização de *cassettes*. Os trabalhos de SCHMITT AND WATERMAN (1991), PEVZNER (1995) e MARTIN (1994) apresentam diferentes aspectos desta ideia. Duas grandes contraindicações do método passam pelo fato desta abordagem ser, muitas vezes, de difícil implementação e flexibilização e por não possuir nenhum algoritmo de criação de mapas a partir das classes que se apresente eficiente, acrescentando este *overhead* a abordagem.

Mudanças na forma de representação do problema, como os grafos de intervalo (WATERMAN AND GRIGGS, 1986) e a utilização da teoria da separação de Pratt (ALLISON AND YEE, 1988) no contexto de erros do processo biológico, foram contribuições à diversificação do estado da arte, mas não forneceram, efetivamente, conteúdo à questão do desempenho, principal dificuldade presente.

Outro ponto apresentado foi a mudança do problema em si, como visto em KAO ET AL (2003), simplificando extremamente a complexidade computacional de resolução e proporcionando, desta forma, o desenvolvimento de algoritmos mais eficientes se comparados aos do DDP. A grande dificuldade inerente a uma abordagem destas está na difusão de uso da técnica, necessitando da alteração de metodologia laboratorial utilizada.

Em outra frente, a modelagem do DDP como um problema de programação linear inteira é uma possibilidade de menor impacto que pode produzir bons resultados como visto em WU AND ZHANG (2008). Apesar de sua escrita como um problema de programação inteira mista não ser trivial, a utilização destes resolvedores costumam apresentar ótima performance. Suas principais dificuldades estão em tratar a existência de múltiplas soluções e em representar erros do processo biológico, fatores que fogem da aplicação de um método de programação inteira tradicional e interferem diretamente no

desempenho do algoritmo.

A aplicação de heurísticas é outra metodologia que oferece resultados relativamente interessantes tanto em relação a tempo, quanto a resultados, sendo o foco das abordagens mais recentes. Algoritmos Genéticos tem sido escolhidos como opções mais frequentes, sejam se utilizando de uma abordagem híbrida (SUR-KOLAY ET AL, 2009) ou tradicional (GANJTABESH ET AL, 2012).

Abordagens híbridas apresentam um caminho interessante para a implementação de boas soluções em geral, mas a utilização de *cassettes* como método exato, como comentado anteriormente e utilizado em SUR-KOLAY ET AL (2009), é um fator que apresenta complicações a questões relacionadas a desempenho. Abordagens tradicionais têm como vantagem não apresentar um gasto computacional excessivo ao se executar uma etapa exata, mas, sem esse auxílio, assumem toda a carga de processamento do problema, tendo, portanto que apresentar suas próprias soluções para uma execução eficiente em todo espaço de busca. GANJTABESH ET AL (2012) assume essa abordagem e consegue bons resultados, superiores inclusive aos do trabalho de SUR-KOLAY ET AL (2009), mas não suficiente a ponto de poder se considerar uma solução definitiva.

Deve-se relativizar esta superioridade, porém. O trabalho desenvolvido por SUR-KOLAY ET AL (2009) é focado na resolução de instâncias relativamente pequenas, da ordem de 10 fragmentos nos multiconjuntos $\mathscr{A}$ e $\mathscr{B}$. Além disto, seu foco está na descoberta de múltiplos mapas físicos. Estes dois fatores estão ausentes na comparação realizada por GANJTABESH ET AL (2012), tendo ele utilizado instâncias de até 150 fragmentos e simplificado a abordagem para a busca de uma única solução.

Outras heurísticas apresentadas na literatura em GOLDSTEIN AND WATERMAN (1987) e GANJTABESH ET AL (2009) podem ser caminhos promissores mas, como seus resultados são omitidos, não se é possível realizar nenhuma análise comparativa.

# 3 Abordagem Proposta

## 3.1 Heurística e Metaheurística

No decorrer deste trabalho serão citados diversos conceitos básicos, mas recorrentes e de grande importância em problemas de otimização. Por este motivo, serão inicialmente apresentados a fim de evitar dificuldades posteriores.

Inicialmente, problema de otimização é todo problema que envolve a procura pela melhor solução possível para um determinado fim em um conjunto de soluções viáveis. Para dimensões pequenas ou para problemas simples, a resolução pode ser trivial, não necessitando de grande análise. Porém, em dados problemas e instâncias, a complexidade é tal que a utilização de métodos determinísticos tradicionais tornaria inviável sua resolução, seja em questão de recursos computacionais, seja em questão de tempo de execução.

Desta forma, muitos algoritmos para problemas de otimização tomam a forma de heurísticas. Ao se considerar a utilização destes métodos, o objetivo passa a ser o desenvolvimento de uma abordagem que seja capaz de se obter uma boa solução do conjunto viável de soluções, mas sem compromisso de que esta se trate da melhor solução possível (solução ótima) ou garantia de algum grau de aproximação desta solução.

O desenvolvimento de heurísticas pode ser feito tanto através de um método construtivo direto, quanto através da exploração do espaço de busca e avaliação contínua da qualidade das soluções encontradas no processo. Em ambos os casos, para se determinar a qualidade das soluções, é comum se utilizar de uma função matemática denominada função objetivo (ou função de avaliação), cujo papel é medir quantitativamente a adequabilidade de uma dada solução em relação ao contexto determinado na elaboração do problema, fornecendo um *feedback* para a execução do algoritmo.

Um tipo específico, metaheurísticas são, de maneira simplificada, heurísticas para o desenvolvimento de heurísticas. Se tratam de estruturas algorítmicas gerais para resolução aproximada de problemas de otimização, sendo aplicadas aos mais diversos contextos e restrições e apresentando, em muitos casos, desempenho extremamente satisfatório.

Este trabalho apresenta um estudo comparativo entre duas diferentes metaheurísticas aplicadas ao *Double Digest Problem*, Algoritmo Genético e Recozimento Simulado. A escolha destas metaheurísticas em particular, ocorre devido a sua grande flexibilidade e bom desempenho em diversas áreas de aplicação, incluindo problemas relacionados à Bioinformática. Além disto, Algoritmos Genéticos e de Recozimento Simulado já foram abordados na literatura para resolução do *Double Digest Problem*, apesar de nunca comparados entre si.

## 3.2 Algoritmo Genético

### 3.2.1 Apresentação da Técnica

Um dos métodos aplicados neste trabalho, o Algoritmo Genético é uma metaheurística baseada nas ideias contidas na teoria da evolução natural de Charles Darwin, segundo a qual, indivíduos mais aptos de uma população têm maior tendência a sobreviver e gerar descendentes.

Desta forma, implementações desta metaheurística trabalham com aspectos oriundos das ideias de Darwin, sendo comum a utilização dos conceitos de população, indivíduo, seleção natural, reprodução e mutação.

Como unidade básica manipulada pelo algoritmo, indivíduo é a representação de uma solução qualquer do espaço de busca. É representado por seu genótipo (a codificação da solução a que corresponde o indivíduo, também conhecido como cromossomo) e fenótipo (representação "física" de sua codificação).

População é o conjunto de indivíduos que serão analisados em uma iteração do algoritmo. Seu tamanho varia conforme o problema e abordagem proposta, não havendo nenhuma regra para sua definição.

Seleção natural é todo o processo que determina quais indivíduos vão repassar seu conteúdo genotípico para futuras gerações. Assim, como no processo biológico, sua contraparte computacional costuma ter forte fator aleatório, mas, em geral, favorecendo aos indivíduos mais aptos.

