UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de São José do Rio Preto

WILLIAM TENÓRIO

Algoritmo genético adaptativo e paralelo para seleção de polimorfismos de nucleotídeo único representativos

São José do Rio Preto
2019

WILLIAM TENÓRIO

Algoritmo genético adaptativo e paralelo para seleção de polimorfismos de nucleotídeo único representativos

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Financiador: Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq. Proc. 133733/2017-3.

Orientador: Prof. Dr. Carlos Roberto Valêncio

São José do Rio Preto
2019

T312a

Tenório, William

Algoritmo genético adaptativo e paralelo para seleção de polimorfismos de nucleotídeo único representativos / William Tenório. -- São José do Rio Preto, 2019

80 f. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto

Orientador: Carlos Roberto Valêncio

1. Ciência da computação. 2. Algoritmos genéticos. 3. Polimorfismo de nucleotídeo único. 4. Processamento paralelo (Computadores). I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

WILLIAM TENÓRIO

Algoritmo genético adaptativo e paralelo para seleção de polimorfismos de nucleotídeo único representativos

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Financiador: Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq. Proc. 133733/2017-3.

Comissão examinadora:

Prof. Dr. Carlos Roberto Valêncio
UNESP – São José do Rio Preto – SP
Orientador

Prof. Dr. Geraldo Francisco Donegá Zafalon
UNESP – São José do Rio Preto – SP

Prof. Dr. Angelo César Colombini
Universidade Federal Fluminense – Niterói - RJ

São José do Rio Preto
03 de setembro de 2019

Agradecimentos

Agradeço a todos aqueles que, de alguma forma, permitiram que este trabalho se concretizasse. Em especial, agradeço:

Aos meus pais, Fátima e Moisés, por todo amor e por nunca terem medido esforços para que eu e meu irmão pudéssemos estudar e realizar nossas conquistas.

Ao meu irmão Wanderson, por sempre ter sido um exemplo de pessoa e profissional.

Ao Prof. Dr. Carlos Roberto Valêncio, meu orientador da vida, pela confiança, paciência e ajuda em todos os momentos da minha trajetória nesta universidade. Sou grato pelas oportunidades que me foram proporcionadas, as quais contribuíram imensamente na construção do ser humano que hoje sou.

Ao Prof. Dr. Angelo César Colombini e Prof. Dr. Geraldo Francisco Donegá Zafalon, e à Profa. Dra. Rogéria Cristiane Gratão de Souza, pelas contribuições dadas no decorrer deste trabalho.

À bibliotecária Luciane Passoni, por todo apoio dado para a formatação deste texto.

Aos queridos companheiros do Grupo de Banco de Dados – GBD, por todos os momentos e experiências que compartilhamos.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq, pela concessão da bolsa de pesquisa, sob o processo nº 133733/2017-3.

"Tenho pensamentos que, se pudesse revelá-los e fazê-los viver, acrescentariam nova luminosidade às estrelas, nova beleza ao mundo e maior amor ao coração dos homens."

Fernando Pessoa (1976)

Resumo

Polimorfismos de Nucleotídeo Único (SNPs) são uma ferramenta promissora nos estudos de doenças. Contudo, a análise de todos os SNPs do genoma humano é uma tarefa custosa computacionalmente. Neste cenário, descobriu-se a possibilidade de utilizar um subconjunto de SNPs, chamado tag SNPs, que fosse representativo o suficiente para ser utilizado em estudos, sem que houvesse necessidade de lidar com o conjunto completo. Devido à sua alta complexidade, diversas abordagens foram propostas para lidar com este problema a partir de meta-heurísticas, como os Algoritmos Genéticos. Contudo, dentro do processo evolutivo destes algoritmos, as soluções são influenciadas pelo contexto do problema, ao qual se atribuiu o nome de pressão seletiva e que pode impactar de maneira negativa os resultados encontrados. Apesar da pressão seletiva poder ser controlada, o processamento adicional para o cálculo destes métodos pode acarretar aumento do custo computacional, o que pode inviabilizar sua aplicação. Desta forma, a contribuição científica deste trabalho está na proposição de um método paralelo para seleção de SNPs representativos baseado em algoritmos genéticos com controle de pressão seletiva que, quando comparado aos demais da literatura, apresenta maior diversidade de indivíduos nas populações, maior velocidade de convergência e, consequentemente, melhor resultado das soluções encontradas pelo algoritmo. Os resultados indicaram que o algoritmo desenvolvido foi capaz de reduzir em 27% a quantidade de iterações até a convergência, assim como aumentar o valor de aptidão das soluções em 11%. Adicionalmente, o algoritmo foi capaz de lidar com maiores volumes de dados e apresenta uma taxa de crescimento do tempo de processamento 3,7 vezes menor do que um algoritmo sequencial tradicional.

