UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

PABLO FELIPE LEONHART

Um Algoritmo Multimemético Auto-Adaptativo para o Problema de Atracamento Molecular

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Márcio Dorn

Porto Alegre
2019

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Leonhart, Pablo Felipe

Um Algoritmo Multimemético Auto-Adaptativo para o Problema de Atracamento Molecular / Pablo Felipe Leonhart. – Porto Alegre: PPGC da UFRGS, 2019.

127 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2019. Orientador: Márcio Dorn.

1. Otimização. 2. Algoritmos Multimeméticos. 3. Algoritmos Auto-adaptativos. 4. Atracamento Molecular. 5. Bioinformática Estrutural. I. Dorn, Márcio. II. Título.

“It is not the most intellectual of the species that survives,
it is not the strongest that survives,
but the species that survives is the one that is able best to adapt
and adjust to the changing environment in which it finds itself.”
— CHARLES DARWIN

AGRADECIMENTOS

Agradeço aos meus pais Milton e Rosa pelo amor, e por sempre me apoiarem de todas as formas possíveis sem medirem esforços, e também à minha irmã Paola pelo companheirismo, e compreensão nos momentos em que estive ausente. Agradeço ao meu orientador professor Dr. Márcio Dorn pela dedicação e apoio nesta importante etapa acadêmica. Também, agradeço em especial à Luísa por todo o carinho, compreensão e incentivo nos momentos difíceis enfrentados durante este período. Agradeço ainda aos meus amigos e colegas de laboratório, em especial ao Bruno Faria, Leonardo Corrêa e Pedro Narloch por todo o companheirismo e tempo dedicados na ajuda do desenvolvimento da minha pesquisa.

RESUMO

Atracamento Molecular é uma metodologia que lida com o problema de prever a ligação de um receptor e um ligante em um nível atômico para formar um complexo estável. Como o espaço de busca de possíveis conformações de ligação é vasto, este problema é classificado na teoria da complexidade computacional como um problema NP-difícil. Por conta da alta complexidade, os métodos exatos não são eficientes e várias metaheurísticas têm sido propostas. No entanto, esses métodos são muito dependentes das configurações de parâmetros e das definições do mecanismo de pesquisa, o que requer abordagens capazes de se adaptarem automaticamente ao longo do processo de otimização. Nesta dissertação, é apresentado um novo modelo de coordenação auto-adaptativa de operadores de busca local em um Algoritmo Multimemético para lidar com o problema de Atracamento Molecular. A abordagem é baseada em uma variante do *Biased Random Key Genetic Algorithm* (BRKGA), funcionando como um operador de busca global, aprimorado com quatro algoritmos de busca local: *Best Improvement, First Improvement, Stochastic Hill Descent*, e *Simulated Annealing*. O algoritmo também engloba um modelo de discretização baseado em pequenos cubos para manter diversa a população de soluções. O mecanismo de auto-adaptação ocorre na escolha de qual método de busca local deve ser aplicado durante a execução e, também, no ajuste do parâmetro de raio de perturbação, que representa o quanto a solução é modificada em cada iteração do processo de busca local. Uma nova função de probabilidades também é apresentada, como parte do núcleo de auto-adaptação, para medir o custo-benefício de cada operador de busca local, e assim guiar o processo de busca. O algoritmo multimemético foi testado em um conjunto composto por 16 estruturas baseadas na HIV-protease e comparado com ferramentas existentes na literatura: AutoDock Vina, DockThor e jMetal. Os resultados obtidos mostram que a abordagem pode predizer a ligação de complexos com conformação similar a estruturas conhecidas, em termos de *Root-Mean-Square Deviation*. Testes estatísticos indicam que o algoritmo apresenta melhores resultados quando comparado a uma abordagem não memética e não adaptativa, e é competitivo com os métodos tradicionais do estado da arte.

Palavras-chave: Otimização. Algoritmos Multimeméticos. Algoritmos Auto-adaptativos. Atracamento Molecular. Bioinformática Estrutural.

A Self-Adaptive Multimeme Memetic Algorithm for the Molecular Docking Problem

ABSTRACT

Molecular Docking is a methodology that deals with the problem of predicting binding of a receptor and a ligand at an atomic level to form a stable complex. Because the search space of possible binding conformations is vast, this problem is classified in computational complexity theory as an NP-difficult problem. Because of the high complexity, the exact methods are not efficient, and several metaheuristics have been proposed. However, these methods are very dependent on parameter settings and search mechanism definitions, which requires approaches that can automatically adapt throughout the optimization process. In this dissertation, a new model of self-adaptive coordination of local search operators is presented in a Multimemetic Algorithm to deal with the problem of Molecular Docking. The approach is based on a variant of the Biased Random Key Genetic Algorithm (BRKGA), running as a global search operator, enhanced with four local search algorithms: Best Improvement, First Improvement, Stochastic Hill Descent, and Simulated Annealing. The algorithm also encompasses a small cube-based discretization model to keep the population of solutions diverse. The self-adaptation mechanism occurs in the choice of which local search method should be applied during execution and also in the perturbation radius parameter setting, which represents how much the solution is modified in each iteration of the local search process. A new probability function is also presented, as part of the self-adaptation core, to measure the cost-effectiveness of each local search operator, and thus guide the search process. The multimemetic algorithm was tested in a set consisting of 16 structures based on HIV-protease and compared with existing tools in the literature: AutoDock Vina, DockThor and jMetal. The results show that the approach can predict the binding of complexes with conformation similar to known structures in terms of Root-Mean-Square Deviation. Statistical tests indicate that the algorithm presents better results when compared to a non-memetic and non-adaptive approach, and is competitive with traditional state-of-the-art methods.

Keywords: Optimization, Multimemetic Algorithms, Self-adaptive Algorithms, Molecular Docking, Structural Bioinformatics.

LISTA DE ABREVIATURAS E SIGLAS

AM	Atracamento Molecular
HIV	*Human Immunodeficiency Virus*
AIDS	*Acquired Immunodeficiency Syndrome*
PDB	*Protein Data Bank*
BNL	*Brookhaven National Laboratories*
EBI	*European Bioinformatics Institute*
VS	*Virtual Screening*
RMN	Ressonância Magnética Nuclear
DM	Dinâmica Molecular
BRKGA	*Biased Random-Key Genetic Algorithm*
RMSD	*Root mean square deviation*
GA	*Genetic Algorithms*
DE	*Differential Evolution*
MA	*Memetic Algorithms*
FI	*First Improvement*
BI	*Best Improvement*
SIM	*Simple Inheritance Mechanism*
PSP	*Protein Structure Prediction*
MMA	*Multimeme Memetic Algorithms*
PSO	*Particle Swarm Optimization*
SA	*Simulated Annealing*
ACO	*Ant Colony Optimization*
AGL	Algoritmo Genético Lamarckiano

LISTA DE SÍMBOLOS

Å Ångströms

Θ Ângulo theta para indicar valores radianos

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A Bioinformática consiste no estudo de problemas biológicos através da criação e emprego de modelos matemáticos e técnicas computacionais (CHOU, 2004). Caracteriza-se como uma importante área do campo das ciências biológicas, mas que por conta da enorme complexidade envolvida nos diversos processos abordados, diferentes outras áreas do conhecimento despendem esforços para, em conjunto, compreendê-los de maneira mais clara e objetiva (VERLI, 2014). De acordo com Luscombe et al. (LUSCOMBE; GREENBAUM; GERSTEIN, 2001) os principais objetivos da bioinformática são: a organização dos dados de uma forma que possibilite aos pesquisadores ter um fácil acesso a estas informações e possam compartilhar novas entradas produzidas; o desenvolvimento de ferramentas e recursos que auxiliem na análise destes dados; o uso destas ferramentas computacionais para analisar dados e interpretar os resultados; além disso, o desenho de novos experimentos, interpretação de fenômenos e construção de modelos.

A Bioinformática pode ser dividida em duas grandes vertentes, Bioinformática baseada em sequência e Bioinformática baseada em estrutura (CHOU, 2004). A primeira linha refere-se ao sequenciamento e estudo de sequências de nucleotídeos e aminoácidos, resultantes principalmente de Projetos Genoma 1 (CONSORTIUM, 2015), o que inclui aplicações com uma ampla gama de métodos analíticos, com o intuito de compreender de maneira mais simples as suas características, funções e processos de evolução. As estratégias aplicadas envolvem métodos de alinhamento de sequências, estratégias de busca em bases de dados, geração de redes metabólicas, entre outros métodos (LANDER et al., 2001; CHOU, 2004). Já a segunda linha, está relacionada a pesquisas com foco na estrutura tridimensional (3-D) de moléculas e macromoléculas, incluindo a predição de estruturas 3-D de proteínas (DORN et al., 2014), atracamento molecular (*docking*) (KITCHEN; FURR J. R., 2004), modelagem molecular e estudos sobre as relações entre estrutura e função de proteínas (WHISSTOCK J. C.; LESK, 2003). As informações estruturais correspondentes a cada tipo de molécula (DNA, proteína, ligante, etc.), podem ser obtidas através da aplicação de variados métodos experimentais, tais como cristalografia de raios-X (MCREE, 1999), Ressonância Magnética Nuclear (RMN) (CAVANAGH et al., 2006) e microscopia eletrônica (ME) (UNWIN; HENDERSON, 1975).

O foco desta pesquisa é o problema do Atracamento Molecular (AM). Esta é uma abordagem computacional utilizada no processo de descobrimento de fármacos para predizer a conformação de uma pequena molécula (ligante) dentro do sítio de ligação de

uma molécula maior (receptor), medindo a afinidade de ligação entre estas moléculas. As principais estruturas utilizadas como receptor são proteínas. Proteínas e polipeptídeos são polímeros formados por 20 diferentes tipos de resíduos de aminoácidos conectados por meio de uma ligação peptídica (LESK, 2005). Cada proteína é definida por sua sequência única de resíduos de aminoácidos que em condições fisiológicas se enovelam numa forma específica conhecida como estado nativo (ANFINSEN, 1973). São estruturas fundamentais para o organismo, suas funções variam desde construção de novos tecidos do corpo humano, transporte de substâncias, atuação no sistema de defesa do organismo, catalisação de reações químicas, regulação de hormônios, entre outros.

Ferramentas de AM buscam encontrar um modelo que descreva a interação entre duas estruturas moleculares, onde o conhecimento da forma tridimensional do receptor e ligante implica na inferência de sua função. Considerando o grande número de possíveis conformações que uma molécula pode assumir, este problema é considerado NP-difícil por conta de sua complexidade computacional (SADJAD; ZSOLDOS, 2011). Por conta disso, o uso de métodos determinísticos de otimização exigiria um elevado tempo de execução, inviabilizando o uso dessas técnicas. Assim, o desenvolvimento de métodos de busca capazes de explorar o espaço de busca de conformações de ligantes é essencial. Dessa maneira, metaheurísticas, como algoritmos meméticos, têm sido propostas e aplicadas na obtenção de boas soluções em tempo de execução razoável. Em Sousa et al. (SOUSA et al., 2013) são apresentados diversos *softwares*, metodologias e parametrizações desenvolvidas na última década de pesquisa na área. Um dos grandes desafios computacionais é trabalhar com a flexibilidade dos complexos, o que inclui os graus de liberdade dos átomos das estruturas. As diferentes abordagens são geralmente divididas em: (i) métodos de receptor e ligante rígidos, (ii) métodos de ligante flexível, e (iii) métodos de receptor e ligante flexíveis.

Na abordagem rígida, são consideradas apenas a translação e rotação das moléculas receptora e ligante. Atualmente, a grande maioria das ferramentas de AM incluem a flexibilidade dos ângulos diedrais do ligante, além de considerar a translação e rotação das estruturas (MAGALHAES, 2006). Nas duas abordagens o receptor é mantido rígido durante o processo, conforme determinado experimentalmente. Estudos recentes incluem a flexibilidade no receptor também (MACHADO et al., 2011; TEODORO; KAVRAKI, 2003; COZZINI et al., 2008; HUANG S.Y.AND ZOU, 2007; WONG, 2008; ALONSO; BLIZNYUK; GREADY, 2006; CHANDRIKA; SUBRAMANIAN; SHARMA, 2009), no entanto, assim como na abordagem de receptor rígido, o *docking* de estruturas de ligantes

grandes e com alta flexibilidade é um grande desafio para estes algoritmos.

O desenvolvimento de ferramentas de Atracamento Molecular envolve duas partes principais: o método de busca, o qual, deve considerar todas as conformações possíveis; e a função de energia, para avaliação da conformação de ligação dos compostos. O algoritmo de busca deve percorrer o sítio de ligação (espaço de busca) com um detalhamento suficiente para encontrar o mínimo global da função de energia. No atracamento rígido, o espaço de busca inclui as conformações a partir da translação e rotação do ligante. Já no atracamento flexível, os graus de liberdade internos da estrutura são adicionados, tornando o modelo conformacional mais realista ao processo como ocorre na natureza. Independente da possibilidade de utilizar metaheurísticas para exploração do espaço de busca, os métodos aplicados são muito dependentes da parametrização e definição dos mecanismos de busca. Dessa forma, mecanismos auto-adaptativos, por exemplo, possibilitam a estes algoritmos a opção de auto-ajustar quais valores de parâmetros, ou mecanismo de busca devem ser utilizados ao longo do processo de otimização (JIN; ZHIHUA; WENYIN, 2014).

Além do mecanismo de busca, outro importante componente é a função de energia responsável por descrever a interação entre receptor e ligante, avaliando diferentes aspectos físico-químicos relacionados ao processo de ligação. A função de avaliação deve ser o mais realista possível para fornecer resultados compatíveis com o complexo determinado experimentalmente (BROOIJMANS, 2003). A função que representa as interações moleculares envolvidas no reconhecimento molecular proteína-ligante incluem: ligações de hidrogênio, interações de *van der Waals*, interações iônicas, interações hidrofóbicas, interações do tipo cátion-π, interações envolvendo anéis aromáticos do tipo π-π e empilhamento-T e coordenadas com íons metálicos (VERLI, 2014). A escolha de uma função de avaliação de energia que represente o sistema e as interações moleculares é de suma importância para o algoritmo de busca, pois é ela que irá distinguir e ranquear diferentes soluções de acordo com a sua energia de ligação.

Problemas de Atracamento Molecular têm enfrentado diversos desafios. De acordo com Sousa et al. (SOUSA et al., 2013) o atracamento proteína-ligante apresenta algumas questões críticas: o tratamento da flexibilidade da proteína, a presença de estruturas moleculares de água e seus efeitos, além da entropia de ligação química. A amostragem do ligante, a flexibilidade da proteína e a função de energia, descritos em Huang et al. (HUANG; ZOU, 2010), são aspectos importantes para a resolução do problema. A amostragem refere-se a geração de orientações e conformações próximas do sítio de ligação. A

avaliação dessa conformação de ligação utilizando uma função de aptidão é fundamental para o algoritmo. Além disso, a complexidade da função de energia infere no custo de execução do algoritmo em si, portanto, a análise da relação entre o cálculo de energia e o custo computacional é um desafio no problema de Atracamento Molecular.

1.1 Motivação

Estudos e avanços de técnicas experimentais têm contribuído para o aumento significativo da quantidade de estruturas de proteínas conhecidas e também o número de estruturas de ligantes (ZHANG et al., 2012). Com isso, surge a necessidade de gerenciar todos estes dados e desenvolver algoritmos que agilizem o processo de descoberta de novos fármacos. Diversas ferramentas de Atracamento Molecular já foram propostas, no entanto, o problema ainda requer uma abordagem mais generalista e acurada no sentido de prever a melhor conformação entre as proteínas receptora e ligante, considerando as deficiências das funções de energia e o excesso de flexibilidade do complexo.

De acordo com Sousa et.al. (SOUSA et al., 2013) estratégias baseadas em algoritmos evolutivos têm apresentado melhores resultados do que algoritmos determinísticos. Tais abordagens permitem melhor explorar o espaço de busca com rápida convergência mesmo em casos com funções objetivos não tão simples, como as multimodais. Algoritmos Genéticos são um exemplo de metaheurística utilizada em diversos estudos e têm se mostrado uma técnica bem sucedida. Além disso, problemas com características similares fazem uso da combinação de técnicas de busca global e local de modo eficiente, e estas mesmas técnicas podem ser estudadas no contexto de Atracamento Molecular. E também, a dimensionalidade e o espaço contínuo das variáveis do problema permitem o estudo e aplicação de diferentes técnicas para melhoria do processo de busca, visto que as melhores características de cada uma delas podem ser exploradas.

Neste sentido, propõe-se o desenvolvimento de um algoritmo memético auto-adaptativo que incorpore um modelo de discretização do espaço conformacional e conceitos de metaheurísticas evolutivas e técnicas de busca local, cujos parâmetros: (i) algoritmo e (ii) intensidade de exploração do mesmo possam ser ajustados em tempo de execução, com o objetivo de utilizar o melhor de cada abordagem para explorar eficientemente o espaço de busca, mantendo a diversidade de soluções para atingir bons resultados no problema de AM. Devido à complexidade apresentada neste problema por conta do vasto número de possíveis conformações de ligação das moléculas, a motivação de adotar um

algoritmo memético vem da possibilidade de flexibilizar o uso de diferentes técnicas de otimização global e local. Com o intuito de facilitar a exploração do espaço conformacional por inteiro, através da diversificação de soluções, propiciada pelo método global e o modelo de discretização, e da intensificação na melhora destas soluções por meio das técnicas locais.

O trabalho apresenta um estudo sobre alguns dos métodos mais relevantes aplicados ao problema de Atracamento Molecular. O foco do trabalho é definido em analisar e estudar os ganhos que a aplicação de buscas locais podem trazer quando combinadas com uma heurística de busca global, a partir disso propor um modelo auto-adaptativo que permite aplicar diferentes algoritmos em diferentes momentos do processo de busca local das soluções, bem como ajustar a intensidade de exploração de cada método, além de comparar com técnicas relevantes já propostas na literatura. As principais contribuições deste trabalho são o desenvolvimento e avaliação de um modelo auto-adaptativo robusto aplicado ao problema de Atracamento Molecular.

Destaca-se que um dos propósitos deste trabalho é conseguir determinar conformações proteína-ligante com acurácia, o que pode proporcionar benefícios em vários campos de pesquisa como: Medicina, Bioinformática e indústria farmacêutica (TRAMONTANO; LESK, 2006). O Atracamento Molecular ainda é uma área que carece de algoritmos robustos e eficazes, permitindo a aplicação de diferentes métodos com possibilidade de avanços científicos significativos.

1.2 Objetivos e metas

O objetivo dessa pesquisa é desenvolver um algoritmo memético auto-adaptativo para o problema de Atracamento Molecular. A escolha das estruturas utilizadas, da representação dos dados, de uma função de energia que descreva bem as interações moleculares e dos parâmetros dos algoritmos são exemplos de variáveis que influenciam na resolução deste problema. Portanto, o desenvolvimento de uma técnica direcionada ao problema de AM inclui a definição destas variáveis mencionadas. As metas a serem alcançadas com o desenvolvimento desta dissertação e necessárias ao cumprimento do objetivo geral são:

1. Estudar as principais características do problema AM, visando a sua modelagem como problema de otimização, bem como suas restrições, limitações e desafios;
2. Estudar as heurísticas e metaheurísticas mais relevantes atualmente, além dos mé-

todos que descrevam o estado da arte para o atracamento molecular;

3. Estabelecer uma metodologia para preparação dos complexos para testes. Nesse processo, é onde se define o conjunto de testes e cada estrutura requer atenção, isto é, verifica-se a necessidade de adicionar ou remover átomos e/ou resíduos das proteínas, bem como o posicionamento das cadeias laterais, entre outras modificações estruturais destes compostos;
4. Implementar o algoritmo memético tendo em vista os algoritmos de busca global e local;
5. Analisar os resultados da abordagem implementada. Verificar se o uso de algoritmos meméticos trazem maiores benefícios em relação a outras abordagens, e nesse caso, qual algoritmo de busca local proporciona melhores resultados;
6. Desenvolver o modelo de auto-adaptação do algoritmo memético. Nesta etapa é necessário adotar uma estratégia que irá guiar o mecanismo auto-adaptativo, de modo que o método de busca local e a intensidade da busca sejam os mais adequados de acordo com o momento atual do processo de busca.

Portanto, espera-se com este trabalho obter o êxito de gerar um modelo auto-adaptativo de otimização que incorpore, de modo inteligente, informações ao longo do processo de busca para melhor guiar o método no refinamento das soluções. Podendo assim, ter uma técnica que consiga maior acurácia no processo de Atracamento Molecular.

1.3 Organização

Os próximos capítulos desta dissertação estão organizados da seguinte forma:

- Capítulo 2: Fundamentação Biológica. Serão apresentados conceitos básicos sobre moléculas, suas funções biológicas e interações, os tipos de funções de cálculo de energia livre e suas definições. Também, os principais bancos de dados que disponibilizam os arquivos de representação dos complexos utilizados neste trabalho. A relação entre métodos de Atracamento Molecular e técnicas de Triagem Virtual é descrita. Finalmente, os tipos de atracamento e suas características são apresentados.
- Capítulo 3: Algoritmos de Atracamento Molecular. Neste capítulo são apresentados as técnicas e algoritmos empregados para a resolução do problema, bem como suas formas de representação e as categorias de métodos aplicados. Por fim, são

apresentados os principais desafios na área de Atracamento Molecular.

- Capítulo 4: Materiais e Métodos. Neste capítulo serão apresentados os algoritmos e estratégias adotados no desenvolvimento desta dissertação, bem como a estruturação do método proposto. Este capítulo tem por objetivo descrever a forma como o desenvolvimento do trabalho foi conduzido, e a metodologia utilizada para isto.
- Capítulo 5: Experimentos e Resultados. O capítulo apresenta os experimentos realizados e os resultados obtidos concernentes à etapa de desenvolvimento do algoritmo memético, proposto para avaliar se a inclusão de busca local gera melhores soluções para o problema, e qual dos métodos seria mais eficiente. Em seguida é discutido o modelo de auto-adaptação proposto, o qual, é baseado numa equação que leva em conta o ganho e efetividade de cada parametrização, para atribuir probabilidades a cada um destes parâmetros e guiar o processo de escolha de uso destes parâmetros. São realizados experimentos com o mesmo conjunto de instâncias de teste. Após isso, é avaliada a performance do algoritmo desenvolvido em relação a aspectos computacionais e significância biológica dos resultados;
- Capítulo 6: Conclusão e Trabalhos Futuros. O capítulo apresenta as considerações finais relativas ao trabalho desenvolvido. São descritas as conclusões formuladas com os resultados obtidos, identificados os objetivos e metas atingidos, apontando as principais dificuldades encontradas. Por fim, são alinhadas recomendações e propostas para futuros trabalhos relacionados ao método desenvolvido e ao problema abordado.

2 FUNDAMENTAÇÃO BIOLÓGICA

Neste capítulo serão apresentados conceitos de fundamentação biológica necessários para o entendimento da área de atracamento molecular. O objetivo é discutir os principais conceitos que envolvem esta pesquisa, visando o embasamento teórico do problema Atracamento Molecular, bem como de questões relativas a este.

2.1 Estruturas Moleculares

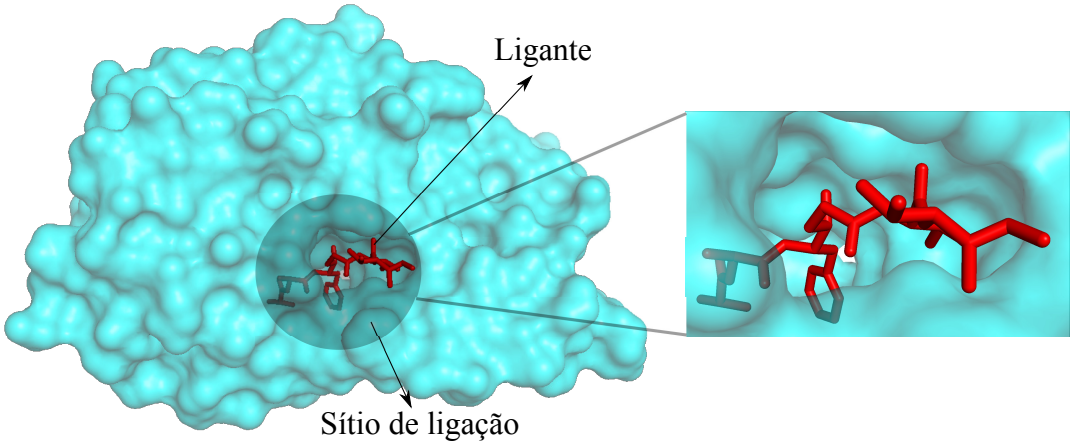
Em Bioinformática Estrutural são estudados problemas biológicos num ponto de vista tridimensional, abrangendo diversas técnicas compreendidas pela química computacional ou modelagem molecular. Um dos principais problemas de pesquisa nesta área é o de Atracamento Molecular, que basicamente consiste no processo de encontrar a melhor ligação entre duas proteínas. Proteínas são polímeros sintetizados pelas células a partir dos aminoácidos, estas biomoléculas, além de serem estáveis, podem adotar diversos arranjos tridimensionais (VERLI, 2014).

De maneira sucinta, no problema de AM existe uma molécula receptora (macromolécula) que se liga a uma segunda molécula (micromolécula), também conhecida como ligante, a qual é uma estrutura capaz de gerar ou bloquear uma reação biológica no organismo (BARREIRO; FRAGA, 2014). Por serem moléculas dinâmicas, suas conformações estão em constante mudança, o que reflete na vibração molecular e em alterações conformacionais nas estruturas. As associações moleculares podem ser entre proteínas, por exemplo, ou outras moléculas biologicamente relevantes, como ácidos nucleicos, hidratos de carbono e lipídios. As características transientes de receptores e ligantes são críticas para a vida, pois permitem a um organismo reagir rapidamente a uma mudança em um ambiente ou por circunstâncias metabólicas (LEHNINGER; NELSON; COX, 2004).

A conexão entre as moléculas ocorre em uma região do receptor chamada de sítio de ligação, que é uma área complementar ao ligante em tamanho, forma, cargas e características hidrofóbicas ou hidrofílicas. Esta interação é muita específica, de modo que a proteína pode discriminar milhares de moléculas e seus ambientes realizando a ligação química com apenas um ou alguns poucos ligantes. Tal seletividade é importante para manter o alto grau de ordem em um sistema vivo. Exemplos de importantes receptores em nosso corpo são: enzimas, e receptores hormonais, de sinalização celular e neurotransmissores (DU et al., 2016).

Neste trabalho, uma das estruturas estudadas é a HIV-protease. Trata-se de uma enzima envolvida na hidrólise da ligação peptídica em retrovírus, que é essencial para o ciclo de vida do HIV (*Human Immunodeficiency Virus*), cuja replicação no organismo causa a AIDS *Acquired Immunodeficiency Syndrome*. Fármacos desenvolvidos têm como propósito se ligar no sítio de ligação desse molécula receptora de modo que essa enzima seja bloqueada, e consequentemente o vírus não possa se reproduzir. A Figura 2.1 representa a estrutura tridimensional do complexo de código `PDB 1AAQ` (DREYER et al., 1992), cuja molécula receptora é a estrutura HIV-protease, são exibidos também, o sítio de ligação e o ligante.

Figura 2.1: Representação tridimensional em *surface* do complexo de código PDB `1AAQ`, com destaque para o sítio de ligação e o ligante (em vermelho)



Fonte: Do Autor.

Portanto, o problema de Atracamento Molecular é definido como a busca pela melhor ligação entre as duas moléculas. Tendo em vista o sítio de ligação da molécula receptora, a complexação ideal do ligante permite que determinada função da proteína seja ativada ou inibida, por exemplo. Características físico-químicas são responsáveis pela afinidade e especificidade de ligante e receptor, enquanto que as estruturais determinam a organização espacial das moléculas, onde as variações nessas estruturas são regidas por mudanças de translação, orientação e rotações de ligações covalentes.

2.2 Interações Ligante-Receptor

A formação de complexos por duas ou mais moléculas promove comunicações intra e intermoleculares entre as partes envolvidas (EISENSTEIN; KATZIR, 2004). O processo de ligação entre proteína e ligante ocorre junto a uma mudança conformaci-

onal na proteína fazendo com que seu sítio de ligação seja complementar à forma do ligante. Essa interação do fármaco com o seu sítio de ligação ocorre durante uma etapa chamada farmacodinâmica sendo determinadas pelas resultantes entre forças intermoleculares atrativas e repulsivas, ou seja, interações hidrofóbicas, eletrostáticas e restrições estéricas (BARREIRO; FRAGA, 2014). Estas interações são fundamentais para quase todos os processos em um organismo vivo (DUNN, 2010).

De acordo com Pauling et.al. (PAULING; DELBRUCK, 1940), as principais interações entre os complexos biomoleculares são: (i) as interações de *van der Waals*, caracterizadas pela atração de moléculas apolares com dipolo induzido; (ii) interações eletroestáticas, cujas forças resultam numa atração ou repulsão entre as cargas e dependem de uma constante dielétrica do meio e da distância intermolecular das cargas; e ligações de hidrogênio, na qual apenas dois elétrons são compartilhados por três átomos. Todas essas interações são importantes para a estabilidade do complexo biomolecular (BENITE; MACHADO; BARREIRO, 2007).

Solventes também são outro importante fator nas interações receptor-ligante, eles são substâncias que permitem a dissolução de outras substâncias em seu meio. Um exemplo de solvente é a água, a qual pode alterar características estruturais dos sítios de ligação devido a sua proximidade às moléculas (PAULING; DELBRUCK, 1940). A grande parte das proteínas passa pelo processo de enovelamento e desempenham suas funções em meio aquoso. Dados estruturais e termodinâmicos indicam que a água pode contribuir para a ligação química entre as partes de um complexo (LADBURY, 1996). A adição do solvente também modifica a entropia do sistema, pois as superfícies apolares liberam e embaralham as moléculas de água. Esse aumento da entropia do solvente combinado com o ocultamento das superfícies apolares é chamado de efeito hidrofóbico (BALDWIN, 2014).

Mudanças na entropia do sistema alteram a estabilidade do complexo como perda da entropia rotacional e translacional, além de variações na entropia vibracional e conformacional da molécula. O solvente também pode interagir na parte interna das proteínas, de modo a preencher parcialmente ou completamente seus canais. Assim, moléculas de água próximas da estrutura da proteína acabam fazendo parte do processo de ligação, pois influenciam na conformação das cadeias laterais expostas, estabilizam o fim das estruturas secundárias, e também ocupam posições nos sítios alvos, influenciando as ligações e catalisação (RICHARDSON, 1981).

Cofatores como coenzimas e grupos prostéticos são substâncias orgânicas (coenzi-

mas) ou inorgânicas necessárias para o funcionamento das enzimas. Estas e muitas outras proteínas somente conseguem realizar sua função bioquímica se conectadas a uma molécula diferente (KEPPEL, 1991). As principais coenzimas são vitaminas, que por sua vez podem estar fortemente conectadas à proteína, por exemplo, íons de metais como zinco e cobre. Estas ligações com metais são capazes de estabilizar uma estrutura 3D de uma proteína. Além disso, os metais neutralizam cargas negativas que poderiam em outra situação se repelir. Estas estruturas são ainda utilizadas como fator catalisador de atividades em enzimas, podendo se agrupar nas proteínas.

2.3 Banco de Dados

Bancos de dados biológicos são bibliotecas de informações científicas coletadas a partir de experimentos e análises computacionais (ATTWOOD et al., 2011). Eles contêm informações de áreas de pesquisa incluindo genômica, proteômica, metabólicas, espressão gênica por *microarray*, e filogenéticas (ALTMAN, 2004). A representação computacional das estruturas biológicas utiliza diversas técnicas. Uma delas é a cristalografia de raio-X, a qual, consiste em aplicar os raios, numa forma de radiação eletromagnética, através de um cristal da substância utilizada. A difusão do feixe em várias direções permite, por radiação, identificar um padrão de intensidades que possibilitam extrair diversas informações sobre a estrutura atômica e molecular do objeto de estudo. Outra técnica é a ressonância magnética nuclear, a qual é uma técnica de pesquisa que explora as propriedades magnéticas de certos núcleos atômicos para determinar propriedades físicas e/ou químicas de átomos ou moléculas nos quais eles estão contidos. Bancos de dados como o *Protein Data Bank* (PDB; http://www.rcsb.org/pdb/) (BERMAN et al., 2000), e ZINC (http://zinc.docking.org/) (IRWIN; SHOICHET, 2005) utilizam, entre outras técnicas, a cristalografia de raios-X e disponibilizam essas moléculas para estudos científicos.

O PDB é o repositório de dados estruturais em 3D de proteínas e ácidos nucléicos mais difundido. Ele foi criado em 1971 pelo *Brookhaven National Laboratories* (BNL) (https://www.bnl.gov/world/) como um banco de estruturas obtidas através da difração de raios-X, ressonância magnética nuclear e crio-microscopia eletrônica, as quais são enviadas por físicos, biólogos e bioquímicos de todo o mundo. As informações que ele apresenta de cada molécula são a representação computacional, método de aquisição, resolução, entre outras importantes informações para análise. Centros de aquisições de estruturas como o *European Bioinformatics Institute* (EBI), e o *Protein Data Bank Ja-*

pan (PDBj) atuam como parceiros do PDB. Todos os dados passam por validação para assegurar a qualidade do modelo atômico a ser disponibilizado.

Já o banco de dados ZINC é uma coleção maior de compostos químicos, comercialmente disponíveis e preparados para o processo de Triagem Virtual. O repositório disponibiliza compostos químicos para alvos biológicos, incluindo fármacos comerciais. Em conjunto com outros 20 bancos de dados, o foco do ZINC é compostos para Atracamento Molecular, disponibilizando resolução, flexibilidade, entre outras informações químicas das estruturas.

2.4 Funções de Energia

Determinar com acurácia e baixo custo computacional a energia de ligação de um complexo molecular é uma importante área de estudos no campo da bioinformática. Cada conformação possível entre receptor e ligante é conhecida como *pose*, isto é, um modo de ligação molecular candidato. Encontrar o menor valor de energia no problema de AM permite o ranqueamento dos *poses* e então determinar se o ligante é propenso a ser sintetizado. O cálculo de energia de ligação de um complexo necessita métodos computacionais robustos (FRENKEL; SMIT, 2002), no entanto, devido à necessidade de uma rápida avaliação, muitas vezes, são utilizadas funções que aproximam essa avaliação. Dessa forma, diferentes funções de energia têm sido utilizadas por programas de Atracamento Molecular, as quais, podem ser classificadas da seguinte forma: funções baseadas em campo de força, funções empíricas e semi-empíricas, e funções baseadas em conhecimento (KITCHEN; FURR J. R., 2004).

2.4.1 Funções baseadas em campo de força

Campo de força pode ser entendido como um campo vetorial que descreve as forças agindo sobre uma partícula em várias posições no espaço. Funções baseadas em campo de força mensuram a soma das energias de interação receptor-ligante e interna do complexo. O primeiro programa de atracamento molecular proposto foi o DOCK (KUNTZ et al., 1982), refinado pelos grupos Shoichet (WEI et al., 2004) e Abagyan (TOTROV; ABAGYAN, 1997), e teve a utilização de uma função baseada em campo de força, o AMBER (WEINER et al., 1984). A maioria desse tipo de função considera o receptor

rígido (veja detalhes na Seção 2.6), causando a omissão do cálculo da energia interna do mesmo, simplificando assim a avaliação da energia de ligação. Outros exemplos de campo de força amplamente utilizados são CHARMM (CORNELL; CIEPLAK, 1995; BROOKS, 1983), e MMFF94 (HALGREN, 1996).

A interação entre ligante e receptor é descrita, frequentemente, por parâmetros de energia de *van der Waals* e eletrostática. O primeiro termo é dado pela energia potencial de *Lennard-Jones*, enquanto que o segundo é inferido pela formulação de *Coulomb* com uma função que avalia a distância entre cargas e suas contribuições individuais. A forma funcional de energia interna do ligante é geralmente bastante similar com a interação receptor-ligante, incluindo também termos de *van der Walls* e eletrostática. A Equação 2.1 descreve os termos acima mencionados:

$$\Delta G_{bind} = \sum_{i=1}^{ligand} \sum_{j=1}^{protein} \left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\epsilon R_{ij}} \right] \tag{2.1}$$

Os parâmetros A e B são definidos para cada par de diferentes combinações de tipos de átomos, R é a distância entre os centros atômicos, q é a carga parcial de cada átomo, e ϵ é a constante dielétrica. A escolha precisa destes parâmetros tem efeitos substanciais no comportamento e performance da função de avaliação. Funções baseadas em campo de força apresentam grandes limitações, já que são originalmente formuladas para a modelagem de contribuições entálpicas para a estrutura e energias, não incluindo assim, solvatação e termos de entropia (KITCHEN; FURR J. R., 2004).

2.4.2 Funções empíricas e semi-empíricas

Esse tipo de função, inicialmente proposta por Böhm et.al. (BÖHM, 1992), é inferida a partir de dados experimentais, a qual analisa a energia de ligação e/ou a conformação como a soma de diversas funções parametrizadas. A formulação destas funções se baseia na ideia de que a energia de ligação pode ser aproximada pela soma de termos individuais não relacionados. Assim, são obtidos coeficientes de vários termos por meio da análise de regressão usando energias de ligação já determinadas experimentalmente, e de informações estruturais obtidas por cristalografia de raios-X.

A formulação das funções empíricas ou semi-empíricas é normalmente mais simples do que as funções por campo de força, apesar de que muitos termos de contribuição individual tenham partes idênticas ao dos termos da mecânica molecular. Como vanta-

gem, esta simplicidade faz com que a avaliação dos termos seja mais fácil. Por outro lado, a desvantagem é ter de usar dados experimentais no desenvolvimento da regressão e adaptação, o que leva a diferentes fatores de pesos para diferentes termos (SCHNEIDER; BöHM, 2002). O resultado disso é que se torna difícil a recombinação de diferentes parametrizações para formular uma nova função de energia de aplicação geral.

A ferramenta GOLD (JONES; WILLETT, 1995) utiliza uma função de energia baseada nos termos de avaliação das ligações de hidrogênio, o potencial de *van der Waals*, e a conformação interna do ligante. A função proposta por Morris et.al. (MORRIS et al., 1998) adota termos semi-empíricos e baseados em campo de força, embora estes também tenham seus pesos multiplicados por termos obtidos experimentalmente. Outra função de energia é o AutoDock (MORRIS et al., 2009), que além de reescalonar os coeficientes nos termos da função de energia da mecânica molecular, incluem dois novos termos. Tais termos refletem o efeito da solvatação na interação das moléculas, produzindo uma estimativa da perda de graus de conformação do ligante quando o mesmo se conecta ao receptor. Outros *softwares* como LUDI (BÖHM, 1992) e FlexX (RAREY et al., 1996) implementam funções empíricas, adicionando termos de ligação de hidrogênio, ponte salina, efeito hidrofóbico e entropia.

Como visto, as funções empíricas têm sua formulação bem variada. Termos para as interações de pares de átomos não-ligados são um exemplo, assim como contribuições não-entálpicas, conhecidas como termos rotor. Esses tipos de termos aproximam as penalidades de entropia da ligação, aumentando o peso do somatório do número de ângulos diedrais nos ligantes. Contudo, termos utilizados atualmente para aproximar a entropia ou energia de solvatação incorporam descrições incompletas desses efeitos em ligações proteína-ligante (SCHNEIDER; BöHM, 2002).

2.4.3 Funções baseadas em conhecimento

Esse tipo de função é formulada a partir da análise de dados experimentais de estruturas. Para inferir a função, os complexos são modelados utilizando relações simples de potenciais de átomos pareados e um número de interações de átomos é definida de acordo com o ambiente molecular. Dessa forma, estas funções tentam, implicitamente, capturar os efeitos da ligação que são difíceis de modelar de forma explícita (WANG; LU; WANG, 2003). Um dos fatores incluídos nestas técnicas são os potenciais de força média (PMF) (MUEGGE; MARTIN, 1999; MUEGGE, 2001) para avaliação da energia

de ligação do complexo.