Operações mais comuns sobre o conteúdo genético dos indivíduos, cruzamento e

mutação se diferem tanto em forma de aplicação, quanto em finalidade. Operações de cruzamento geralmente envolvem dois indivíduos de uma população, previamente selecionados (pais), gerando um ou mais descendentes resultantes da combinação dos genótipos destes pais. Costumam ser utilizados em alta escala na execução do algoritmo com o intuito de combinar boas características de seus antecedentes, gerando melhores descendentes. Operações de mutação, por outro lado, são individuais e, em geral, menos frequentes no fluxo de execução do algoritmo. Tem como principal intuito, incutir variabilidade a população de soluções, evitando uma convergência prematura da heurística.

Como última fase do fluxo de execução, a avaliação da população tem como objetivo definir qual o conjunto de indivíduos será perpetuado para uma nova etapa. Em um contexto fiel ao aspecto biológico, a renovação é completa, eliminando os antigos indivíduos e os substituindo por seus descendentes. Apesar de mais fiel a sua inspiração, essa metodologia é frequentemente substituída por um processo denominado elitismo, onde uma parcela da população mais apta é sempre mantida para o fluxo seguinte, pertencendo ou não a nova geração de indivíduos.

O critério de parada (fim da execução do algoritmo) no Algoritmo Genético pode se dar por diversos fatores, como número máximo de iterações, qualidade da solução atingida, tempo de execução, entre outros, sendo uma das questões de projeto a serem analisadas.

Outros conceitos comuns em implementações de Algoritmos Genéticos são de geração (referente a uma iteração do fluxo principal do algoritmo) e tamanho da população (quantidade de indivíduos mantida de geração para geração). Um pseudocódigo apresentando as etapas aqui listadas está presente no Algoritmo 1.

### 3.2.2 Função objetivo e representação da solução

A representação de uma solução é baseada na definição matemática apresentada na Seção 2.2. Cada solução representa um par de permutações possível ($\psi$,$\omega$) do multiconjunto de fragmentos, sendo composto por duas cadeias, uma referente as permutação dos elementos do multiconjunto $\mathscr{A}$ e outra referente as permutações do multiconjunto $\mathscr{B}$, com o multiconjunto $\mathscr{C}$ utilizado para verificação da qualidade da solução.

**Algoritmo 1:** Algoritmo Genético

**Entrada**: Parâmetros: taxa de cruzamento ($\alpha$), taxa de mutação ($\beta$)
Inicializar população $P_0$;
**repita**
  Seleciona nrCruzamento pares de elementos da população $P$ para reprodução $(p_i, p_j)$;
  **para** *k=1 até nrCruzamento* **faça**
    **se** *probabilidade* $\alpha$ *atendida* **então**
      S $\leftarrow$ S $\cup$ cruzamento$(p_i, p_j)$;
    **fim se**
  **fim para**
  **para todo** $s_i$ *em* $S$ **faça**
    **se** *probabilidade* $\beta$ *atendida* **então**
      $s_i$=mutação$(s_i)$;
    **fim se**
  **fim para todo**
  P $\leftarrow$ avaliação(P,S);
**até** *critério de parada ser atendido*;
**Saída**: Melhor elemento de P encontrado

Tendo isso em mente, o indivíduo é definido como uma possível ordem dos elementos de cada multiconjunto, sendo representado por dois vetores distintos, um correspondente a uma dada ordem dos elementos de $\mathscr{A}$ e outro correspondente a uma outra ordem dos elementos de $\mathscr{B}$.

Funções de avaliação para o DDP geralmente são elaboradas tendo em vista a minimização de algum erro presente no processo biológico, independente das instâncias utilizadas conterem ou não erros. Fato é que esta escolha, em muitas vezes, afeta de forma profunda o desempenho da abordagem, fazendo com que a convergência para um conjunto de soluções adequado ocorra em maior ou menor grau.

Para este trabalho foi desenvolvido o cálculo tendo em vista os erros gerados por cortes incompletos no processo de digestão da cadeia, sendo a abordagem que demonstrou melhor desempenho nas simulações realizadas. Este erro ocorre quando, no processo de aplicação, uma enzima falha em sua ação em determinadas regiões, ocasionando aglutinação de dois ou mais fragmentos em algum dos multiconjuntos fornecidos. A função de avaliação utilizada pode ser observada a seguir:

$$\min f(\psi', \omega') = |\mathscr{C}'(\psi', \omega') \setminus \mathscr{C}|$$

Sendo $\mathscr{C}'(\psi', \omega')$ o multiconjunto induzido, gerado pela ordenação dos multiconjuntos $\mathscr{A}$ e $\mathscr{B}$ através das permutações ($\psi'$, $\omega'$) e tradução destes em um multiconjunto de fragmentos de dupla digestão adequado para a configuração. Como o tratamento de instâncias contendo erros não está no escopo do trabalho, a validade de um mapa físico ocorrerá no momento em que a função objetivo atingir valor zero, verificando-se a igualdade $\mathscr{C}'(\psi', \omega') = \mathscr{C}$.

### 3.2.3 Seleção

A seleção aplicada ao problema é baseada na técnica de *ranking* linear, sendo esta mais robusta se comparado a escolhas tradicionais como a roleta, permitindo uma melhor análise dos indivíduos da população e distribuição mais justa da codificação genética, o que auxilia a geração de uma maior diversidade.

Esta escolha, dado a importância de se manter uma distribuição homogênea em problemas multiresultados, é recomendável para o bom funcionamento da abordagem. Baseada em seu *ranking*, a aptidão de uma solução foi definida de acordo com a fórmula:

$$Fitness(P) = 2 * \frac{T - P + 1}{T - 1}$$

Onde P representa a posição no ranking crescente por valor da função objetivo e T representa o número de indivíduos da população.

São selecionados 100 pares de progenitores nesta fase para a operação de cruzamento.

### 3.2.4 Cruzamento

Métodos comuns de cruzamento, se aplicados ao modelo de cromossomo utilizado, podem gerar descendentes inviáveis. É necessário, portanto, que se mantenha certa cautela na elaboração do operador, observando-se o fato de que cada cromossomo representa uma permutação do multiconjunto de fragmentos, tendo então que possuir referência a todos os fragmentos. O cruzamento utilizado neste trabalho se inspira no modelo apresentado em GANJTABESH ET AL (2012).

Inicialmente, de posse de dois genitores selecionados anteriormente, escolhe-se com probabilidade de 50% uma das cadeias do cromossomo para se aplicar a operação. Em seguida, para cada lócus, aplica-se uma probabilidade de 50% para origem do valor referente a cada um dos genitores. Checa-se o valor em questão não esta presente no cromossomo do descendente, caso esteja, não é atribuído nenhum valor no momento da aplicação, caso não, o valor é atribuído normalmente e segue-se o processo.

Ao termino da cadeia, os pontos onde não foi possível atribuir valor, são complementados com os valores faltantes, seguindo a ordem presente em um dos pais. Note que, com a operação, são gerados dois descendentes, formados por operações complementares (uma dada atribuição em um dos descendentes se segue de outra correspondente a escolha oposta no segundo descendente), tanto em relação ao recebimento de valores, quanto ao complemento final do cromossomo. Note ainda que, para que seja atribuído um valor na primeira fase, é necessário que a condição (não existência do valor no descendente) seja verificada em ambos os descendentes necessariamente, não permitindo que em um dado instante do processo, se atribua a um dos descendentes e não se atribua a outro. Um exemplo do processo, após a seleção de uma das cadeias, pode ser visto na Figura 3.1.