Palavras-chave: Ciência da computação. Algoritmos genéticos. Polimorfismo de nucleotídeo único. Processamento paralelo (Computadores).

Abstract

Single Nucleotide Polymorphisms (SNPs) work as a promising tool to support the study of diseases. Despite this fact, the analysis of all SNPs from human genome is very expensive task from the computational perspective. In that point, genetic related researches found the existence of a small subset of SNPs, named tag SNPs, that can be used to handle the studies, rather than use the complete set of SNPs. Since the problem of finding this subset is a complex task, different metaheuristics were proposed to deal with it, such as Genetic Algorithms. However, a feature named selective pressure can negatively affect the results of these algorithms during the evolutionary process. Even though this feature can be controlled, the additional processing computation performed to deal with it can make its application impracticable. In that sense, the scientific contribution of this work is the proposition of a parallel strategy to find tag SNPs based on genetic algorithms with selective pressure control. This algorithm present higher diversity rate of individuals inside population, higher convergency speed e better solution when compared to related works. The experiments showed that the develop algorithm was able to reduce the number of generations performed until convergence in 27%, as well as increase the fitness of the solutions in 11%. The results also showed that the algorithm is more efficient to deal with a higher volume of data, and present an increasing rate 3.7 times lower than the sequential approach.

Keywords: Computer science. Genetic algorithms. Single nucleotide polymorphism. Parallel processing.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ACO - Otimização por Colônia de Formigas

AG - Algoritmo Genético

bff - *Best Fitness Frequency*

DNA - Ácido Desoxirribonucleico

KDD - Descoberta de Conhecimento em Bases de Dados

LD - Desequilíbrio de Ligação

MR - MapReduce

PSO - Otimização por Enxame de Partículas

RDD - *Resilient Distributed Datasets*

SNP - Polimorfismo de Nucleotídeo Único

Sumário

1 Introdução

Polimorfismos de Nucleotídeo Único (SNPs, lê-se “snips”, do inglês *Single Nucleotide Polymorphisms*) são variações nos blocos construtores do Ácido Desoxirribonucleico (DNA, do inglês *Deoxyribonucleic Acid*), chamados de nucleotídeos, e representam a variação genética mais comumente encontrada na espécie humana. Pesquisadores têm mostrado que os SNPs podem ajudar a predizer o risco de indivíduos ou populações desenvolverem alguma doença, assim como predizer a reação à determinada droga (GRAY; CAMPBELL; SPURR, 2000).

A identificação de variações genéticas que definem a origem e causa de doenças comuns ou complexas é o objetivo primário das pesquisas atuais em epidemiologia, medicina e farmacologia. Contudo, o conhecimento científico de doenças humanas é ainda limitado, uma vez que há um número muito grande de variações genéticas no genoma humano: em média, um SNP é identificado a cada 290 pares de bases do DNA, o que implica na existência de onze milhões de SNPs nos mais de três bilhões de pares de bases do genoma humano (SHERRY et al., 2001). À medida que os experimentos relacionados à identificação de SNPs são realizados e se tornam procedimentos rotineiros, os dados gerados se acumulam rapidamente, o que torna prioritário o desenvolvimento de métodos eficientes que possam explorar o potencial desta técnica. Para atender a esta necessidade, o processo de descoberta de conhecimento em bases de dados (KDD, do inglês *Knowledge Discovery in Databases*) tem sido utilizado, uma vez que permite o tratamento dos dados, assim como a descoberta de informações implícitas e úteis em grandes volumes de dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

1.1 Motivação e escopo

As novas tecnologias de sequenciamento genético, denominadas de tecnologias de sequenciamento de nova geração (do inglês, *Next Generation Sequencing*), começaram a ser comercializadas em 2005 e, deste então, têm evoluído rapidamente (ZEGGINI; MORRIS, 2010). Essas tecnologias promovem o sequenciamento de DNA em plataformas capazes de gerar informação sobre milhões de pares de bases em uma única corrida (KAISLER et al., 2013). Desta forma, o desafio não está mais relacionado à geração de novos dados, mas sim a como estes grandes volumes de dados podem ser eficientemente analisados, a fim de que sejam transformados em informações úteis que contribuam com o crescimento do conhecimento científico.