Um dos *softwares* mais conhecidos, o Rosetta (GRAY et al., 2003), implementa uma função baseada em análises estatísticas do PDB e inclui pesos variados a cada termo. A Drugscore (GOHLKE; HENDLICH; KLEBE, 2000) inclui ainda correções de acessibilidade do solvente para avaliar a interação das moléculas. SMoG (DEWITTE; SHAKHNOVICH, 1996) é outra ferramenta que utiliza a mesma classe em vários termos de sua função de avaliação. A grande vantagem em utilizar esse tipo de função é a simplicidade e o baixo custo computacional, o que facilita uma análise em uma base de dados grande para triagem de fármacos. Entretanto, a derivação dessas funções é basicamente feita sobre informações experimentais de moléculas limitadas, o que requer uma grande quantidade de complexos para sua composição (ZHANG et al., 2005).

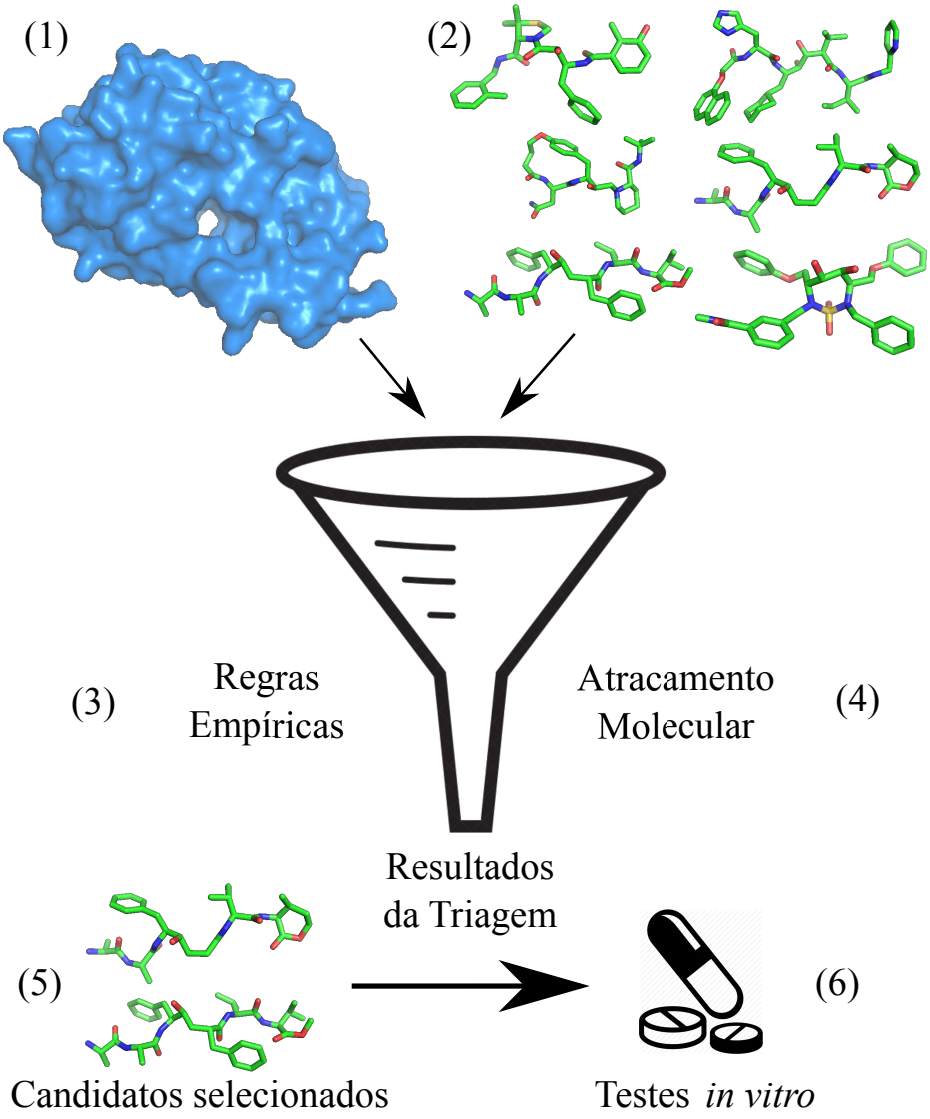
2.5 Triagem Virtual

A Triagem Virtual, *Virtual Screening* (VS), é um processo que auxilia na descoberta de novos fármacos, onde o objetivo é encontrar, com o auxílio de técnicas de AM, estruturas mais propensas a se ligarem em uma molécula alvo, geralmente uma proteína ou enzima. Enormes bases de dados com compostos comercialmente disponíveis são computacionalmente testados contra estruturas alvo conhecidas, onde aquelas que forem melhores preditas, irão ser testadas experimentalmente (REDDY et al., 2007; LAVECCHIA; GIOVANNI, 2013). É uma espécie de filtro que reduz a quantidade de compostos químicos presentes nesses bancos. O processo de seleção de estruturas em base de dados químicas é uma metodologia bem estabelecida para encontrar candidatos a fármacos, a partir da estrutura 3D alvo conhecida (WALTERS; STAHL; MURCKO, 1998). O aumento dos alvos farmacêuticos preditos faz com que a Triagem Virtual tenha um papel fundamental para encontrar os primeiros compostos alvos, principalmente quando não se tem informação sobre potenciais ligantes (BISSANTZ; FOLKERS; ROGNAN, 2000).

O esquema geral dos métodos de VS, ilustrado na Figura 2.2, tem seu processo iniciado com a análise de informações da estrutura 3D de interesse. A estrutura alvo pode ser derivada de dados experimentais como raios-X e ressonância magnética nuclear (RMN), ou modelagem comparativa, ou de simulações de dinâmica molecular (DM). Aspectos fundamentais a serem examinados nessas estruturas são: a drogabilidade do receptor, isto é, a capacidade do receptor em ligar-se com alta afinidade e especificidade a um fármaco, a escolha do sítio de ligação, a incorporação de flexibilidade no receptor (mais discutida

na Seção 2.6), além de diversos aspectos químicos. Outro importante fator é a escolha do conjunto de ligantes que serão testados contra o receptor. Eles também requerem um pré-processamento para garantir propriedades estereoquímicas, tautoméricas, e estados de protonação.

Figura 2.2: Processo de Triagem Virtual: desde a seleção da molécula receptora (1) e um conjunto de ligantes candidatos (2), passando por um processo de seleção composto por regras empíricas (3) e ferramentas de Atracamento Molecular (4), até a seleção de algumas moléculas para a projeção de fármacos (5) e realização de testes *in vitro* (6).



Fonte: Do Autor.

Após isso, ferramentas de Atracamento Molecular são utilizadas para modelar e avaliar possíveis poses de ligação para cada composto. Estas ferramentas exploram três tipos de técnicas para avaliar grandes números de compostos de maneira eficiente: (i) representação da estrutura com menos flexibilidade, para reduzir o tamanho do espaço de busca; (ii) métodos eficientes de exploração do espaço de busca, para identificar possíveis

poses; (iii) funções de energia rápidas, para ranquear os composto em termos de diferenças relativas estimadas na afinidade de ligação. Representações de estruturas dedicadas para o AM geralmente restringem o espaço de busca ao sítio de ligação do receptor e substituem os modelos *full-atom* por representações mais simplificadas. Assim, a função de energia conseguirá avaliar, de forma aproximada e rápida, a energia de ligação das moléculas e então ranquear a biblioteca de ligantes. Em seguida, os compostos são avaliados pela sua conformação, aspectos físico-químicos desejáveis e indesejáveis. Por fim, o resultado é um menor número de compostos selecionados que seguem para ensaios experimentais (LIONTA et al., 2014). A sofisticação das ferramentas de Triagem Virtual e sua dependência do contexto cresce com o conhecimento disponível de uma droga em particular e com o padrão de interação das moléculas (KELLENBERGER et al., 2004).

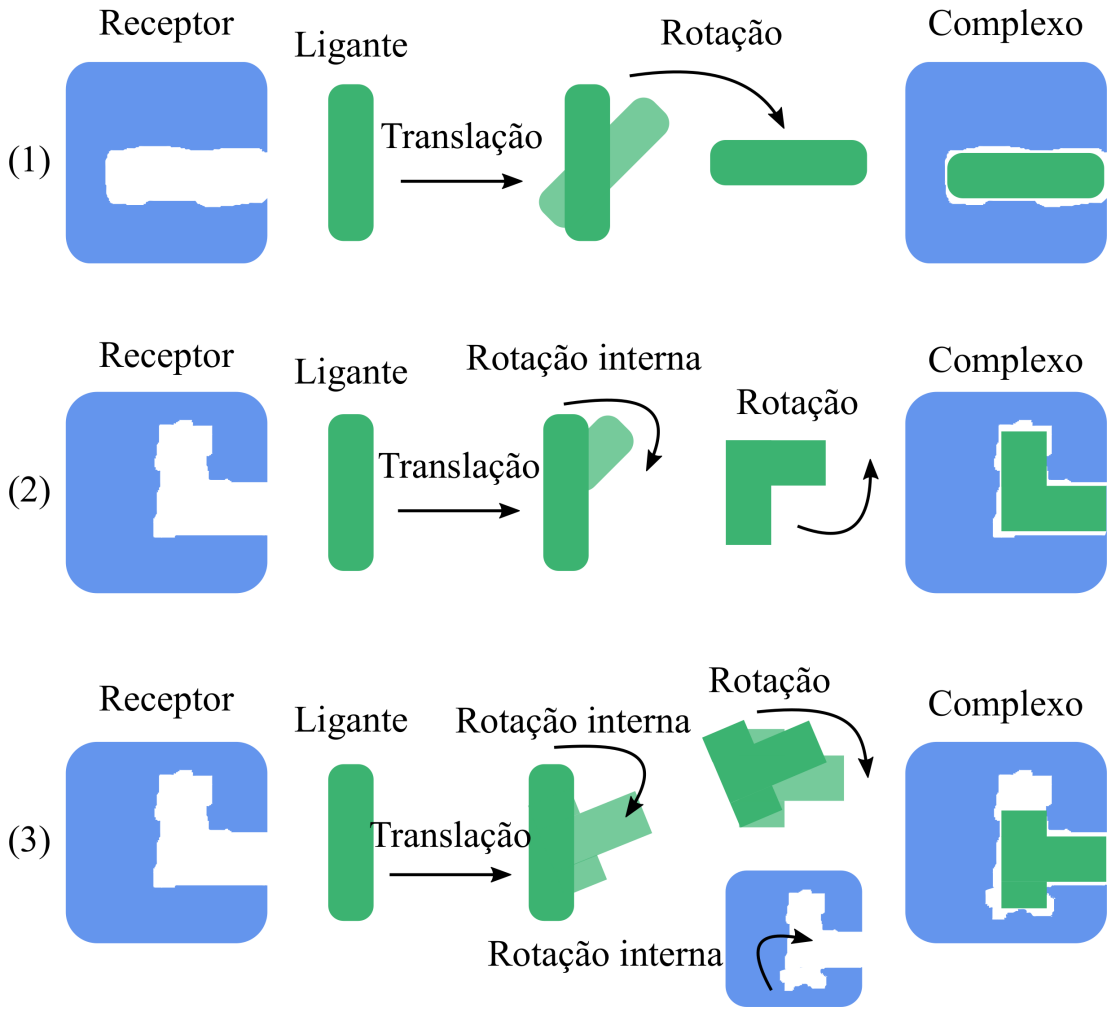
2.6 Tipos de Atracamento Molecular

Um importante fator a ser considerado em um método de Atracamento Molecular é a flexibilidade do receptor e ligante. O processo de complexação dessas moléculas sofre diversas mudanças, pois elas se moldam conforme as ligações químicas acontecem para obter uma maior estabilidade energética. A dificuldade do problema varia com a forma em que receptor e ligante são tratados, isto é, se ambas moléculas forem mantidas rígidas, se somente o receptor permanece rígido, ou se ambos são considerados flexíveis. A inclusão da flexibilidade é importante pois simula de maneira mais realística o complexo molecular no ambiente, no entanto isso aumenta em muito a complexidade do problema, pois aumenta o número de átomos a serem considerados no processo de otimização e todo o cálculo de suas interações para medir a energia de ligação das moléculas, tornando o processo quase inviável (SADJAD; ZSOLDOS, 2011).

No atracamento rígido são consideradas variações apenas na translação e rotação da estrutura do ligante como um corpo rígido. O problema tem sua complexidade diminuída pois a conformação interna do ligante se mantém a mesma durante todo o processo e isso diminui o número de possibilidades a serem testadas. No caso de atracamento flexível é levada em conta essa conformação interna do ligante, onde os graus de liberdade dos ângulos diedrais do ligante são considerados. Atualmente, a grande maioria das ferramentas de AM incluem além da liberdade translacional e rotacional, as rotações diedrais que modificam a conformação da estrutura dos ligantes.

Nas duas abordagens mencionadas, o receptor permanece rígido, isto é, mantido

Figura 2.3: Representação dos tipos de Atracamento Molecular: (1) atracamento rígido; (2) atracamento com ligante flexível; e (3) atracamento com receptor e ligante flexíveis.



Fonte: Do Autor.

fixo na posição determinada por dados experimentais. Entretanto, isso não reflete a realidade biológica, na qual o receptor assim como o ligante, passa por um processo de mudança conformacional. A flexibilidade no receptor é um terceiro tipo de abordagem para AM, porém aumenta ainda mais a complexidade do problema considerando a quantidade de átomos presentes nas estruturas. A Figura 2.3 ilustra os três tipos de atracamentos descritos. Trabalhar com o receptor flexível é importante para compreender os efeitos biológicos que os ligantes exercem sobre o mesmo, assim como suas posições no sítio de ligação, suas orientações, cinética de ligação, metabolismo e transporte (TEAGUE, 2003). Existem estudos que conseguem a inclusão de flexibilidade parcial no receptor (JONES; WILLETT, 1995; CLAUSSEN; BUNING CM.; LENGAUER, 2001; VERDONK et al., 2003), mas isso ainda continua sendo um grande desafio para ferramentas de atracamento.

Outros trabalhos incluem a flexibilidade somente no sítio de ligação ou em algu-

mas partes do receptor, as quais, geralmente se referem às cadeias laterais da molécula. Essa abordagem não representa de forma exata as interações, mas é mais aproximada do modelo biológico. Os estudos que utilizam essa metodologia (TROTT; OLSON, 2010; WEI et al., 2004; FISCHER et al., 2014), mostram resultados bem próximos, em termos estruturais, aos dados experimentais. Outros trabalhos (LEACH, 1994; JACKSON; GABB; STERNBERG, 1998) incluem bibliotecas de rotâmeros (DUNBRACK, 2002), que são valores preferenciais dos ângulos das cadeias laterais de resíduo de aminoácidos. Nessa abordagem o algoritmo realiza uma busca exaustiva sobre todas conformações preferenciais de cada aminoácido.

2.7 Resumo do capítulo

A criação de um novo fármaco demanda um alto custo e tempo para ser realizado em bancada. Esse obstáculo estimula o desenvolvimento de métodos computacionais que possam facilitar e auxiliar nesse processo. Ferramentas de Atracamento Molecular e Triagem Virtual, baseadas em abordagens computacionais, já são aplicadas no processo de descoberta de novos candidatos potenciais a fármacos. No entanto, muitos desafios ainda são enfrentados dada a complexidade do problema, considerando que ambas moléculas possam ser flexíveis no processo. Mas mesmo num cenário em que somente o ligante tem sua flexibilidade considerada, há uma carência de ferramentas que possam predizer com acurácia a complexação mais estável de um composto qualquer. Portanto, existem oportunidades de estudar e aplicar diferentes técnicas para solucionar o problema.

3 TRABALHOS RELACIONADOS

Técnicas em Atracamento Molecular são classificadas em abordagens geométricas ou energéticas. Algoritmos que exploram a geometria dos complexos avaliam alinhamento estrutural entre receptores e ligantes conhecidos, examinam as ligações químicas e avaliam seus efeitos estereoquímicos (KUNTZ et al., 1982). O grande número de ângulos diedrais tornam tais modelos mais simples, mas também menos acurados quando comparados aos métodos de avaliação energética. Em métodos que avaliam energia é realizado o cálculo de energia livre dos complexos, onde diferentes conformações são testadas e o objetivo é encontrar o menor valor potencial energético das estruturas.

Abordagens em AM baseadas em avaliação de energia de ligação utilizam diferentes representações das estruturas 3D e técnicas de otimização. Os complexos passam por uma modelagem que retrata suas interações físico-químicas de modo que seja possível representar as estruturas em uma forma computacional. A busca pelo menor valor de energia equivale a testar diferentes combinações conformacionais receptor-ligante avaliando suas interações atômicas. Se uma função de energia de ligação for suficientemente acurada, a conformação nativa do complexo coincidirá com o mínimo global de energia (COMBS et al., 2013). No entanto, estas funções de energia ainda não são tão acuradas, apresentando diversos desafios a serem solucionados, assim como os algoritmos de otimização que as utilizam (HUANG; ZOU, 2010).

3.1 Representação das Estruturas

Moléculas receptoras e ligantes podem ser representadas computacionalmente de três maneiras: por superfície, por grade ou por átomos (HALPERIN et al., 2002). Na representação por superfície, muito utilizada em ferramentas de AM proteína-proteína, é possível estudar as características das estruturas com base em sua contribuição atômica. Os métodos avaliam pontos da superfície minimizando ângulos entre as superfícies de moléculas opostas (ANDREI et al., 2012). O uso de grades de energia potencial, inicialmente proposto por Goodford (GOODFORD, 1985), pressupõe o armazenamento de dados sobre as contribuições energéticas dos pontos de uma grade, onde os mesmos são lidos durante a avaliação de energia do ligante, de modo a tornar a busca mais rápida não precisando recalcular todos os pontos do *grid*. As informações guardadas são, geralmente, dois tipos de potenciais: eletrostático e *van der Waals* (SCHNEIDER; BöHM,

2002). Por fim, a representação por átomos é utilizada juntamente com uma função de energia potencial no processo de avaliação de aptidão. No entanto, dada a quantidade de átomos presentes em um complexo e o número de interações de pares de átomos, esta abordagem pode ser computacionalmente custosa.

O ligante é geralmente representado pelas coordenadas cartesianas dos átomos e suas ligações químicas. Algumas destas ligações covalentes possuem um ângulo diedral associado. Estes ângulos definem a conformação da estrutura, logo, sua variação reflete na flexibilidade do ligante (SIMONSEN et al., 2013). Variações aleatórias na translação e rotação da estrutura completa e nos ângulos diedrais são feitos dentro do sítio de ligação com o objetivo de encontrar a posição e conformação da estrutura que apresente a menor energia de ligação das moléculas.

Considerando que a estrutura tridimensional é representada pelas coordenadas x, y, z de cada átomo, a operação de translação se dá ao adicionar os valores Δx, Δy e Δz às suas respectivas coordenadas. A Equação 3.1 representa esta operação que translada a molécula ligante como uma estrutura rígida. As variações podem ocorrer em uma, duas ou nas três coordenadas, o que aumenta o número de variações da operação. Os valores associados a estas perturbações devem respeitar o espaço de busca, o qual, deve englobar o sítio de ligação do receptor.

$$(x, y, z) \rightarrow (x + \Delta x, y + \Delta y, z + \Delta z) \tag{3.1}$$

A rotação da molécula ligante é realizada também em cada coordenada da estrutura e é descrita na Equação 3.2. A operação pode ser realizada a partir de quatro valores, três representando um vetor de referência e o quarto indicando um ângulo θ. O vetor de referência é definido a partir das coordenadas de um átomo do ligante, sobre o qual, a estrutura será rotacionada, isto é, esse átomo permanece fixo e a estrutura rotaciona em função desse ponto. A escolha desse ponto geralmente é feita com base no centro de massa da molécula, mas pode ser qualquer átomo da estrutura, pois a operação será feita sobre ele. A partir das coordenadas desse átomo é definido um vetor unitário $u = (u_x, u_y, u_z)$. Com este vetor é possível calcular o quadrivetor $Q = (q_0, q_1, q_2, q_3)$ que irá definir as operações geométricas de rotação. Essas operações são arranjadas e formam a matriz de rotação R, a qual, é multiplicada por cada ponto da molécula, gerando assim

novas coordenadas com os átomos rotacionados em θ radianos.

$$q_0 = \cos\left(\frac{\Theta}{2}\right) \quad q_1 = u_x(1 - q_0 q_0)^{\frac{1}{2}} \quad q_2 = u_y(1 - q_0 q_0)^{\frac{1}{2}} \quad q_3 = u_z(1 - q_0 q_0)^{\frac{1}{2}}$$

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (3.2)$$

Com este equacionamento são minimizadas as operações trigonométricas, já que é necessário fornecer apenas as coordenadas do vetor de referência e o valor do ângulo de rotação (MAGALHAES, 2006). A mesma equação pode ser utilizada para rotacionar os ângulos diedrais, porém a definição do vetor unitário é feita com base na ligação covalente. Assim, os pontos que formam o vetor são as coordenadas dos átomos envolvidos na ligação química.

No atracamento rígido são realizadas somente as operações de translação e de rotação na estrutura como um corpo rígido, e não são consideradas as rotações internas do ligante. Já no atracamento flexível são considerados os diedros. Também, algumas abordagens incluem a flexibilidade parcial ou total da estrutura receptora (COZZINI et al., 2008). Métodos de dinâmica molecular, Monte Carlo e algoritmos evolutivos são utilizados, combinados com bibliotecas de rotâmeros ou grades de energia. No entanto, a inclusão dessa flexibilidade parcial/total aumenta muito a complexidade do problema, exigindo a aplicação de algoritmos de busca mais robustos.

3.2 Métodos de Busca

Algoritmos de busca são aplicados no problema de Atracamento Molecular com o intuito de encontrar o mínimo global de uma função de avaliação de energia de ligação das moléculas. Estes algoritmos podem ser classificados conforme a metodologia aplicada para explorar a flexibilidade do ligante: sistemática, determinística e estocástica (BROOIJMANS, 2003). Na busca sistemática, os métodos consideram todos os graus de liberdade da molécula por meio de um conjunto de valores explorados de modo combinatorial. O ligante é dividido em fragmentos rígidos e flexíveis incorporados no sítio de ligação conectando partes da molécula até obter a estrutura completa. Esse tipo de técnica é também conhecida como construção incremental ou baseada em fragmentos. O

processo se inicia ao adicionar um fragmento núcleo na região alvo do receptor, após isso, para cada novo fragmento, uma busca pela melhor conformação da estrutura é realizada, considerando um conjunto de valores para os graus de liberdade do ligante.

Métodos determinísticos consideram o estado corrente do sistema para determinar as modificações a serem feitas para o próximo estado. O resultado final é muito dependente do estado inicial da estrutura porque uma mesma configuração inicial do sistema e de parâmetros, levam ao mesmo estado final (GUEDES; MAGALHÃES; DARDENNE, 2013). Esse tipo de algoritmo geralmente é utilizado quando existe uma relação entre as características de uma possível solução e sua utilidade em um problema (WEISE, 2009). Em atracamento molecular, esse tipo de algoritmo é utilizado para otimização de energia, bem como métodos de simulação por dinâmica molecular.

Em métodos estocásticos, os graus de liberdade (translação, rotação, e diedros) da molécula são aleatoriamente modificados a cada iteração, gerando uma diversidade de soluções. No problema de AM, algoritmos evolutivos são métodos estocásticos aplicados para encontrar a menor energia de ligação proteína-ligante, exemplos são Algoritmos Genéticos (GA - *Genetic Algorithms*) (LóPEZ-CAMACHO et al., 2013; MAGALHAES; BARBOSA; DARDENNE, 2004), Evolução Diferencial (DE - *Differential Evolution*) (KUKKONEN; LAMPINEN, 2005), Algoritmos Memético (MA - *Memetic Algorithms*) (ROSIN et al., 1997; RUIZ-TAGLE et al., 2017), Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization*) (NEBRO et al., 2009; JANSON; MERKLE; MIDDENDORF, 2008), Arrefecimento Simulado (SA - *Simulated Annealing*) (GOODSELL; OLSON, 1990), Algoritmo de Colônia de Formigas (ACO - *Ant Colony Optimization*) (MEIER et al., 2010), entre outros. O princípio básico dos algoritmos evolutivos é baseado em implementações de seleção, recombinação, mutação e avaliação de aptidão de um conjunto de soluções para um determinado problema. Cada uma destas operações é realizada com o objetivo de diversificar a população e impedir uma convergência prematura.

Esse tipo de algoritmo tem muitas vantagens em relação a outros métodos de otimização lineares, pois trabalham com um conjunto de soluções em um espaço de busca. A partir do valor da função objetivo estes algoritmos conseguem lidar com problema, o qual, pode ser multimodal, apresentando diversos mínimos locais, que podem dificultar o processo de busca ao estagnar em soluções sub-ótimas, além de poder apresentar descontinuidades no espaço de busca, e ruídos nos valores reportados ou problemas de mudanças dinâmicas (DEVI S. SIVA SATHYA, 2015). Diferentes estruturas de dados

podem representar as soluções, combinadas ou não, as quais, são englobadas por um espaço de busca complexo definido (WEISE, 2009). Algoritmos genéticos, em especial, apresentam vantagens pois permitem simplificar a formulação e solução de problemas de otimização. É possível adotar regras de transição probabilísticas, utilizar funções não diferenciáveis, além de não requerer informações adicionais sobre a função a otimizar. Por serem adaptáveis a qualquer problema, é possível combinar a sua aplicação com outras técnicas.

3.3 Metaheurísticas

De acordo com a complexidade de um dado problema estudado, o mesmo pode ser solucionado por métodos exatos (determinísticos) ou por soluções aproximadas. Os métodos exatos podem atingir soluções ótimas, porém quando aplicados em problemas que pertencem à classe de complexidade computacional NP-difícil, eles se tornam inviáveis por apresentarem um tempo de execução não-polinomial (COOK, 1983). Já os métodos aproximados ou heurísticas são capazes de obter boas soluções em um tempo de execução aceitável quando aplicados a problemas reais, apesar de não garantirem que a melhor solução será encontrada (TALBI, 2009).

Levando em conta que muitos dos problemas de otimização existentes, independente do domínio de aplicação, não podem ser resolvidos de maneira ótima e em tempo hábil por conta da alta complexidade que os espaços de busca apresentam, abordagens baseadas em metaheurísticas são utilizadas (DREO et al., 2006; BOUSSAïD; LEPAGNOT; SIARRY, 2013; LUKE, 2013). Conforme Boussaïd et al. (BOUSSAïD; LEPAGNOT; SIARRY, 2013), a maioria das metaheurísticas compartilham algumas características: (i) são inspiradas por fenômenos da natureza, baseados em princípios da biologia, física ou etologia; (ii) incorporam estruturas algorítmicas estocásticas, explorando o conceito de aleatoriedade; (iii) independem do gradiente ou matriz hessiana das funções objetivo; e (iv) apresentam muitos parâmetros que precisam ser ajustados ao problema de aplicação.

A simples aplicação de métodos canônicos, especialmente em problemas de Bioinformática Estrutural, não é suficiente para se obter bons resultados. O grande número de variáveis a serem otimizadas nesse tipo de problema é o principal motivo. Portanto, é importante a incorporação de conhecimento prévio sobre o problema e a exploração de características específicas para aumentar a eficácia dos métodos, diminuindo a complexidade por meio da restrição do espaço de soluções.

Um exemplo de metaheurística é o Algoritmo Memético, que é um algoritmo evolutivo composto de técnicas de busca global e local (MOSCATO, 1989). Algoritmos de busca global podem explorar o espaço de busca como um todo, enquanto que a exploração da vizinhança de uma solução é atribuída a um método de busca local, podendo obter boa precisão (KRASNOGOR; SMITH, 2005). Esta técnica é inspirada nos princípios de Darwin sobre evolução natural e na definição de Dawkin sobre *meme*, o qual, representa uma unidade de evolução cultural que pode realizar refinamentos.

Outros estudos têm aplicado a hibridização com adaptação em MAs. O ajuste de parâmetros e operadores de busca têm se mostrado uma promissora área de pesquisa em algoritmos evolutivos. Esta abordagem pode se autoadaptar para um determinado problema sem conhecimento prévio do mesmo, utilizando assim dados adquiridos para adaptar mecanismos (como por exemplo, o *crossover* no GA), e parâmetros do algoritmo durante o progresso da busca. O desafio em desenvolver um algoritmo memético robusto e eficiente tem algumas questões a serem consideradas: (i) onde e quando a busca local deve ser aplicada; (ii) quais indivíduos devem ser melhorados e como devem ser escolhidos; (iii) o esforço computacional empregado em cada chamada do método de LS; (iv) o equilíbrio entre a taxa de aplicação da busca global e local. Diversos MAs foram desenvolvidos sobre estas questões (JIN; ZHIHUA; WENYIN, 2014; KRASNOGOR, 2002; JAKOB, 2010).

Em Krasnogor et al. (KRASNOGOR; SMITH, 2001) foi desenvolvido um algoritmo memético adaptativo combinando um GA com duas estratégias de busca local: *First Improvement* (FI) e *Best Improvement* (BI) (PAPADIMITRIOU; STEIGLITZ, 1998; AARTS; LENSTRA, 1997). Neste método os autores desenvolveram um simples mas eficiente mecanismo de herança (SIM - *Simple Inheritance Mechanism*) para um problema de busca combinatorial. A estratégia consiste em codificar o material memético na representação dos indivíduos, onde esse material indica qual LS deve ser aplicada. Durante o processo evolutivo, o *crossover* é responsável por escolher qual método deve ser atribuído para a solução filha, de acordo com o melhor *fitness* entre os pais (em caso de empate, sorteio). Os resultados mostraram que este simples esquema é capaz de adaptar o comportamento dos indivíduos na busca independente do problema. Posteriormente, esse método foi aplicado para o problema de Predição de Estruturas de Proteínas (PSP - *Protein Structure Prediction*) (KRASNOGOR, 2002).

No trabalho de Jakob et al. (JAKOB, 2010) foi proposto um algoritmo memético baseado em GA e nos algoritmos Rosenbrock (ROSENBROCK, 1960) e Complex (BOX,

1965) como LS. A proposta é uma abordagem auto-adaptativa para escalar métodos de busca local, frequência e intensidade da busca de acordo com uma função de probabilidade. A ideia é que no início do processo todos os métodos tenham chances iguais de serem selecionados. Durante a evolução, as probabilidades de aplicação de cada algoritmo são atualizadas de acordo com o ganho relativo de *fitness* que cada operador proporciona, levando em consideração a quantidade de avaliações da função objetivo necessárias.

Em Jin et al. (JIN; ZHIHUA; WENYIN, 2014) é proposto um MA adaptativo chamado GADE-DHC que combina algoritmo genético (HOLLAND, 1992; MITCHELL, 1998) e evolução diferencial (STORN; PRICE, 1997) como buscas globais e *Hill-climbing* como LS. Os autores defendem uma estratégia para balancear a intensidade de aplicação das buscas global e local, bem como a taxa de utilização entre GA e DE. Para isso, é aplicada uma função baseada em pesos para medir a contribuição de cada algoritmo de acordo com a sua melhora sobre os indivíduos da população e o estágio da evolução. Os testes da abordagem foram aplicados em um *benchmark* composto por 21 funções, e os resultados se mostraram altamente competitivos comparados com outros algoritmos.

Uma outra abordagem conhecida como Algoritmo Memético Multimeme ou Algoritmo Multimemético (MMA - *Multimeme Memetic Algorithm*) originalmente proposta por Krasnogor e Smith (KRASNOGOR; SMITH, 2001). Uma abordagem memética é composta por um algoritmo de busca global e outro algoritmo de busca local, já no MMA é empregado um conjunto de algoritmos de buscas locais. A ideia desta abordagem é adaptativamente escolher deste conjunto um operador para usar em diferentes instâncias/fases da busca ou indivíduos na população. Nesta abordagem, o indivíduo é representado pelo seu material genético e memético, onde este último especifica qual *meme* será usado para realizar a busca local. Dessa forma, cada solução codifica as variáveis a serem otimizadas no problema, e o operador de busca a ser aplicado nessa solução. Tal operador pode especificar tanto um método de busca local a ser utilizado, quanto qualquer outro parâmetro da mesma, como por exemplo, qual gene sofrerá modificações, a intensidade de cada modificação, entre outros.

No trabalho de Domínguez-Isidro et al. (DOMÍNGUEZ-ISIDRO; MEZURA-MONTES, 2018) é proposto uma coordenação adaptativa de buscas locais, baseada em um esquema de custo-benefício, para um multimeme baseado no algoritmo DE. O mecanismo de coordenação é baseado numa equação que mensura os custos de exploração das buscas locais e o benefício que elas proporcionam. Os algoritmos implementados como LS foram *Hill-climbing* (PAPADIMITRIOU; STEIGLITZ, 1998; AARTS; LENSTRA, 1997),

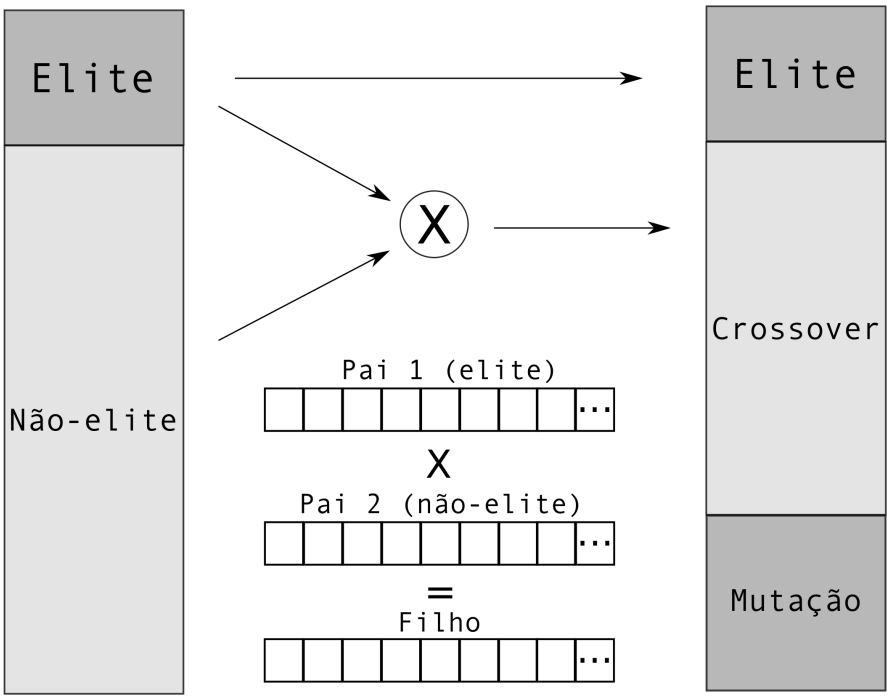
Hooke-Jeeves (HOOKE; JEEVES, 1961) e *Nelder-Mead* (NELDER; MEAD, 1965). O método é testado em um *benchmark* de 36 problemas bem conhecidos, e os resultados numéricos mostram que a coordenação do conjunto de buscas locais se mostra adequada dentro de um esquema memético.

3.3.1 BRKGA

O Algoritmo Genético de Chaves Aleatórias Viciadas (BRKGA - *Biased Random Key Genetic Algorithm*) proposto por Gonçalves e Resende (GONÇALVES; ALMEIDA, 2002; GONÇALVES; RESENDE, 2011) combina conceitos de GA com um esquema de codificação/decodificação que o faz ser aplicável a qualquer problema. O BRKGA utiliza um esquema diferente do GA para codificação do problema através de um vetor de valores reais dentro do intervalo [0, 1]. Essa estratégia de normalização o torna independente da aplicação e, quando necessário, o processo de decodificação é aplicado na solução encontrada para trazê-la de volta ao domínio do problema (BEAN, 1994). A população no BRKGA é organizada em castas de acordo com o valor de *fitness* de seus indivíduos. Após a geração da população inicial, os valores dados pela função objetivo indicam a ordem dos indivíduos dentro de dois grupos, elite e não-elite. O processo de construção da nova população ocorre, primeiramente, por copiar os cromossomos (p_e) que formam a casta elite. Na sequência é realizado o processo de *crossover*, o qual, seleciona um indivíduo de cada casta para gerar novas soluções que irão fazer parte da próxima população. Por fim, a mutação ocorre por meio da geração aleatória de novos indivíduos (p_m). A Figura 3.1 descreve o esquema de castas.

No processo de recombinação do BRKGA, que também difere do GA canônico, são selecionados um indivíduo de cada casta. O primeiro é selecionado do grupo de elite, e o segundo, obrigatoriamente, do restante da população, ambos de forma aleatória. A seleção dos genes para formar a nova solução é feita segundo um fator de probabilidade. Para cada alelo do vetor elite é definido um valor aleatório entre 0 e 1, caso seja menor ou igual a uma probabilidade ρ_e, o valor desse alelo é transmitido para o filho, caso contrário, o gene do segundo pai (não-elite) é repassado. O valor da probabilidade é sugerido pelos autores (GONÇALVES; RESENDE, 2011) para ser entre 0.5 e 0.8, dessa forma o gene do indivíduo de elite tem maior chance de ser selecionado pois é uma solução de melhor *fitness*. O fator que contribui para essa seleção é o cruzamento parametrizado uniforme (*Parametrized Uniform Crossover*) (SPEARS; JONG, 1991) incorporado no BRKGA. A

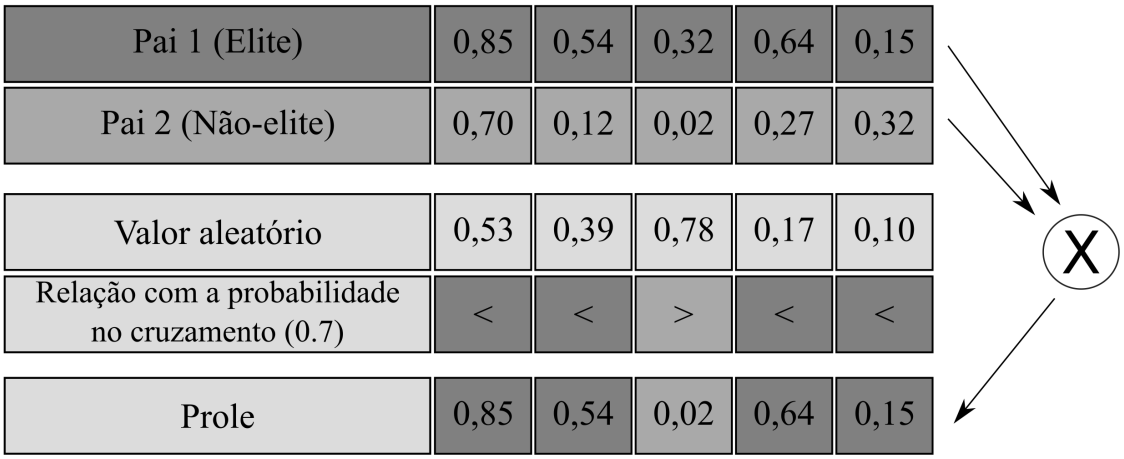
Figura 3.1: Esquema da população do BRKGA e o seu processo de atualização por meio da cópia do grupo elite, e as operações de *crossover* e mutação.



Fonte: Adaptado de Leonhart e Dorn (2019).

Figura 3.2 apresenta um exemplo no qual, a probabilidade de seleção é igual a 0.7. Os indivíduos possuem 5 genes, para os quais são sorteados valores aleatórios e realizada a transferência dos genes para o indivíduo filho. No exemplo dado, os genes 1, 2, 4 e 5 do indivíduo de elite são copiados para a prole.

Figura 3.2: Recombinação parametrizada uniforme no BRKGA.



Fonte: Do Autor.

3.3.2 Algoritmos de Busca Local

A busca local (LS - *Local Search*) é um método heurístico que se move de uma solução para outra, num espaço de candidatos, realizando mudanças locais até atingir um ótimo local ou algum outro critério de parada. Entretanto, este movimento somente é possível se uma relação de vizinhança for definida no espaço de busca das soluções. A geração da vizinhança é feita considerando três importantes aspectos: (i) a ordem de visitação dos genes; (ii) o raio de perturbação, que reflete em quanto cada gene deve ser modificado; (iii) a direção da busca, que indica se o valor de raio deve ser adicionado ou subtraído do valor codificado no indivíduo.