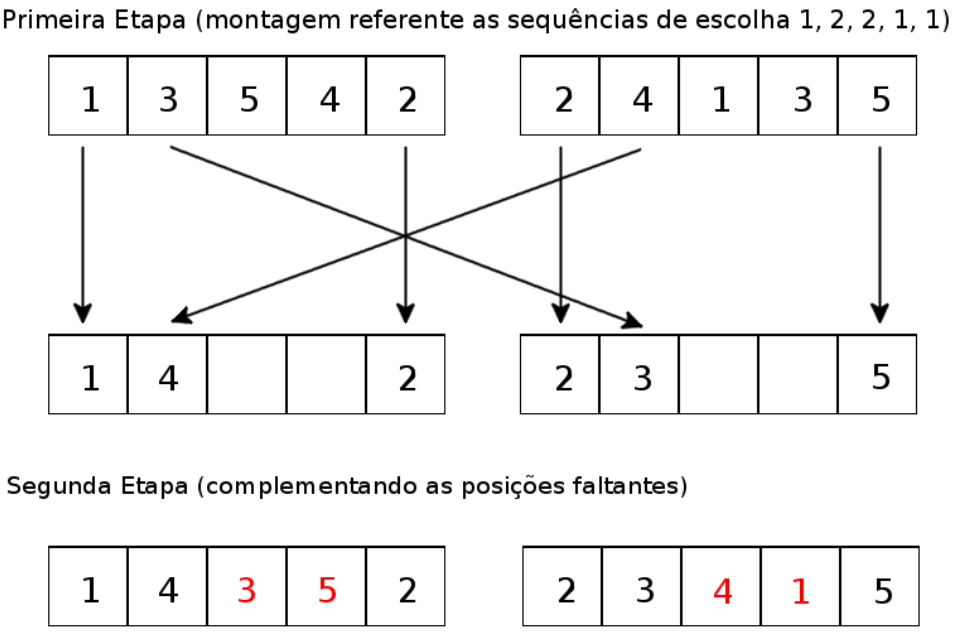


Figura 3.1: Exemplo demonstrando cruzamento implementado

Neste caso foram escolhidos dois cromossomos e uma cadeia a ser alterada, tendo esta seleção configurações (1,3,5,4,2) e (2,4,1,3,5). Foi testada, para cada posição, a origem dos fragmentos, sendo selecionado para o primeiro cromossomo de saída, inicialmente o primeiro progenitor como origem do fragmento, a seguir o segundo, o segundo novamente, o primeiro e, por fim, o primeiro novamente. Para o segundo cromossomo de saída, a sequência foi complementar: segundo, primeiro, primeiro, segundo e segundo.

Como não foi possível montar a cadeia completamente (a terceira e quarta escolhas inviabilizariam pelo menos um dos cromossomos), se passou para a segunda etapa onde, para o primeiro cromossomo de saída, foram identificadas os fragmentos 3 e 5 como faltantes. Como no cromossomo progenitor de entrada associado (o primeiro progenitor), na sequência de fragmentos o 3 antecede o 5, eles são inseridos nesta ordem nas posições vazias. Da mesma forma, é analisado o segundo cromossomo de saída em relação ao segundo cromossomo de entrada.

A probabilidade de ocorrência desta operação é de 85% para cada par de progenitores selecionado.

### 3.2.5 Mutação

Assim como na operação de cruzamento, ao se implementar o operador de mutação deve-se ter em conta a viabilidade dos descendentes gerados. A simples escolha de novos valores fatalmente resultará em elementos duplicados e soluções inviáveis. O método utilizado neste trabalho realiza trocas de posição entre elementos do próprio cromossomo para evitar problemas desta natureza.

A mutação se inicia com a definição de qual cadeia será afetada com probabilidade de 50% para cada. Após esta etapa, são escolhidas aleatoriamente duas posições distintas da cadeia (note que as posições selecionadas podem ser iguais, não ocorrendo, neste caso, nenhuma alteração no cromossomo). Por fim, com as posições definidas, a troca de valores é efetuada entre as duas posições. Um exemplo pode ser observado na Figura 3.2.

Neste exemplo foram selecionadas as posições 2 e 4 do cromossomo e os valores (3 e 4) trocados entre estas posições.

A probabilidade de ocorrência desta operação nos descendentes gerados é de 70%.

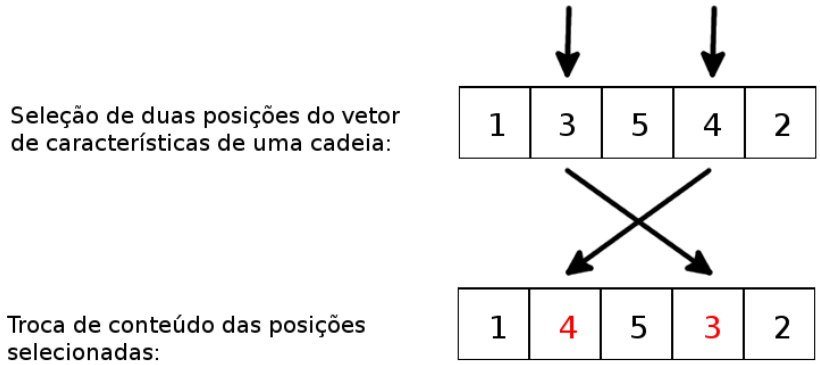


Figura 3.2: Exemplo demonstrando mutação implementada

A alta taxa de mutação utilizada na abordagem, se comparada aos modelos tradicionais de aplicação, é justificada por ocorrer em conjunto a uma operação de baixo impacto, permitindo a passagem de uma solução por múltiplas configurações intermediárias. Sendo esta, devido as características do problema, uma opção mais interessante se comparada a aplicação de uma única operação de resultado final semelhante e de menor frequência.

Além disto, pela necessidade de uma alta taxa de aproveitamento da população, é necessário que esta se mantenha diversa durante toda a execução do algoritmo, impedindo sua convergência para um ponto qualquer no espaço de soluções, outro ponto em que a alta taxa de mutação atua diretamente.

### 3.2.6 Avaliação

A avaliação é feita tendo em vista a existência de múltiplas soluções para o problema. Desta forma, apenas uma parcela da população que permanece não é condicionada à "idade" da solução, em um processo de elitismo, sendo os demais elementos pertencentes exclusivamente à nova geração de indivíduos.

Esta parcela, formada pelos melhores indivíduos entre os que compõem a população atual e os advindos do processo de cruzamento, tem seu tamanho ajustado de acordo com as necessidades de cada execução, possuindo inicialmente tamanho de 25% da população, mas podendo variar entre 0% a 50%, a depender da demanda. O critério de seleção de melhores indivíduos se baseia na classificação, pela função objetivo.

O processo de ajuste é feito de acordo com a taxa de novas soluções descobertas: caso o número de soluções novas seja menor que 10% do número de indivíduos analisados em um intervalo pré-definido, é reduzida a porção de melhores soluções mantidas (em 1%); caso a condição se mantenha e esta redução resulte em melhora (5% a mais de soluções descobertas se comparado à checagem anterior), esta troca é novamente realizada até que a nova condição não aprimore a execução, que a taxa de soluções descoberta seja satisfeita ou que o tamanho do conjunto seja de 0%. Um processo semelhante é considerado para caso o aproveitamento das soluções seja superior a 15%, trocando-se redução por expansão e limitando o processo a 50% do tamanho da população. O ponto de checagem ocorre a cada 250 gerações.

Importante ressaltar o fato de que, na formação deste conjunto, não é permitido que dois descendentes de mesma ancestralidade sejam mantidos ao mesmo tempo, sendo obrigatório o descarte de um destes elementos. Não há restrição alguma quanto a membros da geração anterior.

Para o preenchimento dos demais elementos, o critério de escolha entre os descendentes é semelhante: para cada par gerado na operação de cruzamento o indivíduo de melhor avaliação é incluso na população, sendo seu par eliminado, não há comparação global. Isto é feito, em ambos os casos, com o intuito de melhor explorar o espaço de busca, não mantendo dois descendentes com os mesmos progenitores, tendo estes um maior número de características em comum. O tamanho da população que se perpetua para a próxima geração é sempre de 100 indivíduos e o critério de parada ocorre ao serem atingidas 10.000 gerações.