Um exemplo de técnica de LS é o *Hill Climbing algorithm* (HC) (PAPADIMITRIOU; STEIGLITZ, 1998; AARTS; LENSTRA, 1997), também conhecido como melhoria iterativa ou de descida, que é uma antiga e simples metaheurística. A técnica consiste em iniciar a partir de uma dada solução inicial e, a cada iteração, o método troca a solução atual pela melhor solução vizinha encontrada que melhore o valor da função objetivo. O processo de busca se encerra quando todas as soluções vizinhas são piores do que a solução corrente, indicando que um ótimo local foi encontrado.

Alguns problemas têm representações com grandes vizinhanças, então para acelerar a busca é necessário que se adote uma estratégia para restringir o número de soluções candidatas para um subconjunto do espaço de busca. O algoritmo HC é um tipo de busca local monótona pois permite somente modificações que melhorem o valor da função objetivo. Assim, existem variações do método de acordo com a ordem na qual as soluções vizinhas são geradas (determinística ou estocástica), e a estratégia de seleção da solução. A seguir são descritas três variantes do HC, que também são conhecidas como regras de pivoteamento para selecionar o melhor vizinho:

- *Best improvement* (BI): é uma estratégia que avalia a vizinhança inteira de forma completa e determinística. Assim, a exploração das soluções candidatas é exaustiva porque explora todos os possíveis movimentos a partir da solução atual. Esta abordagem pode consumir bastante tempo computacional para grandes vizinhanças, mas garante a seleção do melhor candidato a cada iteração.
- *First Improvement* (FI): esta estratégia escolhe o primeiro candidato que tenha melhor *fitness* do que a solução atual. Portanto, um candidato melhor avaliado é imediatamente selecionado para substituir a melhor solução conhecida. A avaliação da vizinhança é realizada de maneira determinística seguindo uma ordem pré-definida

para geração das soluções, isto é, a ordem de modificação dos genes é sorteada aleatoriamente no início do processo. Assim, apesar do método visitar parte dos candidatos, é uma abordagem mais rápida do que o BI. Somente no pior caso é que uma geração completa da vizinhança é feita, o que consequentemente significa que nenhum candidato melhor existe.

- *Stochastic Hill Descent* (SHD): esta variante é quase idêntica ao método FI. A estratégia de seleção da melhor solução é a mesma, a diferença está na ordem de geração dos vizinhos, que é realizada de forma aleatória a cada iteração do processo de busca. Assim, a estratégia evita avaliar os candidatos sempre da mesma forma, garantindo que todas as regiões da vizinhança da solução tenham condições iguais de serem exploradas.

Outro algoritmo que comumente é utilizado como busca local é o *Simulated Annealing* (SA) (KIRKPATRICK; GELATT; VECCHI, 1983; ČERNÝ, 1985), um método estocástico que permite, em algumas condições, aceitar soluções piores do que a atual e se fundamenta numa analogia com a termodinâmica (KIRKPATRICK; GELATT; VECCHI, 1983; ČERNÝ, 1985). A técnica é uma metáfora do processo térmico conhecido como *annealing* ou recozimento, utilizado em metalurgia para obtenção de estados de baixa energia em sólidos. Assim, quando o metal é aquecido em altas temperaturas, seus átomos fazem movimentos desordenados de grandes amplitudes. À medida que o metal é resfriado progressivamente, os átomos reduzem seus movimentos e tendem a se estabilizar em torno de uma região de mínima energia. Esse processo permite aos metalúrgicos modelar e corrigir defeitos dos materiais.

Analogamente, no SA a solução corrente é substituída por outra vizinha de acordo com uma função objetivo e uma variável T, de temperatura. Quanto maior for o valor de T mais aleatória será a geração da solução candidata, e à medida que for diminuindo, o algoritmo tende a convergir para um ótimo local. A ideia é evitar ficar preso em subótimos locais e ter uma convergência mais lenta do processo de busca. A partir de uma dada solução, o SA gera uma solução vizinha aleatória a cada iteração do processo. Esse candidato R é sempre aceito se for melhor avaliado pela função objetivo do que a solução atual S. Caso contrário, se o *fitness* for pior, existe uma regra que permite aceitar a solução de acordo com uma probabilidade, calculada conforme a Equação 3.3 mostra:

$$\begin{aligned} \Delta E &= \frac{Qualidade(R) - Qualidade(S)}{T} \\ P(T, R, S) &= e^{\Delta E} \end{aligned} \tag{3.3}$$

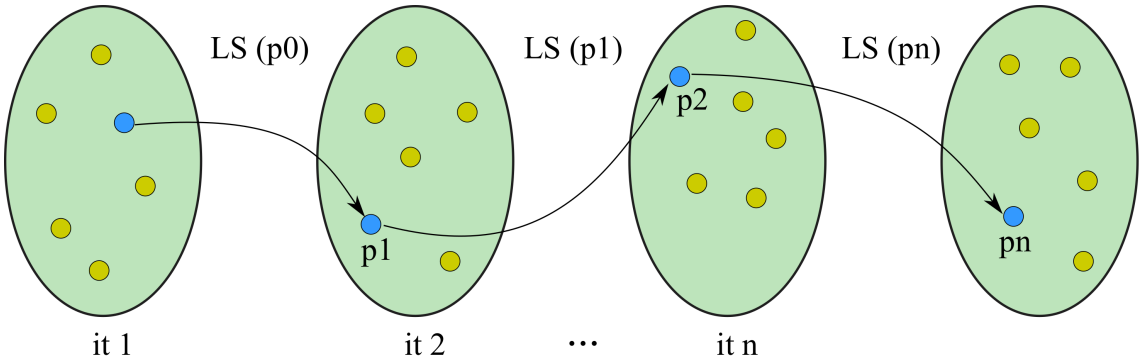
onde $T >= 0$ e ΔE representam a diferença de qualidade entre as soluções R e S. De acordo com a temperatura atual e a degradação da função objetivo, a solução candidata tem uma probabilidade de ser aceita ou não. Se R apresentar um valor de *fitness* muito diferente do valor apresentado por S, a fração será maior e então a probabilidade de aceitação é quase nula. Se R for próximo de S, então a probabilidade é quase igual a 1, aumentando as chances de R ser selecionada. O fator de temperatura também tem um importante papel, se T for próximo de 0, então a probabilidade é também quase 0. Por outro lado, se T for um valor alto, a probabilidade de aceitação será próxima de 1. Assim, a ideia é que no início o método funcione como uma caminhada aleatória no espaço de busca, aceitando soluções independentemente de quão boas são. Quando T decresce, a probabilidade também diminui e então o processo age de maneira semelhante ao algoritmo HC.

3.3.3 Encadeamento de Buscas Locais

Uma recente *feature* no âmbito de buscas locais é o conceito de encadeamento (*LS chains*) proposto por Molina et al. (MOLINA; LOZANO; HERRERA, 2009). A alta dimensionalidade dos problemas de otimização têm aumentado, em particular nos algoritmos meméticos, o espaço de busca ao redor de cada solução, o que exige que métodos de busca local sejam aplicados com alta intensidade. Assim, a ideia é que, dependendo das características do problema, diferentes aplicações de LS sejam encadeadas compartilhando as informações acerca daquela região de busca. Um método de busca local pode não explorar toda a vizinhança de uma solução, então o estágio final de uma chamada de LS pode se tornar o ponto de partida de uma subsequente aplicação de busca local nesse indivíduo. Esta estratégia permite que os operadores de busca local sejam estendidos em algumas áreas promissoras, e evitam que diferentes algoritmos avaliem candidatos já visitados por meio do encadeamento das buscas. A Figura 3.3 exibe o encadeamento de uma busca local formado por um algoritmo de LS e um conjunto de parâmetros p.

A ideia é que a cada iteração em que a busca local for aplicada seja utilizada a parametrização final da aplicação anterior do algoritmo naquela mesma solução. De acordo com Molina et al. (MOLINA et al., 2011), existem alguns aspectos importantes a serem considerados na gestão de LS encadeadas. A intensidade de busca deve ser fixa, para garantir que cada busca local tenha o mesmo esforço computacional aplicado. Outra questão é salvar as configurações (ordem de visitação, por exemplo) que guiam a busca

Figura 3.3: Exemplo de encadeamento de busca local. p_{i+1} é o parâmetro final alcançado pelo algoritmo de LS quando iniciado com uma parametrização p_i. A estratégia inicial é definida por p_0.



Fonte: Adaptado de Molina et al. (2009).

e o estado corrente ao final da aplicação da LS. Fazendo uma relação com os algoritmos descritos na subseção anterior, nas variantes do HC o último vizinho gerado é salvo, e especificamente, nas abordagens FI e SHD também são salvas a ordem de visitação dos genes. No caso do SA, o encadeamento da busca só pode ser feito ao compartilhar o status final da aplicação, já a ordem de visitação da vizinhança não, pois ela é feita de forma aleatória a cada iteração do algoritmo.

3.4 metaheurísticas Aplicadas ao problema AM

Inicialmente, a maioria das técnicas de Atracamento Molecular é aplicada no atracamento rígido de estruturas ligantes. Este problema, mais simples, consiste em um processo de busca pela melhor ligação do composto realizando apenas operações de translação nas moléculas. Os compostos utilizados já apresentam a melhor ligação e assim permitem verificar a eficácia do algoritmo em predizer a conformação das moléculas. O atracamento rígido utilizado nessa etapa é um meio de simplificar o problema e garantir que o algoritmo consiga resolvê-lo de maneira ótima. A flexibilidade do ligante é então incluída ao adicionar seus graus de liberdade gradualmente. Devido a enorme complexidade e multidimensionalidade apresentadas pelo problema de atracamento molecular flexível e à carência de algoritmos eficazes, muitos métodos baseados em metaheurísticas têm sido aplicados para tentar obter soluções ótimas para estes problemas. No trabalho de Camacho et al. (LóPEZ-CAMACHO et al., 2014) é feito um estudo comparativo mostrando as principais técnicas utilizadas em AM: Algoritmos Genéticos (LóPEZ-CAMACHO et al., 2013; MAGALHAES; BARBOSA; DARDENNE, 2004), Evolução Diferencial (KUK-

KONEN; LAMPINEN, 2005) e PSO (NEBRO et al., 2009; JANSON; MERKLE; MIDDENDORF, 2008).

Em Morris et al. (MORRIS et al., 2009) foi desenvolvida a ferramenta AutoDock4, uma das principais referências em *softwares* de Atracamento Molecular. O método desenvolvido é baseado em um Algoritmo Genético Lamarckiano (AGL) (WHITLEY; GORDON; MATHIAS, 1994) e utiliza uma função de energia semi-empírica. Testes iniciais foram feitos com 7 complexos com adição gradual de flexibilidade, e obtiveram resultados com RMSD abaixo de 1,14Å com uma média de 0,88Å (Ångströms - unidade de medida de comprimento que se relaciona com o metro: $1\text{Å} = 10^{-10}m$). Na etapa seguinte foram utilizadas 170 estruturas, das quais, 100 obtiveram valores de RMSD abaixo de 3,5Å. Importante destacar que foi utilizada uma grade de energia potencial na qual é adicionada a flexibilidade parcial da proteína, e uma função de energia própria.

Em Meier et al. (MEIER et al., 2010) foi desenvolvido o *framework* ParaDocks que implementa os algoritmos PSO (KENNEDY; EBERHART, 1995; SHI; EBERHART, 1998) e ACO (DORIGO; CARO, 1999). A ferramenta utiliza paralelamente uma Unidade de Processamento Gráfico (GPU) e uma Unidade Central de Processamento (CPU) para realizar a predição das conformações. São utilizadas no trabalho diferentes funções de energia em 13 complexos. Foram obtidos resultados de RMSD abaixo de 2Å para 73% das instâncias. A função de energia PMF04 (MUEGGE, 2006) se mostrou mais acurada entre as demais funções testadas: GOLD (VERDONK et al., 2003), BLEEP (MITCHELL et al., 1999), e DRUGSCORE (GOHLKE; HENDLICH; KLEBE, 2000). Assim, os resultados mostram a eficácia de metaheurísticas para o problema, além de fazer o comparativo entre as funções de energia.

No trabalho de López-Camacho et al. (LóPEZ-CAMACHO et al., 2014) são comparadas 3 metaheurísticas: PSO (KENNEDY; EBERHART, 1995; SHI; EBERHART, 1998), DE (STORN; PRICE, 1997) e GA (HOLLAND, 1992; MITCHELL, 1998), e avaliadas quanto à sua convergência. Os autores desenvolveram um *framework* que incorpora a avaliação de energia do AutoDock 4.2. Foram realizados testes com 83 estruturas de ligantes (HIV-protease) com diferentes tamanhos e flexibilidades, obtendo resultados com valores abaixo de 10Å em um total de 30 execuções de 1.500.000 avaliações de energia cada. O algoritmo evolução diferencial obteve uma convergência mais tardia mas com melhores resultados. O algoritmo genético teve rápida convergência, no entanto as soluções estagnaram após de 250.000 avaliações de energia.

Recentemente, no trabalho de Spieler et al. (SPIELER, 2016) foi proposto um al-

goritmo BRKGA combinado com um modelo de discretização em cubos para aplicação no problema de atracamento molecular. Na abordagem, é proposta uma discretização do espaço de busca em subcubos, onde as soluções são classificadas de acordo com suas coordenadas xyz. A ideia desse modelo de representação é manter a diversidade da população, pois pelo menos um indivíduo estará em cada região, isto é, em cada subcubo. Os testes foram realizados com 43 instâncias baseadas na proteína HIV-protease, e a função de energia utilizada foi a AutoDock Vina (TROTT; OLSON, 2010). O método considera o problema de atracamento como semi-rígido, pois os graus de liberdade do ligante são limitados a movimentos de $\frac{\pi}{4}\Theta$, ficando apenas translação e rotação da estrutura a serem otimizados. Os resultados obtidos foram comparados com os métodos DockThor (MAGALHÃES et al., 2014) e AutoDock Vina, e apresentaram ganhos, dada a condição mencionada acima, com valores de RMSD abaixo de 1.3Å para 88% dos casos de teste.

Métodos híbridos também foram aplicados em AM, como em Rosin et al. (ROSIN et al., 1997) onde foi implementado um GA hibridizado com o algoritmo Solis-Wets (SW) trabalhando como operador de busca local. A aplicação da busca local foi realizada em indivíduos aleatórios com uma frequência de 7%. A abordagem memética teve duas variações referentes à intensidade de aplicação do SW, as quais, foram comparadas com o GA e *Simulated Annealing* sem busca local. Os testes foram executados com a função de energia do AutoDock com 1.500.000 avaliações de energia sobre um conjunto de 6 instâncias de teste. Os resultados mostraram que a aplicação de busca local trouxe melhores resultados em comparação com abordagens não meméticas.

Em Tagle et al. (RUIZ-TAGLE et al., 2017) são implementadas três variações do SA empregadas como método de busca local em uma abordagem memética. As abordagens de LS se diferenciam no modo de exploração do espaço de busca, onde uma delas explora toda a área de busca, a segunda diminui o espaço de acordo com a queda de temperatura do *Simulated Annealing*, e a última realiza perturbações apenas na rotação do ligante e rotação de seus diedros. Os testes foram executados em um conjunto de 9 complexos com um total de 500.000 avaliações de energia, utilizando a função do AutoDock Vina. Os resultados mostraram que a terceira abordagem que realiza somente rotações obteve melhores valores de energia em 90% das instâncias, mas quando comparado o RMSD houve uma distribuição dos melhores valores entre todas abordagens. O trabalho também comparou os resultados com as ferramentas DockThor e AutoDock Vina, se mostrando promissora a ideia de realizar pequenas modificações para melhorar a qualidade das soluções em algoritmos evolutivos.

As técnicas de otimização baseadas em metaheurísticas variam em vários aspectos. Além do método em si, a sua parametrização, e a representação dos dados têm influência direta nas soluções obtidas. Neste trabalho são comparados os resultados com as ferramentas AutoDock Vina (TROTT; OLSON, 2010), DockThor (MAGALHÃES et al., 2014) e jMetal (DURILLO; NEBRO, 2011), os quais são apresentados a seguir.

3.4.1 AutoDock Vina

O AutoDock Vina (http://vina.scripps.edu/) (TROTT; OLSON, 2010) é uma ferramenta de Atracamento Molecular e Triagem Virtual que fornece uma função de energia para predição de conformações proteína-ligante. Ao utilizar *multithreading* para explorar o paralelismo de *hardware* com compartilhamento de memória, obtém uma rápida resposta na avaliação de energia, e sua alta performance tem tornado a ferramenta uma das mais citadas na área. Em seu desenvolvimento diversos algoritmos foram testados, entre eles GA, PSO e SA, até se chegar no algoritmo de busca local iterada (ILS - *Iterated local search algorithm*) (LOURENÇO; MARTIN; STÜTZLE, 2003). Esse algoritmo utiliza diversos passos que incluem operações de mutação e otimização local. A quantidade dos passos varia de acordo com a complexidade do problema.

O algoritmo de otimização mantém diversos mínimos locais relevantes encontrados e os combina em execuções distintas para realizar um processo de refinamento e agrupamento. O formato de arquivos é compatível com as demais versões do *software*, além de ferramentas auxiliares como o AutoDock Tools (MORRIS et al., 2009), utilizada na preparação e configuração dos parâmetros necessários para execução. Estes parâmetros incluem número de átomos das estruturas, número de ângulos de rotação do ligante, tamanho do espaço de busca e seu ponto central, entre outros. O AutoDock Vina calcula seu próprio *grid map* e realiza agrupamentos e ranqueamentos dos resultados. A realização dos testes incluiu 190 complexos, nos quais o receptor foi tratado como rígido e o ligante flexível com seus graus de liberdade variando de 0 a 32. Os resultados obtiveram valores de RMSD menores do que 2Å para 78% das instâncias de teste.

3.4.2 DockThor

O DockThor (https://dockthor.lncc.br/v2/) (MAGALHÃES et al., 2014) implementa o algoritmo *Steady State Genetic Algorithm* (SSGA) (WHITLEY; KAUTH, 1988) para o problema de AM. A representação dos dados é feita por meio de um vetor com valores referentes às translações, rotações e conformações da estrutura ligante. O processo se inicia com a geração aleatória de vetores de soluções. A partir da seleção de indivíduos e formação de uma recombinação, um método de torneio de seleção define se a nova solução é inserida na população. O método utiliza um critério de inserção baseado em similaridade e um torneio dinâmico para preservar boas soluções e aumentar a diversidade na população do GA. A exploração da diversidade de soluções é importante pois os novos indivíduos substituem soluções similares para aumentar a capacidade de busca do algoritmo. O critério de parada é definido como um número máximo de avaliações da função de energia.

Testes realizados nessa ferramenta incluíram o atracamento rígido de 5 complexos baseados na proteína HIV-protease, com variação de 12 a 20 na quantidade de ângulos diedrais. O método também foi testado em um *benchmark* composto por 34 complexos proteína-ligante de 18 famílias diferentes de proteínas. A performance foi comparada com outras ferramentas: GOLD (VERDONK et al., 2003), AutoDock Vina (TROTT; OLSON, 2010) e GLIDE (REPASKY; SHELLEY; FRIESNER,). Considerando um limiar de RMSD em 2,5Å, o DockThor obteve sucesso em 91,2% das instâncias enquanto que GOLD e Vina atingiram 82,4%, e GLIDE 97% de sucesso para o mesmo conjunto. Os resultados mostram que a técnica é eficaz, produzindo uma boa diversidade de soluções.

3.4.3 jMetal

O jMetal (http://jmetal.github.io/jMetal/) (LóPEZ-CAMACHO et al., 2013) é um *framework* de otimização, inicialmente escrito em Java e na sua mais recente versão em C++, que faz integração com o AutoDock para utilização da sua função de energia. A ferramenta incorpora algoritmos mono-objetivos como PSO, diversas variações de GA, e DE, além de técnicas multi-objetivos como *Non-dominated Sorting Genetic Algorithm-II* (NSGA), PSO com limitação de velocidade e *Multiobjective Evolutionary Algorithm Based on Decomposition* (MOEA/D). Todos algoritmos fazem uso da função de energia do AutoDock para avaliar suas soluções.

A ideia do *framework* é ser simples para o usuário, permitindo a inclusão de novos componentes e reutilização dos mesmos. As configurações para execução de um dado algoritmo são definidas em arquivo de parâmetros. A execução dos algoritmos evolutivos ocorre por gerar soluções e enviá-las para o AutoDock calcular seu *fitness*, retornando o valor de energia da mesma. Ao final do processo, a melhor ou as melhores soluções são retornadas e um arquivo de *log* com os resultados finais é gerado.

Os algoritmos mono-objetivos foram testados em duas etapas, na primeira com um *benchmark* composto por 7 complexos, e na segunda um conjunto de instâncias que envolvem flexibilidade nas cadeias laterais do resíduo arg-8 do receptor HIV-protease. As técnicas multi-objetivos também foram testadas, mas com 4 funções de problemas restritos. Em ambos, os resultados se mostraram competitivos, fazendo do *framework* uma ferramenta vantajosa.

3.5 Desafios em Atracamento Molecular

O problema de Atracamento Molecular enfrenta diversos desafios seja tanto na complexidade matemática quanto na capacidade de representação das interações físico-químicas entre as moléculas. No trabalho de Sousa et al. (SOUSA et al., 2013) são listados os principais desafios para AM: o tratamento da flexibilidade da proteína, a presença de moléculas de água e seus efeitos, e a entropia de ligação. Em Huang et al. (HUANG; ZOU, 2010) são debatidos os desafios de amostragem do ligante e funções de energia acuradas para o problema. Verdonk et al. (VERDONK et al., 2007) discutem a representação das estruturas moleculares, o papel de moléculas de água nas interações químicas, bem como métodos de busca e suas convergências.

A representação 3D de receptores e ligantes deve considerar os estados tautoméricos da molécula, em receptores os diversos estados de protonação, e em ligantes a sua mudança de conformação ao se ligar com as proteínas. Muitas ferramentas realizam o processo de atracamento com o ligante rígido e uma série de conformações pré-definidas, outras abordagens consideram aspectos geométricos durante testes iniciais com o intuito de reduzir o número de graus de liberdade para otimizar no algoritmo de busca. Em relação ao receptor, é possível definir manualmente os possíveis estados com base em análises do sítio de ligação e de conformações conhecidas do ligante. Quanto à flexibilidade da molécula, muitas ferramentas a consideram como rígida, outras já consideram certas conformações, isso porém, não considera a influência do ligante na mudança de conformação

da molécula receptora.

Moléculas de água medeiam interações entre receptores e ligantes, e muitas vezes são consideradas como parte da proteína em *Docking*. Nesse caso, o desafio é determinar quando essa molécula de água deve ser incluída ou removida, já que isso varia de acordo com o ligante. Diferentes opções são adotadas pelas ferramentas: eliminar moléculas de água, permitir aquelas que relativamente contribuem energeticamente para o sistema e, incluir todas as moléculas na avaliação de energia. A entropia contribui muito para o cálculo de energia, incluindo a redução de graus de liberdade rotacionais e translacionais do ligante, mudanças na forma da proteína e ligante, e no arranjo de camadas de água sobre os solutos. Muitas funções ignoram a entropia, por conta do alto custo computacional, com o objetivo de simplificar o cálculo de energia. Existem tentativas de inclusão de entropia em funções, mas a formulação desses termos ainda é uma questão aberta.

Outro desafio é encontrar uma função de energia acurada, pois muitas delas apresentam avaliações de aptidão inadequadas. A maioria das funções são multi-modais, apresentando valores de energia que na verdade não condizem com o RMSD, isto é, nem sempre um baixo valor de energia reflete em um RMSD baixo, e vice-versa. Funções presentes em diversas ferramentas conseguem reproduzir as ligações experimentais com uma precisão de 70-80% dos complexos, em termos de RMSD. Porém, a adição de flexibilidade no receptor e ligante, suas topologias e valência geométrica das moléculas, tornam a avaliação destas funções longe do ideal. Por outro lado, uma função rigorosa seria custosa computacionalmente, dada a análise de diversas ligações atômicas. Portanto, funções de energia apresentam simplificações para obter uma avaliação boa em um tempo curto.

3.6 Resumo do capítulo

Muitos trabalhos têm sido aplicados para o Atracamento Molecular, no entanto o problema ainda requer uma solução mais genérica. A chave para desenvolver uma técnica capaz de resolver de maneira ótima esse problema passa por utilizar bons algoritmos e funções de energia acuradas. Portanto, o desenvolvimento de um método requer a análise das abordagens já utilizadas, verificando seus prós e contras e buscar aspectos ainda não explorados. A representação das estruturas, categorias de métodos e os desafios encontrados foram analisados e ponderados para desenvolver uma metodologia capaz de contribuir para a resolução do problema.

4 MATERIAIS E MÉTODOS

O objetivo deste capítulo é descrever os algoritmos e estratégias adotados no desenvolvimento desta dissertação, assim como a metodologia e estruturação do método proposto. Inicialmente, é realizada a preparação e análise das estruturas escolhidas para os testes, e definida uma representação para as soluções no algoritmo. Como visto, o processo de atracamento molecular requer a escolha e uso de uma função de energia para avaliação das possíveis conformações proteína-ligante. Além disso, foi adotado um modelo de discretização do espaço de busca, implementado por Spieler et al. (SPIELER, 2016; LEONHART et al., 2018), com o intuito de melhor explorar as soluções para o problema. Por fim, o desenvolvimento do método em si ocorreu em duas etapas: (i) implementação de uma abordagem memética e, (ii) proposta e implementação de um modelo auto-adaptativo para coordenação dos operadores de busca local.

Na primeira etapa, foi desenvolvido um algoritmo baseado no BRKGA para exploração do espaço de busca. A partir dessa implementação, foram incorporados métodos de busca local e assim formado um algoritmo memético. Essa etapa serviu para testar e verificar se uma abordagem híbrida traz vantagens em relação a uma versão não-memética. Em decorrência dos resultados obtidos, na segunda etapa, foi proposto um modelo de auto-adaptação sobre o método memético. Nesse modelo, a ideia é que parâmetros relativos às buscas locais possam ser ajustáveis, durante a execução do algoritmo, de acordo com as características do problema. Por exemplo, no início do processo de busca, modificações de maior impacto na estrutura do ligante podem ser mais interessantes para exploração do sítio de ligação, enquanto que na parte final da execução, pequenas modificações podem ser melhores para o refinamento do *pose*. A coordenação destes parâmetros, que incluem o método de busca local a ser aplicado e o raio de perturbação adotado em cada um deles, trabalha de acordo com uma função de probabilidades, proposta neste trabalho, que avalia o custo-benefício de cada operador no processo de busca. Todos os procedimentos preliminares e necessários para a execução dos algoritmos, bem como suas implementações, são detalhados nas próximas seções.

4.1 Preparação e representação das estruturas moleculares

As estruturas utilizadas em todas as etapas de teste foram obtidas do PDB (*Protein Data Bank*) (BERMAN et al., 2000) e são detalhadas na Seção 5.2. Cada arquivo

adquirido passou por um processo de preparação para que se obtivesse uma representação mais fiel dos complexos, e consequentemente do problema biológico. Esse processo é composto pela remoção/adição de átomos, inclusão de cargas nos átomos, definição de ângulos diedrais de rotação (torção) do ligante, conversões de arquivos, para compatibilidade entre as ferramentas auxiliares, e definição do ambiente para execução dos algoritmos desenvolvidos.

Primeiramente, a ferramenta *Open Babel tools*(O'BOYLE et al., 2011) foi utilizada para converter arquivos PDB para o formato *.mdl*, o qual, representa a estrutura química dos átomos. Em seguida, a correção das moléculas por meio da adição de átomos de hidrogênio e remoção de átomos de água, foi feita com a ferramenta PyMOL (SCHRÖDINGER, 2015) a qual permite a visualização das estruturas. Moléculas pequenas, como solventes, íons sem interação com o complexo, água, entre outros, foram removidas. Esse procedimento é importante pois ajuda na simplificação do cálculo de energia, devido à complexidade da inclusão desses elementos na formulação. Assim, a proteína é considerada no vácuo com a remoção destes átomos.

Na etapa seguinte de preparação, a ferramenta AutoDockTools (ADT) (MORRIS et al., 2009) foi utilizada para gerar arquivos com as coordenadas de cada átomo e obter o átomo central do ligante. O ADT foi escolhido pois faz parte de um pacote de ferramentas bastante difundido na área de atracamento molecular. O formato dos arquivos (PDBQT) representa o ligante adicionado de valores de cargas para cada átomo, bem como a inclusão de átomos de hidrogênio na estrutura, além de informações sobre as ligações químicas presentes na molécula e os ângulos diedrais ativos para rotação. A adição de átomos de hidrogênio foi realizada em ambas estruturas, receptor e ligante, e nesse processo as moléculas de carbono e hidrogênio são unidas representando uma molécula de carga equivalente. Este processo é necessário porque os métodos de representação das estruturas moleculares não conseguem descrever os átomos de hidrogênio.

O número máximo de ângulos torcionais foi definido em 10, para que o problema tenha uma complexidade máxima definida e as soluções tenham uma representação bem definida. Existem estruturas ligantes com menos de 10 ângulos diedrais, por conta do tamanho e forma de sua estrutura, logo todos estes ângulos são considerados na otimização. Além disso, em ligantes com mais de 10 diedros, foi necessária a escolha daqueles que seriam otimizados no processo. A seleção destes 10 ângulos de torção seguiu a ideia de manter o núcleo do ligante fixo. Assim, foram escolhidos os ângulos que menos movimentam átomos da estrutura, mantendo a região central da molécula fixa (MORRIS et al.,

2009).

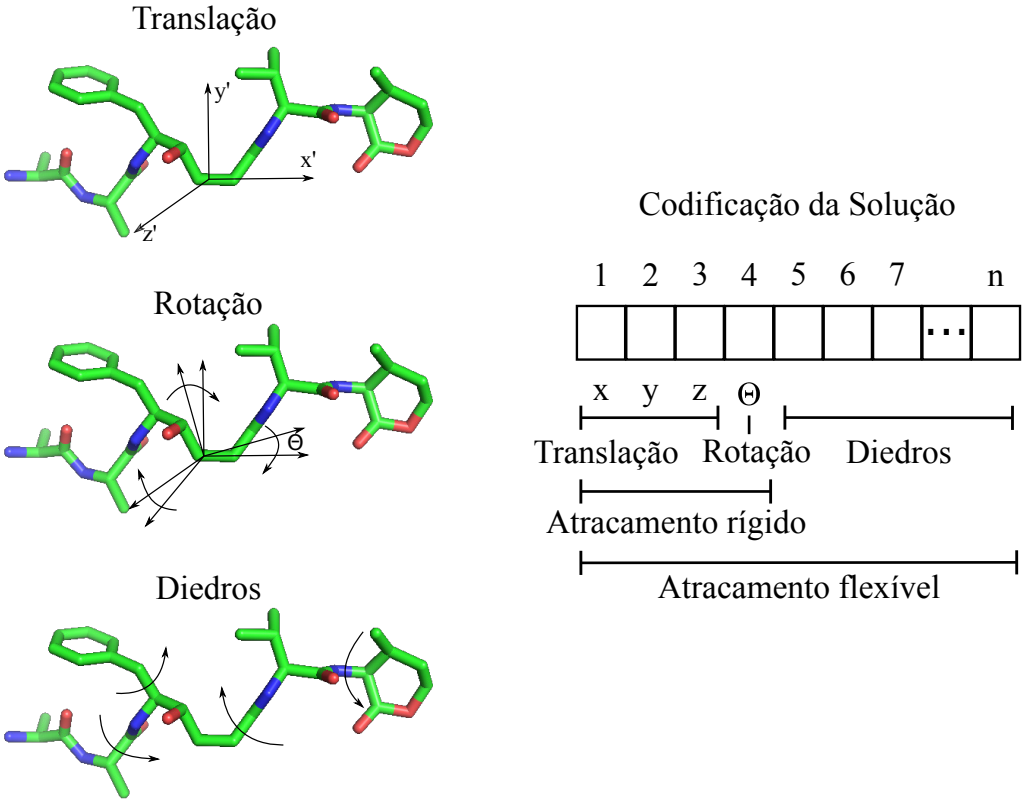
Após a preparação das estruturas ligante e receptor, foi necessária a geração de alguns arquivos, especificados na sequência, de acordo com a necessidade da função de energia utilizada, a qual é descrita na Seção 4.2. Foi utilizado o *PyRosetta toolkit* (ADOLF-BRYFOGLE; JR., 2013) na criação dos arquivos a serem utilizados pela função de energia. Foi gerado a partir do arquivo *.mdl* um arquivo de parâmetros (*.params*), o qual, contém informações químicas sobre o ligante. Além disso, um arquivo com ambas moléculas foi criado. Dessa forma, é possível carregar os complexos e avaliá-los de acordo com a função de energia do Rosetta. Entretanto, é importante ressaltar que o algoritmo desenvolvido é o responsável por produzir as modificações na estrutura ligante e repassar as coordenadas atualizadas para a interface da função de energia.

Por fim, aspectos relativos à execução do algoritmo de busca sobre o espaço conformacional são definidos. O espaço de busca é representado como uma caixa sobre o sítio de ligação da molécula receptora. Essa caixa precisa ter dimensões suficientes para permitir englobar a área de interação do complexo. Para as instâncias testadas neste trabalho foram utilizadas as mesmas dimensões de caixa, 11Å em cada dimensão. A definição do ponto central da caixa foi feita a partir da obtenção do átomo central do ligante, obtido com a ferramenta ADT. Portanto, cada complexo tem um ponto central, a partir do qual, são definidas as proporções da caixa.

Na Figura 4.1 é representado o vetor solução adotado para o problema. A codificação é representada pelas coordenadas x, y e z de translação da molécula, o parâmetro Θ referente ao ângulo no qual, a estrutura será rotacionada, além dos ângulos diedrais representando os graus de liberdade interna do ligante. A operação de rotação é feita nas ligações covalentes, porém o vetor de referência é fixo na direção da ligação química, assim é preciso inferir apenas um ângulo de rotação na codificação da solução. A mesma ideia se aplica à operação de rotação da estrutura como um todo, o vetor referência é definido a partir do átomo central do ligante e um de seus átomos mais próximo. Assim, cada complexo tem um vetor de referência diferente de acordo com seu ligante, e somente o ângulo de rotação precisa ser otimizado.

A Figura 4.1 também representa as operações sobre a estrutura ligante. Além disso, é destacado onde cada processo de Atracamento Molecular age. No atracamento rígido, somente as operações de translação e rotação da estrutura como um corpo rígido são realizadas. Enquanto que no atracamento flexível, os valores para os diedros são incluídos no processo de otimização. Cada vetor solução destes tem uma quantidade de

Figura 4.1: Codificação da solução: x,y e z são valores de translação do ligante, Θ representa o ângulo de rotação para a molécula, e os valores seguintes representam os ângulos diedrais referentes à rotação das ligações covalentes da estrutura ligante.



Fonte: Do Autor.

posições n variável, conforme o número de diedros do respectivo ligante.

4.2 Função de energia utilizada

A metodologia proposta neste trabalho para o problema de *Docking* utiliza uma abordagem de avaliação das soluções baseada em energia. Incluir no cálculo todos os graus de liberdade do complexo proteína-ligante e as variáveis químico-físicas teriam um custo computacional bem elevado (PEARLMAN; CHARIFSON, 2001). Portanto, simplificações e aproximações são realizadas em modelos de energia para se ter um tempo de execução viável, mas com o requisito mínimo de que a função utilizada seja responsável pela complementaridade hidrofílica e hidrofóbica da superfície. Considerando que métodos computacionais em atracamento molecular são parte do processo de desenvolvimento de fármacos, sua automatização é de suma importância.

Para avaliar a qualidade de uma solução é utilizada uma função de aptidão no

algoritmo que opera a busca. Neste trabalho é utilizada a função de energia e pesos *RosettaLigand* extendida por Meiler et al. (MEILER; BAKER, 2006) a partir da função estabelecida por Gray et.al. (GRAY et al., 2003). Os autores tentaram montar um modelo de energia livre que melhor discriminasse o atracamento das estruturas ao capturar as interações de *van der Waals*, solvatação, ligações de hidrogênio, e a eletrostática, além das energias internas locais, como a deformação dos ângulos de torção. A Equação 4.1 mostra detalhes da função de energia utilizada.

$$\begin{aligned} S = w_{atr}S_{atr} + w_{rep}S_{rep} + w_{sol}S_{sol} + w_{sasa}S_{sasa} + w_{hb}S_{hb} + w_{dun}S_{dun} + \\ w_{pair}S_{pair} + w_{elec}{}^{sr-rep}S_{elec}{}^{sr-rep} + w_{elec}{}^{sr-atr}S_{elec}{}^{sr-atr} + \\ w_{elec}{}^{lr-rep}S_{elec}{}^{lr-rep} + w_{elec}{}^{lr-atr}S_{elec}{}^{lr-atr} \end{aligned} \quad (4.1)$$

A afinidade de ligação de duas moléculas é medida por uma combinação linear que inclui as pontuações atraente (S_{atr}) e repulsiva (S_{rep}) de *van der Waals*, uma pontuação implícita de solvatação (S_{sol}), um termo de solvatação baseado na área da superfície (S_{sasa}), uma pontuação para ligações de hidrogênio (S_{hb}), um termo de probabilidade rotâmera (S_{dun}), um termo de probabilidade de pares de resíduos (S_{pair}), e termos eletrostáticos simples divididos em componentes atraentes e repulsivos de curto e longo alcance ($S_{elec}{}^{sr-rep}$, $S_{elec}{}^{sr-atr}$, $S_{elec}{}^{lr-rep}$, $S_{elec}{}^{lr-atr}$). A Tabela 4.1 exibe os pesos finais utilizados na função, os quais foram obtidos algumas etapas de teste com complexos para realização do melhor ajuste.

Tabela 4.1: Pesos utilizados na função de energia *Rosetta*

Termo	**Peso**	**Termo**	**Peso**
Repulsivo *van der Waals*	0.080	Probabilidade par-a-par de resíduos	0.164
Atrativo *van der Waals*	0.338	Repulsivo de curto alcance	0.025
Solvatação da superfície	0.344	Atrativo de curto alcance	0.025
Exclusão de solvência Gaussiana	0.279	Repulsivo de longo alcance	0.098
Probabilidade rotâmera	0.069	Atrativo de longo alcance	0.002
Ligações de hidrogênio	0.441		

Fonte: Adaptado de Gray et al. (2003).

Para a utilização da função descrita, são necessários arquivos em formato compatível com a interface oferecida pelo *Rosetta*. Como descrito na seção anterior, diversas ferramentas foram utilizadas para criação e conversão destes arquivos, que têm por objetivo representar as coordenadas de cada átomo, além de informações das ligações químicas e valores de carga em cada estrutura. O objetivo do método de otimização é encontrar o mínimo global da função de energia. Tendo em vista que muitas soluções são geradas

durante o processo, esta tarefa de avaliação se torna a mais custosa computacionalmente.

4.3 Descrição do espaço de busca

O problema de Atracamento Molecular requer a definição de um espaço de busca que englobe o sítio de ligação da molécula receptora, mas que também permita as operações de translação e rotações na estrutura do ligante. Como já discutido anteriormente, a representação deste espaço segue a ideia de uma caixa e tem por objetivo delimitar a quantidade de possíveis soluções para o problema. Neste trabalho foi adotada uma representação por cubos para definir o espaço de busca, proposta por Spieler et al. (SPIELER, 2016). O centro do campo de busca é definido a partir do átomo central do ligante, a partir do qual, são adicionados valores em cada direção de cada eixo para formar as dimensões da caixa, e dos subcubos, medidas em Angström (Å).

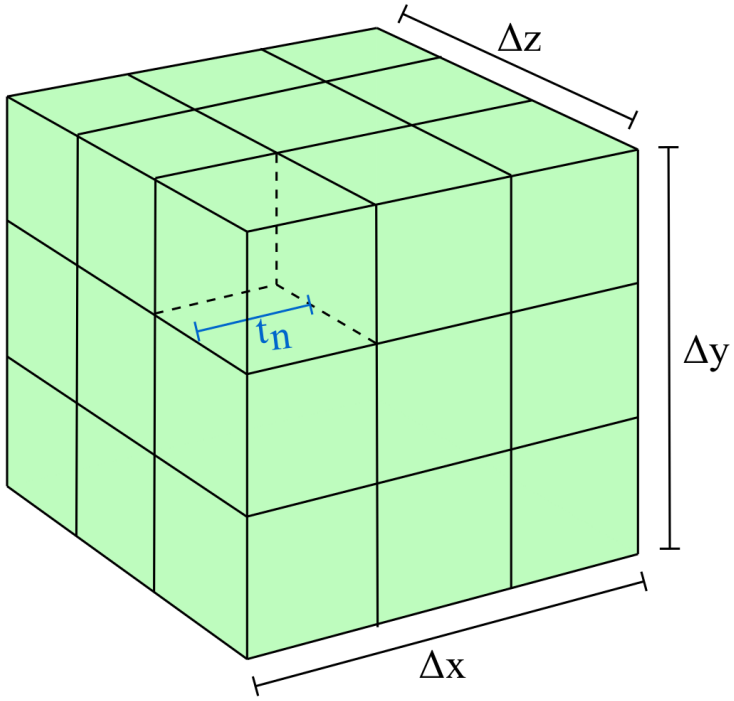
A partir de três variáveis: Δx, Δy, e Δz, que correspondem a altura, largura e comprimento da caixa, adicionadas às coordenadas p_{cx}, p_{cy}, e p_{cz}, que representam o ponto central da caixa, é possível calcular o volume dos subcubos. Na Equação 4.2 é demonstrado como o cálculo das dimensões do cubo central é feito. A variável s indica o número de subcubos em cada eixo da caixa, e a constante multiplicada de valor igual a 2 indica que são definidos os limites de cada região nas duas direções a partir do seu ponto central.

$$\begin{aligned} V_{c1} = [|p_{cx} + \frac{\Delta x}{s \cdot 2}| + |p_{cx} - \frac{\Delta x}{s \cdot 2}|] \times \\ [|p_{cy} + \frac{\Delta y}{s \cdot 2}| + |p_{cy} - \frac{\Delta y}{s \cdot 2}|] \times \\ [|p_{cz} + \frac{\Delta z}{s \cdot 2}| + |p_{cz} - \frac{\Delta z}{s \cdot 2}|] \end{aligned} \tag{4.2}$$

A Figura 4.2 ilustra a representação do espaço de busca em cubos. Em Spieler et al. (SPIELER, 2016) foi definido que o valor de s é igual a 3, gerando assim 3 cubos por eixo, somando 27 cubos dentro da área total da caixa.

A partir dessa abordagem, as soluções são geradas de modo a ocupar todas as regiões da caixa. Com isso, objetiva-se manter uma maior diversidade de indivíduos. Assim, o algoritmo gera soluções aleatórias, de acordo com a representação proposta na Seção 4.1, nas quais, os três valores de translação são limitados pelo espaço de busca, pois o ligante deve ficar dentro dessa área de busca. Essa estratégia de discretização do espaço de busca, além de explorar melhor as soluções, permite a formação de agrupamentos com

Figura 4.2: Discretização do espaço de busca proposto por Spieler et al. (SPIELER, 2016). Em destaque, um cubo n na superior esquerda da caixa com a identificação de seu tamanho t calculado a partir das dimensões da caixa.



Fonte: Do Autor.

base no critério de similaridade das soluções, que neste caso é o valor de translação a partir do ponto central.

4.4 Método de otimização proposto

O método de otimização proposto neste trabalho para lidar com o problema de Atracamento Molecular, consiste em duas etapas principais: (i) implementação de uma abordagem memética (Seção 4.5); e (ii) proposta e implementação de um modelo auto-adaptativo para coordenação dos operadores de busca local (Seção 4.6).

4.5 Etapa I: Implementação do algoritmo memético

Em problemas de otimização busca-se o mínimo ou máximo de uma função em um domínio discreto ou contínuo. A função de energia do *Rosetta* (MEILER; BAKER, 2006), descrita na Seção 4.2, utilizada para avaliação das soluções no problema de atracamento tem por características ser multi-modal, contínua, diferenciável e sujeita a restrições do

espaço de busca devido à limitação do espaço biológico. Dessa forma, a solução $x^* \in S \subset R^n$, onde S é a região definida como espaço de busca. O resultado ótimo é formulado como: $f(x^*) \leq f(x) | \forall x \in S$, onde a função objetivo é definida como $f : S \longrightarrow R$. A função $f(x)$ para otimização e suas restrições são apresentadas nas Equações 4.3 e 4.4:

$$min \quad f(x), x = x_1, x_2, ..., x_n \tag{4.3}$$

Sujeito a:

$$\begin{aligned} g_i(x) &\subseteq \left[\tfrac{-\Delta x}{2}, \tfrac{\Delta x}{2}\right] \\ h_i(x) &\subseteq \left[\tfrac{-\Delta y}{2}, \tfrac{\Delta y}{2}\right] \\ j_i(x) &\subseteq \left[\tfrac{-\Delta z}{2}, \tfrac{\Delta z}{2}\right] \end{aligned} \tag{4.4}$$

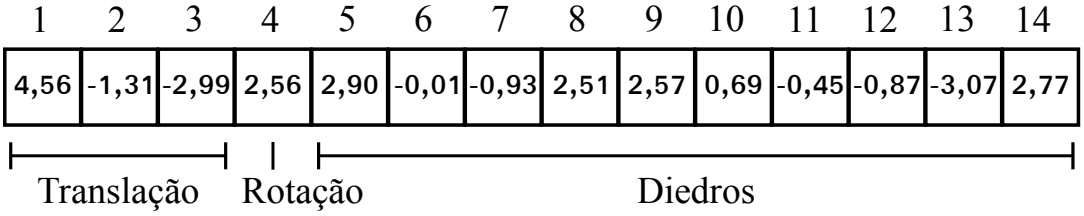
As restrições $g_i(x), h_i(x)$ e $j_i(x)$ referem-se ao espaço de busca delimitado pelo tamanho que varia de $\frac{-\Delta}{2}$ a $\frac{\Delta}{2}$ para cada eixo. A partir da função a ser otimizada e suas restrições, além da função de energia adotada, propõe-se neste trabalho, inicialmente, o desenvolvimento de um algoritmo memético.

4.5.1 Algoritmo Genético de Chaves Aleatórias Viciadas

A primeira etapa, o algoritmo memético, envolveu desenvolver um algoritmo baseado no Algoritmo Genético de Chaves Aleatórias Viciadas (BRKGA - *Biased Random-Key Genetic Algorithms*) (GONÇALVES; ALMEIDA, 2002; GONÇALVES; RESENDE, 2011). A proposta original do BRKGA indica que a codificação do problema é feita por meio de um vetor de valores reais dentro do intervalo [0, 1]. Porém, neste trabalho, cada gene do indivíduo é representado com o valor real diretamente, sem a necessidade de realizar uma conversão. A Figura 4.3 mostra um exemplo de indivíduo com seus possíveis valores. Entretanto, as demais características do algoritmo BRKGA foram preservadas, como o conceito de castas, e as operações de recombinação e mutação.

A primeira etapa, antes da execução em si do algoritmo, requer a preparação do ligante para que o processo de busca não seja tendencioso. Cada execução do algoritmo lê os arquivos com as coordenadas do ligante e receptor conforme eles estão no PDB. Considerando que a caixa, que delimita o espaço de busca, colocada no sítio de ligação da proteína tem seu centro definido nas coordenadas do átomo central do ligante, é necessário que essa molécula seja colocada em outra posição. Assim, ao carregar a estrutura ligante, a mesma sofre uma aleatorização, que consiste na geração de um indivíduo com valores

Figura 4.3: Exemplo de codificação de um indivíduo no algoritmo BRKGA proposto. Os valores simulam uma solução cujas restrições são: [-5,5] para a translação, e [-3,14, 3,14] para as operações de rotação da estrutura.



Fonte: Do Autor.

aleatórios para as operações de translação e rotação da estrutura. Esse indivíduo, cujos valores respeitam as restrições do problema, é aplicado sobre o ligante de entrada e então geradas as novas coordenadas. Essa nova estrutura passa então a ser o alvo da otimização do método proposto.

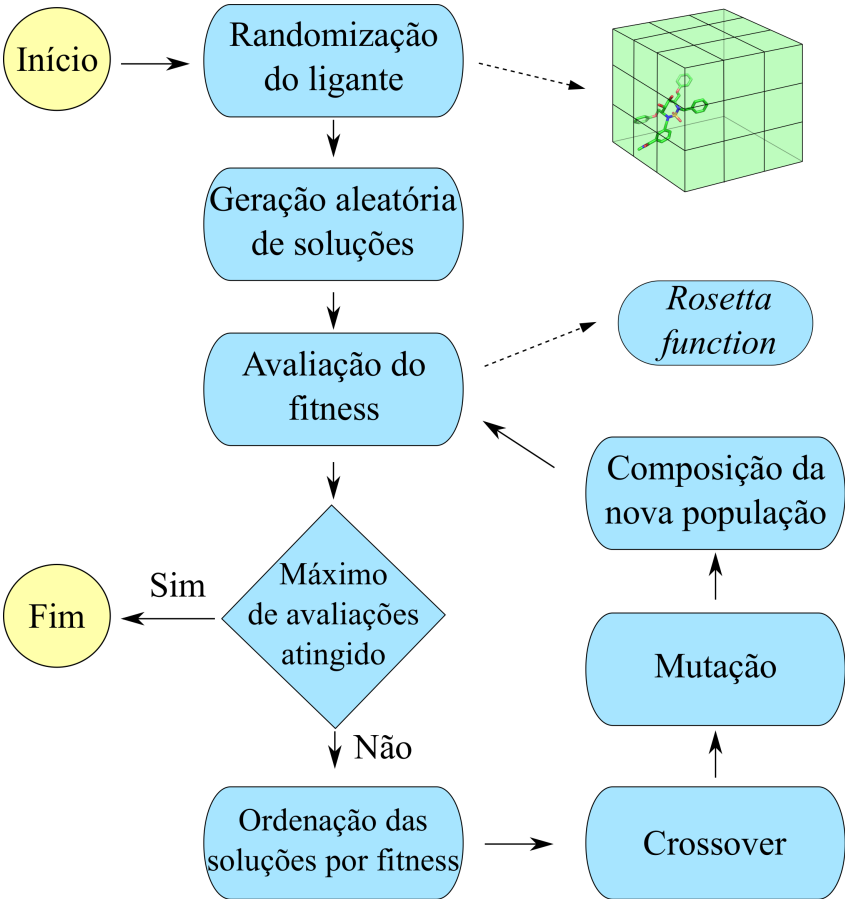
O algoritmo trabalha com uma população p de vetores solução, os quais evoluem a cada iteração, também conhecida por geração. Cada componente da solução é chamado de gene e é gerado de forma independente. A população inicial é composta por indivíduos gerados aleatoriamente, respeitando as restrições dos possíveis movimentos que o ligante pode assumir e do espaço de busca. Este último segue uma representação por cubos(Seção 4.3), e portanto o total de indivíduos é gerado de forma distribuída entre todos os subcubos. Inicialmente, tem-se uma população bem diversa, mas no decorrer das gerações a tendência é que o algoritmo convirja para determinadas regiões. No entanto, outro parâmetro do algoritmo definido como percentual de indivíduos não-migrantes (p_{nm}) é adotado para que cada subcubo do espaço discretizado tenha uma quantidade mínima de indivíduos. Assim, embora o algoritmo deva convergir, a ideia é que a população continue minimamente diversa com soluções ocupando toda a região de busca.

Após a geração da população inicial, os indivíduos são avaliados pela função de energia e ordenados de acordo com seu *fitness*. A população é então dividida em dois grupos: elite (p_e) e não-elite ($p - p_e$). A partir daí ocorre o processo de geração da nova população, onde o primeiro passo é copiar os indivíduos da elite para a nova população. Em seguida, o processo de *crossover* é realizado combinando soluções de ambos conjuntos. São selecionados um indivíduo pai do grupo de elite e outro do grupo não-elite. Um valor sorteado define as chances dos genes do indivíduo da elite serem selecionados, de acordo com a probabilidade ρ_e, e então compor a solução filha. É importante ressaltar que aqui não existe a restrição dos subcubos, portanto a solução gerada pode ou não pertencer à mesma região de seus pais. Assim, pode ocorrer que alguns subcubos percam soluções

e venham a ficar vazios.

Na sequência, ocorre o processo de mutação. Como visto anteriormente, a operação de recombinação pode deixar cubos vazios, os quais são então preenchidos nesta etapa do algoritmo. O processo de mutação se dá por gerar soluções aleatórias para serem inseridas na população. Utilizando-se desta característica definiu-se que essa geração pode ser direcionada para as regiões vazias do espaço de busca. Após a operação de *crossover* tem-se uma lista dos possíveis subcubos vazios, os quais são então preenchidos. Após o preenchimento, se ainda não for atingida a quantidade p_m de soluções, o algoritmo irá gerar o restante das soluções independente da região do espaço de busca. Assim, uma restrição da parametrização é que a quantidade de indivíduos não-migrantes de todos os cubos não some um valor maior do que a quantidade de soluções definida para ser gerada na mutação. Finalmente, é formada a nova população. A Figura 4.4 ilustra o fluxograma de execução do processo implementado no BRKGA. A próxima etapa é a inclusão dos mecanismos de busca local para composição do algoritmo memético.

Figura 4.4: Fluxograma de execução do BRKGA, representado em alto nível, desde a preparação dos dados passando pelas etapas do algoritmo até o seu encerramento.



Fonte: Do Autor.

4.5.2 Algoritmo Memético

O algoritmo memético é uma abordagem evolutiva composta por técnicas de busca global e local (MOSCATO, 1989). A busca global tem a tarefa de explorar o espaço de busca como um todo, enquanto que a busca local consegue explorar áreas próximas de uma determinada solução com o intuito de obter uma maior precisão (KRASNOGOR; SMITH, 2005). A ideia de utilizar busca local no problema de atracamento molecular pode ser justificada pela intenção de realizar pequenas correções numa solução. Tais correções podem evitar o que se chama de *clash* de energia, isto é, choques entre os átomos das moléculas, e assim com uma simples modificação obter uma boa conformação proteína-ligante. Em nossa proposta o algoritmo responsável por realizar a busca global é o BRKGA, enquanto que os métodos adotados como busca local são três variações do algoritmo *Hill climbing* e o *Simulated Annealing*.

Como discutido na Seção 3.3.2. existem aspectos a serem considerados no desenvolvimento de um algoritmo memético. De acordo com Molina et al. (MOLINA; LOZANO; HERRERA, 2009), para que uma busca local tenha sucesso em sua aplicação combinada com um algoritmo evolutivo, algumas questões precisam ser resolvidas, como: qual busca local escolher, a seleção dos indivíduos para aplicação da LS, e o esforço computacional gasto em cada aplicação. Além destas questões, decisões a respeito da definição de uma vizinhança e como os algoritmos de busca local operam suas buscas são discutidas.

Recentemente, tem-se reconhecido que a particular escolha de um algoritmo de busca local exerce um grande impacto sobre a eficácia de MAs (SMITH, 2007). A grande variedade de algoritmos de busca local disponível torna quase impossível saber qual deles é mais relevante para um problema quando se tem um limitado conhecimento da sua superfície de custo antes de começar (ONG; KEANE, 2004). Além disso, os algoritmos de LS por si só são conhecidos por funcionar de maneira muito diferente para diferentes problemas, inclusive aqueles de mesmo domínio. Dependendo da complexidade de um determinado problema, os algoritmos que se mostraram eficientes no passado podem não funcionar bem em outros problemas (MOLINA; LOZANO; HERRERA, 2009). De qualquer forma, a escolha dos algoritmos aplicados nesse trabalho seguiu o que geralmente tem sido utilizado na literatura.

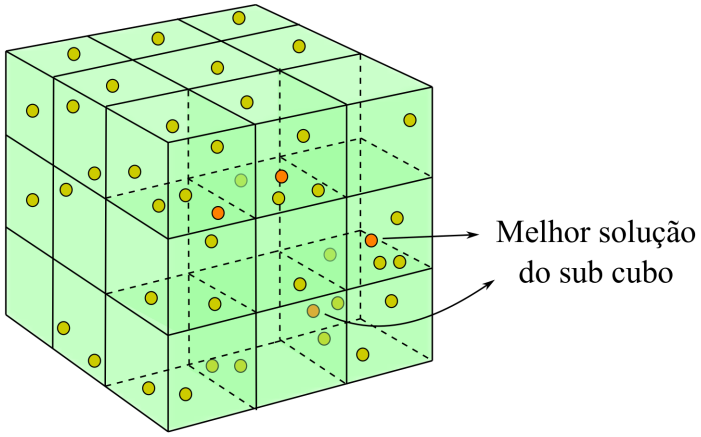
A seleção dos indivíduos a serem melhorados pela busca local é outro importante fator na eficácia de um algoritmo memético. Uma escolha errada faz com que o MA possa

não explorar adequadamente algumas regiões promissoras do espaço de busca. Normalmente, algoritmos de LS são aplicados para refinar soluções recém criadas na aplicação do *crossover* e mutação. No entanto, o aumento de avaliações da função de *fitness* requiridas pode aumentar o custo computacional do MA, e ser um impeditivo para alguns problemas. Como solução, uma abordagem proposta por Hart et al. (HART, 1994) utiliza um parâmetro (p_{LS}) que determina a probabilidade de aplicar a LS no cromossomo criado. Outras alternativas também podem ser consideradas, como em Chelouah et al. (CHELOUAH; SIARRY, 2003), onde o método de busca local é aplicado após o algoritmo global detectar uma nova região promissora, e em Liang et al. (LIANG; SUGANTHAN, 2005), a LS melhora, periodicamente, os N melhores indivíduos da população.

Em nossa proposta de algoritmo memético, a escolha dos indivíduos a serem melhorados está relacionada com o esquema de discretização do espaço de busca. No momento em que o algoritmo de busca local deve ser aplicado, é preparado um conjunto de indivíduos candidatos a serem melhorados (ir_{LS} - *individuals to run LS*). Primeiramente, é feita uma ordenação dos cubos, de acordo com a quantidade de soluções que cada um possui, de forma decrescente. Logo, os N melhores indivíduos que fazem parte dos M cubos mais populosos irão formar o conjunto. Entretanto, esse processo de seleção segue a restrição de que somente indivíduos ainda não melhorados pela busca local podem ser escolhidos. Assim, caso o melhor indivíduo de um determinado cubo já foi melhorado por um operador de LS, a seguinte melhor solução, não melhorada, do cubo é selecionada. Além disso, como o processo de busca local pode ocorrer em meio a etapa de *crossover* do BRKGA, os indivíduos da nova população que pertençam a algum cubo mais populoso e que sejam melhores do que as soluções da população atual, farão parte do conjunto de indivíduos a serem melhorados. A Figura 4.5 ilustra o processo de seleção dos indivíduos para aplicação da busca local. Essa estratégia de seleção dos indivíduos visa refinar aquelas soluções melhores avaliadas e que estão nas regiões mais promissoras encontradas pelo algoritmo.

Outro aspecto importante na implementação de um algoritmo memético é a definição de quando aplicar a busca local e a sua intensidade. O número de avaliações de *fitness* requiridas pelo algoritmo de LS determina o seu custo. No trabalho de Morris et al. (MORRIS et al., 2009) os autores se referem a esse número como a intensidade da LS (I_{ls} - *LS intensity*). É importante ajustar bem este parâmetro, porque um valor baixo pode ser insuficiente para explorar a vizinhança de uma solução, e um valor alto pode também, consumir tempo desnecessário na avaliação das soluções. O comum é adotar

Figura 4.5: Processo de seleção de indivíduos para aplicação de busca local. Os subcubos mais populosos têm suas melhores soluções (destacadas em laranja) selecionadas para o refinamento.



Fonte: Do Autor.

um único valor para a intensidade da busca, o qual, é mantido durante todo o processo de otimização. Em Lozano et al. (LOZANO et al., 2004) foi definida a taxa de busca local/global ($r_{L/G}$), como sendo o percentual de avaliações gasto em busca local sobre o total de avaliações atribuído ao algoritmo memético. Esse parâmetro está relacionado ao valor de intensidade da LS, e consequentemente sua relação define a frequência com que a busca local é aplicada (n_{frec} - *number of evaluations*). A Equação 4.5 mostra como o cálculo é feito.

$$n_{frec} = I_{ls}\left(\frac{1 - r_{L/G}}{r_{L/G}}\right) \tag{4.5}$$

Com esse mecanismo, assim que o BRKGA realizar a busca global por n_{frec} avaliações, imediatamente a busca local será aplicada. Devido ao número de indivíduos na população, é possível que esse valor de avaliações seja atingido durante os processos de recombinação e mutação do BRKGA. Por isso que, em algumas situações, uma solução recém criada pelo *crossover* pode vir a ser escolhida para a aplicação da LS, conforme discutiu-se anteriormente.

Uma definição bastante relevante no projeto de um MA é a definição da vizinhança a ser considerada no processo de busca local. Considerando que o problema de atracamento molecular é formulado em um espaço contínuo de soluções, apresentando inúmeras possibilidades, é necessário tomar algumas decisões para restringir essa área de busca. Para uma dada solução S tem-se um total de n genes que, como visto, é dependente da estrutura do ligante. Em cada gene o algoritmo de busca local irá realizar uma modificação

que consiste em adicionar ou remover um valor do gene x. Esse valor é chamado de raio de perturbação (r_{pt}), e nesta primeira etapa ele terá um valor fixo para qualquer algoritmo aplicado. A orientação da busca ocorre nos dois sentidos, ao subtrair e adicionar o r_{pt} do valor de cada gene. Portanto, nossa vizinhança segue a seguinte definição:

$$x \pm r_{pt} | \forall x \in S \tag{4.6}$$

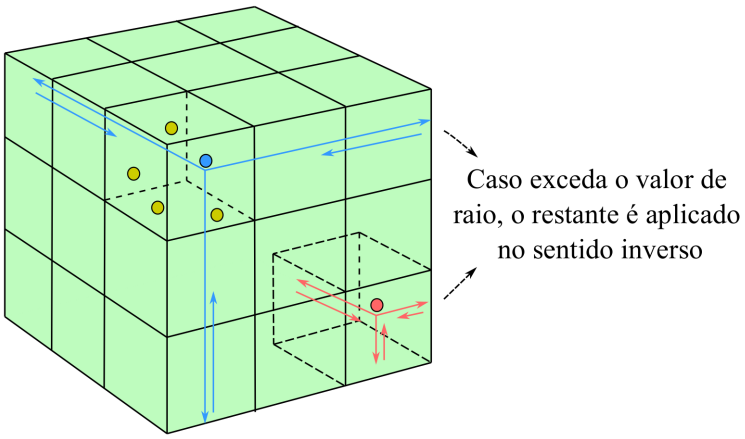
desse modo, cada solução tem um total de $2n$ soluções vizinhas a serem exploradas, considerando as duas operações aplicadas sobre cada posição do vetor.

A partir de todas estas definições foram implementados quatro algoritmos de busca local para trabalhar com o BRKGA, três variações do *Hill climbing* e o *Simulated Annealing*. A implementação canônica dos algoritmos é descrita na Seção 3.3.2. Em nossa versão ocorreram adaptações por conta de se trabalhar com um problema contínuo e pela discretização do espaço de busca em cubos. A aplicação da busca local envolve, além do vetor solução, a informação sobre a possibilidade ou não do indivíduo migrar de cubo. Dessa forma, se a quantidade de soluções naquela determinada região for menor ou igual ao limite mínimo, o indivíduo não pode migrar, caso contrário sim.

Com a informação da migração a execução das buscas locais passa a obedecer restrições quanto à translação. Caso a solução não possa migrar de cubo, os valores podem apenas estar entre aqueles definidos para cada área. Se uma alteração de valor ultrapassar o limite mínimo ou máximo do cubo, o valor excedente é então somado ou subtraído, respectivamente, do limite. Conforme ilustrado na Figura 4.6, o processo funciona como um espelhamento, ao invés do valor exceder o limite do cubo ou simplesmente parar no mesmo, ele volta para o seu interior de modo a movimentar a solução de acordo com o valor do r_{pt}. No exemplo dado, a solução em azul tem total liberdade de movimentação por não ser a única em seu cubo, enquanto que a solução em vermelho, por ser a única naquela região, fica restrita a não exceder os limites do cubo.

Já no caso de uma solução poder se movimentar livremente, a única restrição será a do espaço de busca como um todo. Ocorre o mesmo cálculo de posição só que em um cubo maior. Para as perturbações nos valores que correspondem às rotações, também existe uma estratégia em relação aos limites. Estas operações trabalham com valores mínimo e máximo para realizar a conformação da estrutura em determinado ângulo. Como tais operações independem da discretização por cubos, o procedimento de validação é mais simples, de modo que ao atingir um limite, o valor excedente é incrementado ou decrementado do limite oposto. Dessa forma, os valores são percorridos de forma circular

Figura 4.6: Exemplo de movimentação das soluções no espaço de busca seguindo as restrições de migração de cada região. A solução em azul possui liberdade total de movimentação, enquanto que a vermelha só pode mover-se dentro do seu cubo.



Fonte: Do Autor.

respeitando o valor de perturbação aplicado.

Conforme descrito na Seção 3.3.3, o conceito de encadeamento de buscas locais proposto por Molina et al. (MOLINA et al., 2011) também é implementado em nossa proposta de algoritmo memético. Em todos os métodos de LS desenvolvidos, as configurações da aplicação são salvas para permitir que uma futura chamada de busca local possa utilizá-las. Estas configurações ficam salvas junto à respectiva solução e são compostas pela ordem de visitação da vizinhança e qual o último gene visitado. Para o algoritmo *Best Improvement* apenas o último gene visitado é relevante, já que a visitação é feita por completo não existindo uma ordem diferente. Nos métodos *First Improvement* e *Stochastic Hill Descent* tanto o último gene modificado, quanto a ordem de geração dos vizinhos são salvos ao final da aplicação.

Para o *Simulated Annealing* não existe uma ordem de visitação pois as soluções são geradas aleatoriamente, no entanto, o número de passos do algoritmo foi ajustado. Com base em testes experimentais, foi definido um número máximo de passos (iterações) (n_{steps}) para o SA de modo que ele tenha uma quantidade suficiente para melhoria e ao mesmo tempo não gaste muitas iterações numa mesma região. Também, estruturas auxiliares monitoram se caso todos os vizinhos sejam gerados e não ocorra atualização da solução, então o procedimento é encerrado. Assim, essas limitações permitem que, em algumas situações, possa ocorrer o encadeamento de aplicações de busca local. É importante salientar que somente soluções cuja vizinhança não foi totalmente explorada é que ficam aptas ao encadeamento de LS, caso contrário, a solução é considerada como

melhorada e não será mais selecionada para aplicação de busca local. O pseudocódigo 1 descreve como cada um dos métodos de busca são implementados.

Algoritmo 1 - Pseudocódigo das buscas locais

Entrada: S, método de LS, r_{pt} ▷ S = solução, r_{pt} = raio de perturbação

```
ordem = null, geneCorrente = 0
se S.config != null então:        ▷ Carrega a configuração da aplicação de LS anterior, se houver
    ordem = carregarOrdem(), geneCorrente = obterGeneCorrente()
fim se
executarBuscaLocal()
```

Saída: S

***Hill Climbing* e variantes**

```
S* = S                            ▷ S* = melhor solução da vizinhança
ordem = [1, 2, 3, ..., n]         ▷ A ordem depende da variação: BI, FI ou SHD. Esse ordenamento sequencial é do BI, já no FI a ordem é gerada de forma aleatória uma vez, e no caso do SHD, a ordem é gerada dentro do 'enquanto'
enquanto S não for ótimo local E avaliações < maxAvaliações faça:
    para geneCorrente ← ordem faça:   ▷ Este laço itera o conjunto de ordenação da geração dos vizinhos, e pode ser interrompido no FI e SHD se S for melhorado
        S' = pertubarGene(S, geneCorrente)
        se f(S') < f(S) então:
            S* = S'
        fim se
        avaliações += 1
    fim para
    S = S*
fim enquanto
```

Simulated Annealing

```
minTemperatura = 2.5, maxTemperatura = 25000, passos = 0
fatorTemperatura = -log(maxTemperatura / minTemperatura)
temperatura = maxTemperatura
enquanto S não for ótimo local E passo < maxPassos faça:
    gene = gerarGeneAleatorio()
    R = pertubarGene(S, gene)
    delta = f(R) - f(S)
    exp = exp(delta / temperatura)
    se (delta > 0 E exp < random.random()) OU (delta <= 0) então:   ▷ Se R é pior do que S mas exp é menor do que um valor aleatório, ou se R é melhor do que S, atualiza a solução corrente
        S = R
    fim se
    temperatura = maxTemperatura * exp(fatorTemperatura * passos / maxPassos)
    passos += 1
fim enquanto
```

No BRKGA, que comanda a aplicação da busca local, é controlado também a quantidade de avaliações de energia realizadas, não podendo exceder o valor definido em I_{ls}. Uma aplicação de LS pode ou não requerer todo o montante de avaliações, caso não utilize, outras soluções são submetidas ao processo com a restrição do valor de aplicação atualizado. As soluções retornadas pelo processo de busca local são imediatamente atualizadas na população, bem como a informação populacional dos cubos. O pseudocódigo 2 apresenta como cada parte do algoritmo memético foi projetada e implementada.

Algoritmo 2 - Pseudocódigo do algoritmo memético desenvolvido

Entrada: P, P_e, P_m,n, ρ_e, L, I_{ls}

$P \leftarrow$ inicializa com n vetores de valores reais
enquanto não alcançar o número máximo de avaliações de energia **faça:**
    Agrupa as soluções em L grupos
    Avalia a energia de cada solução em P
    Divide P em P_e e $P_{\overline{e}}$
    Copia o conjunto elite para a próxima geração: $P^+ \leftarrow P_e$
    $P^+ \leftarrow P^+ \bigcup$ *crossover()* $\bigcup$ *mutação()*
fim enquanto

Saída: Melhor solução X

Operação de *crossover*

para todo $i \leftarrow 1$ a $|P| - |P_e| - |P_m|$ **faça:**
    Seleciona aleatoriamente um pai a de P_e; Seleciona aleatoriamente um pai b de $P_{\overline{e}}$;
    para todo $j \leftarrow 1$ to n **faça:**
        Atribui um valor booleano à variável B de acordo com a probabilidade ρ
        se $B == True$ **então:**
            $c[j] \leftarrow a[j]$
        senão
            $c[j] \leftarrow b[j]$
        fim se
    fim para
    avaliarSolução(c)
fim para

Processo de mutação

enquanto existir cubos vazios **faça:**
    avaliarSolução(novoIndivíduo) ▷ De acordo com a identificação do cubo
fim enquanto
Preenche o restante do conjunto com indivíduos gerados em cubos aleatórios

Avaliação de soluções

fitness = *calcularEnergia*(solução) ▷ Resultado obtido com a função de energia adotada
se avaliaçõesCorrente é múltiplo de I_{ls} **então:**
    prepararBuscaLocal()
fim se
retorna *fitness*

Preparação para execução da busca local

Compõe a lista de potenciais candidadtos para aplicação da `LS` ▷ De acordo com o valor de I_{ls}
enquanto número de avaliações < I_{str} **faça:**
    executarBuscaLocal() ▷ Itera sobre a lista anterior
fim enquanto

Processo de busca local

Carrega as configurações da solução ▷ Mantendo as parametrizações utilizadas anteriormente
Aplica o método de `LS`
Salva as configurações do processo de busca na solução ▷ Permite realizar o encadeamento das `LS`

Por fim, essa etapa de desenvolvimento do algoritmo memético tem sua análise e comparação realizada na Seção 5. Onde cada algoritmo de busca local é comparado entre si e com o BRKGA em sua versão não memética.

4.6 Etapa II: Implementação do modelo auto-adaptativo

A segunda etapa do método proposto consiste no desenvolvimento de um Algoritmo Multimemético Auto-adaptativo a partir do Algoritmo Memético proposto e implementado na etapa anterior. Proposto por Krasnogor et al. (KRASNOGOR; SMITH, 2001), o MMA tem por objetivo aplicar diferentes operadores de busca local a diferentes indivíduos, durante uma mesma execução. Essa adaptação requer um mecanismo para gerenciar o conjunto de operadores e decidir qual deles deve ser escolhido em determinado momento. No trabalho dos autores foi utilizado o SIM, um mecanismo de herança simples em que os indivíduos de melhor *fitness* transmitem seu conteúdo memético para seus filhos na operação de *crossover*.

Neste trabalho é proposta a implementação de uma função de probabilidades que é aplicada como mecanismo de avaliação dos operadores de busca local e, consequentemente adapta o processo de escolha dos mesmos. Inicialmente manteve-se o parâmetro de raio de perturbação fixo, e criou-se um *pool* com os 4 métodos de busca local implementados. O modelo de adaptação foi então aplicado para combinar as buscas locais, e comparado com a implementação do mecanismo SIM (KRASNOGOR; SMITH, 2001), e uma versão aleatória de seleção dos métodos. Na sequência, decidiu-se por testar outros valores para o raio de perturbação, e a partir disso também foi criado um conjunto desses valores para serem adaptados no algoritmo. Assim, a versão final do algoritmo auto-adaptativo inclui a variação dos métodos de busca local e do raio de perturbação para geração da vizinhança de uma solução.

É importante salientar que em nosso método a representação do material memético do indivíduo não é feita no vetor-solução, isto é, não está presente na codificação das operações do ligante. O objetivo da implementação desse modelo auto-adaptativo é verificar a viabilidade e os ganhos que podem ser obtidos ao combinar mais de uma técnica de busca local com diferentes valores de perturbação em uma mesma execução do algoritmo memético. Os detalhes da função proposta e como o modelo de adaptação funciona são descritos a seguir.

4.6.1 Função de probabilidade

Desde o início da pesquisa e desenvolvimento do modelo auto-adaptativo para melhor utilização das buscas locais, sempre se idealizou um mecanismo que fosse o mais justo possível na escolha daquele método que melhor desempenho apresentasse. Baseando-se em análises de execuções prévias e em propostas do estado da arte, decidiu-se por propor e implementar uma função de probabilidade para guiar nosso esquema de auto-adaptação. Essa função é composta por dois termos e tem sua inspiração nos estudos realizados por Jakob (JAKOB, 2006; JAKOB, 2010) e Domínguez et al. (DOMÍNGUEZ-ISIDRO; MEZURA-MONTES, 2018).

A Equação 4.7 mostra os termos de taxa de ganho de *fitness* (rfg - *ratio fitness gain*), e taxa de sucesso de aplicação (rs_{app} - *ratio of success application*). O primeiro termo é uma medida do quanto o operador de busca local melhora a aptidão de uma solução, considerando a relação dessa melhoria com a média de aptidão de uma parte da população, o conjunto elite. Já o segundo termo é mais simples e representa o benefício alcançado por uma LS através da relação de quantas soluções foram melhoradas em sua aplicação.

$$rfg = \left| \frac{f_{original} - f_{LS}}{f_{original} - f_{meanA}} \right| \qquad rs_{app} = \frac{n_{improvements}}{n_{app}} \tag{4.7}$$

O rfg é obtido a partir da aplicação de busca local em um indivíduo, onde $f_{original}$ é o *fitness* da solução antes da melhora, f_{LS} representa o *fitness* alcançado pelo algoritmo de LS, e f_{meanA} é a média de *fitness* de todas as soluções da casta A no momento em que a busca local foi chamada. O objetivo desse termo é medir a efetividade da busca local sobre uma solução comparando a melhoria com os melhores indivíduos da população naquele momento. O valor de rs_{app} representa a relação entre o número de melhorias ($n_{improvements}$ - *number of improvements*) e o número de aplicações (n_{app} - *number of applications*) de um dado operador de busca local. Para manter os métodos competitivos e cooperativos, adotou-se pesos a cada um dos termos da equação, ideia inspirada no trabalho de Jin et al. (JIN; ZHIHUA; WENYIN, 2014). A Equação 4.8 mostra a proposta:

$$\begin{gathered} factor_{rfg} = \left(\frac{\sum_{i=1}^{n_{app}} rfg_i}{n_{app}} \right) \cdot w_{rfg} \qquad factor_{rs_{app}} = rs_{app} \cdot w_{rs_{app}} \\ weight_X = factor_{rfg} \cdot factor_{rs_{app}} \end{gathered} \tag{4.8}$$

Para medir a efetividade de uma busca local considera-se o seu histórico de melhorias através da média dos valores rfg obtidos multiplicada por um valor fixo, que é o peso w_{rfg}. Da mesma forma, à taxa de sucesso é atribuído outro peso, o $w_{rs_{app}}$. A soma destes pesos deve ser igual a 1, e a tendência é que seja utilizado um valor ligeiramente maior para o primeiro termo pelo fato de representar melhor a contribuição de uma LS. Assim, a obtenção dos fatores de cada termo é realizada, e a sua multiplicação nos dá o peso final $weight_X$ da busca local X. Após cada aplicação de um algoritmo de LS, o seu respectivo peso é calculado e então sua probabilidade de aplicação pode ser calculada, conforme a Equação 4.9 mostra:

$$prob_X = \frac{weight_X}{\sum_{j=1}^{n_{LS}} weigth_j} \tag{4.9}$$

A probabilidade de uma dada busca local ($prob_X$) é obtida calculando seu percentual de contribuição sobre a soma dos pesos de todos os métodos de busca (n_{LS}). Assim, cada operador terá um valor entre 0 e 1, o que permite montar uma roleta com as proporções de cada algoritmo e então sortear um número para escolha do método.

É importante destacar que no início da execução do MMA, mais especificamente na primeira aplicação de busca local, todos os métodos de busca têm o mesmo número de avaliações da função objetivo. Isso garante que os pesos iniciais sejam calibrados de forma justa e igual entre todos os algoritmos. Durante a execução, o método que se mostrar melhor, terá uma probabilidade maior de ser selecionado para novas aplicações. Esse controle inicial da divisão de esforço computacional entre cada método, bem como o controle do conjunto de operadores de busca utilizado, e todo o processo de cálculo e atualização dos fatores e probabilidades é realizado pelo mesmo módulo que preparava e executava as buscas locais no algoritmo memético. A implementação do MMA proposto é descrita no Algoritmo 3.

4.6.2 Fase A - Implementação de modelos adaptativos para comparação

Inicialmente, para comparar nosso modelo de auto-adaptação baseado em probabilidades, decidiu-se adotar o método SIM (KRASNOGOR; SMITH, 2001), e um método aleatório. O SIM é um mecanismo simples de herança que transmite o material memético por meio do *crossover*. Nesta implementação foram necessárias poucas mudanças, adicionou-se um gene na representação do indivíduo para indicar o método de busca a ser

Algoritmo 3 - Pseudocódigo do algoritmo multimemético auto-adaptativo

Entrada: P, P_e, P_m,n, I_{ls}

1: $P \leftarrow$ inicializa com n vetores de valores reais
2: **enquanto** não alcançar o número máximo de avaliações de energia **faça:**
3: Avalia as soluções de P e divide em P_e e $P_{\overline{e}}$
4: $P^+ \leftarrow P_e \bigcup$ *crossover()* $\bigcup$ *mutação()* ▷ Composição da próxima população
5: **se** avaliaçõesCorrente é múltiplo de I_{ls} **então:**
6: *prepararBuscaLocal()* ▷ Indivíduos melhorados são introduzidos em P^+
7: **fim se**
8: **fim enquanto**

Saída: Melhor solução X

Preparação para execução da busca local

9: Composição da lista de potenciais candidatos para aplicação da `LS` ▷ De acordo com o valor de I_{ls}
10: **enquanto** número de avaliações < I_{str} **faça:**
11: indivíduo = obterPróximoIndivíduo() ▷ Itera sobre a lista construída anteriormente
12: método, raio = escolherOperadores() ▷ Valores sorteados definem a seleção
13: *executarBuscaLocal(indivíduo, método, raio)*
14: rfg e rs_{app} são calculados ▷ De acordo com o método e o raio de perturbação
15: *atualizarProbabilidades()* ▷ Probabilidades dos operadores são atualizadas de acordo com seus respectivos pesos
16: **fim enquanto**

utilizado, o qual, é gerado de forma aleatória no início da execução e depois é atualizado na recombinação das soluções. No momento da aplicação da busca local, o indivíduo selecionado já terá em sua representação o método a ser utilizado.