## 3.3 Recozimento Simulado

### 3.3.1 Apresentação da Técnica

Outra abordagem utilizada, o algoritmo Recozimento Simulado é uma metaheurística que se baseia no processo de recozimento (Annealing) de um metal, onde este é fundido a altas temperaturas e resfriado lentamente para que se obtenha a configuração de suas estruturas moleculares com menor nível de energia possível.

A reprodução deste evento tem em mente o estado de desordem molecular da estrutura em temperaturas elevadas, contrastando com um nível de organização maior ao se atingir temperaturas mais baixas, com menor movimento das mesmas. Desta forma, o algoritmo em posse de uma solução, ao se encontrar em uma temperatura alta, tem maior tendência a, em uma movimentação por sua vizinhança, aceitar este deslocamento mesmo que não ocorra redução do nível de energia. Por outro lado, em temperaturas baixas, a probabilidade de aceitação de uma solução da vizinhança que não apresente redução no nível de energia é pequena, promovendo, quase que em absoluto, apenas movimentos de melhora neste estado de execução.

A probabilidade de aceitação ou rejeição de uma solução de piora é dada através de uma função conhecida como fator de Boltzmann, sendo obtida pela formula $P(\Delta E, T) = e^{-\frac{\Delta E}{T}}$, onde $\Delta E$ é a diferença entre o nível de energia da nova solução em relação ao nível de energia da solução corrente e $T$ a temperatura do sistema no momento da análise. Repare que uma solução de melhora sempre é aceita pelo algoritmo.

Movimentos por uma vizinhança são alterações, geralmente pequenas, efetuadas em uma dada solução com o objetivo de explorar o entorno desta no espaço de busca, à procura de uma configuração próxima mais adequada. Em implementações de Recozimento Simulado, movimentos são os principais operadores aplicados a uma dada solução corrente, tendo lugar, em muitos casos, a utilização de múltiplos movimentos distintos para um melhor desempenho.

Nesta metaheurística, a configuração de uma solução é equivalente ao genótipo e o nível de energia à aptidão no Algoritmo Genético, mas, diferente dele, não é utilizado um conjunto de soluções (população) para a execução do algoritmo. Alguns outros fatores de ajustes importantes para o Recozimento Simulado são sua função de resfriamento da temperatura e a forma de determinação da temperatura inicial. Sendo a primeira responsável por determinar o valor da temperatura em um novo fluxo de execução do algoritmo e a segunda por escolher em qual valor de temperatura o algoritmo deve começar sua execução. O término do algoritmo ocorre quando a temperatura da execução atinge um valor arbitrariamente próximo de zero (outras abordagens são possíveis para definição do critério de parada). Um pseudocódigo com a implementação padrão apresentando as

etapas aqui listadas pode ser visto no Algoritmo 2.

**Algoritmo 2:** Recozimento Simulado

**Entrada**: Temperatura Inicial ($T_{max}$)
Construção da solução inicial $s_0$;
$T = T_{max}$;
**repita**
  **repita**
    Gerar um vizinho aleatório s';
    $\Delta E = f(s') - f(s)$;
    **se** $\Delta E \leq 0$ **então**
      s=s';
    **senão**
      Aceita s' com probabilidade $e^{-\frac{\Delta E}{T}}$
    **fim se**
  **até** *Até a condição de equilíbrio*;
  T=resfriamento(T);
**até** *critério de parada ser atendido* ($T < T_{min}$);
**Saída**: Melhor solução encontrada

### 3.3.2 Função objetivo e representação da solução

A função objetivo, representação de solução e do indivíduo utilizadas são equivalentes ao que foi apresentado anteriormente para o Algoritmo Genético.

O nível de energia de uma solução equivale ao valor de sua função objetivo.

### 3.3.3 Definição da temperatura inicial

Um dos parâmetros a se avaliar no desenvolvimento de uma abordagem de Recozimento Simulado, existem diversas técnicas na literatura de como se definir um bom valor da temperatura inicial.

Para esta abordagem, o método de ajuste da temperatura inicial escolhido foi o cálculo da média dos valores das funções de aptidão de um conjunto de vizinhos gerados por um determinado movimento.

Desta forma, inicialmente é definida uma temperatura arbitrariamente alta (60) e, com este valor, são gerados certo número de vizinhos (500 para esta abordagem) através da operação de troca de elementos entre as extremidades da cadeia (descrita a seguir), desenvolvendo a solução inicial como em uma execução tradicional do algoritmo. A seguir,

são avaliados os valores dos custos das soluções encontradas, calculando-se a média destes valores e atribuindo o resultado à temperatura inicial.

### 3.3.4 Atualização e Iterações por temperatura

A função de resfriamento utilizada foi apresentada por LUNDY AND MEES (1986). Nesta abordagem, substitui-se a execução de múltiplas iterações por valor de temperatura, por um resfriamento mais lento, ocorrendo um ajuste extremamente gradual nas probabilidades de aceitação de soluções de piora.

Desta forma, a implementação realiza apenas um movimento por estado de temperatura, atualizando-a logo a seguir, sendo a combinação que apresentou melhores resultados. A função de resfriamento pode ser vista abaixo, foi definido $\beta$ como 0.0006 para as execuções:

$$T_k = \frac{T_{k-1}}{1 - T_{k-1} * \beta}$$

O procedimento termina quando a temperatura é menor que o valor 0.001.

### 3.3.5 Estratégias de Movimentação

Movimentos pela vizinhança são os principais mecanismos utilizados por algoritmos de Recozimento Simulado para o desenvolvimento da solução corrente, sendo a qualidade destes, portanto, vital para bom desempenho do algoritmo.

Nesta implementação foram desenvolvidas duas movimentações complementares para exploração do espaço de busca. Estas não ocorrem em uma mesma temperatura, sendo a seleção para cada estado feita por sorteio. A primeira operação (Troca de trechos em uma mesma extremidade da cadeia) tem maior probabilidade ocorrência (80%), enquanto a segunda operação (Troca de elementos entre as extremidades da cadeia) ocorre nas demais execuções (20%). Segue uma descrição de cada um dos movimentos implementados.

### Troca de trechos em uma mesma extremidade da cadeia

Utilizando-se da existência de simetria entre as soluções geradas, a operação trabalha apenas com uma extremidade por execução, realizando trocas entre seus elementos.

Para isso, a cada operação, é selecionada, inicialmente, tanto a cadeia quanto a extremidade que será trabalhada. Tendo posse destas informações, são selecionados aleatoriamente dois trechos de uma mesma extremidade, com comprimento máximo estipulado (valor 5 para as execuções realizadas).

Finalizando o processo, ocorre a troca dos segmentos gerando uma nova solução. Um exemplo pode ser visto na Figura 3.3.