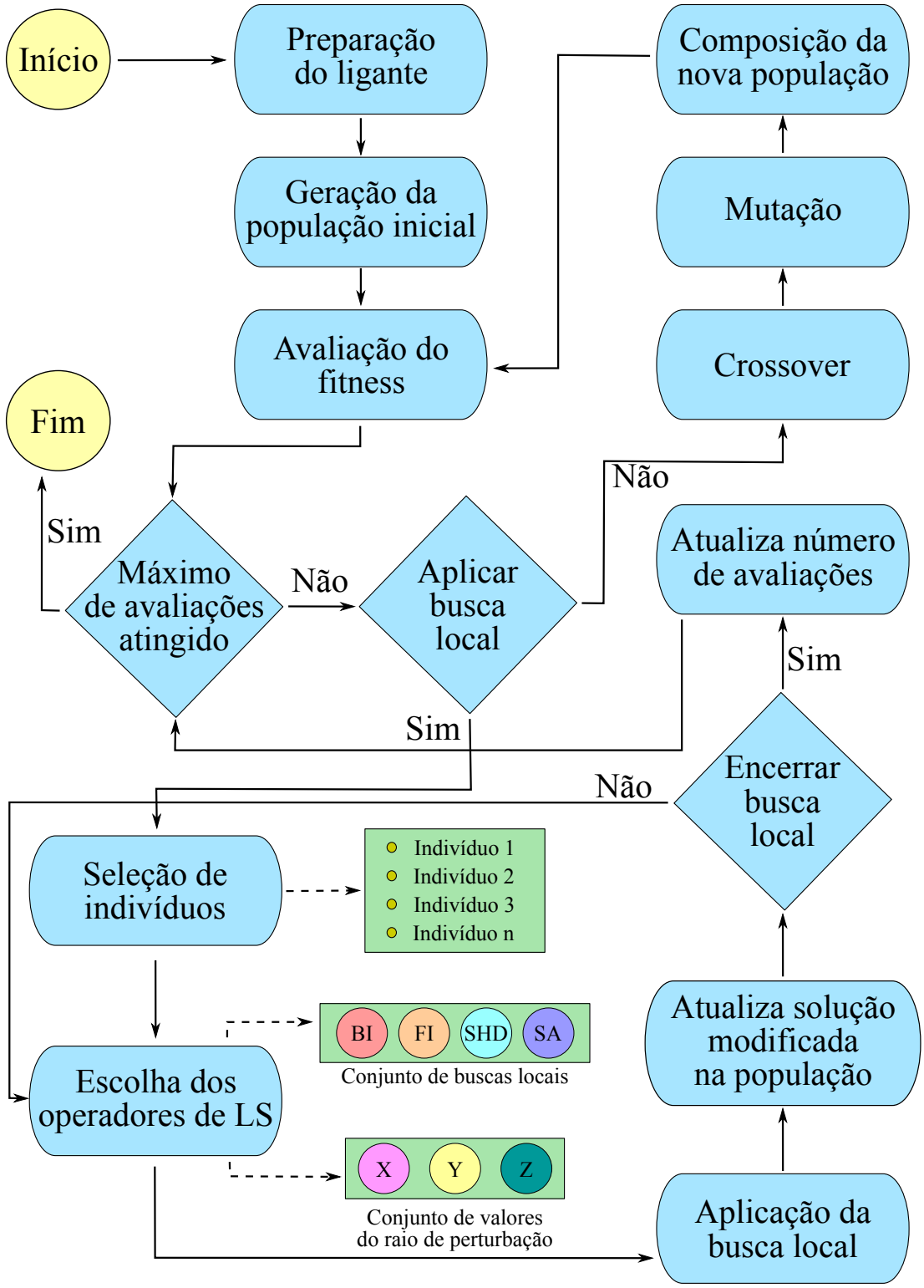
O método aleatório mencionado trata-se de uma abordagem na qual, ao iniciar o processo de aplicação da busca local, é sorteado um dos algoritmos e aplicado na busca. Sua implementação não exigiu muito esforço, apenas esse sorteio antes da aplicação da LS. A ideia dessa fase de implementação e comparação com outras abordagens, é validar o nosso modelo de auto-adaptação baseado em probabilidades. Nesta fase de testes e análises somente foram variados os algoritmos de busca local, enquanto que os demais parâmetros permaneceram com valor fixo.

4.6.3 Fase B - Variação e implementação de adaptação do raio de perturbação

Após a implementação e validação do modelo auto-adaptativo com os algoritmos de busca local, decidiu-se dar atenção a um parâmetro específico do processo de LS, o raio de perturbação. Até então, as implementações do algoritmo memético e do MMA consideravam esse parâmetro com valor fixo, variando apenas o algoritmo de aplicação. Assim, inicialmente definiu-se um conjunto de valores para o parâmetro, e testou-se cada um deles separado. Em seguida, estendeu-se a função de probabilidades para esse conjunto pré-definido e foi aplicado o MMA, de modo que fosse possível mensurar qual valor de perturbação tem melhor desempenho.

A ideia da combinação de diferentes valores para perturbação no processo de busca local é permitir uma melhor exploração das soluções. No início os algoritmos podem realizar modificações maiores com o objetivo de encontrar regiões mais promissoras, seguindo a ideia do componente de busca global do algoritmo memético só que em menor escala. E no final da execução, menores modificações podem se mostrar mais interessantes para um melhor refinamento dos indivíduos. Por fim, combinando os conjuntos de algoritmos de LS e de raio de perturbação tem-se o modelo final de auto-adaptação baseado em probabilidades. A Figura 4.7 ilustra o fluxograma de execução do algoritmo multimemético proposto.

Figura 4.7: Fluxograma de execução do algoritmo MMA auto-adaptativo, representado em alto nível, com o esquema de seleção dos operadores para aplicação da busca local nas soluções.



Fonte: Do Autor.

4.7 Resumo do capítulo

Neste capítulo foram apresentados a metodologia do desenvolvimento do método proposto, além de suas etapas e detalhes de implementação. Inicialmente, um modelo memético com o intuito de melhor explorar o espaço de soluções foi proposto para ser avaliado em relação a uma abordagem não híbrida. A partir disso, foi proposto um modelo de coordenação auto-adaptativa dos operadores de busca local, com o objetivo de melhor utilizar as características dos mesmos em relação ao problema aplicado. Em seguida, diferentes valores de raio de perturbação na geração de soluções vizinhas foram testados e combinados, de modo a serem adaptados durante a execução do processo de busca. Finalmente, o algoritmo multimemético auto-adaptativo foi formulado por completo. No próximo capítulo serão apresentados os resultados e as análises obtidos em cada uma das etapas de desenvolvimento.

5 EXPERIMENTOS E RESULTADOS

Nesse capítulo serão apresentados os experimentos realizados aplicando a metodologia desenvolvida. São discutidas as formas de avaliação dos métodos, bem como o conjunto de instâncias adotado e a parametrização necessária para a execução dos algoritmos em cada uma das etapas de desenvolvimento do trabalho. Todos os testes foram executados em um servidor IBM X3650 M5 - Intel Xeon E5-2650V4 30 MB, 4 CPUs, 2.2Ghz, 96 cores/threads, 128 GB ram e disco com 4 TB. Além disso, todo o código foi desenvolvido em linguagem Python.

5.1 Avaliação dos métodos

No processo de atracamento molecular o objetivo é encontrar a melhor conformação de ligação entre uma molécula receptora alvo e uma molécula ligante. Geralmente, as informações sobre essa conformação não são conhecidas, e portanto, informações disponíveis sobre a ligação entre complexos existentes são utilizadas. Os algoritmos de atracamento molecular são guiados por uma função de energia, a qual indica a afinidade de cada conformação, permitindo o ranqueamento destes *poses*. Assim, um valor de energia mais baixo apresenta um estado energicamente favorável para ligação entre as moléculas, mas não necessariamente representa a estrutura obtida experimentalmente. Assim, o *pose* nativo do complexo pode ter um valor de energia de ligação maior do que o apresentado por outra conformação.

Um critério de avaliação comumente utilizado é o desvio médio quadrático - RMSD (*Root Mean Square Deviation*) (ZHANG; SKOLNICK, 2004). Essa métrica permite avaliar a acurácia das soluções obtidas em relação às estruturas cristalográficas ou a métodos de terceiros. Para duas estruturas a e b, com um total de n átomos, o RMSD é definido pela Equação 5.1, onde $a_{ix}, b_{ix}, a_{iy}, b_{iy}, a_{iz}$ e b_{iz} representam as coordenadas xyz dos i-ésimos átomos de a e b.

$$RMSD_{ab} = \sqrt{\frac{\sum_{i=1}^{j}((a_{ix} - b_{ix})^2 + (a_{iy} - b_{iy})^2 + (a_{iz} - b_{iz})^2)}{n}} \tag{5.1}$$

5.2 Instâncias para testes

Para a realização dos experimentos foram selecionados 16 complexos contendo moléculas receptora e ligante. Estas estruturas são baseadas no receptor HIV-protease (DAVIES, 1990) e foram previamente classificadas no trabalho de Morris et al. (MORRIS et al., 2009). A proteína HIV-protease apresenta um sítio de ligação em forma de túnel, característica que facilita a definição do centro da área de busca a partir do átomo central do ligante. O conjunto de teste foi obtido junto ao PDB e classificado conforme o tamanho do ligante em 4 grupos: pequeno, médio, grande, e cíclico. Todos os complexos passaram pela mesma preparação descrita na Seção 4.1. Nessas estruturas foram selecionados até 10 ângulos diedrais para ficarem ativos no processo de rotação interna do ligante. A identificação de cada molécula, seu código PDB e resolução são apresentados na Tabela 5.1.

Tabela 5.1: Conjunto de complexos classificados de acordo com o tamanho do ligante. Identificação das moléculas, referências, respectivos códigos PDB, resolução e número de diedros adotados também são exibidos.

Tamanho	Complexo proteína-ligante	ID PDB	Resolução (Å)	Ref.	Diedros
Pequeno	HIV-1 protease/Hydroxyethylene Isostere inhibitor	1AAQ	2.50	(DREYER et al., 1992)	10
	HIV-1 protease/Macrocyclic Peptidomimetic inhibitor 4	1B6L	1.75	(MARTIN et al., 1999)	8
	HIV-1 protease/Hydroxyethylene-based inhibitor	1HEG	2.20	(MURTHY et al., 1992)	10
	JE-2147-HIV protease	1KZK	1.09	(REILING, 2002)	10
Médio	HIV-1 protease/Macrocyclic Peptidomimetic inhibitor	1B6J	1.85	(MARTIN et al., 1999)	10
	HIV-1 protease/Hydroxyethylene-based inhibitor	1HEF	2.20	(MURTHY et al., 1992)	10
	HIV-1 protease/Multi-drug resistant	1K6P	2.20	(KING et al., 2002)	10
	HIV-1 protease/Lopinavir	1MUI	2.80	(STOLL et al., 2002)	10
Grande	HIV-1 protease/Dihydroethylene inhibitor	1HIV	2.00	(THANKI et al., 1992)	10
	HIV-1 protease/KNI-272	1HPX	2.00	(BALDWIN et al., 1995)	10
	HIV-1 protease/HOE/BAY 793 orthorhombic form	1VIK	2.40	(LANGE-SAVAGE et al., 1997)	10
	HIV-1 protease/C2 symmetric inhibitor	9HVP	2.80	(ERICKSON et al., 1990)	10
Cíclico	HIV-1 protease/AHA006	1AJX	2.00	(BäCKBRO et al., 1997)	10
	HIV-1 protease/XV638	1BV9	2.00	(ALA et al., 1998)	10
	HIV-1 protease/AHA047	1G2K	1.95	(SCHAAL et al., 2002)	10
	HIV-1 protease/Q8261	1HVH	1.80	(JADHAV et al., 1998)	10

Fonte: Do Autor.

5.3 Parametrização

Inicialmente, foi executado o algoritmo BRKGA (detalhado na Seção 4.5.1) para cada uma das instâncias e salvos a população inicial e a conformação inicial do ligante em cada uma das execuções. Conforme descrito na Seção 4.5.1, o ligante tem sua posição e conformação modificados, em relação à estrutura cristalográfica, para não tornar o processo de busca tendencioso. Assim, para que seja justa a comparação entre os algoritmos,

cada execução de cada instância tem o mesmo ponto de partida, a mesma semente. Também, como limite de execução é definido um total de 1.000.000 de avaliações da função de energia.

Em seguida foram definidos os parâmetros utilizados em cada algoritmo. No BRKGA adotou-se os valores recomendados de acordo com aqueles propostos por Gonçalves (GONÇALVES; RESENDE, 2011). O tamanho da população é de 150 indivíduos, dos quais 20% representam o grupo de elite, enquanto que 50% dos indivíduos são gerados no *crossover*, e os demais 30% são obtidos no processo de mutação. A probabilidade de herança do alelo de elite no processo de escolha aleatória é definido entre 0,5 e 0,7. Além disso, considerando que a discretização do espaço de busca por cubos tem um total de 27 regiões, o percentual de indivíduos não-migrantes foi definido em 30% da população. Esse valor respeita a restrição $p_{nm} \leq p_m$, e garante que no mínimo 1 indivíduo estará presente em cada cubo ao longo da execução do algoritmo.

Em relação ao algoritmo memético, os parâmetros das buscas locais foram definidos como 0,50 para o raio de perturbação, fixo nesta primeira etapa. O valor de intensidade de cada busca local foi definido em 500 avaliações, e a taxa de aplicação local/global igual a 0,50 conforme sugerido pelos autores em (MOLINA; LOZANO; HERRERA, 2009; MOLINA et al., 2011). A partir disso, pode-se calcular a quantidade de avaliações globais, que é igual ao definido para as LS, 500. Também, foi definida a quantidade de indivíduos selecionados para aplicação de busca local como igual a 18. Essa definição vem de experiências obtidas em testes e pelo simples cálculo de quantas soluções poderiam ser avaliadas considerando-se que o caso em que somente uma visita completa na vizinhança de uma solução fosse feita. Nesse caso, um indivíduo com 10 diedros (número máximo), teria um total de 14 genes e portanto 28 vizinhos, então 500/28 $\approx$ 18.

Entre os métodos de busca local implementados, somente o *Simulated Annealing* tem parâmetros específicos a serem definidos. O número de passos foi configurado para ter, na primeira etapa, o número de avaliações disponíveis para cada aplicação de LS. Já para a segunda etapa, o parâmetro foi definido em 1/3 do número de avaliações destinado a cada janela de aplicação da busca local. Esse valor foi considerado suficiente para se gastar na melhora de um indivíduo, e também permite a aplicação do encadeamento da aplicação do SA com os demais algoritmos de LS. É importante lembrar que esses são os valores máximos, no entanto, se for detectado que a busca não está melhorando (atingiu um mínimo local), o processo pode encerrar antes de atingir essa quantidade de passos. Esta detecção se dá por controlar os genes modificados, isto é, se cada gene for alterado

em ambas direções e a solução não for aceita em nenhuma das vezes, o processo do SA é então encerrado. Além disso, a temperatura mínima foi definida em 2,5 e a máxima em 25.000, com base em diversos experimentos realizados.

No desenvolvimento do algoritmo MMA poucos parâmetros adicionais foram necessários. O raio de perturbação, que antes era fixo, agora passa a ser definido como um conjunto de valores: 0,10, 0,25 e 0,50. Estes valores foram definidos de forma experimental em testes preliminares com a geração e avaliação de indivíduos aleatórios, e tais valores se mostraram mais atrativos para aplicação no refinamento das soluções. Além disso, os pesos utilizados na função de probabilidades foram definidos em 0,6 para o termo que mensura o benefício do operador de LS, e 0,4 para o termo que mede a taxa de sucesso. Tais valores foram adotados pois considerou-se que o primeiro termo tem ligeira relevância sobre o segundo. O uso de pesos também foi utilizado no trabalho de Jin et al. (JIN; ZHIHUA; WENYIN, 2014).

A definição das restrições das operações realizadas com o ligante também são importantes. A operação de translação está relacionada com a definição do tamanho do espaço de busca, que foi valorado em 11Å para todos os eixos da caixa. Assim, os limites de cada aresta são [-5,5, 5,5]. Já tanto a rotação do ligante, como um todo, e suas rotações internas têm seus valores representados em radianos e vão de $-\pi$ a π, o que representa $[-180, 180]$ em graus. Por exemplo, para um raio de perturbação igual a 0,25, tem-se uma variação de $\approx \pm 28,65^{\circ}$ na rotação da estrutura. Todos os parâmetros e definições discutidos são sumarizados na Tabela 5.2.

Tabela 5.2: Conjunto de parâmetros definidos para o BRKGA e os algoritmos de busca local nas implementações das abordagens memética e multimemética.

Parâmetros gerais		
max_evals	Máximo de avaliações	1.000.000
n_cubes	Número de cubos	27
cube_size	Dimensões do espaço de busca	11x11x11 (Å)
rotation	Intervalo de valores para rotação	$[-\pi, \pi]$
n_diedral	Número máximo de diedros	10
Parâmetros do BRKGA		
p	Tamanho da população	150
p_e	População de Elite	0,20
p_m	População Mutante	0,30
ρ_e	Probabilidade do alelo de elite	0,5 - 0,7
p_{nm}	População não-migrante	0,30
Parâmetros das buscas locais		
r_{pt}	Raio de perturbação	0,50 - Etapa I [0,10, 0,25, 0,50] - Etapa II
i_{LS}	Intensidade de aplicação da LS	500
$r_{L/G}$	Taxa de aplicação local/global	0,50
ir_{LS}	Número de indivíduos para uma LS	18
Parâmetros do *Simulated Annealing*		
n_{steps}	Número máximo de passos	500 - Etapa I 167 - Etapa II
min_t	Temperatura mínima	2,50
max_t	Temperatura máxima	25.000
Parâmetros do algoritmo multimemético		
w_{rfg}	Peso da taxa de ganho do fitness	0,6
$w_{rs_{app}}$	Peso da taxa de sucesso de aplicação	0,4

Fonte: Do Autor.

5.4 Análises - Etapa I

A etapa I engloba o desenvolvimento do algoritmo de busca global baseado no BRKGA, e dos algoritmos de busca local para a composição do algoritmo memético. O intuito nesta fase é avaliar se uma abordagem memética traz vantagens em relação a uma implementação que não utiliza busca local, e caso se confirme, qual dos algoritmos de busca local se mostra mais eficiente na implementação do MA. Assim, os resultados obtidos e as análises realizadas referentes aos mesmos, envolvendo os diferentes passos de execução que fazem parte desta etapa, serão apresentados na sequência.

A Tabela 5.3 apresenta os resultados obtidos em 31 execuções com 1 milhão de avaliações de energia cada para cada combinação de algoritmo e instância. Nas primeiras colunas são identificadas as estruturas testadas e o método aplicado, em seguida são

apresentados o menor valor de energia alcançado (*kcal/mol*) e o respectivo RMSD (Å). As demais colunas apresentam as médias e desvios padrões das métricas para as 31 execuções de cada abordagem. O menor valor de energia alcançado deveria, em tese, representar o ligante de menor RMSD comparado com a estrutura cristalográfica. No entanto, por conta da característica multi-modal da função de energia adotada, e por suas possíveis falhas em não representar a realidade biológica, podem existir soluções com menor RMSD mas valor de energia maior se comparado com o mínimo encontrado.

Tabela 5.3: Resultados obtidos do BRKGA, *Best Improvement*, *First Improvement*, *Stochastic Hill Descent*, *Simulated Annealing*. O menor valor de energia e seu respectivo RMSD são apresentados, bem como suas médias e desvios padrões das 31 execuções. Células em cinza destacam as melhores soluções obtidas em cada instância, e células em azul, as melhores médias.

ID	**Método**	**Melhor solução**		**Média das 31 execuções**	
		Energia	**RMSD**	**Energia**	**RMSD**
1AAQ	BRKGA	36,697	1,135	42,748 ± 5,890	3,118 ± 2,474
	BI	-63,402	1,454	-62,825 ± 0,703	1,812 ± 1,379
	FI	36,728	1,145	38,732 ± 3,117	1,455 ± 1,396
	SHD	36,880	1,132	37,938 ± 0,470	1,093 ± 0,136
	SA	37,036	1,139	39,117 ± 2,682	1,593 ± 1,508
1AJX	BRKGA	-246,846	4,400	-239,679 ± 1,874	12,239 ± 1,502
	BI	-250,801	3,102	-169,539 ± 54,021	9,400 ± 4,183
	FI	-249,351	1,382	-173,483 ± 53,985	8,579 ± 4,705
	SHD	-250,513	0,979	-243,683 ± 4,388	8,591 ± 4,746
	SA	-250,622	1,219	-242,208 ± 3,842	10,318 ± 4,219
1B6J	BRKGA	-328,688	1,107	-316,829 ± 7,495	5,852 ± 2,329
	BI	-324,976	1,417	-156,281 ± 121,733	5,180 ± 2,789
	FI	-328,665	1,116	-156,631 ± 122,768	4,908 ± 2,358
	SHD	-328,700	1,094	-322,425 ± 3,264	5,059 ± 2,758
	SA	-328,266	1,180	-320,807 ± 3,987	5,942 ± 2,216
1B6L	BRKGA	-317,455	3,090	-312,622 ± 3,113	6,641 ± 1,935
	BI	-65,036	3,146	-60,165 ± 1,017	2,508 ± 1,324
	FI	-318,691	2,974	-249,884 ± 111,609	4,130 ± 2,267
	SHD	-319,214	2,800	-316,638 ± 1,835	3,542 ± 1,986
	SA	-318,572	2,955	-315,723 ± 2,344	4,424 ± 2,413
1BV9	BRKGA	-80,066	0,742	-16,327 ± 44,554	7,273 ± 2,857
	BI	-69,071	0,773	-62,516 ± 3,854	6,690 ± 3,394
	FI	-68,984	0,783	-63,072 ± 2,315	7,206 ± 3,206
	SHD	-80,637	0,670	-57,174 ± 18,625	6,825 ± 4,844
	SA	-80,408	0,704	-53,440 ± 22,764	6,897 ± 4,586

Continua na próxima página

Tabela 5.3 – continuação da página anterior

ID	Método	Melhor solução		Média das 31 execuções	
		Energia	RMSD	Energia	RMSD
1G2K	BRKGA	-455,767	1,339	-436,888 ± 9,444	6,037 ± 2,049
	BI	-456,726	0,844	-254,381 ± 140,560	4,405 ± 2,578
	FI	-456,196	2,888	-255,586 ± 142,089	4,195 ± 2,635
	SHD	-456,710	0,881	-452,564 ± 5,603	2,829 ± 2,275
	SA	-456,624	0,781	-450,822 ± 6,384	3,431 ± 2,358
1HEF	BRKGA	175,777	13,476	176,690 ± 0,575	12,254 ± 0,565
	BI	-24,815	10,020	-23,725 ± 0,392	11,943 ± 0,777
	FI	173,658	10,137	175,527 ± 1,008	11,701 ± 0,627
	SHD	173,688	11,426	175,260 ± 1,002	11,790 ± 0,666
	SA	173,688	11,360	175,456 ± 0,867	11,998 ± 0,801
1HEG	BRKGA	357,760	6,911	361,201 ± 1,448	8,672 ± 1,682
	BI	0,893	8,845	128,639 ± 171,284	8,144 ± 1,780
	FI	1,178	8,773	128,629 ± 171,321	7,898 ± 1,669
	SHD	357,184	6,449	358,999 ± 0,592	7,534 ± 1,361
	SA	356,765	5,564	359,114 ± 1,319	7,773 ± 1,824
1HIV	BRKGA	-164,164	2,923	-150,521 ± 3,102	6,646 ± 1,111
	BI	-112,952	7,319	-71,644 ± 7,549	2,892 ± 1,015
	FI	-164,985	3,367	-162,899 ± 3,573	2,774 ± 1,346
	SHD	-164,937	3,410	-163,665 ± 0,819	2,276 ± 0,552
	SA	-165,152	3,388	-163,188 ± 2,671	2,654 ± 1,037
1HPX	BRKGA	-348,396	5,277	-346,813 ± 1,591	5,870 ± 1,044
	BI	-356,473	2,059	-186,002 ± 120,916	5,770 ± 1,320
	FI	-355,669	2,863	-186,040 ± 121,044	5,578 ± 1,015
	SHD	-357,341	3,150	-352,560 ± 3,995	4,039 ± 1,357
	SA	-356,789	2,972	-351,626 ± 3,730	4,659 ± 1,491
1HVH	BRKGA	428,012	8,939	434,158 ± 2,595	10,335 ± 2,225
	BI	-67,692	8,766	-66,588 ± 0,728	8,535 ± 1,299
	FI	427,554	5,827	431,650 ± 2,829	9,385 ± 2,533
	SHD	427,293	8,384	429,985 ± 2,451	8,652 ± 1,622
	SA	427,003	8,417	430,809 ± 2,263	8,657 ± 1,802
1K6P	BRKGA	-382,381	4,419	-375,894 ± 4,853	6,885 ± 1,755
	BI	-382,643	5,399	-228.434 ± 112.202	5.252 ± 1.724
	FI	-384,006	6,426	-236,122 ± 114,480	5,599 ± 1,483
	SHD	-383,783	0,827	-380,573 ± 2,097	5,669 ± 1,778
	SA	-383,077	5,360	-380,127 ± 1,849	6,288 ± 1,561
1KZK	BRKGA	-440,323	1,914	-422,159 ± 17,595	5,680 ± 2,143
	BI	-440,883	0,335	-231,964 ± 151,633	3,428 ± 2,026
	FI	-440,990	0,844	-230,890 ± 150,249	3,441 ± 2,199
	SHD	-441,164	1,536	-438,450 ± 2,990	2,829 ± 1,970
	SA	-441,030	1,551	-435,519 ± 5,892	3,944 ± 2,350

Continua na próxima página

Tabela 5.3 – continuação da página anterior

ID	Método	Melhor solução		Média das 31 execuções	
		Energia	RMSD	Energia	RMSD
1MUI	BRKGA	-36,393	1,826	-26,570 ± 6,681	5,853 ± 3,019
	BI	-99,679	2,051	-38,436 ± 20,209	3,726 ± 2,840
	FI	-99,748	1,901	-98,857 ± 1,029	3,208 ± 2,681
	SHD	-36,794	1,823	-34,989 ± 1,856	1,723 ± 0,679
	SA	-36,566	1,930	-33,668 ± 2,641	2,534 ± 2,206
1VIK	BRKGA	174,700	2,177	228,249 ± 50,900	6,853 ± 2,817
	BI	-75,557	3,657	22,288 ± 126,302	6,699 ± 2,927
	FI	-75,813	2,794	22,266 ± 128,464	4,922 ± 2,914
	SHD	173,464	2,204	186,056 ± 16,519	3,815 ± 2,814
	SA	173,346	2,158	187,988 ± 18,040	4,587 ± 2,926
9HVP	BRKGA	359,987	1,090	372,039 ± 14,288	4,982 ± 3,386
	BI	360,011	1,053	364,812 ± 5,599	4,032 ± 3,632
	FI	-56,665	1,581	-55,895 ± 1,200	2,679 ± 2,404
	SHD	360,256	1,457	362,908 ± 4,197	3,113 ± 3,438
	SA	360,086	1,050	362,945 ± 4,748	2,503 ± 2,461

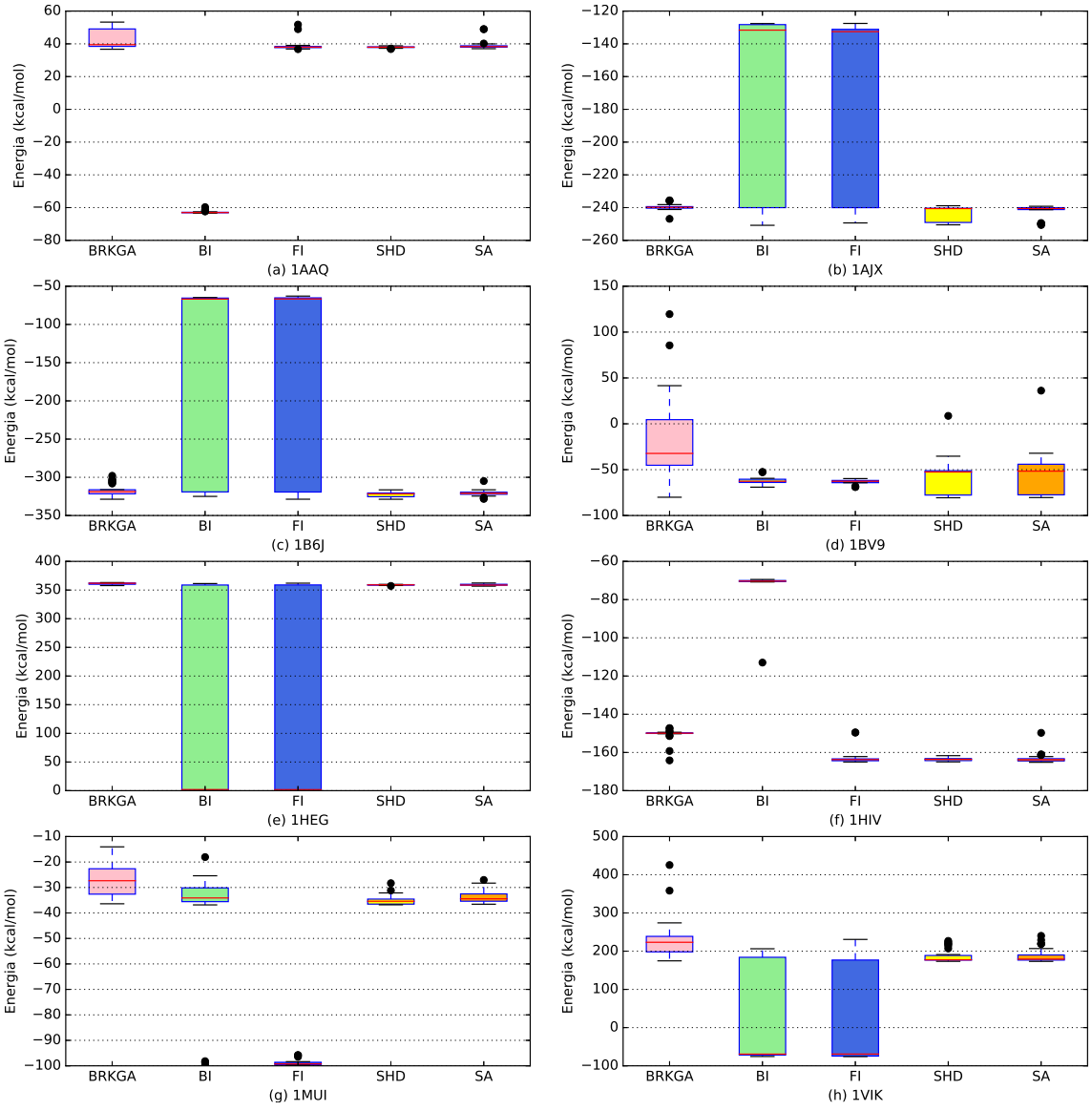
Fonte: Do Autor.

A partir dos resultados é possível destacar que para praticamente todos os casos de teste, os algoritmos meméticos em si obtiveram melhores resultados quando comparados ao BRKGA. Em termos de energia, a abordagem sem uso de busca local não foi melhor em nenhum dos casos, enquanto que para o RMSD, apenas na instância 1HIV foi alcançado um valor superior em relação aos demais métodos, porém a média mostra que esse valor é um *outlier*. Analisando os menores valores de energia obtidos, é obtida uma distribuição dos melhores resultados entre as variantes do algoritmo *Hill climbing*. Na comparação dos menores RMSDs também há uma divisão nos melhores resultados obtidos entre os métodos BI, SA e SHD, com destaque para o último que foi superior em 43% dos casos de teste. Em relação aos valores de médias obtidos, também houve uma distribuição dos melhores resultados para energia e RMSD, entre as variações do algoritmo HC. No entanto, é importante destacar que em metade das instâncias, o método SHD se mostrou superior aos demais.

Apesar de o algoritmo SA não ter obtido os melhores resultados entre os menores valores e as médias obtidos, o mesmo não apresenta uma variação dos valores tão grande quanto os métodos *Best Improvement* e *First Improvement*. Pode-se observar na Tabela 5.3 que em pelo menos 50% das instâncias os métodos BI e FI apresentaram um desvio padrão, para os valores de energia, muito alto em relação aos demais algoritmos. A

Figura 5.1 exibe os diagramas de caixa dos valores de energia para metade das instâncias do conjunto testado: 1AAQ, 1AJX, 1B6J, 1BV9, 1HEG, 1HIV, 1MUI e 1VIK.

Figura 5.1: Diagrama de caixa das estrtururas (a) 1AAQ, (b) 1AJX, (c) 1B6J, (d) 1BV9, (e) 1HEG, (f) 1HIV, (g) 1MUI e (h) 1VIK comparando os valores de energia para os cinco algoritmos implementados.

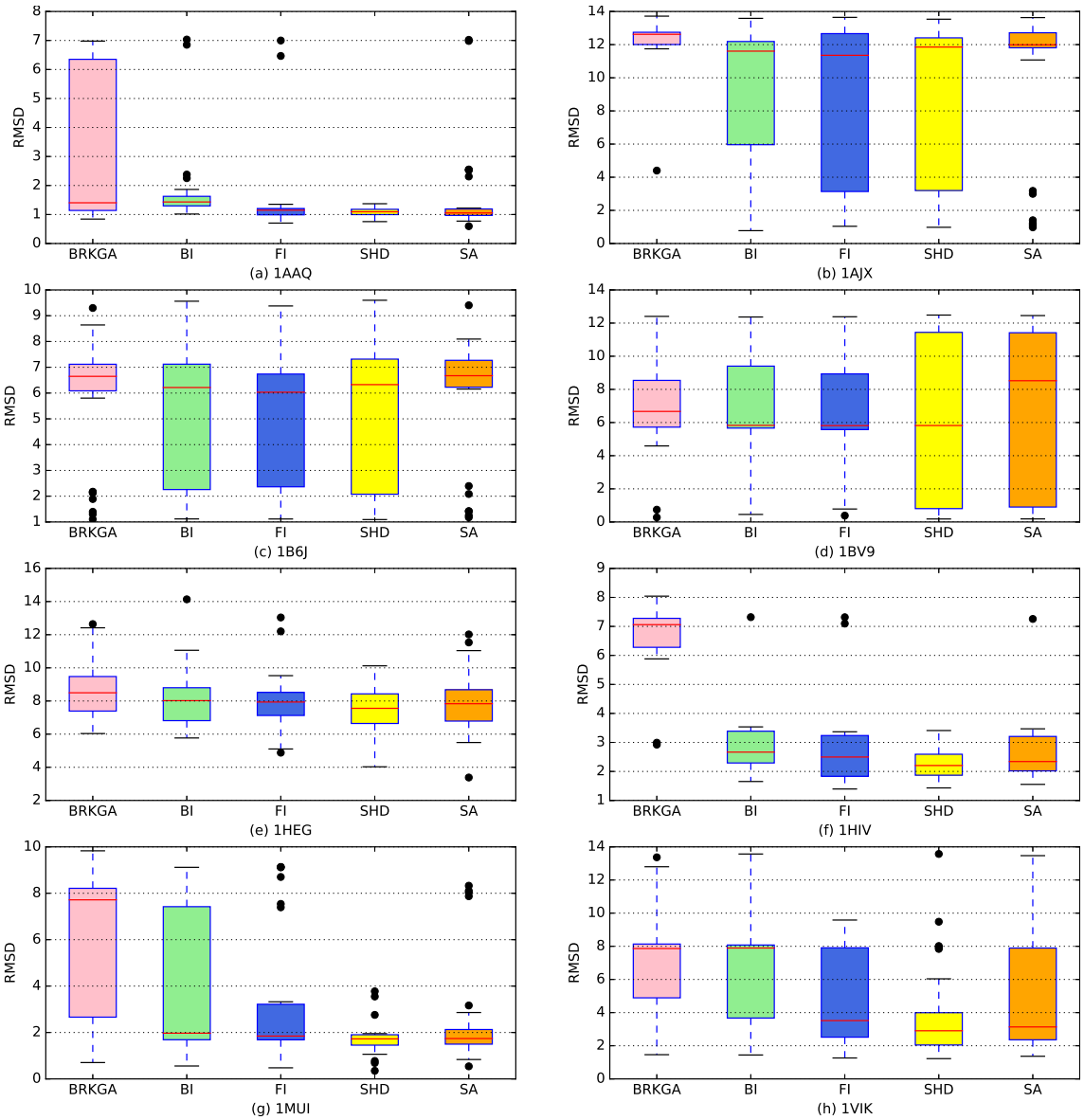


Fonte: Do Autor.

Para a grande maioria dos complexos exibidos, os valores de mediana dos algoritmos com busca local são menores do que as medianas alcançadas apenas com o BRKGA. Constata-se que os métodos SHD e SA apresentam uma distribuição padrão em pelo menos 6 das 8 estruturas aqui mostradas. Além disso, é possível identificar uma maior distribuição dos valores de energia nos métodos BI e FI, tanto para piores resultados, casos 1AJX e 1B6J, quanto para os melhores, como o complexo 1HEG. No entanto, essa

distribuição reflete a rugosidade do espaço de busca trabalhado, isto é, diversos mínimos locais podem ser encontrados pela função de energia, que por sua vez também apresenta característica multi-modal. Assim, tais valores dos métodos BI e FI não necessariamente são ruins quando se analisa a estrutura das soluções obtidas. Na Figura 5.2 são ilustrados os diagramas de caixa dos valores de RMSD obtidos pelos algoritmos sobre as mesmas estruturas analisadas anteriormente.

Figura 5.2: Diagrama de caixa das estruturas (a) 1AAQ, (b) 1AJX, (c) 1B6J, (d) 1BV9, (e) 1HEG, (f) 1HIV, (g) 1MUI e (h) 1VIK comparando os valores de RMSD para os cinco algoritmos implementados.

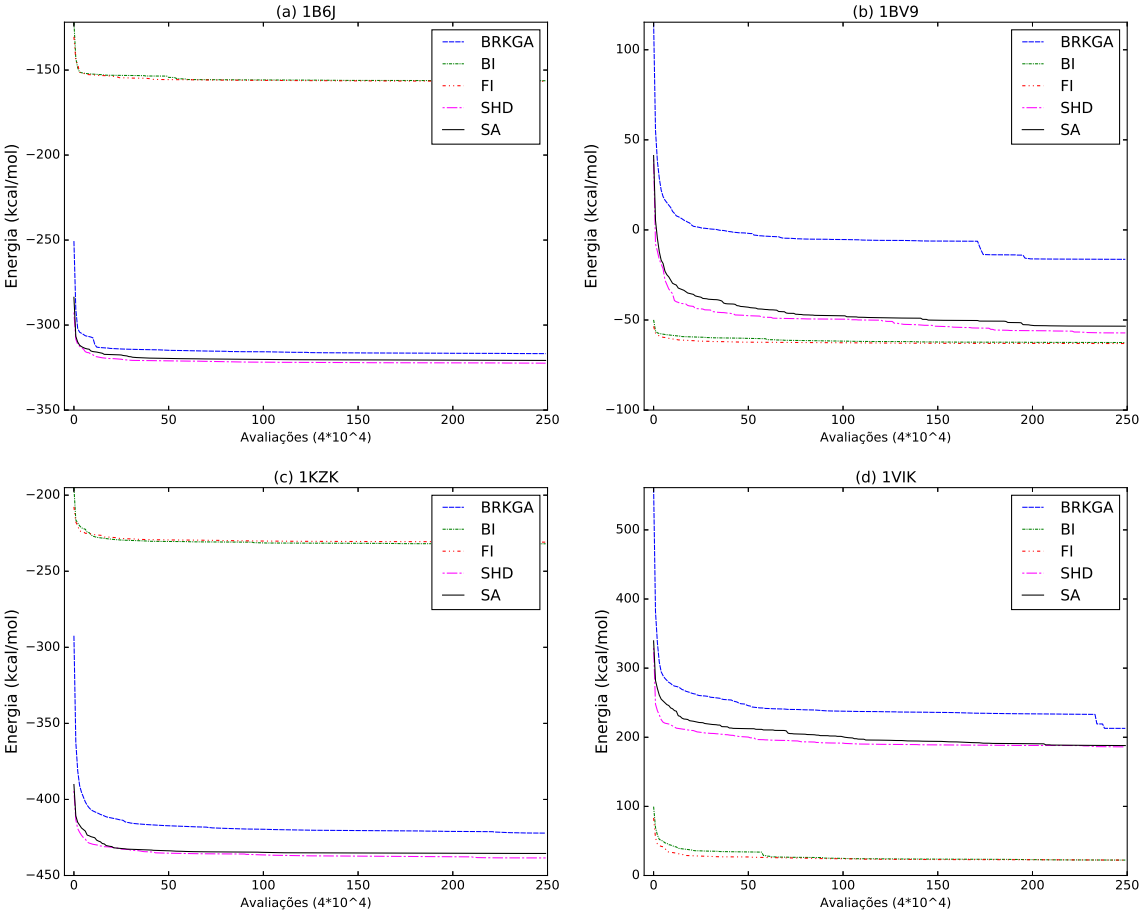


Fonte: Do Autor.

Na Figura 5.2 é possível ver que os métodos BI e FI, embora apresentem uma distribuição maior, seus valores de mediana são menores em relação aos valores do BRKGA.

Os métodos SHD e SA também apresentam uma distribuição mais variada para algumas estruturas abordadas, mas da mesma forma, possuem baixos valores para mediana. Além disso, estes últimos dois algoritmos se mostraram competitivos ao alcançar os melhores valores em pelo menos metade dos casos de teste. Uma outra análise sobre estes resultados é em relação a convergência dos algoritmos implementados. A Figura 5.3 exibe os gráficos de convergência das instâncias 1B6J, 1BV9, 1KZK e 1VIK.

Figura 5.3: Curvas de convergência dos valores de energia obtidos pelos algoritmos para as estruturas (a) 1B6J, (b) 1BV9, (c) 1KZK e (d) 1VIK.



Fonte: Do Autor.

Os gráficos confirmam os resultados já observados anteriormente, os métodos *Best Improvement* e *First Improvement* apresentam comportamento semelhante ao explorar os mínimos locais do espaço de busca. Os algoritmos *Stochastic Hill descent* e *Simulated Annealing* possuem uma curva ligeiramente mais suavizada em comparação com os demais métodos, mostrando que conseguem melhorar o resultado ao longo do processo de otimização. Além disso, pode-se observar que o BRKGA por si só não atinge mínimos de energia como os demais algoritmos. Também, é notável que, de modo geral, todos os métodos apresentam uma rápida convergência no início do processo, em torno de 150 mil

avaliações, e depois tendem a estagnar no valor ótimo alcançado.

Para analisar a diferença significativa dos dados obtidos pelos métodos, foi aplicado o Teste de Kruskal-Wallis (KRUSKAL; WALLIS, 1952), um teste não paramétrico utilizado na comparação de três ou mais amostras independentes. O teste utiliza os valores numéricos das amostras transformados em postos e agrupados num único conjunto de dados. A comparação dos grupos é feita através da média desses postos. Esse teste indica se existe diferença significativa entre pelo menos dois grupos. A hipótese nula é formulada sobre a tendência dos grupos apresentarem valores similares da variável em questão, e o nível de significância para rejeição dessa hipótese é de $\alpha \leq 0.05$.

A Tabela 5.4 exibe os resultados do teste aplicado para os valores de energia e RMSD obtidos pelos algoritmos. Valores maiores do que 0,05 são destacados em cinza na tabela. Pode-se notar que para todas instâncias existe diferença entre os grupos testados para a métrica de energia. Já em relação ao RMSD, apenas 4 instâncias (1B6J, 1BV9, 1HEG e 9HVP) apresentam um *p-value* indicando que não existe diferença entre os algoritmos.

Tabela 5.4: Resultados de aplicação do teste Kruskal-Wallis para os valores de energia e RMSD dos algoritmos. Para cada instância são exibidos os *p-values* resultantes da comparação dos grupos com um nível de significância menor igual a 5%.

ID	**Energia**	**RMSD**	**ID**	**Energia**	**RMSD**
1AAQ	5,38e-19	1,23e-07	1AJX	8,65e-11	4,00e-04
1B6J	2,10e-10	4,33e-01	1B6L	1,89e-17	2,71e-09
1BV9	4,02e-11	9,73e-01	1G2K	8,12e-15	3,99e-05
1HEF	2,13e-20	8,46e-03	1HEG	1,19e-13	1,45e-01
1HIV	5,52e-23	6,81e-14	1HPX	8,71e-15	3,06e-06
1HVH	2,55e-19	1,52e-03	1K6P	1,80e-10	9,96e-03
1KZK	5,20e-13	3,34e-05	1MUI	1,78e-19	2,39e-06
1VIK	3,92e-13	9,11e-05	9HVP	1,69e-16	8,01e-02

Fonte: Do Autor.

Com base na rejeição da hipótese nula para a grande maioria das estruturas, foi aplicado o teste *post hoc* de Dunn (DUNN, 1964) para descobrir quais grupos apresentam diferenças. Este é também um teste não paramétrico aplicado sobre amostras independentes. A Tabela 5.5 exibe os valores de comparação de energia entre os algoritmos.

Tabela 5.5: Análise dos resultados de energia com um nível de significância $p \leq 0.05$. As células em destaque indicam as comparações que não apresentam diferença significativa.

ID	Método	BRKGA	BI	FI	SHD	ID	BRKGA	BI	FI	SHD
1AAQ	BI	0,000	---	---	---	1AJX	0,331	---	---	---
	FI	0,007	0,000	---	---		0,623	1,000	---	---
	SHD	0,002	0,000	1,000	---		0,004	0,000	0,000	---
	SA	1,000	0,000	0,829	0,348		0,090	0,000	0,000	1,000
1B6J	BI	0,036	---	---	---	1B6L	0,000	---	---	---
	FI	0,089	1,000	---	---		1,000	0,000	---	---
	SHD	0,062	0,000	0,000	---		0,001	0,000	0,029	---
	SA	0,844	0,000	0,000	1,000		0,043	0,000	0,549	1,000
1BV9	BI	0,000	---	---	---	1G2K	0,655	---	---	---
	FI	0,000	1,000	---	---		0,814	1,000	---	---
	SHD	0,000	0,698	0,341	---		0,000	0,000	0,000	---
	SA	0,002	0,130	0,053	1,000		0,005	0,000	0,000	1,000
1HEF	BI	0,000	---	---	---	1HEG	0,000	---	---	---
	FI	0,006	0,000	---	---		0,000	1,000	---	---
	SHD	0,000	0,000	1,000	---		0,001	0,015	0,018	---
	SA	0,001	0,000	1,000	1,000		0,001	0,019	0,022	1,000
1HIV	BI	0,020	---	---	---	1HPX	0,219	---	---	---
	FI	0,000	0,000	---	---		0,304	1,000	---	---
	SHD	0,000	0,000	1,000	---		0,001	0,000	0,000	---
	SA	0,000	0,000	1,000	1,000		0,004	0,000	0,000	1,000
1HVH	BI	0,000	---	---	---	1K6P	0,346	---	---	---
	FI	0,137	0,000	---	---		0,729	1,000	---	---
	SHD	0,000	0,000	0,775	---		0,005	0,000	0,000	---
	SA	0,014	0,000	1,000	1,000		0,080	0,000	0,000	1,000
1KZK	BI	0,576	---	---	---	1MUI	0,032	---	---	---
	FI	0,338	1,000	---	---		0,000	0,000	---	---
	SHD	0,000	0,000	0,000	---		0,000	0,789	0,000	---
	SA	0,053	0,000	0,000	1,000		0,010	1,000	0,000	1,000
1VIK	BI	0,000	---	---	---	9HVP	0,509	---	---	---
	FI	0,000	1,000	---	---		0,000	0,000	---	---
	SHD	0,002	0,039	0,010	---		0,885	1,000	0,000	---
	SA	0,013	0,008	0,002	1,000		0,240	1,000	0,000	1,000

Fonte: Do Autor.

Os resultados da análise estatística mostram que numa comparação entre BRKGA e métodos meméticos, em todas instâncias existe um algoritmo de busca local que alcança melhores valores de energia em relação à abordagem não-memética. Destaque para o método SHD que apresenta diferença significativa em 14 das 16 instâncias. Entretanto, SHD e SA não apresentam diferenças para todos os seus resultados. E na comparação entre todos os métodos de busca local também não há predominância de um algoritmo sobre os demais. Da mesma forma, foram comparados e analisados os valores de RMSD entre todos os métodos, conforme exibe a Tabela 5.6.

Tabela 5.6: Análise dos resultados de RMSD com um nível de significância $p \leq 0.05$. As células em destaque indicam as comparações que não apresentam diferença significativa.

ID	Método	BRKGA	BI	FI	SHD	ID	BRKGA	BI	FI	SHD
1AAQ	BI	1.000	---	---	---	1AJX	0.001	---	---	---
	FI	0.015	0.001	---	---		0.006	1.000	---	---
	SHD	0.001	0.000	1.000	---		0.007	1.000	1.000	---
	SA	0.005	0.000	1.000	1.000		0.540	0.483	1.000	1.000
1B6J	BI	1.000	---	---	---	1B6L	0.000	---	---	---
	FI	1.000	1.000	---	---		0.002	0.054	---	---
	SHD	1.000	1.000	1.000	---		0.000	1.000	1.000	---
	SA	1.000	1.000	1.000	1.000		0.001	0.105	1.000	1.000
1BV9	BI	1.000	---	---	---	1G2K	0.109	---	---	---
	FI	1.000	1.000	---	---		0.051	1.000	---	---
	SHD	1.000	1.000	1.000	---		0.000	0.224	0.428	---
	SA	1.000	1.000	1.000	1.000		0.003	1.000	1.000	1.000
1HEF	BI	1.000	---	---	---	1HEG	1.000	---	---	---
	FI	0.058	0.896	---	---		0.761	1.000	---	---
	SHD	0.013	0.313	1.000	---		0.189	1.000	1.000	---
	SA	0.313	1.000	1.000	1.000		0.437	1.000	1.000	1.000
1HIV	BI	0.000	---	---	---	1HPX	1.000	---	---	---
	FI	0.000	1.000	---	---		1.000	1.000	---	---
	SHD	0.000	0.095	1.000	---		0.001	0.000	0.002	---
	SA	0.000	1.000	1.000	1.000		0.142	0.014	0.287	1.000
1HVH	BI	0.007	---	---	---	1K6P	0.012	---	---	---
	FI	0.400	1.000	---	---		0.062	1.000	---	---
	SHD	0.021	1.000	1.000	---		0.241	1.000	1.000	---
	SA	0.003	1.000	1.000	1.000		1.000	0.470	1.000	1.000
1KZK	BI	0.009	---	---	---	1MUI	0.151	---	---	---
	FI	0.002	1.000	---	---		0.034	1.000	---	---
	SHD	0.000	1.000	1.000	---		0.000	0.060	0.245	---
	SA	0.050	1.000	1.000	0.437		0.000	0.619	1.000	1.000
1VIK	BI	1.000	---	---	---	9HVP	1.000	---	---	---
	FI	0.171	0.252	---	---		1.000	1.000	---	---
	SHD	0.001	0.001	1.000	---		0.402	1.000	1.000	---
	SA	0.070	0.107	1.000	1.000		0.110	1.000	0.651	1.000

Fonte: Do Autor.

A análise confirma que, em 75% dos casos de teste, pelo menos um dos métodos de busca local supera o BRKGA. Novamente, destaque para o algoritmo SHD que apresenta diferença significativa em 11 das 16 instâncias testadas. Entretanto, entre os métodos de LS não há diferença de RMSD para a grande maioria das estruturas.

Portanto, a partir dos resultados obtidos nesta primeira etapa do trabalho concluiu-se que a abordagem memética traz sim benefícios em relação à utilização apenas do algoritmo BRKGA. Além disso, a busca local SHD apresentou ter uma ligeira vantagem em relação aos demais métodos. Porém, em uma análise estatística dos dados é possível afirmar que, de modo geral, todos os métodos se mostraram eficientes. Assim, pode-se

justificar uma análise da combinação destas técnicas para melhor explorar as características de cada uma delas e do espaço de busca em si.

5.5 Análises - Etapa II

A etapa II do desenvolvimento deste trabalho é referente à implementação do algoritmo multimemético auto-adaptativo. Nesta fase o objetivo é analisar se a combinação de mais de um algoritmo de busca local e diferentes valores de raio de perturbação em uma mesma execução melhoram os resultados obtidos para o problema de Atracamento Molecular. Para coordenar esse modelo de auto-adaptação foi desenvolvida uma função de probabilidades, descrita na Seção 4.6.1, que mensura o quanto cada parametrização contribui no refinamento das soluções.

Inicialmente, nosso MMA foi aplicado para adaptar os 4 algoritmos de busca, BI, FI, SHD e SA, durante uma mesma execução. Nessa fase, o parâmetro raio de perturbação permaneceu fixo em 0,50. Nossa abordagem, chamada de RPT_050, foi comparada com a implementação SIM (KRASNOGOR; SMITH, 2001), e um simples método aleatório proposto também neste trabalho. A Tabela 5.7 apresenta os resultados obtidos em 31 execuções com 1 milhão de avaliações da função de energia para cada combinação dos algoritmos e instâncias. São detalhados os menores valores de energia e o respectivos RMSDs encontrados por cada método, além dos valores de média e desvio padrão para cada complexo.

Tabela 5.7: Resultados obtidos da aplicação dos algoritmos RPT_050, RANDOM e SIM. O menor valor de energia e o respectivo RMSD são apresentados, bem como suas médias e desvios padrões das 31 *runs*. Células em cinza destacam as melhores soluções obtidas em cada instância, e células em azul, as melhores médias.

ID	Método	Melhor solução		Média das 31 execuções	
		Energia	RMSD	Energia	RMSD
1AAQ	RPT_050	36,907	1,127	38,105 ± 0,607	1,082 ± 0,114
	RANDOM	36,710	0,891	37,914 ± 0,590	1,190 ± 0,284
	SIM	37,217	1,265	38,498 ± 2,047	1,304 ± 1,059
1AJX	RPT_050	-251,118	0,973	-241,961 ± 3,736	10,800 ± 3,612
	RANDOM	-250,214	1,241	-241,520 ± 3,183	10,822 ± 3,595
	SIM	-250,638	1,396	-242,941 ± 4,063	9,339 ± 4,405
1B6J	RPT_050	-329,106	1,107	-321,935 ± 2,401	5,158 ± 2,624
	RANDOM	-327,592	2,014	-322,049 ± 3,851	5,172 ± 2,769
	SIM	-328,144	1,188	-322,086 ± 2,652	5,119 ± 2,447

Continua na próxima página

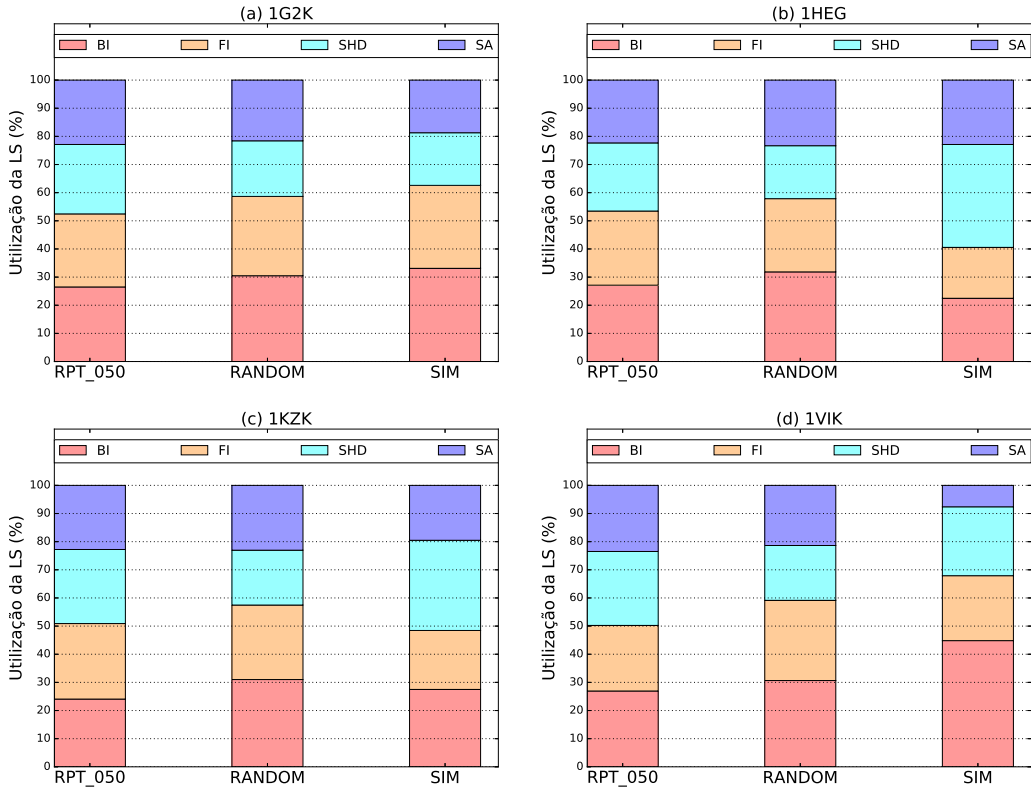
Tabela 5.7 – continuação da página anterior

ID	Método	Melhor solução		Média das 31 execuções	
		Energia	RMSD	Energia	RMSD
1B6L	RPT_050	-318,991	2,758	-315,540 ± 2,251	4,617 ± 2,402
	RANDOM	-318,854	2,749	-316,079 ± 2,308	3,820 ± 1,959
	SIM	-318,634	2,922	-315,457 ± 2,739	4,413 ± 2,096
1BV9	RPT_050	-80,343	0,196	-49,264 ± 15,662	7,362 ± 3,965
	RANDOM	-80,070	0,217	-43,537 ± 27,498	6,762 ± 3,834
	SIM	-80,654	0,673	-46,917 ± 12,278	8,578 ± 3,292
1G2K	RPT_050	-456,699	0,827	-452,307 ± 4,817	3,271 ± 2,384
	RANDOM	-456,639	1,058	-449,484 ± 7,584	3,793 ± 2,544
	SIM	-456,657	1,055	-449,360 ± 7,727	3,848 ± 2,551
1HEF	RPT_050	173,715	11,392	175,633 ± 1,003	11,994 ± 0,463
	RANDOM	173,641	11,399	175,475 ± 0,941	11,861 ± 0,740
	SIM	173,964	11,511	175,699 ± 0,888	11,944 ± 0,494
1HEG	RPT_050	357,536	9,440	359,213 ± 1,039	8,262 ± 1,624
	RANDOM	357,479	9,508	359,693 ± 1,275	8,149 ± 2,187
	SIM	355,376	5,396	358,990 ± 1,233	7,616 ± 1,730
1HIV	RPT_050	-165,054	3,126	-163,039 ± 2,728	2,608 ± 0,877
	RANDOM	-165,228	3,408	-163,630 ± 0,917	2,430 ± 0,601
	SIM	-165,034	3,030	-163,242 ± 1,378	2,630 ± 0,566
1HPX	RPT_050	-356,998	2,727	-350,969 ± 3,816	4,296 ± 1,432
	RANDOM	-357,302	2,972	-350,111 ± 3,968	4,811 ± 1,814
	SIM	-356,659	3,410	-349,605 ± 3,225	4,971 ± 1,316
1HVH	RPT_050	427,548	8,414	432,148 ± 2,819	9,393 ± 2,615
	RANDOM	427,482	8,389	431,016 ± 2,445	9,056 ± 2,029
	SIM	427,511	8,383	430,878 ± 2,402	8,723 ± 1,498
1K6P	RPT_050	-383,463	0,338	-380,396 ± 2,416	5,340 ± 1,955
	RANDOM	-383,381	5,262	-380,670 ± 2,019	5,822 ± 1,895
	SIM	-383,629	4,321	-380,436 ± 2,227	5,486 ± 1,597
1KZK	RPT_050	-441,030	1,534	-436,945 ± 6,119	3,236 ± 1,931
	RANDOM	-441,020	1,554	-436,927 ± 3,883	3,619 ± 2,198
	SIM	-441,095	1,534	-436,114 ± 7,421	3,264 ± 2,191
1MUI	RPT_050	-36,682	1,956	-32,745 ± 3,346	3,590 ± 3,021
	RANDOM	-36,537	1,653	-32,095 ± 3,704	3,943 ± 3,336
	SIM	-36,691	1,646	-32,564 ± 3,016	3,387 ± 2,539
1VIK	RPT_050	173,935	2,150	192,759 ± 22,209	5,016 ± 3,146
	RANDOM	175,244	1,589	186,453 ± 16,547	3,802 ± 2,166
	SIM	173,329	2,290	193,747 ± 22,998	5,118 ± 3,177
9HVP	RPT_050	360,070	1,104	363,295 ± 4,577	3,143 ± 3,201
	RANDOM	360,058	1,094	363,726 ± 5,142	3,295 ± 3,358
	SIM	360,020	1,042	363,853 ± 5,280	3,327 ± 3,491

Fonte: Do Autor.

Em termos de energia, os melhores valores foram obtidos pelo método SIM em quase metade das instâncias, no entanto, os métodos RPT_050 e RANDOM apresentam as melhores médias. Já em termos de RMSD, os menores valores foram alcançados pelo método multimemético em metade dos casos de teste e melhores médias em 44% deles. De modo geral, o algoritmo RPT_050 proposto se mostrou competitivo na melhora dos valores de energia, e ainda apresentou bons resultados em termos estruturais. Foi analisada ainda a taxa de aplicação dos método de busca local em cada algoritmo para cada instância. A Figura 5.4 exibe gráficos de barras empilhadas com o percentual de cada algoritmo de busca local utilizado por cada método nas instâncias 1G2K, 1HEG, 1KZK, e 1VIK.

Figura 5.4: Percentual do uso dos algoritmos de busca local por cada método nas estruturas: (a) 1G2K, (b) 1HEG, (c) 1KZK e (d) 1VIK.



Fonte: Do Autor.

Os gráficos nos mostram a distribuição de aplicação de cada algoritmo de LS em cada técnica auto-adaptativa. Na instância 1G2K, o método RPT_050 se destacou com uma aplicação do BI de 26,5% e do FI com 26%. Também, na instância 1KZK, o método RPT_050 aplicou os métodos FI e SHD com cerca de 26,5% das avaliações cada. Já na instância 1HEG, o método SIM obteve bons resultados com uma aplicação de 37% do método SHD. E para o complexo 1VIK, a variação aleatória foi superior ao aplicar os

algoritmos BI e FI com 30,7% e 28,5% respectivamente. Percebe-se que, de modo geral, uma maior aplicação do algoritmo SHD rende bons resultados, e que altas aplicações dos métodos BI ou FI, por exemplo, podem não atingir os mínimos desejados.

A partir dos resultados obtidos com o algoritmo multimemético RPT_050, o qual, se mostrou eficiente em melhorar os valores de energia, decidiu-se testar esta versão auto-adaptativa com diferentes valores de raio de perturbação. Nas etapas anteriores este parâmetro foi considerado com valor fixo igual a 0,50, já nesta fase são aplicados também os valores 0,10 e 0,25. A ideia é verificar se diferentes modificações nos genes podem melhorar ou não a performance da busca local aplicada. Assim, o algoritmo multimemético foi executado da mesma forma que na etapa anterior, com a combinação dos métodos de LS, só que com diferentes valores de r_{pt}. A Tabela 5.8 exibe a comparação das 3 versões (RPT_010, RPT_025, RPT_050) do MMA proposto.

Tabela 5.8: Resultados obtidos do MMA com a variação dos 3 valores de raio de perturbação: 0,10 (RPT_010), 0,25 (RPT_025) e 0,50 (RPT_050). O menor valor de energia e o respectivo RMSD são apresentados, bem como suas médias e desvios padrões das 31 *runs*. Células em cinza destacam as melhores soluções obtidas em cada instância, e células em azul, as melhores médias.

ID	Método	Melhor solução		Média das 31 execuções	
		Energia	RMSD	Energia	RMSD
1AAQ	RPT_010	36,681	1,124	39,151 ± 4,125	1,948 ± 1,925
	RPT_025	36,780	1,104	38,011 ± 2,066	1,293 ± 1,025
	RPT_050	36,907	1,127	38,105 ± 0,607	1,082 ± 0,114
1AJX	RPT_010	-251,185	1,391	-242,282 ± 3,626	10,705 ± 3,469
	RPT_025	-251,197	0,596	-243,045 ± 4,476	9,510 ± 4,359
	RPT_050	-251,118	0,973	-241,961 ± 3,736	10,800 ± 3,612
1B6J	RPT_010	-329,480	1,123	-323,471 ± 2,887	5,049 ± 2,654
	RPT_025	-328,135	2,213	-322,760 ± 3,032	5,278 ± 2,540
	RPT_050	-329,106	1,107	-321,935 ± 2,401	5,158 ± 2,624
1B6L	RPT_010	-319,419	2,926	-315,643 ± 2,601	4,972 ± 2,101
	RPT_025	-319,444	2,522	-317,504 ± 1,761	3,328 ± 1,839
	RPT_050	-318,991	2,758	-315,540 ± 2,251	4,617 ± 2,402
1BV9	RPT_010	-80,826	0,307	-34,468 ± 36,582	7,055 ± 2,992
	RPT_025	-80,791	0,248	-48,993 ± 20,715	7,531 ± 4,010
	RPT_050	-80,343	0,196	-49,264 ± 15,662	7,362 ± 3,965
1G2K	RPT_010	-456,687	0,780	-442,502 ± 8,815	5,612 ± 2,217
	RPT_025	-456,717	1,055	-451,086 ± 6,879	3,483 ± 2,280
	RPT_050	-456,699	0,827	-452,307 ± 4,817	3,271 ± 2,384

Continua na próxima página

Tabela 5.8 – continuação da página anterior

ID	Método	Melhor solução		Média das 31 execuções	
		Energia	RMSD	Energia	RMSD
1HEF	RPT_010	173,335	9,968	174,949 ± 1,355	11,777 ± 0,669
	RPT_025	173,660	11,395	174,624 ± 0,880	11,662 ± 0,621
	RPT_050	173,715	11,392	175,633 ± 1,003	11,994 ± 0,463
1HEG	RPT_010	356,014	4,953	359,028 ± 1,482	8,262 ± 2,095
	RPT_025	356,828	5,732	359,245 ± 1,138	7,899 ± 1,877
	RPT_050	357,536	9,440	359,213 ± 1,039	8,262 ± 1,624
1HIV	RPT_010	-165,084	3,010	-160,625 ± 6,329	3,685 ± 2,216
	RPT_025	-165,585	3,427	-164,011 ± 1,019	2,650 ± 0,556
	RPT_050	-165,054	3,126	-163,039 ± 2,728	2,608 ± 0,877
1HPX	RPT_010	-355,419	1,237	-348,558 ± 1,588	5,648 ± 1,255
	RPT_025	-357,160	3,456	-350,176 ± 3,448	5,089 ± 1,319
	RPT_050	-356,998	2,727	-350,969 ± 3,816	4,296 ± 1,432
1HVH	RPT_010	427,250	8,880	431,777 ± 2,386	9,223 ± 2,261
	RPT_025	427,135	8,372	431,128 ± 2,642	9,319 ± 2,287
	RPT_050	427,548	8,414	432,148 ± 2,819	9,393 ± 2,615
1K6P	RPT_010	-384,307	4,117	-380,652 ± 2,434	5,691 ± 1,770
	RPT_025	-383,852	4,111	-380,968 ± 2,118	5,782 ± 1,875
	RPT_050	-383,463	0,338	-380,396 ± 2,416	5,340 ± 1,955
1KZK	RPT_010	-441,119	1,554	-433,123 ± 9,903	4,712 ± 2,314
	RPT_025	-441,121	0,843	-436,987 ± 6,039	3,655 ± 2,375
	RPT_050	-441,030	1,534	-436,945 ± 6,119	3,236 ± 1,931
1MUI	RPT_010	-36,833	1,959	-32,579 ± 4,639	4,240 ± 3,129
	RPT_025	-36,754	1,790	-33,423 ± 3,417	3,616 ± 2,821
	RPT_050	-36,682	1,956	-32,745 ± 3,346	3,590 ± 3,021
1VIK	RPT_010	173,003	2,163	194,467 ± 16,849	6,150 ± 3,275
	RPT_025	172,994	2,160	192,218 ± 21,084	4,851 ± 2,567
	RPT_050	173,935	2,150	192,759 ± 22,209	5,016 ± 3,146
9HVP	RPT_010	359,960	1,085	367,330 ± 9,096	5,582 ± 3,893
	RPT_025	359,981	1,077	361,852 ± 3,851	2,149 ± 2,417
	RPT_050	360,070	1,104	363,295 ± 4,577	3,143 ± 3,201

Fonte: Do Autor.

A partir dos resultados obtidos destaca-se que, em termos de energia, os métodos RPT_010 e RPT_25 obtiveram os melhores valores em metade das instâncias cada. Porém, quando avaliadas as médias de energia alcançadas, o método RPT_25 foi superior em 66% dos casos de teste. Em relação ao RMSD das estruturas, os melhores valores foram distribuídos entre os 3 métodos, sem que algum deles se sobressaísse, no entanto, a abordagem RPT_050 obteve melhores médias em 43% dos complexos.

A significância dos resultados obtidos foi analisada aplicando o teste de Kruskal-

Wallis. A Tabela 5.9 exibe os *p-values* para comparações de energia e RMSD entre os algoritmos executados. Valores maiores do que 0,05, destacados em cinza na tabela, indicam que não existe diferença significativa entre qualquer par de método.

Tabela 5.9: Resultados de aplicação do teste Kruskal-Wallis para os valores de energia e RMSD das versões com variação no raio de perturbação no algoritmo multimemético. Para cada instância são exibidos os *p-values* resultantes da comparação dos grupos com um nível de significância menor igual a 5%.

ID	Energia	RMSD	ID	Energia	RMSD
1AAQ	1,27e-02	9,53e-02	1AJX	6,72e-02	4,31e-02
1B6J	8,14e-02	9,48e-01	1B6L	8,21e-04	2,01e-02
1BV9	9,82e-02	6,98e-01	1G2K	3,43e-04	1,42e-03
1HEF	2,14e-03	9,47e-03	1HEG	3,76e-01	5,03e-01
1HIV	2,58e-01	3,04e-01	1HPX	5,49e-01	1,88e-03
1HVH	3,01e-01	9,92e-01	1K6P	7,30e-01	7,45e-01
1KZK	3,77e-01	5,64e-02	1MUI	4,10e-01	6,62e-02
1VIK	7,77e-01	5,63e-01	9HVP	6,07e-02	3,07e-02

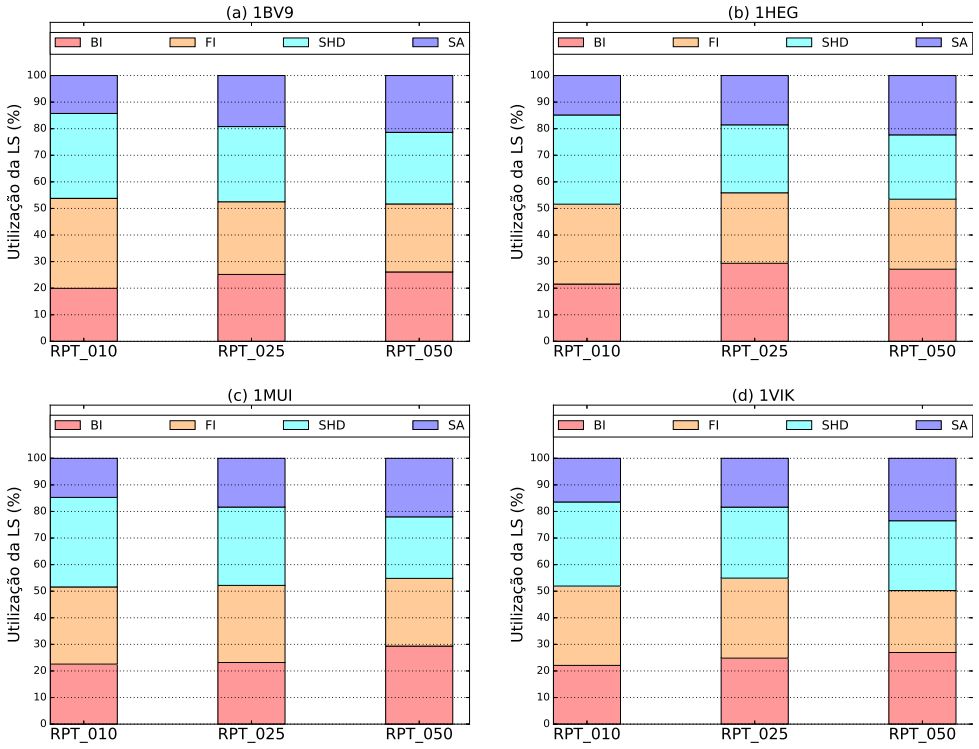
Fonte: Do Autor.

Os resultados nos mostram que a grande maioria das instâncias não apresentam significância na diferença dos valores obtidos pelos métodos comparados. Somente nos complexos 1B6L, 1G2K e 1HEF que ambos valores, de energia e RMSD, se mostram significativamente diferentes, mas sem que algum método se destaque entre eles. Dados os resultados, optou-se por não aplicar o teste Dunn para verificar quais grupos se diferenciam uns dos outros.

Assim como na etapa anterior, também analisou-se a taxa de utilização dos algoritmos de busca local em cada variação do algoritmo aplicado. A Figura 5.5 exibe os gráficos de barras empilhadas ilustrando o percentual de aplicação de cada algoritmo de busca local para cada método proposto nas instâncias 1BV9, 1HEG, 1MUI e 1VIK.

Os resultados mostram que, de forma geral, os métodos com melhores resultados utilizaram uma ou duas buscas locais com cerca de 30% do esforço total em cada uma. Para a instância 1BV9, o método RPT_010 obteve melhor média de RMSD aplicando os algoritmos FI (34%) e SHD (32%), enquanto que o método RPT_050 atingiu melhor média de energia distribuindo a aplicação dos algoritmos BI, FI e SHD com cerca de 26% cada. Na estrutura 1HEG o método RPT_010 se destacou aplicando os algoritmos FI e SHD com 30% e 33,5% respectivamente. Já para a instância 1MUI, o método RPT_025 obteve melhor média de energia aplicando os algoritmos FI e SHD com cerca de 29% cada, e o método RPT_050 conseguiu melhor média de RMSD aplicando 29,3% do esforço no algoritmo BI. E no caso de teste 1VIK, o método de destaque foi o RPT_025

Figura 5.5: Percentual do uso dos algoritmos de busca local por cada método nas estruturas: (a) 1BV9, (b) 1HEG, (c) 1MUI e (d) 1VIK.



Fonte: Do Autor.

com aplicação dos algoritmos FI (30%) e SHD (26,7%).

Na fase em que o MMA foi comparado com os métodos RANDOM e SIM, o parâmetro r_{pt} foi configurado em 0,5, e os métodos de busca local que se destacaram foram as variações do algoritmo *Hill climbing*. Da mesma forma, na fase de teste dos diferentes valores de raio de busca, podemos ver que estas diferentes combinações de aplicação das buscas locais e raio de perturbação são responsáveis pelas melhores soluções. Portanto, uma combinação de diferentes algoritmos de LS e diferentes valores de parâmetro da busca, numa mesma execução, passa a ser uma abordagem a ser analisada.

A partir dessa distribuição das melhores soluções passou-se a considerar a auto-adaptação do valor de raio de perturbação, considerando o *pool* de valores igual a [0,10, 0,25, 0,50]. A função de probabilidades que já mensurava o benefício de cada algoritmo de busca local, agora também é aplicada na avaliação do parâmetro de raio de perturbação. As avaliações do algoritmo de LS e r_{pt} são feitas de forma independente, isto é, um mesmo algoritmo é avaliado quando aplicado com qualquer valor de raio, e este, da mesma forma, é avaliado independente do algoritmo utilizado. Com esta abordagem, diferentes valores de r_{pt} produzem diferentes níveis de refinamento, e essa combinação pode ser benéfica para alcançar resultados melhores ao longo do processo de busca.

Os resultados da versão final do MMA proposto são exibidos na Tabela 5.10 e comparados com algoritmos das etapas anteriores, BRKGA (Seção 4.5.1) e variações do MA (Seção 4.5.2).

Tabela 5.10: Resultados de comparação dos algoritmos BRKGA, SHD, SA, RPT_050 e a versão final do MMA. Os menores valores de energia e o RMSD associado são apresentados, assim como suas médias e desvios padrões das 31 execuções de cada algoritmo. Células em cinza destacam as melhores soluções obtidas em cada instância, e células em azul, as melhores médias.