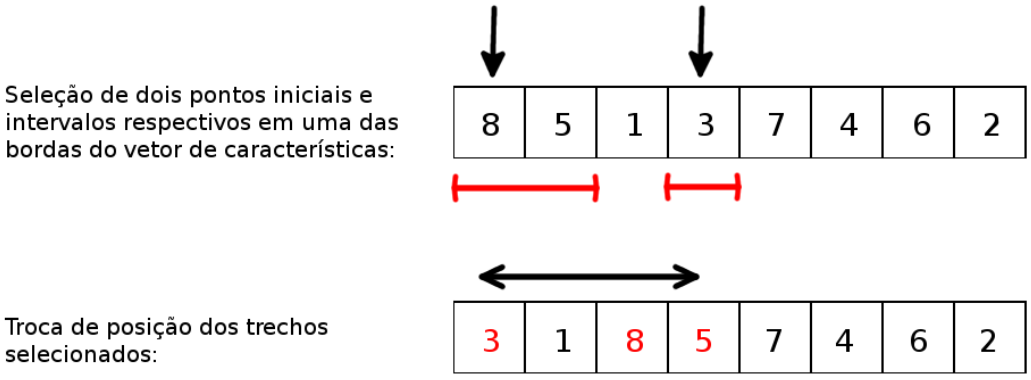


Figura 3.3: Exemplo demonstrando operação de troca em uma mesma extremidade

Neste exemplo, foi selecionada a primeira extremidade do vetor de características para ser operada. Desta extremidade, foram definidas as posições iniciais 1 e 4, com intervalos de comprimento 2 e 1 respectivamente, gerando os trechos (8,5) e (3), ambos pertencentes a mesma borda. Estes intervalos têm, então, seus posicionamentos trocados originando uma solução com esta nova configuração.

### Troca de elementos entre as extremidades da cadeia

A operação apresentada anteriormente não é capaz de, a partir de uma solução inicial, atingir todas as configurações do espaço de busca, dado que não altera a distribuição dos fragmentos em cada uma das extremidades.

Portanto, como complemento, foi desenvolvida uma operação para troca dos frag-

mentos entre as bordas da solução.

O processo consiste inicialmente na seleção da cadeia onde a operação será aplicada. Em seguida, são selecionados dois fragmentos quaisquer para a troca, um em cada extremidade da cadeia. Estes fragmentos são trocados, gerando assim uma nova solução. A Figura 3.4 mostra um esquema do processo.

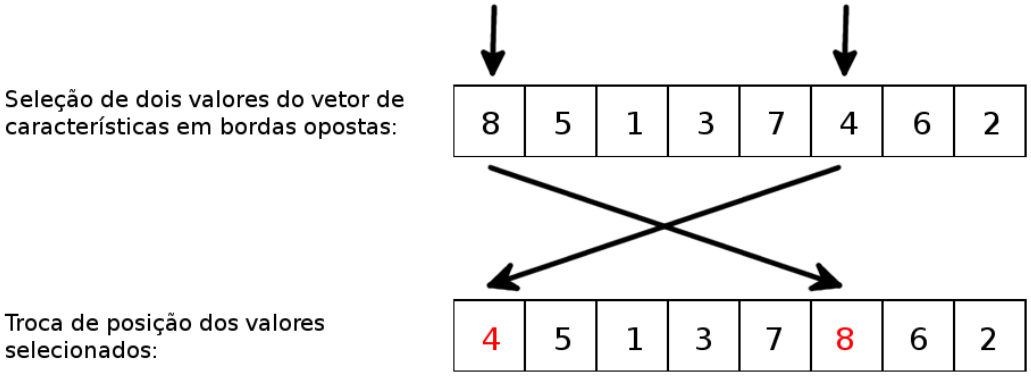


Figura 3.4: Exemplo demonstrando operação de troca entre extremidades

Neste exemplo, foram selecionadas as posições 1 e 6 (valores 8 e 4, respectivamente) de extremidades opostas no vetor de configuração. Em seguida, os valores são trocados gerando uma nova solução válida.

# 4 Resultados

Para geração dos resultados, todos os algoritmos mencionados neste capítulo foram implementados em C++ e executados em máquina AMD Athlon(tm) II X2 250 Processor com 3 GB de RAM.

Os testes foram gerados em duas modalidades. Em um primeiro momento, fiel ao modelo do problema que foi apresentado até aqui, foi objetivado a busca por múltiplas soluções dado uma instância do problema.

Em seguida, foi testado o desempenho destas mesmas abordagens na busca por uma única solução, simplificação recorrente na literatura em abordagens para este problema.

## 4.1 Instâncias

As instâncias foram geradas através do método proposto na literatura por GOLDSTEIN AND WATERMAN (1987), com algumas modificações quando há representação de instâncias sem cortes coincidentes. No modelo original, é determinado um comprimento de cadeia e, para cada ponto do intervalo inteiro, aplica-se probabilidade $p$ de se ocorrer um corte, independente tanto em relação a cadeia, quanto aos cortes anteriores. Para o caso onde se representa instâncias sem cortes coincidentes, a adaptação consiste em, na ocorrência de aceite de cortes em um mesmo lócus para ambas as cadeias, se definir aleatoriamente, com probabilidade de 50% para cada cadeia, em qual delas este corte será representado.

Cada instância foi nomeada de acordo com os parâmetros que a originou: o nome inicia-se com o tamanho da cadeia, tendo em seguida a probabilidade (em porcentagem) de ocorrência de corte e, por fim, o sufixo ncc, indicando, nas instâncias em que está presente, a não existência de cortes coincidentes entre as aplicações das enzimas. A Tabela 4.1 mostra as instâncias geradas com o respectivo número de fragmentos de cada multiconjunto.

Tabela 4.1: Instâncias de teste

| Nome da Instância | $\lvert\mathscr{A}\rvert$ | $\lvert\mathscr{B}\rvert$ | $\lvert\mathscr{C}\rvert$ | Tamanho da Cadeia |
|---|---|---|---|---|
| DDP50-30 | 12 | 22 | 27 | 50 |
| DDP50-30ncc | 17 | 14 | 30 | 50 |
| DDP50-50 | 23 | 23 | 32 | 50 |
| DDP50-50ncc | 20 | 19 | 38 | 50 |
| DDP100-30 | 27 | 28 | 44 | 100 |
| DDP100-30ncc | 30 | 24 | 53 | 100 |
| DDP100-50 | 52 | 56 | 77 | 100 |
| DDP100-50ncc | 35 | 41 | 75 | 100 |
| DDP200-30 | 61 | 63 | 103 | 200 |
| DDP200-30ncc | 52 | 67 | 118 | 200 |
| DDP200-50 | 92 | 96 | 141 | 200 |
| DDP200-50ncc | 75 | 79 | 153 | 200 |
| DDP300-30 | 92 | 96 | 158 | 300 |
| DDP300-30ncc | 91 | 72 | 162 | 300 |
| DDP300-50 | 160 | 152 | 234 | 300 |
| DDP300-50ncc | 125 | 104 | 228 | 300 |
| DDP400-30 | 119 | 135 | 210 | 400 |
| DDP400-30ncc | 99 | 127 | 225 | 400 |
| DDP400-50 | 203 | 220 | 308 | 400 |
| DDP400-50ncc | 141 | 158 | 298 | 400 |
| DDP500-30 | 152 | 164 | 264 | 500 |
| DDP500-30ncc | 135 | 137 | 271 | 500 |
| DDP500-50 | 258 | 241 | 371 | 500 |
| DDP500-50ncc | 201 | 174 | 374 | 500 |

Garantir a inexistência de cortes coincidentes é um fator interessante para avaliação destas técnicas, tanto por, como mencionado anteriormente, ser uma condição de possível garantia do ponto de vista biológico através da escolha das enzimas a serem utilizadas, quanto por tornar a busca por mapas mais complexa para um mesmo tamanho de instância e probabilidade de corte, devido ao menor número de mapas existentes e maior complexidade de geração destes mapas (a instância possui uma quantidade semelhante de fragmentos em $\mathscr{C}$ de sua contraparte mas com número reduzido de fragmentos de $\mathscr{A}$ e $\mathscr{B}$).

## 4.2 Resultados da busca por múltiplos mapas

Para comparação e análise de desempenho, foram utilizadas métricas comuns ao contexto deste problema: tempo de execução do algoritmo (tempo de parede), número de soluções distintas encontradas em uma execução do algoritmo e valor da melhor solução encontrada (para as execuções incapazes de atingir alguma solução).

Os resultados foram adquiridos através da execução dos algoritmos por 10 vezes para cada instância, calculando a média e desvio padrão de cada característica, bem como o coeficiente de variação. O desvio padrão e o coeficiente de variação são utilizados como métrica para análise, uma vez que permitem medidas de robustez de cada técnica.

A Tabela 4.2 mostra o número médio de mapas físicos obtidos nas execuções da estratégia baseada em Algoritmo Genético e da baseada em Recozimento Simulado com o coeficiente de variação destas amostras, bem como o valor de objetivo do melhor indivíduo da população (nos casos onde não se foi possível encontrar soluções), tempo médio das execuções e desvio padrão desta medida. A marcação de tempo exclui as operações de entrada e saída de dados.

Ambas as abordagens apresentam desempenho satisfatório para o problema. Como poderia ser esperado, devido à característica de cada método, em um número maior de instâncias e, em especial, nas de mais simples resolução, a abordagem Genética gera resultados melhores em termo de número de mapas por seu viés populacional permitir a exploração de um espaço de busca mais amplo o que, com a necessidade de um pequeno esforço no refinamento dos indivíduos iniciais, pode favorecer o seu desempenho. Por

Tabela 4.2: Execuções Multiresultado

| | Algoritmo Genético | | | | | Recozimento Simulado | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Instâncias | Mapas Físicos (Média) | Mapas Físicos (CV (%)) | Melhor (Média) | Tempo (Média) | Tempo (DP) | Mapas Físicos (Média) | Mapas Físicos (CV (%)) | Melhor (Média) | Tempo (Média) | Tempo (DP) |
| DDP50-30 | **719.932,40** | 14,55 | 0,00 | 7,04 | 0,05 | 306.612,20 | 3,26 | 0,00 | **4,76** | 0,19 |
| DDP50-30ncc | 96.354,20 | 47,83 | 0,00 | 6,60 | 0,15 | **101.060,60** | 3,03 | 0,00 | **4,91** | 0,11 |
| DDP50-50 | **650.724,80** | 24,35 | 0,00 | 8,18 | 0,07 | 260.954,60 | 1,65 | 0,00 | **5,47** | 0,09 |
| DDP50-50ncc | 18.762,80 | 77,26 | 0,00 | 7,78 | 0,25 | **146.186,40** | 15,83 | 0,00 | **6,04** | 0,10 |
| DDP100-30 | **494.116,80** | 11,36 | 0,00 | 10,38 | 0,08 | 163.691,20 | 4,47 | 0,00 | **7,50** | 0,09 |
| DDP100-30ncc | **222.586,20** | 59,58 | 0,00 | 11,00 | 0,14 | 106.031,40 | 24,86 | 0,00 | **8,47** | 0,13 |
| DDP100-50 | 270.022,60 | 11,55 | 0,00 | 17,48 | 0,10 | **421.184,00** | 0,73 | 0,00 | **12,65** | 0,11 |
| DDP100-50ncc | **58.891,60** | 78,14 | 0,20 | 13,92 | 0,31 | 57.149,80 | 34,29 | 1,00 | **11,42** | 0,07 |
| DDP200-30 | **247.301,60** | 14,65 | 0,00 | 21,29 | 0,19 | 126.973,80 | 2,70 | 0,00 | **17,18** | 0,10 |
| DDP200-30ncc | **260.579,00** | 19,17 | 0,00 | 22,53 | 0,21 | 98.778,60 | 35,71 | 1,40 | **18,77** | 0,06 |
| DDP200-50 | **412.097,40** | 10,39 | 0,00 | 29,50 | 0,14 | 245.971,20 | 0,56 | 0,00 | **22,92** | 0,11 |
| DDP200-50ncc | 42.427,60 | 135,51 | 0,80 | 27,41 | 0,38 | **67.008,00** | 18,13 | 0,00 | **23,64** | 0,11 |
| DDP300-30 | **164.608,40** | 32,24 | 0,00 | 31,78 | 0,18 | 104.997,80 | 4,52 | 0,00 | **26,30** | 0,04 |
| DDP300-30ncc | **110.888,40** | 52,13 | 0,00 | 30,64 | 0,21 | 81.656,40 | 14,90 | 0,00 | **26,23** | 0,07 |
| DDP300-50 | **337.177,60** | 3,22 | 0,00 | 48,33 | 0,17 | 267.229,60 | 0,79 | 0,00 | **38,58** | 0,06 |
| DDP300-50ncc | 702,40 | 156,27 | 1,80 | 40,93 | 0,18 | **22.581,40** | 68,88 | 15,70 | **35,81** | 0,18 |

outro lado, em instâncias mais complexas, o algoritmo Recozimento Simulado supera este viés ao ser capaz de atingir com mais facilidade soluções de um maior grau de dificuldade, mas possuindo, ainda assim, uma boa exploração do espaço pela abordagem aqui proposta, conseguindo se igualar e, em alguns momentos, superar o Algoritmo Genético.

A abordagem baseada em Recozimento Simulado também apresenta maior robustez em relação ao número de mapas encontrados, o que pode ser devido a sua maior facilidade em, dado um ponto inicial, se buscar diversas soluções na vizinhança e proximidades do ponto, "corrigindo" com maior facilidade um ponto de partida de qualidade duvidosa, mas com região vizinha promissora. Neste aspecto, o Algoritmo Genético pode demonstrar maior dificuldade, necessitando certo esforço até atingir um conjunto de pontos de boa qualidade.

O tempo de execução é outro fator em que o algoritmo Recozimento Simulado possui certa vantagem, com a diferença entre as abordagens crescendo gradualmente conforme o número de fragmentos da instância aumenta.

Em um último ponto, por mais que haja certa dificuldade na abordagem Genética para se encontrar soluções para certas instâncias, suas execuções sempre se mostram próximas a um desses pontos. Fato oposto ocorre no Recozimento Simulado, onde, apesar das poucas ocorrências em que não há solução, estas podem se apresentar distantes de um ponto promissor, podendo se tratar de uma dificuldade comum em técnicas de aprimoramento de única solução: a necessidade de grande esforço para se "escapar" de regiões amplas de baixa qualidade.

## 4.3 Resultados da busca por um único mapa

A simplificação do problema com a busca por um único mapa que satisfaça a distribuição de fragmentos não é suficiente para a resolução do aspecto biológico do DDP, mas, devido à complexidade advinda do processo e a perda de interesse em sua resolução, resultado de sua substituição por métodos laboratoriais mais modernos, precisos e eficientes, esta abordagem se tornou recorrente na literatura.

Desta forma, a seção corrente apresenta o desempenho dos métodos desenvolvidos para o problema multiresultado nesta variante simplificada, comparados com o método

proposto na literatura por GANJTABESH ET AL (2012) que se utiliza deste artifício.

Como a descrição de seu trabalho peca na falta de detalhes em relação ao método de avaliação da população, serão adaptadas as duas metodologias mais comuns nesta fase em implementações de Algoritmo Genético. O algoritmo que será denominado aqui como GANJTABESH_Total, aplica a renovação completa da população com ranqueamento global da qualidade dos descendentes da iteração corrente e manutenção apenas dos indivíduos mais aptos. O segundo método implementado, denominado por GANJTABESH_Filhos, realiza renovação completa da população com concorrência apenas entre os descendentes de mesma origem, mantendo o mais apto.

As abordagens propostas neste trabalho não possuirão quaisquer modificações, exceto a inclusão de uma nova condição de parada relativa à descoberta de um mapa físico, de forma a tornar justa a comparação de tempo de execução entre as técnicas.