ID	**Método**	**Melhor solução**		**Média das 31 execuções**	
		Energia	**RMSD**	**Energia**	**RMSD**
1AAQ	BRKGA	36.697	1.135	42.748 ± 5.890	3.118 ± 2.474
	SHD	36.880	1.132	37.938 ± 0.470	1.093 ± 0.136
	SA	37.036	1.139	39.117 ± 2.682	1.593 ± 1.508
	RPT_050	36.907	1.127	38.105 ± 0.607	1.082 ± 0.114
	MMA	36.718	1.240	38.759 ± 3.728	1.641 ± 1.698
1AJX	BRKGA	-246.846	4.400	-239.679 ± 1.874	12.239 ± 1.502
	SHD	-250.513	0.979	-243.683 ± 4.388	8.591 ± 4.746
	SA	-250.622	1.219	-242.208 ± 3.842	10.318 ± 4.219
	RPT_050	-251.118	0.973	-241.961 ± 3.736	10.800 ± 3.612
	MMA	-250.886	3.082	-243.002 ± 4.555	9.422 ± 4.569
1B6J	BRKGA	-328.688	1.107	-316.829 ± 7.495	5.852 ± 2.329
	SHD	-328.700	1.094	-322.425 ± 3.264	5.059 ± 2.758
	SA	-328.266	1.180	-320.807 ± 3.987	5.942 ± 2.216
	RPT_050	-329.106	1.107	-321.935 ± 2.401	5.158 ± 2.624
	MMA	-329.000	1.127	-322.167 ± 3.891	5.630 ± 2.841
1B6L	BRKGA	-317.455	3.090	-312.622 ± 3.113	6.641 ± 1.935
	SHD	-319.214	2.800	-316.638 ± 1.835	3.542 ± 1.986
	SA	-318.572	2.955	-315.723 ± 2.344	4.424 ± 2.413
	RPT_050	-318.991	2.758	-315.540 ± 2.251	4.617 ± 2.402
	MMA	-319.555	2.482	-315.656 ± 2.245	5.097 ± 2.429
1BV9	BRKGA	-80.066	0.742	-16.327 ± 44.554	7.273 ± 2.857
	SHD	-80.637	0.670	-57.174 ± 18.625	6.825 ± 4.844
	SA	-80.408	0.704	-53.440 ± 22.764	6.897 ± 4.586
	RPT_050	-80.343	0.196	-49.264 ± 15.662	7.362 ± 3.965
	MMA	-80.811	0.707	-53.518 ± 16.200	7.484 ± 4.364
1G2K	BRKGA	-455.767	1.339	-436.888 ± 9.444	6.037 ± 2.049
	SHD	-456.710	0.881	-452.564 ± 5.603	2.829 ± 2.275
	SA	-456.624	0.781	-450.822 ± 6.384	3.431 ± 2.358
	RPT_050	-456.699	0.827	-452.307 ± 4.817	3.271 ± 2.384
	MMA	-456.782	1.100	-451.144 ± 7.353	3.209 ± 2.523

Continua na próxima página

Tabela 5.10 – continuação da página anterior

ID	Método	Melhor solução		Média das 31 execuções	
		Energia	RMSD	Energia	RMSD
1HEF	BRKGA	175.777	13.476	176.690 ± 0.575	12.254 ± 0.565
	SHD	173.688	11.426	175.260 ± 1.002	11.790 ± 0.666
	SA	173.688	11.360	175.456 ± 0.867	11.998 ± 0.801
	RPT_050	173.715	11.392	175.633 ± 1.003	11.994 ± 0.463
	MMA	173.582	11.403	174.806 ± 1.188	11.664 ± 0.335
1HEG	BRKGA	357.760	6.911	361.201 ± 1.448	8.672 ± 1.682
	SHD	357.184	6.449	358.999 ± 0.592	7.534 ± 1.361
	SA	356.765	5.564	359.114 ± 1.319	7.773 ± 1.824
	RPT_050	357.536	9.440	359.213 ± 1.039	8.262 ± 1.624
	MMA	356.907	9.525	358.989 ± 1.112	8.190 ± 1.491
1HIV	BRKGA	-164.164	2.923	-150.521 ± 3.102	6.646 ± 1.111
	SHD	-164.937	3.410	-163.665 ± 0.819	2.276 ± 0.552
	SA	-165.152	3.388	-163.188 ± 2.671	2.654 ± 1.037
	RPT_050	-165.054	3.126	-163.039 ± 2.728	2.608 ± 0.877
	MMA	-165.125	3.380	-163.346 ± 3.583	2.851 ± 1.389
1HPX	BRKGA	-348.396	5.277	-346.813 ± 1.591	5.870 ± 1.044
	SHD	-357.341	3.150	-352.560 ± 3.995	4.039 ± 1.357
	SA	-356.789	2.972	-351.626 ± 3.730	4.659 ± 1.491
	RPT_050	-356.998	2.727	-350.969 ± 3.816	4.296 ± 1.432
	MMA	-357.495	3.520	-349.827 ± 3.396	5.085 ± 1.173
1HVH	BRKGA	428.012	8.939	434.158 ± 2.595	10.335 ± 2.225
	SHD	427.293	8.384	429.985 ± 2.451	8.652 ± 1.622
	SA	427.003	8.417	430.809 ± 2.263	8.657 ± 1.802
	RPT_050	427.548	8.414	432.148 ± 2.819	9.393 ± 2.615
	MMA	427.953	8.413	431.740 ± 2.498	9.059 ± 2.792
1K6P	BRKGA	-382.381	4.419	-375.894 ± 4.853	6.885 ± 1.755
	SHD	-383.783	0.827	-380.573 ± 2.097	5.669 ± 1.778
	SA	-383.077	5.360	-380.127 ± 1.849	6.288 ± 1.561
	RPT_050	-383.463	0.338	-380.396 ± 2.416	5.340 ± 1.955
	MMA	-384.233	4.071	-381.266 ± 1.597	5.727 ± 1.930
1KZK	BRKGA	-440.323	1.914	-422.159 ± 17.595	5.680 ± 2.143
	SHD	-441.164	1.536	-438.450 ± 2.990	2.829 ± 1.970
	SA	-441.030	1.551	-435.519 ± 5.892	3.944 ± 2.350
	RPT_050	-441.030	1.534	-436.945 ± 6.119	3.236 ± 1.931
	MMA	-441.190	1.539	-437.009 ± 6.579	3.469 ± 2.308
1MUI	BRKGA	-36.393	1.826	-26.570 ± 6.681	5.853 ± 3.019
	SHD	-36.794	1.823	-34.989 ± 1.856	1.723 ± 0.679
	SA	-36.566	1.930	-33.668 ± 2.641	2.534 ± 2.206
	RPT_050	-36.682	1.956	-32.745 ± 3.346	3.590 ± 3.021
	MMA	-36.846	1.951	-33.825 ± 3.377	3.352 ± 2.749

Continua na próxima página

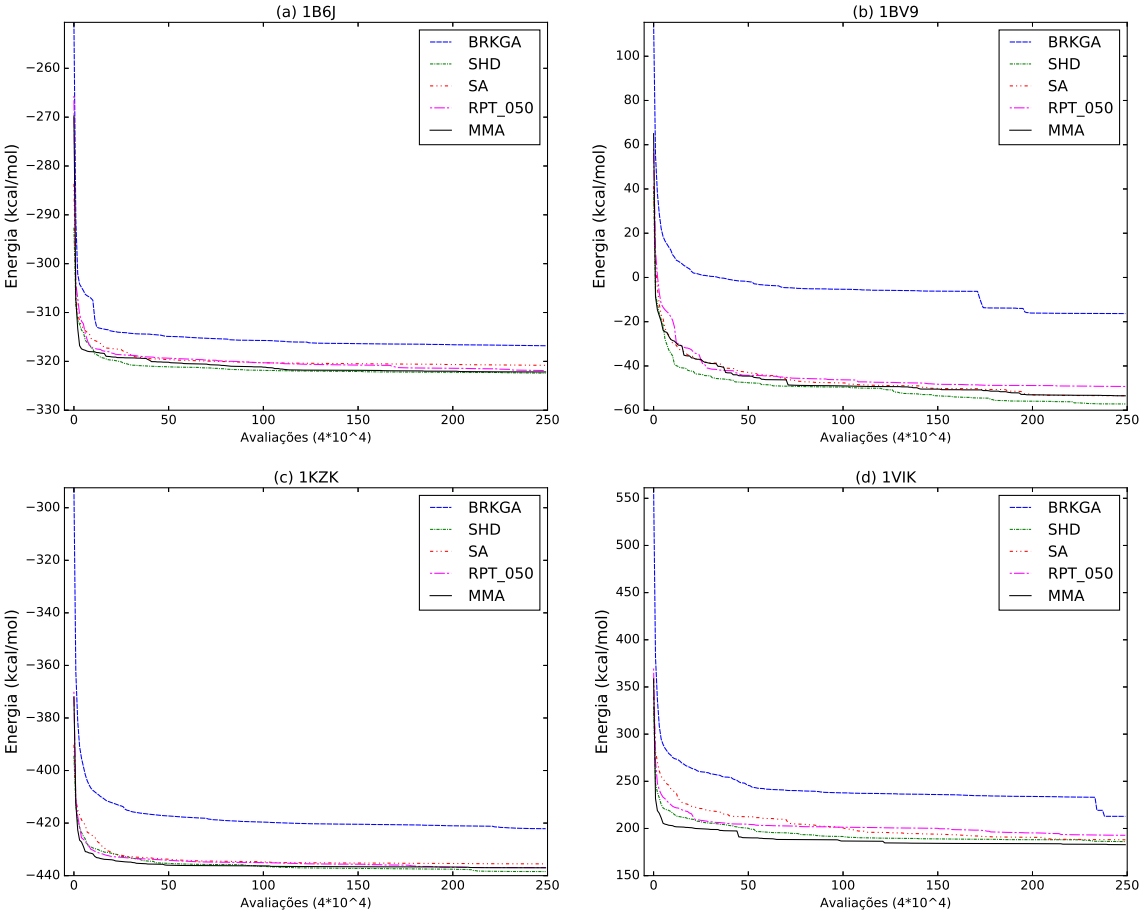
Tabela 5.10 – continuação da página anterior

ID	Método	Melhor solução		Média das 31 execuções	
		Energia	RMSD	Energia	RMSD
1VIK	BRKGA	174.700	2.177	228.249 ± 50.900	6.853 ± 2.817
	SHD	173.464	2.204	186.056 ± 16.519	3.815 ± 2.814
	SA	173.346	2.158	187.988 ± 18.040	4.587 ± 2.926
	RPT_050	173.935	2.150	192.759 ± 22.209	5.016 ± 3.146
	MMA	173.345	2.147	182.827 ± 13.129	4.203 ± 2.658
9HVP	BRKGA	359.987	1.090	372.039 ± 14.288	4.982 ± 3.386
	SHD	360.256	1.457	362.908 ± 4.197	3.113 ± 3.438
	SA	360.086	1.050	362.945 ± 4.748	2.503 ± 2.461
	RPT_050	360.070	1.104	363.295 ± 4.577	3.143 ± 3.201
	MMA	359.972	1.112	363.167 ± 4.779	3.357 ± 3.395

Fonte: Do Autor.

Os resultados da comparação entre os diferentes algoritmos implementados apresentam uma melhoria obtida à medida que o processo de desenvolvimento foi avançando. Em termos de energia, os melhores valores foram obtidos pelo MMA em 62% dos complexos testados, mas considerando as médias obtidas, o algoritmo SHD foi superior em 75% das instâncias, enquanto que a versão multimemética nos 25% restantes. Já em relação ao RMSD, os melhores resultados se dividiram entre SHD, SA e RPT_050 com mais instâncias, mas quando considerada a média, o algoritmo SHD, novamente, se destaca em 75% dos complexos avaliados. A partir disso, foram analisadas as curvas de convergência destes algoritmos nas instâncias 1B6J, 1BV9, 1KZK e 1VIK, conforme ilustrado na Figura 5.6.

Figura 5.6: Curvas de convergência dos valores de energia obtidos pelos algoritmos BRKGA, SHD, SA, RPT_050 e MMA para as estruturas (a) 1B6J, (b) 1BV9, (c) 1KZK e (d) 1VIK.



Fonte: Do Autor.

Os gráficos nos mostram que os algoritmos SHD e MMA apresentam uma curva mais suavizada e conseguem atingir melhores valores de energia ao longo do processo de busca. Por outro lado, o algoritmo BRKGA apresenta uma curva com rápida estagnação para todos os complexos analisados. Para confirmar a veracidade destes dados, o teste de Kruskal-Wallis foi aplicado para verificar se existem diferenças significativas. A Tabela 5.11 apresenta os resultados de energia e RMSD, onde as células em cinza indicam que não há diferença entre qualquer algoritmo avaliado.

Tabela 5.11: Resultados de aplicação do teste Kruskal-Wallis para os valores de energia e RMSD dos algoritmos. Para cada instância são exibidos os *p-values* resultantes da comparação dos grupos com um nível de significância menor igual a 5%.

ID	Energia	RMSD	ID	Energia	RMSD
1AAQ	1,58e-08	7,62e-04	1AJX	3,18e-06	1,81e-03
1B6J	7,64e-04	6,31e-01	1B6L	3,70e-06	1,03e-05
1BV9	1,51e-06	9,80e-01	1G2K	1,82e-10	1,89e-05
1HEF	1,21e-09	3,83e-04	1HEG	2,75e-08	1,06e-01
1HIV	6,24e-14	1,62e-13	1HPX	7,09e-08	1,07e-04
1HVH	8,15e-07	1,56e-02	1K6P	3,76e-07	1,88e-02
1KZK	7,89e-08	5,18e-05	1MUI	1,37e-08	1,04e-05
1VIK	1,65e-08	1,19e-03	9HVP	5,14e-05	2,09e-01

Fonte: Do Autor.

Assim como na primeira etapa, aqui há diferença em todas as instâncias para os valores de energia, e somente em 4 delas (1B6J, 1BV9, 1HEG e 9HVP) não há diferença significante nos dados de RMSD. Para verificar quais algoritmos apresentam diferenças estatísticas, aplicou-se o teste Dunn. A Tabela 5.12 exibe os valores de comparação de energia entre os métodos testados.

Tabela 5.12: Análise dos resultados de energia com um nível de significância $p \leq 0.05$. As células em destaque indicam as comparações que não apresentam diferença significativa.

ID	Método	BRKGA	SHD	SA	RPT_050	ID	BRKGA	SHD	SA	RPT_050
	SHD	0.000	---	---	---		0.000	---	---	---
	SA	0.643	0.123	---	---		0.007	1.000	---	---
1AAQ	RPT_050	0.005	1.000	1.000	---	1AJX	0.010	1.000	1.000	---
	MMA	0.000	1.000	0.000	0.133		0.000	1.000	1.000	0.854
	SHD	0.004	---	---	---		0.000	---	---	---
	SA	0.348	1.000	---	---		0.001	1.000	---	---
1B6J	RPT_050	0.014	1.000	1.000	---	1B6L	0.010	0.486	1.000	---
	MMA	0.002	1.000	0.956	1.000		0.002	1.000	1.000	1.000
	SHD	0.000	---	---	---		0.000	---	---	---
	SA	0.000	1.000	---	---		0.000	1.000	---	---
1BV9	RPT_050	0.024	0.480	1.000	---	1G2K	0.000	1.000	1.000	---
	MMA	0.000	1.000	1.000	1.000		0.000	1.000	0.422	1.000
	SHD	0.000	---	---	---		0.000	---	---	---
	SA	0.000	1.000	---	---		0.000	1.000	---	---
1HEF	RPT_050	0.006	0.864	1.000	---	1HEG	0.000	1.000	1.000	---
	MMA	0.000	1.000	0.680	0.031		0.000	1.000	1.000	1.000
	SHD	0.000	---	---	---		0.000	---	---	---
	SA	0.000	1.000	---	---		0.000	1.000	---	---
1HIV	RPT_050	0.000	1.000	1.000	---	1HPX	0.000	1.000	1.000	---
	MMA	0.000	0.607	0.885	0.647		0.000	1.000	1.000	1.000
	SHD	0.000	---	---	---		0.000	---	---	---
	SA	0.000	1.000	---	---		0.007	1.000	---	---
1HVH	RPT_050	0.114	0.032	0.844	---	1K6P	0.000	1.000	1.000	---
	MMA	0.018	0.184	1.000	1.000		0.000	1.000	0.365	1.000
	SHD	0.000	---	---	---		0.000	---	---	---
	SA	0.004	0.584	---	---		0.000	0.747	---	---
1KZK	RPT_050	0.000	1.000	1.000	---	1MUI	0.007	0.124	1.000	---
	MMA	0.000	1.000	0.928	1.000		0.000	1.000	1.000	0.547
	SHD	0.000	---	---	---		0.509	---	---	---
	SA	0.001	1.000	---	---		0.105	1.000	---	---
1VIK	RPT_050	0.016	1.000	1.000	---	9HVP	0.216	1.000	1.000	---
	MMA	0.000	1.000	0.331	0.027		0.000	0.027	0.168	0.080

Fonte: Do Autor.

Os dados de comparação nos mostram que em 75% das instâncias o BRKGA perde para qualquer outro algoritmo, e entre estes, o MMA supera o BRKGA em todos os complexos. Por outro lado, a comparação entre as variações dos algoritmos memético e multimemético não existe diferença significativa nos resultados para praticamente todos os casos de teste. De maneira similar, foram comparados e analisados os valores de RMSD entre todos os métodos, conforme exibe a Tabela 5.13.

Tabela 5.13: Análise dos resultados de RMSD com um nível de significância $p \leq 0.05$. As células em destaque indicam as comparações que não apresentam diferença significativa.

ID	Método	BRKGA	SHD	SA	RPT_050	ID	BRKGA	SHD	SA	RPT_050
	SHD	0.005	---	---	---		0.007	---	---	---
	SA	0.004	1.000	---	---		0.530	1.000	---	---
1AAQ	RPT_050	0.003	1.000	1.000	---	1AJX	1.000	0.458	1.000	---
	MMA	0.025	1.000	1.000	1.000		0.004	1.000	1.000	0.331
	SHD	1.000	---	---	---		0.000	---	---	---
	SA	1.000	1.000	---	---		0.001	1.000	---	---
1B6J	RPT_050	1.000	1.000	1.000	---	1B6L	0.001	1.000	1.000	---
	MMA	1.000	1.000	1.000	1.000		0.095	0.240	1.000	1.000
	SHD	1.000	---	---	---		0.000	---	---	---
	SA	1.000	1.000	---	---		0.008	1.000	---	---
1BV9	RPT_050	1.000	1.000	1.000	---	1G2K	0.003	1.000	1.000	---
	MMA	1.000	1.000	1.000	1.000		0.000	1.000	1.000	1.000
	SHD	0.006	---	---	---		0.187	---	---	---
	SA	0.229	1.000	---	---		0.458	1.000	---	---
1HEF	RPT_050	0.901	0.839	1.000	---	1HEG	1.000	0.918	1.000	---
	MMA	0.000	1.000	0.627	0.146		1.000	1.000	1.000	1.000
	SHD	0.000	---	---	---		0.000	---	---	---
	SA	0.000	1.000	---	---		0.044	1.000	---	---
1HIV	RPT_050	0.000	1.000	1.000	---	1HPX	0.002	1.000	1.000	---
	MMA	0.000	0.891	1.000	1.000		0.470	0.166	1.000	0.789
	SHD	0.065	---	---	---		0.249	---	---	---
	SA	0.014	1.000	---	---		1.000	1.000	---	---
1HVH	RPT_050	0.979	1.000	1.000	---	1K6P	0.023	1.000	0.784	---
	MMA	0.226	1.000	1.000	1.000		0.092	1.000	1.000	1.000
	SHD	0.000	---	---	---		0.000	---	---	---
	SA	0.063	0.526	---	---		0.000	1.000	---	---
1KZK	RPT_050	0.004	1.000	1.000	---	1MUI	0.007	1.000	1.000	---
	MMA	0.003	1.000	1.000	1.000		0.050	0.327	1.000	1.000
	SHD	0.001	---	---	---		0.859	---	---	---
	SA	0.139	1.000	---	---		0.285	1.000	---	---
1VIK	RPT_050	0.233	0.907	1.000	---	9HVP	1.000	1.000	1.000	---
	MMA	0.009	1.000	1.000	1.000		0.580	1.000	1.000	1.000

Os resultados da análise estatística confirma que nas instâncias 1B6J, 1BV9, 1HEG e 9HVP nenhum método se destaca. Nos demais complexos, o algoritmo SHD se mostra superior ao BRKGA em cerca de 62% deles. Enquanto que os algoritmos SA, RPT_050 e MMA são melhores em aproximadamente metade dos complexos analisados. Além disso, em nenhuma instância, existe diferença entre qualquer par de algoritmos meméticos e multimeméticos.

5.5.1 Comparação com outras ferramentas

Os resultados de atracamento obtidos com o algoritmo multimemético foram comparados com as ferramentas AutoDock Vina (TROTT; OLSON, 2010), DockThor (MAGALHÃES et al., 2014), e jMetal (LóPEZ-CAMACHO et al., 2013). É importante salientar que nenhuma destas ferramentas utiliza a mesma função de energia do *Rosetta* aplicada no algoritmo proposto, logo a comparação fica por conta dos resultados estruturais alcançados (em termos de RMSD). A execução dos testes nas 3 ferramentas seguiu a mesma definição com 31 execuções para cada uma das 16 estruturas do *benchmark* utilizado, e o critério de parada igual a 1 milhão de avaliações da função objetivo. Além disso, os mesmos ângulos diedrais de cada ligante, o centro do campo de busca e a dimensão desta caixa foram definidos para as 3 ferramentas. A parametrização específica de cada um dos algoritmos foi mantida conforme definida pelos autores. A Tabela 5.14 apresenta os resultados da comparação entre BRKGA, MMA e as ferramentas da literatura.

Tabela 5.14: Resultados de comparação dos algoritmos BRKGA e MMA com as ferramentas AutoDock Vina, DockThor e jMetal. Os menores valores de energia e RMSD são destacados.

ID	**Método**	**Melhor solução**		**Média das 31 execuções**	
		Energia	**RMSD**	**Energia**	**RMSD**
1AAQ	BRKGA	36.697	1.135	42.748 ± 5.890	3.118 ± 2.474
	MMA	36.718	1.240	38.759 ± 3.728	1.641 ± 1.698
	Vina	3.932	9.753	9.074 ± 5.045	9.059 ± 0.857
	DockThor	3.520	0.944	5.119 ± 1.407	9.254 ± 4.920
	jMetal	-16.000	2.900	-12.562 ± 3.480	2.111 ± 1.147
1AJX	BRKGA	-246.846	4.400	-239.679 ± 1.874	12.239 ± 1.502
	MMA	-250.886	3.082	-243.002 ± 4.555	9.422 ± 4.569
	Vina	-10.739	1.522	-9.821 ± 0.421	6.215 ± 2.825
	DockThor	48.679	0.855	49.329 ± 2.302	0.881 ± 0.089
	jMetal	-18.000	4.400	-12.836 ± 4.354	3.881 ± 1.639
1B6J	BRKGA	-328.688	1.107	-316.829 ± 7.495	5.852 ± 2.329
	MMA	-329.000	1.127	-322.167 ± 3.891	5.630 ± 2.841
	Vina	-9.123	2.894	-2.386 ± 3.641	6.917 ± 3.077
	DockThor	38.783	0.612	38.958 ± 0.109	0.615 ± 0.057
	jMetal	-18.000	3.470	-11.449 ± 6.787	2.487 ± 0.714
1B6L	BRKGA	-317.455	3.090	-312.622 ± 3.113	6.641 ± 1.935
	MMA	-319.555	2.482	-315.656 ± 2.245	5.097 ± 2.429
	Vina	-12.712	0.892	-12.018 ± 1.291	2.217 ± 3.068
	DockThor	30.512	0.464	30.723 ± 0.126	0.433 ± 0.025
	jMetal	-16.000	2.170	-13.141 ± 3.677	2.169 ± 0.629

Continua na próxima página

Tabela 5.14 – continuação da página anterior

ID	Método	Melhor solução		Média das 31 execuções	
		Energia	RMSD	Energia	RMSD
1BV9	BRKGA	-80.066	0.742	-16.327 ± 44.554	7.273 ± 2.857
	MMA	-80.811	0.707	-53.518 ± 16.200	7.484 ± 4.364
	Vina	14.561	5.781	20.648 ± 2.639	8.407 ± 1.488
	DockThor	55.688	0.890	56.592 ± 1.039	0.945 ± 0.051
	jMetal	-20.000	2.200	-14.613 ± 5.645	2.453 ± 1.306
1G2K	BRKGA	-455.767	1.339	-436.888 ± 9.444	6.037 ± 2.049
	MMA	-456.782	1.100	-451.144 ± 7.353	3.209 ± 2.523
	Vina	-10.013	4.409	-9.343 ± 0.860	6.046 ± 1.768
	DockThor	15.462	0.356	16.098 ± 1.486	0.426 ± 0.166
	jMetal	-20.000	2.600	-12.836 ± 5.313	3.009 ± 1.688
1HEF	BRKGA	175.777	13.476	176.690 ± 0.575	12.254 ± 0.565
	MMA	173.582	11.403	174.806 ± 1.188	11.664 ± 0.335
	Vina	-1.785	9.355	1.425 ± 2.257	8.774 ± 0.562
	DockThor	67.677	1.754	68.069 ± 0.364	1.832 ± 0.149
	jMetal	-22.000	6.600	-10.743 ± 6.252	5.811 ± 1.673
1HEG	BRKGA	357.760	6.911	361.201 ± 1.448	8.672 ± 1.682
	MMA	356.907	9.525	358.989 ± 1.112	8.190 ± 1.491
	Vina	-5.850	5.501	-5.512 ± 0.262	6.005 ± 0.978
	DockThor	58.366	2.749	60.502 ± 1.905	4.018 ± 1.488
	jMetal	-14.500	3.160	-9.146 ± 3.905	5.239 ± 1.734
1HIV	BRKGA	-164.164	2.923	-150.521 ± 3.102	6.646 ± 1.111
	MMA	-165.125	3.380	-163.346 ± 3.583	2.851 ± 1.389
	Vina	-0.292	7.493	12.010 ± 18.697	8.086 ± 0.919
	DockThor	55.134	0.286	55.321 ± 0.139	0.291 ± 0.039
	jMetal	-32.000	7.930	-18.140 ± 3.675	2.582 ± 1.447
1HPX	BRKGA	-348.396	5.277	-346.813 ± 1.591	5.870 ± 1.044
	MMA	-357.495	3.520	-349.827 ± 3.396	5.085 ± 1.173
	Vina	-9.084	5.736	-6.523 ± 2.183	6.181 ± 1.026
	DockThor	88.642	7.812	96.046 ± 9.689	7.073 ± 2.105
	jMetal	-18.000	4.860	-11.728 ± 4.775	3.840 ± 0.942
1HVH	BRKGA	428.012	8.939	434.158 ± 2.595	10.335 ± 2.225
	MMA	427.953	8.413	431.740 ± 2.498	9.059 ± 2.792
	Vina	-8.647	7.339	-7.043 ± 1.124	5.976 ± 1.714
	DockThor	127.176	8.341	130.224 ± 2.764	6.099 ± 1.577
	jMetal	-18.000	2.440	-12.448 ± 4.419	2.887 ± 0.522
1K6P	BRKGA	-382.381	4.419	-375.894 ± 4.853	6.885 ± 1.755
	MMA	-384.233	4.071	-381.266 ± 1.597	5.727 ± 1.930
	Vina	-5.164	5.223	-0.528 ± 3.197	7.445 ± 2.094
	DockThor	143.304	1.731	150.404 ± 6.755	2.099 ± 1.878
	jMetal	-20.000	2.380	-14.685 ± 4.770	3.199 ± 0.847

Continua na próxima página

Tabela 5.14 – continuação da página anterior

ID	Método	Melhor solução		Média das 31 execuções	
		Energia	RMSD	Energia	RMSD
1KZK	BRKGA	-440.323	1.914	-422.159 ± 17.595	5.680 ± 2.143
	MMA	-441.190	1.539	-437.009 ± 6.579	3.469 ± 2.308
	Vina	-9.853	2.369	-8.046 ± 0.831	5.735 ± 2.574
	DockThor	27.607	0.788	27.791 ± 0.162	0.754 ± 0.133
	jMetal	-24.000	9.020	-10.437 ± 12.788	7.954 ± 1.635
1MUI	BRKGA	-36.393	1.826	-26.570 ± 6.681	5.853 ± 3.019
	MMA	-36.846	1.951	-33.825 ± 3.377	3.352 ± 2.749
	Vina	-8.224	6.682	-6.293 ± 1.147	7.484 ± 1.085
	DockThor	17.060	0.337	17.419 ± 0.153	0.458 ± 0.199
	jMetal	-20.000	4.650	-13.043 ± 4.163	2.575 ± 0.983
1VIK	BRKGA	174.700	2.177	228.249 ± 50.900	6.853 ± 2.817
	MMA	173.345	2.147	182.827 ± 13.129	4.203 ± 2.658
	Vina	96.973	11.193	140.083 ± 29.699	9.178 ± 2.020
	DockThor	165.542	1.381	267.462 ± 123.245	2.968 ± 2.598
	jMetal	-40.000	4.900	-19.123 ± 14.436	4.009 ± 0.983
9HVP	BRKGA	359.987	1.090	372.039 ± 14.288	4.982 ± 3.386
	MMA	359.972	1.112	363.167 ± 4.779	3.357 ± 3.395
	Vina	-3.736	10.092	0.503 ± 4.026	9.393 ± 0.954
	DockThor	25.957	1.189	26.045 ± 0.047	1.211 ± 0.048
	jMetal	-22.000	4.170	-14.894 ± 4.135	2.717 ± 1.009

Fonte: Do Autor.

Os resultados de energia, como mencionado anteriormente, são comparados somente entre o BRKGA e o MMA. Este último é superior em todas as instâncias analisando-se as médias de energia, e só perde no caso 1AAQ quando visto os menores valores obtidos. Em relação ao RMSD, os menores valores foram obtidos pela ferramenta DockThor, superior em 75% dos casos de teste, e em 81% das instâncias considerando as médias alcançadas. Para confirmar as diferenças entre os resultados de RMSD, o teste Kruskal-Wallis foi aplicado. A Tabela 5.15 exibe os resultados da aplicação do teste para os valores de RMSD, onde aqueles superiores a 0,05 são destacados.

Tabela 5.15: Resultados de aplicação do teste Kruskal-Wallis para os valores de RMSD dos algoritmos BRKGA, MMA, Vina, DockThor e jMetal. Para cada instância são exibidos os *p-values* resultantes da comparação dos grupos com um nível de significância menor igual a 5%.

ID	**RMSD**	**ID**	**RMSD**	**ID**	**RMSD**	**ID**	**RMSD**
1AAQ	3,83e-14	1AJX	3,93e-24	1B6J	2,85e-20	1B6L	4,22e-23
1BV9	1,28e-16	1G2K	6,41e-21	1HEF	6,53e-30	1HEG	6,40e-20
1HIV	3,77e-27	1HPX	7,40e-18	1HVH	1,25e-21	1K6P	4,52e-22
1KZK	1,34e-20	1MUI	6,41e-21	1VIK	3,70e-15	9HVP	1,25e-16

Fonte: Do Autor.

Os resultados indicam que existe diferença significativa entre algum par de métodos para todas as instâncias avaliadas. Para verificar quais métodos se sobressaem, foi aplicado o teste *post-hoc* Dunn (DUNN, 1964). A Tabela 5.16 exibe os *p-values* da comparação de todas as ferramentas. Valores maiores do que 0,05 (destacados em cinza) indicam que não existe diferença significativa entre os métodos para certa instância.

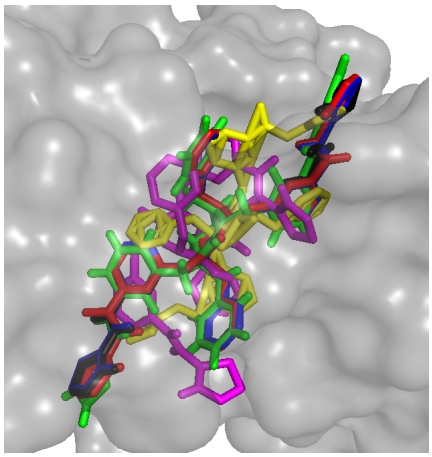
Tabela 5.16: Análise dos resultados de RMSD com um nível de significância $p \leq 0.05$. As células em destaque indicam as comparações que não apresentam diferença significativa.

ID	Método	BRKGA	MMA	Vina	DockThor	ID	BRKGA	MMA	Vina	DockThor
	MMA	0.573	---	---	---		0.083	---	---	---
	Vina	0.000	0.000	---	---		0.000	0.346	---	---
1AAQ	DockThor	0.000	0.000	1.000	---	1AJX	0.000	0.000	0.000	---
	jMetal	1.000	1.000	0.000	0.000		0.000	0.005	1.000	0.001
	MMA	1.000	---	---	---		1.000	---	---	---
	Vina	1.000	1.000	---	---		0.000	0.001	---	---
1B6J	DockThor	0.000	0.000	0.000	---	1B6L	0.000	0.000	0.000	---
	jMetal	0.005	0.017	0.000	0.001		0.000	0.032	1.000	0.000
	MMA	1.000	---	---	---		0.003	---	---	---
	Vina	0.951	0.780	---	---		1.000	0.001	---	---
1BV9	DockThor	0.000	0.000	0.000	---	1G2K	0.000	0.000	0.000	---
	jMetal	0.001	0.002	0.000	0.279		0.005	1.000	0.002	0.000
	MMA	0.849	---	---	---		1.000	---	---	---
	Vina	0.000	0.011	---	---		0.000	0.002	---	---
1HEF	DockThor	0.000	0.000	0.000	---	1HEG	0.000	0.000	0.001	---
	jMetal	0.000	0.000	0.084	0.058		0.000	0.000	1.000	0.243
	MMA	0.007	---	---	---		0.477	---	---	---
	Vina	0.089	0.000	---	---		1.000	0.008	---	---
1HIV	DockThor	0.000	0.000	0.000	---	1HPX	0.061	0.000	1.000	---
	jMetal	0.001	1.000	0.000	0.000		0.000	0.002	0.000	0.000
	MMA	1.000	---	---	---		0.968	---	---	---
	Vina	0.000	0.004	---	---		1.000	0.208	---	---
1HVH	DockThor	0.000	0.017	1.000	---	1K6P	0.000	0.000	0.000	---
	jMetal	0.000	0.000	0.000	0.000		0.000	0.002	0.000	0.133
	MMA	0.171	---	---	---		0.066	---	---	---
	Vina	1.000	0.041	---	---		1.000	0.001	---	---
1KZK	DockThor	0.000	0.000	0.000	---	1MUI	0.000	0.000	0.000	---
	jMetal	0.043	0.000	0.180	0.000		0.016	1.000	0.000	0.000
	MMA	0.010	---	---	---		0.663	---	---	---
	Vina	0.202	0.000	---	---		0.000	0.000	---	---
1VIK	DockThor	0.000	0.358	0.000	---	9HVP	0.000	0.175	0.000	---
	jMetal	0.021	1.000	0.000	0.214		1.000	1.000	0.000	0.008

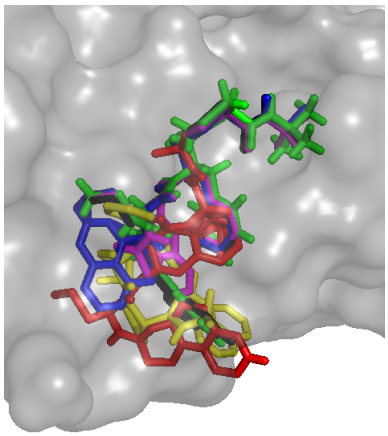
Os resultados da comparação de RMSD entre os métodos indicam que nossa abordagem MMA é competitiva com as ferramentas Vina e jMetal em pelo menos 25 e 38% respectivamente, dos casos de teste. Em seguida, realizou-se a análise das melhores estruturas de ligante obtidas por cada uma das ferramentas. A Figura 5.7 exibe as conformações de cada algoritmo de acordo com as diferentes cores: preto para a conformação experimental; em vermelho, o resultado do BRKGA; em azul, o resultado do MMA; cor magenta para a ferramenta AutoDock Vina; em verde, o resultado do DockThor; e em amarelo, o resultado obtido pelo jMetal. É importante destacar que os ligantes ilustrados são referentes aos melhores resultados obtidos, isto é, aqueles de menor energia. Portanto, tal análise não reflete a média das execuções nem a distribuição das soluções, apesar de

que uma execução qualquer possa obter uma estrutura de menor RMSD mas que não corresponde ao menor valor de energia encontrado.

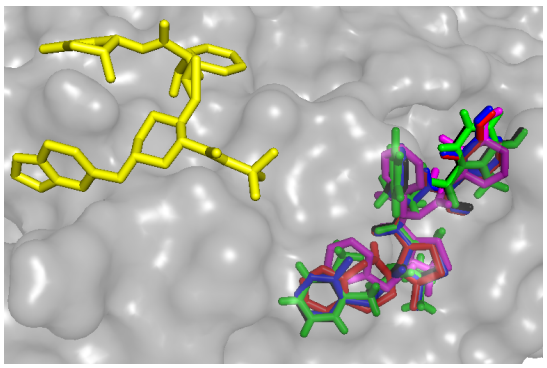
Figura 5.7: Melhores resultados para os testes de comparação das estruturas 1BV9, 1HPX, 1KZK e 1MUI. Em preto a estrutura experimental, em vermelho, o resultado do algoritmo BRKGA, em azul, o resultado obtido pelo MMA, em magenta, o resultado do AutoDock Vina, em verde o resultado da ferramenta Dockthor, e em amarelo, o resultado obtido com o jMetal.



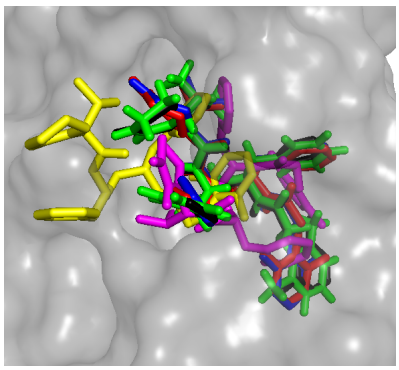
(a) 1BV9



(b) 1HPX



(c) 1KZK



(d) 1MUI

A Figura 5.7 (a) exibe os resultados para a estrutura 1BV9, onde pode-se observar que os algoritmos BRKGA, MMA e DockThor obtêm estruturas bem semelhantes ao ligante do cristal, enquanto que as ferramentas jMetal e Vina apresentam valores de RMSD mais elevados. Já para o complexo 1HPX (b), nenhum método apresenta resultado próximo da estrutura do cristal, o algoritmo MMA possui o menor RMSD e a ferramenta DockThor o maior. Na instância 1KZK (c), apenas o jMetal apresenta um alto valor de RMSD, os demais algoritmos alcançam um resultado bem próximo do cristal. E por fim, o complexo 1MUI (d) apresenta, novamente, os algoritmos BRKGA, MMA e DockThor com resultados próximos do objetivo, enquanto que Vina e jMetal possuem um RMSD superior a 4,6 Å.

5.6 Resumo do capítulo

Neste capítulo foram apresentados os resultados obtidos concernentes ao método de otimização proposto quando da otimização de um conjunto de testes de complexos proteína-ligante. A metaheurística desenvolvida consiste, primeiramente, da Etapa I de desenvolvimento de um algoritmo memético, que combina o BRKGA como operador de busca global, e os algoritmos BI, FI, SHD e SA como técnicas de busca local. Nesta etapa foram avaliados os desempenhos de cada um dos algoritmos e observado que, em geral, as variações da técnica *Hill climbing* se sobressaíram em relação aos demais métodos, com destaque para a variante *Stochastic Hill Descent*.

Após os resultados obtidos, foi implementado, na segunda etapa, o algoritmo multimemético auto-adaptativo. Nesta fase, tanto o algoritmo de busca local quanto o parâmetro de raio de perturbação para geração de vizinhos no processo local de refinamento foram auto-adaptados ao longo do processo de busca. Cada um destes operadores de LS possui uma probabilidade de seleção ao longo da execução, a qual é atualizada de acordo com uma equação, proposta neste trabalho, que mensura o custo-benefício do operador no refinamento das soluções. Os resultados mostraram que o MMA atinge resultados melhores do que apenas utilizar o BRKGA, ou então um algoritmo de seleção aleatória dos operadores de LS. Entretanto, os resultados não superaram aqueles obtidos com as variações do algoritmo memético. Concluiu-se, então, que o método MMA pode ser considerado uma contribuição efetiva para a área de atracamento molecular, mas que ainda necessita melhorias para que os resultados obtidos possam ser mais competitivos com outras ferramentas já existentes.

6 CONCLUSÃO E TRABALHOS FUTUROS

Apesar dos avanços de métodos computacionais em atracamento molecular, há ainda uma crescente demanda pela exploração de novas abordagens que consigam manipular os dados estruturais a partir das estruturas moleculares determinadas experimentalmente. Visto que atualmente ainda não exista um método capaz de obter a conformação de ligação ótima para o problema de AM, o desenvolvimento de novos métodos robustos e de aplicação geral, a investigação e adaptação destas metodologias, com o objetivo de reunir e aplicar todas estes dados experimentais disponíveis, de maneira eficaz nos processos de atracamento molecular, são realmente uma necessidade, além de representar uma relevante área de pesquisa relacionada à Bioinformática Estrutural.

Considera-se também que uma técnica aplicada ao problema de AM requer uma função de energia acurada, bem como um método de busca eficiente na exploração do espaço de busca de soluções, aliado a estratégias que possibilitem incorporar conhecimento do problema em si para contornar as adversidades encontradas pela complexidade do mesmo e da função de avaliação. Acredita-se que a partir do estudo e desenvolvimento de metaheurísticas adaptadas para lidar com questões biológicas endereçadas pelo AM, seja possível extrair o potencial dos métodos de busca e obter bons resultados para o problema.

Desta forma, foi proposto e desenvolvido um método de otimização voltado para o problema de Atracamento Molecular, e que possa auxiliar no processo de Triagem Virtual, e consequentemente o descobrimento de novos fármacos. O método consiste em Algoritmo Multimemético Auto-Adaptativo, o qual, baseia-se em uma função de probabilidades para avaliar e adaptar parâmetros do processo de busca. O algoritmo teve sua implementação dividida em duas etapas principais: (i) o desenvolvimento de um algoritmo memético com aplicação de diferentes métodos de busca local; e (ii) a incorporação do conceito multimeme na estrutura do algoritmo e a adaptação dos parâmetros de busca local a partir do desenvolvimento de um a função que mensura o custo-benefício de cada um deles. Os resultados e análises concernentes a estes embasam as considerações finais sobre o método proposto.

Na primeira etapa, foi implementada uma variação do algoritmo BRKGA, desconsiderando a codificação dos genes no intervalo [0,1], como método de busca global, e 4 algoritmos de busca local: BI, FI, SHD e SA, os quais, foram aplicados de forma independente mas combinados com o BRKGA. A execução do MA foi ajustada ao modelo de

discretização por cubos adotada, de modo que há um limite mínimo de soluções por cada região (subcubo). Dessa forma, regras de movimentação das soluções, isto é, limites de valores na perturbação dos genes foram definidos para que se pudesse ter um algoritmo capaz de explorar todo o sítio de ligação ao manter a diversidade populacional, mas que também, tivesse a habilidade de explorar regiões promissoras ao aplicar métodos de busca local.

Os resultados obtidos nessa primeira etapa mostram que, de modo geral, uma abordagem memética apresenta melhores resultados do que um algoritmo não-memético. As diferenças dos valores de energia e RMSD foram significativos para praticamente todos os casos de teste avaliados, confirmando que a aplicação de busca local trouxe melhorias para o processo de busca por boas soluções no espaço conformacional dos complexos. Referente aos métodos de LS aplicados, nenhum deles obteve grande destaque na resolução do problema. Entretanto, o algoritmo SHD se mostrou ligeiramente melhor, por exemplo, do que as outras duas variações do algoritmo *Hill climbing*, muito por conta de sua característica aleatória em visitar a vizinhança das soluções no espaço de busca. Este comportamento evita buscas cíclicas, isto é, iniciando e encerrando pelos mesmos pontos, e possibilita que cada gene tenha iguais condições de ser melhorado. Também, o *Simulated Annealing* tem um processo similar ao HC na aceitação de soluções, mas permite soluções piores com uma certa probabilidade. Esta característica e o comportamento aleatório na perturbação e geração de novos indivíduos durante a busca, tornam o método competitivo com o *Stochastic Hill Descent*. Conforme as análises dos *boxplots* e curvas de convergência destes algoritmos reportadas na Seção 5.4, destaca-se uma ligeira vantagem do método SHD na obtenção de menores valores de energia.

Entretanto, os resultados desta fase inicial mostram que nenhum dos métodos pode ser considerado o melhor na aplicação do algoritmo memético, e portanto, levaram à decisão de aplicar mais de um algoritmo de busca local combinados durante o processo de busca. Portanto, a segunda fase de desenvolvimento deste trabalho se deu por implementar um algoritmo multimemético auto-adaptativo. Como passo inicial, foi proposta e desenvolvida uma equação de probabilidades com o intuito de avaliar e medir o custo-benefício que cada busca local representa na execução do algoritmo. A equação é composta por dois termos: (i) o primeiro mede a taxa de ganho de *fitness*; e (ii) o segundo, a taxa de sucesso de aplicação e melhora das soluções, ambos termos aplicados sobre cada método de LS. À cada termo foi associado um peso, e o produto deles representa o peso que o método tem no MA. Assim, cada algoritmo possui uma probabilidade de aplicação, de

acordo com o peso calculado, que é ajustada à cada aplicação do processo de busca local.

O modelo auto-adaptativo desenvolvido foi comparado inicialmente com outras duas versões de MMA, uma aleatória e outra baseada no mecanismo SIM (KRASNOGOR; SMITH, 2001). Os resultados mostraram que a abordagem implementada foi competitiva com os demais métodos e apresentou bons resultados em termos estruturais. Em seguida, decidiu-se por estender a auto-adaptação para outro parâmetro da busca local, o raio de perturbação. Até então, esse parâmetro tinha valor fixado em 0,50, e nesta fase passou a ter ainda os valores 0,10 e 0,25. A mesma equação de probabilidades foi aplicada para mensurar o impacto desta parametrização e então ajustá-la durante a execução do algoritmo. Assim, o algoritmo MMA teve sua versão final alcançada, e sua aplicação comparada com as versões anteriores implementadas. Os resultados mostraram que a abordagem multimemética alcança os melhores mínimos de energia em mais da metade das instâncias, mas que em termos de média destes valores e na comparação de RMSD se mostrou equivalente com os métodos meméticos.

A explicação para a equivalência dos resultados pode estar no fato de que no MMA são utilizados, de maneira distribuída, 4 algoritmos de busca local ao invés de apenas 1, como no MA, em todo processo de busca. Neste sentido, pode-se estar perdendo esforço computacional com métodos que não contribuem para encontrar os ótimos locais. Além disso, no MA é adotado um valor fixo para o raio de perturbação do processo de busca local, enquanto que no MMA são utilizados 3 valores auto-adaptáveis. Esta caraterística é interessante pois os métodos podem explorar diferentes vizinhanças em diferentes estágios da evolução do algoritmo, e então compensar uma possível perda de esforço computacional. Assim, a combinação de um conjunto de métodos de busca local combinado com um conjunto de valores para o raio de perturbação confirma ser uma interessante abordagem na busca de boas soluções para o problema de AM.

Conclui-se, então, que o algoritmo MMA desenvolvido nesta dissertação, pode ser considerado uma contribuição efetiva, porém inicial, para a área de atracamento molecular. Entende-se que o método ainda necessita de melhorias, testes e validações mais robustas e próximas às ferramentas estado da arte, para que se melhore os resultados obtidos. De forma geral, julga-se que o objetivo tenha sido alcançado, mostrando que a inclusão de buscas locais auxilia na obtenção de melhores resultados para o problema, e que a combinação delas, de maneira auto-adaptável, e de parâmetros específicos de LS também podem trazer benefícios na obtenção de boas soluções. Como principais contribuições tem-se: (i) a implementação de uma função de probabilidades para avaliar, em

execução, os impactos das buscas locais para melhor adaptar-se e guiar o processo de busca; e (ii) o desenvolvimento de um algoritmo que combina diferentes algoritmos e parâmetros de modo a adaptá-los seguindo a aplicação da função proposta.

Como trabalhos futuros entende-se que seja importante considerar a aplicação de diferentes algoritmos de busca global, como por exemplo, Algoritmos Genéticos, Evolução Diferencial e Otimização por Enxame de Partículas, assim como outros algoritmos de busca local, como *Solis and Wets* e *Nelder-Mead*. Em relação à abordagem auto-adaptativa desenvolvida, é possível modificar o conjunto de buscas locais utilizando apenas os algoritmos SHD e SA, por exemplo, já que obtiveram bom desempenho na primeira etapa. Os valores para o parâmetro raio de perturbação também podem ser modificados, assim como a aplicação de diferentes valores para genes específicos, isto é, a operação de translação pode ter valores específicos, diferentes dos utilizados na operação de rotação da estrutura ligante e dos seus ângulos diedrais.

Outra questão, a frequência de aplicação das buscas locais pode ser ajustada de acordo com Bambha (BAMBHA et al., 2004), que propõe um mecanismo dinâmico que varia a intensidade da LS durante o processo de otimização, gerando assim baixa acurácia do método no início e alta acurácia no final da execução. O objetivo é focar, inicialmente, na busca global para encontrar regiões de busca promissoras, e depois gastar mais tempo de execução no final do processo para refinar as soluções com o algoritmo de busca local. Além disso, na função de probabilidades implementada existem questões que podem ser exploradas, desde os valores de pesos utilizados nos termos até mesmo outros detalhes. Por exemplo, a opção por considerar a média dos valores de *fitness* dos indivíduos da casta 'A' pode ser trocada pelo valor de *fitness* da melhor solução apenas, ou o valor médio de todas as soluções, ou a média dos melhores indivíduos de cada subcubo. Outra possibilidade é implementar uma janela de tempo para avaliar as contribuições de cada operador de busca local, considerando as últimas 100 mil avaliações, por exemplo. Finalmente, trabalhos futuros podem explorar as modificações mencionadas para tornar esta abordagem melhor para encontrar soluções ótimas para o problema de Atracamento Molecular.

7 PUBLICAÇÕES E PRODUÇÃO TÉCNICA

Neste capítulo serão apresentados os trabalhos desenvolvidos durante o período do Mestrado, abrangendo as áreas de inteligência artificial, otimização e metaheurísticas, e atracamento molecular.

7.1 Artigos completos publicados em periódicos

- **LEONHART, P. F.**; SPIELER, E.; LIGABUE-BRAUN, R; DORN, M.. A biased random key genetic algorithm for the protein–ligand docking problem. In: Soft Computing, p. 1-22, 2018. Qualis: A2.

7.2 Artigos completos aceitos para publicação

- **LEONHART, P. F.**; NARLOCH, P. H.; DORN, M.. A Self-Adaptive Local Search Coordination in Multimeme Memetic Algorithm for Molecular Docking. In: International Conference on Computational Science - ICCS, 2019. Qualis: A1.
- **LEONHART, P. F.**; DORN, M.. A Biased Random Key Genetic Algorithm with Local Search Chains for Molecular Docking. In: Applications of Evolutionary Computation. Springer International Publishing, 2019. Qualis: B1.

Esta dissertação de Mestrado está inserida no contexto do Programa de Nucleação de Grupos de Pesquisa - PRONUPEQ - Projeto FAPERGS 16-2551/0000520-6.

REFERÊNCIAS

AARTS, E.; LENSTRA, J. K. (Ed.). **Local Search in Combinatorial Optimization**. 1st. ed. New York, NY, USA: John Wiley & Sons, Inc., 1997.

ADOLF-BRYFOGLE, J.; JR., R. L. D. The pyrosetta toolkit: A graphical user interface for the rosetta software suite. **PLOS ONE**, Public Library of Science, v. 8, n. 7, p. 1–8, 2013.

ALA, P. et al. Counteracting hiv-1 protease drug resistance: Structural analysis of mutant proteases complexed with xv638 and sd146, cyclic urea amides with broad specificities. **Biochemistry**, v. 37, p. 15042–9, 1998.

ALONSO, H.; BLIZNYUK, A. A.; GREADY, J. E. Combining docking and molecular dynamic simulations in drug design. **Med. Res. Rev.**, v. 26, n. 5, p. 531–568, 2006.

ALTMAN, R. B. Editorial: Building successful biological databases. **Briefings in Bioinformatics**, v. 5, n. 1, p. 4–5, 03 2004.

ANDREI, R. M. et al. Intuitive representation of surface properties of biomolecules using bioblender. **BMC bioinformatics**, v. 13, n. 4, p. 1, 2012.

ANFINSEN, C. B. Principles that govern the folding of protein chains. **Science**, v. 181, n. 4096, p. 223–230, 1973.

ATTWOOD, T. et al. Concepts, historical milestones and the central place of bioinformatics in modern biology: A european perspective. In: MAHDAVI, M. A. (Ed.). **Bioinformatics**. Rijeka: IntechOpen, 2011. cap. 1.

BALDWIN, E. T. et al. Structure of hiv-1 protease with kni-272, a tight-binding transition-state analog containing allophenylnorstatine. **Structure**, v. 3, n. 6, p. 581 – 590, 1995.

BALDWIN, R. L. Dynamic hydration shell restores kauzmann's 1959 explanation of how the hydrophobic factor drives protein folding. **Proc. Natl. Acad. Sci.**, v. 111, n. 36, p. 13052–13056, 2014.

BAMBHA, N. K. et al. Systematic integration of parameterized local search into evolutionary algorithms. **Evolutionary Computation, IEEE Transactions on**, v. 8, n. 2, p. 137 – 155, 2004.

BARREIRO, E. J.; FRAGA, C. A. M. **Química Medicinal-: As bases moleculares da ação dos fármacos**. [S.l.: s.n.], 2014.

BEAN, J. C. Genetic algorithms and random keys for sequencing and optimization. **ORSA J. Comp.**, v. 6, n. 2, p. 154–160, 1994.

BENITE, A. M. C.; MACHADO, S. d. P.; BARREIRO, E. J. Uma visão da química bioinorgânica medicinal. **Química Nova**, Scielo, v. 30, p. 2062–2067, 2007.

BERMAN, H. M. et al. The protein data bank. **Nucleic Acids Res.**, Oxford University Press, v. 28, n. 1, p. 235–242, 2000.

BISSANTZ, C.; FOLKERS, G.; ROGNAN, D. Protein-based virtual screening of chemical databases. 1. evaluation of different docking/scoring combinations. **J. Med. Chem.**, v. 43, n. 25, p. 4759–4767, 2000.

BÖHM, H. J. Ludi: rule-based automatic design of new substituents for enzyme inhibitor leads. **J. Comput.-Aided Mol. Des.**, v. 6, n. 6, p. 593–606, 1992.

BOUSSAïD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. **Information Sciences**, v. 237, p. 82 – 117, 2013. Prediction, Control and Diagnosis using Advanced Neural Computations.

BOX, M. J. A New Method of Constrained Optimization and a Comparison With Other Methods. **The Computer Journal**, v. 8, n. 1, p. 42–52, 1965.

BROOIJMANS, I. D. K. N. Molecular recognition and docking algorithms. **Annu. Rev. Biophys. Biomol. Struct.**, v. 32, n. 1, p. 335–373, 2003.

BROOKS, B. R. Charmm: A program for macromolecular energy, minimization and dynamics calculations. **J. Comput. Chem.**, v. 4, n. 2, p. 187–217, 1983.

BäCKBRO, K. et al. Unexpected binding mode of a cyclic sulfamide hiv-1 protease inhibitor. **Journal of medicinal chemistry**, v. 40, n. 6, p. 898—902, 1997.

CAVANAGH, J. et al. **Protein NMR spectroscopy: principles and practice.** 2. ed. New York, USA: Academic Press, 2006. 912 p.

ČERNÝ, V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. **Journal of Optimization Theory and Applications**, v. 45, n. 1, p. 41–51, 1985.

CHANDRIKA, B. R.; SUBRAMANIAN, J.; SHARMA, S. D. Managing protein flexibility in docking and its applications. **Drug Discovery Today**, v. 14, n. 7, p. 394–400, 2009.

CHELOUAH, R.; SIARRY, P. Genetic and nelder–mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. **European Journal of Operational Research**, v. 148, n. 2, p. 335 – 348, 2003.

CHOU, K.-C. Structural bioinformatics and its impact to biomedical science. **Current Medicinal Chemistry**, Bentham Science Publishers Ltd., v. 11, n. 16, p. 2105–2134, 2004.

CLAUSSEN, H.; BUNING CM., R. M.; LENGAUER, T. Flexe: efficient molecular docking considering protein structure variations1. **J. Mol. Biol.**, v. 308, n. 2, p. 377 – 395, 2001.

COMBS, S. A. et al. Small-molecule ligand docking into comparative models with rosetta. **Nat. Protoc.**, v. 8, n. 7, p. 1277–1299, 2013.

CONSORTIUM, . G. P. A global reference for human genetic variation. **Nature**, Nature Publishing Group, v. 526, n. 7571, p. 68–74, 2015.

COOK, S. A. An overview of computational complexity. **Commun. ACM**, ACM, New York, NY, USA, v. 26, n. 6, 1983.

CORNELL, W. D.; CIEPLAK, P. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. **J. Am. Chem. Soc.**, v. 117, n. 19, p. 5179–5197, 1995.

COZZINI, P. et al. Target flexibility: An emerging consideration in drug discovery and design†. **J. Med. Chem.**, v. 51, n. 20, p. 6237–6255, 2008.

DAVIES, D. R. The structure and function of the aspartic proteinases. **Annual Review of Biophysics and Biophysical Chemistry**, v. 19, n. 1, p. 189–215, 1990.

DEVI S. SIVA SATHYA, M. S. C. R. V. Evolutionary algorithms for de novo drug design – a survey. **Appl. Soft Comput.**, v. 27, p. 543 – 552, 2015.

DEWITTE, R. S.; SHAKHNOVICH, E. I. Smog: de novo design method based on simples, fast, and accurate free energy estimates. 1. methodology and supporting evidence. **J. Am. Chem. Soc.**, v. 118, n. 47, p. 11733–11744, 1996.

DOMÍNGUEZ-ISIDRO, S.; MEZURA-MONTES, E. A cost-benefit local search coordination in multimeme differential evolution for constrained numerical optimization problems. **Swarm Evol. Comput.**, v. 39, p. 249 – 266, 2018.

DORIGO, M.; CARO, G. D. The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo and F. Glover, editors, New Ideas in Optimization, McGraw-Hill, p. 11–32, 1999.

DORN, M. et al. Three-dimensional protein structure prediction: Methods and computational strategies. **Computational Biology and Chemistry**, v. 53, p. 251 – 276, 2014.

DREO, J. et al. **Metaheuristics for hard optimization : methods and case studies**. [S.l.]: Springer-Verlag, 2006. 372 p.

DREYER, G. B. et al. Hydroxyethylene isostere inhibitors of human immunodeficiency virus-1 protease: structure-activity analysis using enzyme kinetics, x-ray crystallography, and infected t-cell assays. **Biochemistry**, v. 31, n. 29, p. 6646–6659, 1992.

DU, X. et al. Insights into protein–ligand interactions: Mechanisms, models, and methods. **International Journal of Molecular Sciences**, v. 17, n. 2, 2016.

DUNBRACK, R. L. Rotamer libraries in the 21st century. **Current Opinion in Structural Biology**, v. 12, n. 4, p. 431 – 440, 2002.

DUNN, M. F. Protein–ligand interactions: General description. In: **eLS**. [S.l.]: American Cancer Society, 2010. cap. Dunn MF.

DUNN, O. J. Multiple comparisons using rank sums. **Technometrics**, Taylor & Francis, v. 6, n. 3, p. 241–252, 1964.

DURILLO, J.; NEBRO, A. jmetal: a java framework for multi-objective optimization. **Adv. Eng. Softw.**, v. 42, p. 760 – 771, 2011.

EISENSTEIN, M.; KATZIR, E. K. On proteins, grids, correlations, and docking. **Comptes Rendus Biologies**, v. 327, n. 5, p. 409 – 420, 2004.

ERICKSON, J. R. et al. Design, activity, and 2.8 a crystal structure of a c2 symmetric inhibitor complexed to hiv-1 protease. **Science**, v. 249 4968, p. 527–33, 1990.

FISCHER, M. et al. Incorporation of protein flexibility and conformational energy penalties in docking screens to improve ligand discovery. **Nat. Chem.**, v. 6, n. 7, p. 575–583, 2014.

FRENKEL, D.; SMIT, B. Chapter 7 - free energy calculations. In: FRENKEL, D.; ; SMIT, B. (Ed.). **Understanding Molecular Simulation (Second Edition)**. Second edition. San Diego: Academic Press, 2002. p. 167 – 200.

GOHLKE, H.; HENDLICH, M.; KLEBE, G. Knowledge-based scoring function to predict protein–ligand interactions. **J. Mol. Biol**, v. 2000, p. 337–356, 2000.

GONÇALVES, J. F.; ALMEIDA, J. R. de. A hybrid genetic algorithm for assembly line balancing. **Journal of Heuristics**, v. 8, n. 6, p. 629–642, 2002.

GONÇALVES, J. F.; RESENDE, M. G. C. Biased random-key genetic algorithms for combinatorial optimization. **Journal of Heuristics**, v. 17, n. 5, p. 487–525, 2011.

GOODFORD, P. J. A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. **J. Med. Chem.**, ACS Publications, v. 28, n. 7, p. 849–857, 1985.

GOODSELL, D. S.; OLSON, A. J. Automated docking of substrates to proteins by simulated annealing. **Proteins: Structure, Function, and Bioinformatics**, v. 8, n. 3, p. 195–202, 1990.

GRAY, J. J. et al. Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. **Journal of Molecular Biology**, v. 331, n. 1, p. 281 – 299, 2003.

GUEDES, I. A.; MAGALHÃES, C. S. d.; DARDENNE, L. E. Receptor-ligand molecular docking. **International Union for Pure and Applied Biophysics (IUPAB) and Springer-Verlag Berlin Heidelberg 2013**, n. 6, p. 75–87, 2013.

HALGREN, T. A. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. **J. Comput. Chem.**, v. 17, n. 5-6, p. 490–519, 1996.

HALPERIN, I. et al. Principles of docking: An overview of search algorithms and a guide to scoring functions. **Proteins: Struct., Funct., Bioinf.**, v. 47, n. 4, p. 409–443, 2002.

HART, W. E. **Adaptive global optimization with local search**. Tese (Doutorado) — University of California, San Diego, 1994.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence**. Cambridge, MA, USA: MIT Press, 1992.

HOOKE, R.; JEEVES, T. Direct search solution of numerical and statistical problems. **Journal of the ACM**, v. 8, n. 2, p. 212–229, 1961.

HUANG, S.-Y.; ZOU, X. Advances and challenges in protein-ligand docking. **Int. J. Mol. Sci.**, v. 11, n. 8, p. 3016, 2010.

HUANG S.Y.AND ZOU, X. Ensemble docking of multiple protein structures: considering protein structural variations in molecular docking. **Proteins: Struct., Funct., Bioinf.**, v. 66, n. 2, p. 399–421, 2007.

IRWIN, J. J.; SHOICHET, B. K. Zinc - a free database of commercially available compounds for virtual screening. **J. Chem. Inf. Model.**, v. 45, n. 1, p. 177–182, 2005.

JACKSON, R. M.; GABB, H. A.; STERNBERG, M. J. Rapid refinement of protein interfaces incorporating solvation: application to the docking problem1. **J. Mol. Biol.**, v. 276, n. 1, p. 265–285, 1998.

JADHAV, K. et al. Nonpeptide cyclic cyanoguanidines as hiv-1 protease inhibitors: Synthesis, structure-activity relationships, and x-ray crystal structure studies. **Journal of medicinal chemistry**, v. 41, p. 1446–55, 1998.

JAKOB, W. Towards an adaptive multimeme algorithm for parameter optimisation suiting the engineers' needs. In: PARALLEL PROBLEM SOLVING FROM NATURE - PPSN IX. **Proceedings...** [S.l.]: Springer Berlin Heidelberg, 2006. p. 132–141.

JAKOB, W. A general cost-benefit-based adaptation framework for multimeme algorithms. **Memetic Computing**, v. 2, n. 3, p. 201–218, 2010.

JANSON, S.; MERKLE, D.; MIDDENDORF, M. Molecular docking with multi-objective particle swarm optimization. **Appl. Soft Comput.**, v. 8, n. 1, p. 666–675, 2008.

JIN, X.; ZHIHUA, C.; WENYIN, G. An adaptive strategy to adjust the components of memetic algorithms. In: 2014 IEEE 26TH INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE. **Proceedings...** [S.l.], 2014. p. 55–62.

JONES, G.; WILLETT, P. Docking small-molecule ligands into active sites. **Curr. Opin. Biotechnol.**, v. 6, n. 6, p. 652–656, 1995.

KELLENBERGER, E. et al. Comparative evaluation of eight docking tools for docking and virtual screening accuracy. **Proteins: Structure, Function, and Bioinformatics**, v. 57, n. 2, p. 225–242, 2004.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: PROCEEDINGS OF ICNN'95 - INTERNATIONAL CONFERENCE ON NEURAL NETWORKS. **Proceedings...** [S.l.], 1995. v. 4, p. 1942–1948 vol.4.

KEPPEL, G. **Design and analysis: A researcher's handbook**. 3. ed. [S.l.]: Prentice-Hall, Inc, 1991.

KING, N. M. et al. Lack of synergy for inhibitors targeting a multi-drug-resistant hiv-1 protease. **Protein Science**, v. 11, n. 2, p. 418–429, 2002.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **Science**, v. 220 4598, p. 671–80, 1983.

KITCHEN, D. B.; FURR J. R., B. J. Docking and scoring in virtual screening for drug discovery: methods and applications. **Nat. Rev. Drug. Discov.**, v. 3, n. 2, p. 935 – 949, 2004.

KRASNOGOR, N. **Studies on the Theory and Design Space of Memetic Algorithms**. Tese (Doutorado) — University of the West of England, 2002.

KRASNOGOR, N.; SMITH, J. Emergence of profitable search strategies based on a simple inheritance mechanism. **Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)**, Morgan Kaufmann, San Francisco, USA, p. 432–439, 2001.

KRASNOGOR, N.; SMITH, J. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. **IEEE Trans. Evol. Comput.**, v. 9, n. 5, p. 474–488, 2005.

KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion variance analysis. **Journal of the American Statistical Association**, Taylor & Francis, v. 47, n. 260, p. 583–621, 1952.

KUKKONEN, S.; LAMPINEN, J. Gde3: The third evolution step of generalized differential evolution. In: CONGRESS ON EVOLUTIONARY COMPUTATION (IEEE CEC). **Proceedings...** [S.l.], 2005. p. 443–450.

KUNTZ, I. D. et al. A geometric approach to macromolecule-ligand interactions. **Journal of Molecular Biology**, v. 161, n. 2, p. 269 – 288, 1982.

LADBURY, J. E. Just add water! the effect of water on the specificity of protein-ligand binding sites and its potential application to drug design. **Chem. and Bio.**, v. 3, n. 12, p. 973 – 980, 1996.

LANDER, E. et al. Initial sequencing and analysis of the human genome. **Nature**, Nature Publishing Group, v. 409, n. 6822, p. 860–921, 2001.

LANGE-SAVAGE, G. et al. Structure of hoe/bay 793 complexed to human immunodeficiency virus (hiv-1) protease in two different crystal forms structure/function relationship and influence of crystal packing. **European journal of biochemistry / FEBS**, v. 248, p. 313–22, 1997.

LAVECCHIA, A.; GIOVANNI, C. D. Virtual screening strategies in drug discovery: A critical review. **Current Medicinal Chemistry**, v. 20, n. 23, p. 2839–2860, 2013.

LEACH, A. R. Ligand docking to proteins with discrete side-chain flexibility. **J. Mol. Biol.**, v. 235, n. 1, p. 345–356, 1994.

LEHNINGER, A.; NELSON, D. L.; COX, M. M. **Principles of Biochemistry**. 4. ed. New York, NY, USA: W.H. Freeman, 2004.

LEONHART, P. F. et al. A biased random key genetic algorithm for the protein–ligand docking problem. **Soft Computing**, p. 1–22, 2018.

LESK, A. M. **Introduction to Bioinformatics**. 2. ed. [S.l.]: Oxford University Press, 2005.

LIANG, J. J.; SUGANTHAN, P. N. Dynamic multi-swarm particle swarm optimizer with local search. In: 2005 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION. **Proceedings...** [S.l.], 2005. v. 1, p. 522–528.

LIONTA, E. et al. Structure-based virtual screening for drug discovery: Principles, applications and recent advances. **Current Topics in Medicinal Chemistry**, v. 14, n. 16, p. 1923–1938, 2014.

LóPEZ-CAMACHO, E. et al. jmetalcpp: optimizing molecular docking problems with a c++ metaheuristic framework. **Bioinformatics**, v. 30, n. 3, p. 437–438, 2013.

LóPEZ-CAMACHO, E. et al. Solving molecular flexible docking problems with metaheuristics: A comparative study. **Appl. Soft Comput.**, v. 28, n. 28, p. 379–393, 2014.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: GLOVER, F.; KOCHENBERGER, G. A. (Ed.). **Handbook of Metaheuristics**. Boston, MA: Springer US, 2003. p. 320–353.

LOZANO, M. et al. Real-coded memetic algorithms with crossover hill-climbing. **Evolutionary Computation**, v. 12, n. 3, p. 273–302, 2004.

LUKE, S. **Essentials of Metaheuristics**. 2. ed. Department of Computer Science, George Mason University: Lulu, 2013. 263 p.

LUSCOMBE, N. M.; GREENBAUM, D.; GERSTEIN, M. What is Bioinformatics? A proposed definition and overview of the field. **Methods Inf. Med.**, New Haven, CT, USA., v. 40, n. 4, p. 346–358, 2001.

MACHADO, K. S. et al. Fredows: a method to automate molecular docking simulations with explicit receptor flexibility and snapshots selection. **BMC genomics**, v. 12, n. 4, p. 1, 2011.

MAGALHAES, C. S. D. **Algoritmos Geneticos para o Problema de Docking Proteina-Ligante**. Tese (Doutorado) — Laboratorio Nacional de Computacao Cientifica, Petropolis, RJ, Brasil, 2006.

MAGALHAES, C. S. d.; BARBOSA, H. A. J.; DARDENNE, L. E. A genetic algorithm for the ligand-protein docking problem. **Genet. Mol. Biol.**, v. 27, p. 605 – 610, 2004.

MAGALHÃES, C. S. de et al. A dynamic niching genetic algorithm strategy for docking highly flexible ligands. **Inf. Sci.**, v. 289, p. 206–224, 2014.

MARTIN, J. L. et al. Molecular recognition of macrocyclic peptidomimetic inhibitors by hiv-1 protease † , ‡. **Biochemistry**, v. 38, p. 7978–88, 1999.

MCREE, D. E. **Practical protein crystallography**. 2. ed. London, UK: Academic Press, 1999. 477 p.

MEIER, R. et al. Paradocks: A framework for molecular docking with population-based metaheuristics. **J. Chem. Inf. Model.**, v. 50, n. 5, p. 879–889, 2010.

MEILER, J.; BAKER, D. Rosettaligand: Protein–small molecule docking with full side-chain flexibility. **Proteins: Structure, Function, and Bioinformatics**, v. 65, n. 3, p. 538–548, 2006.

MITCHELL, J. B. O. et al. Bleep—potential of mean force describing protein–ligand interactions: Ii. calculation of binding energies and comparison with experimental data. **J. Comput. Chem.**, v. 20, n. 11, p. 1177–1185, 1999.

MITCHELL, M. **An Introduction to Genetic Algorithms**. Cambridge, MA, USA: MIT Press, 1998.

MOLINA, D.; LOZANO, M.; HERRERA, F. Memetic algorithm with local search chaining for continuous optimization problems: A scalability test. **ISDA 2009 - 9th International Conference on Intelligent Systems Design and Applications**, p. 1068–1073, 2009.

MOLINA, D. et al. Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains. **Soft Computing**, v. 15, n. 11, p. 2201–2220, 2011.

MORRIS, G. M. et al. Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. **Journal of Computational Chemistry**, v. 19, n. 14, p. 1639–1662, 1998.

MORRIS, G. M. et al. Autodock4 and autodocktools4: Automated docking with selective receptor flexibility. **J. Comput. Chem.**, v. 30, n. 16, p. 2785–2791, 2009.

MOSCATO, P. **On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts - Towards Memetic Algorithms**. California Institute of Technology, Pasadena, 1989. 68 p.

MUEGGE, I. Effect of ligand volume correction on pmf scoring. **J. Comput. Chem.**, v. 22, n. 4, p. 418–425, 2001.

MUEGGE, I. Pmf scoring revisited. **J. Med. Chem.**, v. 49, n. 20, p. 5895–5902, 2006.

MUEGGE, I.; MARTIN, Y. C. A general and fast scoring function for protein-ligand interactions: a simplified potential approach. **J. Med. Chem.**, v. 42, n. 5, p. 791–804, 1999.

MURTHY, K. et al. The crystal structures at 2.2-a resolution of hydroxyethylene-based inhibitors bound to human immunodeficiency virus type 1 protease show that the inhibitors are present in two distinct orientations. **The Journal of biological chemistry**, v. 267, p. 22770–8, 1992.

NEBRO, A. et al. Smpso: A new pso-based metaheuristic for multi-objective optimization. In: 2009 IEEE SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE IN MULTICRITERIA DECISION-MAKING. **Proceedings...** [S.l.], 2009. p. 66–73.

NELDER, J.; MEAD, R. A simplex method for function minimization. **Comput. J.**, v. 7, n. 4, p. 308–313, 1965.

O'BOYLE, N. M. et al. Open babel: An open chemical toolbox. **J. Cheminf.**, v. 3, n. 1, p. 1–14, 2011.

ONG, Y. S.; KEANE, A. J. Meta-lamarckian learning in memetic algorithms. **IEEE Transactions on Evolutionary Computation**, v. 8, n. 2, p. 99–110, 2004.

PAPADIMITRIOU, C. H.; STEIGLITZ, K. **Combinatorial optimization: algorithms and complexity**. [S.l.]: Courier Corporation, 1998.

PAULING, L.; DELBRUCK, M. The nature of the intermolecular forces operative in biological processes. **Science**, Am. Assoc. Adv. Sci., v. 92, n. 2378, p. 77–79, 1940.

PEARLMAN, D. A.; CHARIFSON, P. S. Are free energy calculations useful in practice? a comparison with rapid scoring functions for the p38 map kinase protein system. **J. Med. Chem.**, v. 44, n. 21, p. 3417–3423, 2001.

RAREY, M. et al. A fast flexible docking method using an incremental construction algorithm. **J. Mol. Biol.**, Elsevier, v. 261, n. 3, p. 470–489, 1996.

REDDY, A. S. et al. Virtual screening in drug discovery - a computational perspective. **Current Protein & Peptide Science**, v. 8, n. 4, p. 329–351, 2007.

REILING, K. K. Anisotropic dynamics of the je-2147-hiv protease complex: Drug resistance and thermodynamic binding mode examined in a 1.09 inch structure. 01 2002.

REPASKY, M. P.; SHELLEY, M.; FRIESNER, R. A. Flexible ligand docking with glide. **Current Protocols in Bioinformatics**, v. 18, n. 1, p. 8.12.1–8.12.36.

RICHARDSON, J. S. Advances in protein chemistry. In: **The Anatomy and Taxonomy of Protein Structure**. [S.l.]: Academic Press, 1981. v. 34, p. 167 – 339.

ROSENBROCK, H. H. An Automatic Method for Finding the Greatest or Least Value of a Function. **The Computer Journal**, v. 3, n. 3, p. 175–184, 1960.

ROSIN, C. D. et al. A comparison of global and local search methods in drug docking. In: IN PROCEEDINGS OF THE SEVENTH INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS. **Proceedings...** [S.l.]: Morgan Kaufmann, 1997. p. 221–228.

RUIZ-TAGLE, B. et al. Evaluating the use of local search strategies for a memetic algorithm for the protein-ligand docking problem. In: 2017 36TH INTERNATIONAL CONFERENCE OF THE CHILEAN COMPUTER SCIENCE SOCIETY (SCCC). **Proceedings...** [S.l.], 2017. p. 1–12.

SADJAD, B.; ZSOLDOS, Z. Toward a robust search method for the protein-drug docking problem. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, v. 8, p. 1120–1133, 2011.

SCHAAL, W. et al. Synthesis and comparative molecular field analysis (comfa) of symmetric and nonsymmetric cyclic sulfamide hiv-1 protease inhibitors. **Journal of medicinal chemistry**, v. 44, 2002.

SCHNEIDER, G.; BöHM, H. J. Virtual screening and fast automated docking methods. **Drug Discovery Today**, v. 7, n. 1, p. 64 – 70, 2002.

SCHRÖDINGER, L. **The AxPyMOL Molecular Graphics Plugin for Microsoft PowerPoint**. 2015.

SHI, Y.; EBERHART, R. A modified particle swarm optimizer. In: 1998 IEEE INTERNATIONAL CONFERENCE ON EVOLUTIONARY COMPUTATION PROCEEDINGS. IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE (CAT. NO.98TH8360). **Proceedings...** [S.l.], 1998. p. 69–73.

SIMONSEN, M. et al. Gpu-accelerated high-accuracy molecular docking using guided differential evolution. **Nat. Comput. Ser.**, p. 349–368, 2013.

SMITH, J. E. Coevolving memetic algorithms: A review and progress report. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, v. 37, n. 1, p. 6–17, 2007.

SOUSA, S. et al. Protein-ligand docking in the new millennium a retrospective of 10 years in the field. **Curr. Med. Chem.**, v. 20, n. 18, p. 2296–2314, 2013.

SPEARS, V. M.; JONG, K. A. D. On the virtues of parameterized uniform crossover. In: IN PROCEEDINGS OF THE FOURTH INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS. **Proceedings...** [S.l.], 1991. p. 230–236.

SPIELER, E. de O. **Um algoritmo genético de chaves aleatórias viciadas para o problema de atracamento molecular**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, 2016.

STOLL, V. et al. X-ray crystallographic structure of abt-378 (lopinavir) bound to hiv-1 protease. **Bioorganic & Medicinal Chemistry**, v. 10, n. 8, p. 2803 – 2806, 2002.

STORN, R.; PRICE, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, 1997.

TALBI, E. ghazali. Common concepts for metaheuristics. In: **Metaheuristics**. [S.l.]: John Wiley & Sons, Ltd, 2009. cap. 1, p. 1–86.

TEAGUE, S. J. Implications of protein flexibility for drug discovery. **Nat. Rev. Drug Discovery**, v. 2, n. 7, p. 527–541, 2003.

TEODORO, M. L.; KAVRAKI, L. E. Conformational flexibility models for the receptor in structure based drug design. **Curr. Pharm. Des.**, Bentham Science Publishers, v. 9, n. 20, p. 1635–1648, 2003.

THANKI, N. et al. Crystal structure of a complex of hiv-1 protease with a dihydroxyethylene-containing inhibitor: Comparisons with molecular modeling. **Protein science : a publication of the Protein Society**, v. 1, p. 1061–72, 1992.

TOTROV, M.; ABAGYAN, R. Flexible protein–ligand docking by global energy optimization in internal coordinates. **Proteins: Structure, Function, and Bioinformatics**, v. 29, n. S1, p. 215–220, 1997.

TRAMONTANO, A.; LESK, A. M. **Protein structure prediction: concepts and applications**. 1. ed. Weinheim, Germany: [s.n.], 2006.

TROTT, O.; OLSON, A. J. Autodock vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. **J. Comput. Chem.**, v. 31, n. 2, p. 455–461, 2010.

UNWIN, P.; HENDERSON, R. Molecular structure determination by electron microscopy of unstained crystalline specimens. **Journal of Molecular Biology**, v. 94, n. 3, p. 425 – 440, 1975.

VERDONK, M. L. et al. Improved protein-ligand docking using gold. **Proteins: Struct., Funct., Bioinf.**, v. 52, n. 4, p. 609–623, 2003.

VERDONK, M. L. et al. Illustration of current challenges in molecular docking. In: ____. **Structure-Based Drug Discovery**. Dordrecht: Springer Netherlands, 2007. p. 201–221.

VERLI, H. Níveis de informação biológica. In: **Bioinformática: da Biologia à Flexibilidade Moleculares**. [S.l.: s.n.], 2014. cap. 2, p. 14–37.

WALTERS, W. P.; STAHL, M. T.; MURCKO, M. A. Virtual screening—an overview. **Drug Discovery Today**, v. 3, n. 4, p. 160–178, 1998.

WANG, R.; LU, Y.; WANG, S. Comparative evaluation of 11 scoring functions for molecular docking. **J. Med. Chem.**, v. 46, n. 12, p. 2287–2303, 2003.

WEI, B. Q. et al. Testing a flexible-receptor docking algorithm in a model binding site. **Journal of Molecular Biology**, v. 337, n. 5, p. 1161–1182, 2004.

WEINER, S. J. et al. A new force field for molecular mechanical simulation of nucleic acids and proteins. **J. Am. Chem. Soc.**, v. 106, n. 3, p. 765–784, 1984.

WEISE, T. Global optimization algorithms-theory and application. **Learning Technology – LTTF Newsletter**, p. 26–28, 2009.

WHISSTOCK J. C.; LESK, A. M. Prediction of protein function from protein sequence and structure. Q. Rev. Biophys., Cambridge University Press, v. 36, n. 3, p. 307–340, 2003.

WHITLEY, D.; GORDON, V. S.; MATHIAS, K. Lamarckian evolution, the baldwin effect and function optimization. In: DAVIDOR, Y.; SCHWEFEL, H.-P.; MÄNNER, R. (Ed.). **Proceedings...** Berlin, Heidelberg: Springer Berlin Heidelberg, 1994. p. 5–15.

WHITLEY, L. D.; KAUTH, J. Genitor: a different genetic algorithm. In: TECHNICAL REPORT (COLORADO STATE UNIVERSITY. DEPARTMENT OF COMPUTER SCIENCE). **Proceedings...** [S.l.]: Fort Collins, Colorado, 1988. p. 88–101.

WONG, C. F. Flexible ligand–flexible protein docking in protein kinase systems. **Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics**, v. 1784, n. 1, p. 244–251, 2008.

ZHANG, C. et al. A knowledge-based energy function for protein-ligand, protein-protein, and protein-dna complexes. **J. Med. Chem.**, v. 48, n. 7, p. 2325–2335, 2005.

ZHANG, Y.; SKOLNICK, J. Scoring function for automated assessment of protein structure template quality. **Proteins: Structure, Function, and Bioinformatics**, v. 57, n. 4, p. 702–710, 2004.

ZHANG, Y. et al. Using ensemble methods to deal with imbalanced data in predicting protein-protein interactions. **Comput. Biol. Chem.**, v. 36, p. 36–41, 2012.