Abaixo segue o desempenho de cada algoritmo desenvolvido. Os resultados foram adquiridos seguindo as mesmas diretrizes anteriores, através da execução do algoritmo por 10 vezes para cada instância. Levou-se em conta na análise a proporção de execuções em que o algoritmo foi capaz de encontrar uma solução, a proximidade relativa da solução atingida, em média, e o tempo médio para se encontrar uma solução (relativo apenas as execuções onde foi possível se encontrar solução).

A Tabela 4.3 apresenta o desempenho dos dois algoritmos implementados tendo como base o trabalho de GANJTABESH ET AL (2012) e a Tabela 4.4 apresenta a comparação de desempenho entre as técnicas aqui propostas e GANJTABESH_Filhos (técnica ligeiramente superior na comparação entre as duas variações).

Fazendo-se ressalvas relativas à existência de aproximações na implementação da técnica proposta por GANJTABESH ET AL (2012), é nítida a superioridade das abordagens propostas por este trabalho, mesmo além de seu escopo inicial.

A capacidade de encontrar soluções da abordagem Genética aqui proposta excede ao que foi apresentado pelos algoritmos GANJTABESH_Total e GANJTABESH_Filhos, apresentando inclusive tempo de execução inferior na maior parte dos casos, sendo capaz de encontrar soluções para todas as instâncias e quase sempre se apresentando mais próximo de uma solução, nesse ponto superando inclusive o Recozimento Simulado.

Tabela 4.3: Comparação das implementações de GANJTABESH ET AL (2012)

| | GANJTABESH_Total | | | GANJTABESH_Filhos | | |
|---|---|---|---|---|---|---|
| Instâncias | Taxa de Acerto (%) | Imediação da Solução (Média) | Tempo em busca (Média) | Taxa de Acerto (%) | Imediação da Solução (Média) | Tempo em busca (Média) |
| DDP100-30 | **100,00** | **100,00** | 0,24 | **100,00** | **100,00** | **0,13** |
| DDP100-30ncc | **100,00** | **100,00** | **0,40** | 90,00 | 99,62 | 0,59 |
| DDP100-50 | **100,00** | **100,00** | **0,00** | **100,00** | **100,00** | **0,00** |
| DDP100-50ncc | **100,00** | **100,00** | **1,18** | **100,00** | **100,00** | 1,35 |
| DDP200-30 | **100,00** | **100,00** | 0,34 | **100,00** | **100,00** | **0,32** |
| DDP200-30ncc | **100,00** | **100,00** | **3,27** | **100,00** | **100,00** | 5,36 |
| DDP200-50 | **100,00** | **100,00** | **0,02** | **100,00** | **100,00** | **0,02** |
| DDP200-50ncc | **60,00** | **99,48** | **5,68** | 30,00 | 98,89 | 9,85 |
| DDP300-30 | **100,00** | **100,00** | **2,57** | **100,00** | **100,00** | 3,65 |
| DDP300-30ncc | 70,00 | 99,63 | **6,75** | **100,00** | **100,00** | 12,09 |
| DDP300-50 | **100,00** | **100,00** | 0,07 | **100,00** | **100,00** | **0,06** |
| DDP300-50ncc | **10,00** | **98,90** | **29,05** | **10,00** | **98,90** | 53,59 |
| DDP400-30 | 60,00 | 99,81 | **7,01** | **100,00** | **100,00** | 11,80 |
| DDP400-30ncc | 30,00 | 99,38 | **18,47** | **40,00** | **99,42** | 20,82 |
| DDP400-50 | **100,00** | **100,00** | **0,32** | **100,00** | **100,00** | 0,34 |
| DDP400-50ncc | 0,00 | 99,13 | - | **10,00** | **99,16** | **106,65** |
| DDP500-30 | 90,00 | 99,96 | **24,73** | **100,00** | **100,00** | 39,23 |
| DDP500-30ncc | **70,00** | **99,78** | **22,12** | 30,00 | 99,48 | 44,93 |
| DDP500-50 | **100,00** | **100,00** | **4,64** | **100,00** | **100,00** | 6,19 |
| DDP500-50ncc | 0,00 | 98,85 | - | 0,00 | **99,06** | - |

Tabela 4.4: Execuções buscando uma única solução

| | GANJTABESH_Filhos | | | Algoritmo Genético | | | Recozimento Simulado | | |
|---|---|---|---|---|---|---|---|---|---|
| Instâncias | Taxa de Acerto (%) | Imediação da Solução (Média) | Tempo em busca (Média) | Taxa de Acerto (%) | Imediação da Solução (Média) | Tempo em busca (Média) | Taxa de Acerto (%) | Imediação da Solução (Média) | Tempo em busca (Média) |
| DDP100-30 | **100,00** | **100,00** | 0,13 | **100,00** | **100,00** | 0,03 | **100,00** | **100,00** | **0,01** |
| DDP100-30ncc | 90,00 | 99,62 | 0,59 | **100,00** | **100,00** | 0,35 | **100,00** | **100,00** | **0,04** |
| DDP100-50 | **100,00** | **100,00** | **0,00** | **100,00** | **100,00** | **0,00** | **100,00** | **100,00** | **0,00** |
| DDP100-50ncc | **100,00** | **100,00** | 1,35 | **100,00** | **100,00** | 0,54 | 70,00 | 95,20 | **0,07** |
| DDP200-30 | **100,00** | **100,00** | 0,32 | **100,00** | **100,00** | 0,08 | **100,00** | **100,00** | **0,01** |
| DDP200-30ncc | **100,00** | **100,00** | 5,36 | **100,00** | **100,00** | 0,96 | 80,00 | 97,54 | **0,21** |
| DDP200-50 | **100,00** | **100,00** | 0,02 | **100,00** | **100,00** | **0,01** | **100,00** | **100,00** | **0,01** |
| DDP200-50ncc | 30,00 | 98,89 | 9,85 | **70,00** | **99,61** | 6,03 | 60,00 | 93,14 | **0,40** |
| DDP300-30 | **100,00** | **100,00** | 3,65 | **100,00** | **100,00** | 0,21 | **100,00** | **100,00** | **0,06** |
| DDP300-30ncc | **100,00** | **100,00** | 12,09 | 90,00 | 99,88 | 3,36 | 70,00 | 96,17 | **0,30** |
| DDP300-50 | **100,00** | **100,00** | 0,06 | **100,00** | **100,00** | **0,02** | **100,00** | **100,00** | **0,02** |
| DDP300-50ncc | 10,00 | 98,90 | 53,59 | 30,00 | **99,17** | 33,30 | **80,00** | 95,53 | **0,92** |
| DDP400-30 | **100,00** | **100,00** | 11,80 | **100,00** | **100,00** | 0,38 | **100,00** | **100,00** | **0,10** |
| DDP400-30ncc | 40,00 | 99,42 | 20,82 | 80,00 | **99,82** | 12,09 | **90,00** | 98,44 | **0,58** |
| DDP400-50 | **100,00** | **100,00** | 0,34 | **100,00** | **100,00** | 0,04 | **100,00** | **100,00** | **0,03** |
| DDP400-50ncc | 10,00 | 99,16 | 106,65 | 20,00 | **99,30** | 39,45 | **70,00** | 94,23 | **2,24** |
| DDP500-30 | **100,00** | **100,00** | 39,23 | **100,00** | **100,00** | 1,64 | **100,00** | **100,00** | **0,18** |
| DDP500-30ncc | 30,00 | 99,48 | 44,93 | 70,00 | **99,78** | 16,26 | **90,00** | 98,67 | **2,28** |
| DDP500-50 | **100,00** | **100,00** | 6,19 | **100,00** | **100,00** | 0,34 | **100,00** | **100,00** | **0,08** |
| DDP500-50ncc | 0,00 | 99,06 | - | 20,00 | **99,39** | 56,50 | **60,00** | 91,93 | **2,64** |

Para a abordagem de Recozimento Simulado, a vantagem é ainda mais evidente, sendo capaz de encontrar soluções em 60% das execuções para uma dada instância na qual as abordagens de GANJTABESH ET AL (2012) não o conseguem em nenhuma. Além disso, apresenta ou a melhor taxa de acerto ou um percentual próximo deste este valor na maior parte das instâncias, superando as demais abordagens. O tempo, da mesma forma, é, em geral, muito inferior nas execuções onde se encontrou solução.

A única contraparte desta abordagem está na baixa qualidade das soluções quando há falha na busca, fator onde é superada pelas demais. Esse ponto é um possível indicativo de uma elevada distância das soluções supracitadas em relação a um ponto promissor mas, levando-se em conta a alta taxa de sucesso do algoritmo, se torna de pouca relevância e que não diminui os méritos da técnica.

# 5 Conclusão e Trabalhos Futuros

O DDP é, de fato, uma técnica laboratorial já em desuso e não foi objetivo, em nenhum momento neste trabalho, se reverter essa situação. Porém trata-se de um problema que, apesar de estar em um nicho específico, possui certas semelhanças com diversos outros problemas combinatórios envolvendo permutação de elementos e possui um viés não muito explorado, em geral, nestes problemas: a busca por um número diverso de resultados. Este fator torna o problema uma interessante fonte de estudo para o desempenho das mais diversas heurísticas sob essa ótica.

Neste contexto, o trabalho aqui apresentado foi capaz de mostrar a adaptação de duas metaheurísticas comuns em problemas de otimização para o contexto multiresultado, demonstrando a adequabilidade das mesmas.

Dado o tempo de execução a que os algoritmos foram submetidos e ao tamanho das instâncias, o resultado atingido pode ser considerado aceitável, tendo cada técnica apresentado melhor resultado para uma determinada condição. Esta adequabilidade pode ser observada na comparação da busca por um único mapa, mesmo fora do escopo inicial das técnicas implementadas. A comparação, porém, é prejudicada devido à ausência de trabalhos submetidos às mesmas condições e objetivos inicialmente estipulados, o que poderia permitir um melhor panorama da qualidade de cada abordagem.

Ainda pode ser observado que, apesar de uma relativa superioridade da abordagem Genética para o problema envolvendo múltiplas soluções, como esperado, a diferença não é tão elevada quanto poderia ser imaginado inicialmente, trazendo o Recozimento Simulado como, de fato, uma opção nestes casos.

Por ser um tema com poucos trabalhos atuais, existem diversas possibilidades de continuação para este trabalho. Entre elas estão o desenvolvimento de outras heurísticas ou metaheurísticas para efeito de comparação com as aqui apresentadas, o aprimoramento das técnicas utilizadas, seja com métodos de geração de soluções iniciais para melhorar a robustez, seja com hibridização com outros métodos exatos ou construtivos ou, até mesmo a modificação das etapas propostas, melhorando o desempenho de algum fator

de interesse. Em sentido mais imediato, uma possibilidade seria analisar o efeito da integração do algoritmo baseado em Recozimento Simulado aqui proposto com a Busca Local Iterada em conjunto a um aumento na temperatura de parada, com o objetivo de auxiliar na principal deficiência observada na técnica original para a busca por um único mapa. Além destes, em um contexto mais prático, é possível se objetivar a resolução de instâncias contendo algum erro do processo biológico, mas sempre tendo em conta o pouco interesse no problema.

# Referências Bibliográficas


Adleman, L. M. Molecular computation of solutions to combinatorial problems. **Science (New York, N.Y.)**, v.266, n.5187, p. 1021–4, 1994.

Allison, L.; Yee, C. N. Restriction site mapping is in separation theory. **Comput. Appl. Biol. Sci**, v.4, p. 97–101, 1988.

Blazewicz, J.; Formanowicz, P. ; Kasprzak, M. Selected combinatorial problems of computational biology. **European Journal of Operational Research**, v.161, n.3, p. 585–597, 2005.

Cieliebak, M.; Eidenbenz, S. ; Woeginger, G. J. **Double digest revisited: complexity and approximability in the presence of noisy data**. Technical report, ETH Zürich, Department of Computer Science, Institute for Theoretical Computer Science, Zürich, 2002.

Cieliebak, M. **Algorithms and hardness results for DNA physical mapping, protein identification, and related combinatorial problems**. Zürich, 2003. Tese de Doutorado - Technische Wissenschaften ETH Zürich.

Ganjtabesh, M.; Ahrabian, H. ; Nowzari-Dalini, A. Molecular solutions for double and partial digest problems in polynomial time. **Computing and Informatics**, v.28, n.5, p. 599–618, 2009.

Ganjtabesh, M.; Ahrabian, H.; Nowzari-Dalini, A. ; Kashani Moghadam, Z. R. Genetic algorithm solution for double digest problem. **Bioinformation**, v.8, n.10, p. 453–6, Jan. 2012.

Goldstein, L.; Waterman, M. S. Mapping dna by stochastic relaxation. **Advances in Applied Mathematics**, v.8, n.2, p. 194 – 207, 1987.

Kao, M.-Y.; Samet, J. ; Sung, W.-K. The enhanced double digest problem for dna physical mapping. **Journal of Combinatorial Optimization**, v.7, n.1, p. 69–78, 2003.

Lundy, M.; Mees, A. Convergence of an annealing algorithm. **Mathematical Programming**, v.34, n.1, p. 111–124, 1986.

Martin, D. R. Equivalence classes for the double-digest problem with coincident cut sites. **Journal of Computational Biology**, v.1, n.3, p. 241–253, 1994.

Mneimneh, S. **Lecture 12: Physical mapping by restriction mapping**. Lecture Notes: Introduction to Computational Biology, 2008.

Pearson, W. R. Automatic construction of restriction site maps. **Nucleic Acids Research**, v.10, n.1, p. 217–227, 1982.

Pevzner, P.; Waterman, M. **Open combinatorial problems in computational molecular biology**. In: Proceedings Third Israel Symposium on the Theory of Computing and Systems, p. 158–173. IEEE Comput. Soc. Press, 1995.

Pevzner, P. A. Dna physical mapping and alternating eulerian cycles in colored graphs. **Algorithmica**, v.13, n.1/2, p. 77–105, 1995.

Pevzner, P. A. **Computational Molecular Biology An Algorithmic Approach**. MIT Press, 2000.

Schmitt, W.; Waterman, M. S. Multiple solutions of dna restriction mapping problems. **Advances in Applied Mathematics**, v.12, n.4, p. 412 – 427, 1991.

Setubal, J. a.; Meidanis, J. a. **Introduction to computational molecular biology**. PWS Boston, 1997.

Stefik, M. Inferring dna structures from segmentation data. **Artificial Intelligence**, v.11, n.1-2, p. 85 – 114, 1978.

Sur-Kolay, S.; Banerjee, S.; Mukhopadhyaya, S. ; Murthy, C. A. The double digest problem: finding all solutions. **IJBRA**, v.5, n.5, p. 570–592, 2009.

Watson, J. D.; Crick, F. H. Genetical implications of the structure of deoxyribonucleic acid. **Nature**, v.171, n.4361, p. 964–7, 1953.

Waterman, M.; Griggs, J. Interval graphs and maps of dna. **Bulletin of Mathematical Biology**, v.48, n.2, p. 189–195, 1986.

Wu, Z.; Zhang, Y. Solving large double digestion problems for dna restriction mapping by using branch-and-bound integer linear programming. **Int. J. Bioinformatics Res. Appl.**, v.4, n.4, p. 351–362, Nov. 2008.