Neste cenário, a análise de todos os SNPs do genoma humano se apresenta como uma tarefa custosa do ponto de vista computacional, além de desnecessária (LIAO et al., 2017). Isto por que foi descoberta a possibilidade de selecionar um pequeno número de SNPs que é capaz de representar o conjunto de SNPs total, ao qual se atribuiu o nome de Seleção de SNPs Representativos, ou tag SNPs. Assim, surgiu o processo de seleção de SNPs representativos, que tem como objetivo selecionar um subconjunto de SNPs que apresente informações suficientes para conduzir estudos genômicos e que, ao mesmo tempo, seja pequeno o suficiente para reduzir a sobrecarga das etapas de processamento e análise.

Este problema, tal como demostrado por Bafna et al. (2003), tem complexidade *NP-hard*. Diante disso, foram propostas diversas abordagens com base na aplicação de meta-heurísticas para sua resolução, como Algoritmos Genéticos (AG) (MOUAWAD; MANSOUR, 2014; MAHDEVAR et al., 2010), Otimização por Colônia de Formigas (LIAO et al., 2012), Recozimento Simulado (ÜSTÜNKAR et al., 2012), entre outras. Dentre elas, os Algoritmos Genéticos se destacam, superando-as pelas suas boas propriedades de convergência (DAS; MULLICK; SUGANTHAN, 2016).

Os Algoritmos Genéticos (AG) são técnicas de busca por soluções em problemas complexos desenvolvidas na década de 1960 com base na teoria da evolução das espécies, descrita por Charles Darwin. A teoria diz respeito a um processo originado pela criação de uma população, seguido da aplicação de operadores de seleção, cruzamento e mutação dos indivíduos que evoluem no decorrer de gerações (LINDEN, 2008).

Dentro desse processo evolutivo há a influência do contexto no qual o problema está inserido, que atua sobre os indivíduos da população. Essa influência é denominada pressão seletiva (BÄCK, 1994). A pressão seletiva imposta a uma população pode ser controlada por meio de métodos de controle, o que, se realizado de maneira coesa, pode proporcionar

melhores resultados no decorrer do processo evolutivo (BÄCK, 1994). Na maioria dos trabalhos que fazem uso de AGs encontrados na literatura científica, não há discussão sobre o controle da pressão seletiva: os mecanismos que possuem influência sobre este aspecto, como estratégias evolutivas e parâmetros iniciais do algoritmo, são definidos de maneira empírica, muitas vezes sem a presença de discussões a respeito (ALETI; MOSER, 2016).

Por outro lado, existe também a dificuldade de se realizar a análise de um grande conjunto de soluções, mesmo para as técnicas de otimização que fazem uso de meta-heurísticas. A análise de uma grande quantidade de soluções, aliada ao processamento adicional dos métodos de controle de pressão seletiva, podem acarretar em tempo de processamento elevados. Neste cenário, a computação paralela e o processamento de dados em memória têm se tornado alternativas viáveis, pois permitem a resolução de problemas complexos de maneira mais rápida.

Os AGs, tal como constatado por Linden (2008), se apresentam como intrinsecamente paralelos: a geração e avaliação dos novos indivíduos durante o processo evolutivo, assim como o processo evolutivo em si, podem ser executadas em paralelo. Dessa forma pode ser possível reduzir o tempo de execução adicionando mais processadores e, além disso, permitir a avaliação de várias populações simultaneamente, a fim de explorar de forma mais aprofundada o espaço de soluções.

1.2 Objetivos gerais e específicos

O presente trabalho tem como objetivo a implementação de um algoritmo genético eficiente que faça o uso do processamento paralelo de dados em memória e de métodos de controle de pressão seletiva como forma de otimização do tempo de execução e aceleração da convergência para resolução de problemas de seleção de SNPs representativos.

Para tal, primeiramente, foi verificada a influência que o controle de pressão seletiva tem sobre a evolução dos algoritmos genéticos. Dessa forma, verificou-se o impacto da utilização dos métodos de controle da pressão seletiva em algoritmos genéticos por meio da comparação dos resultados produzidos na presença e ausência destes mecanismos de controle.

Posteriormente, verificou-se a influência da aplicação das técnicas de processamento paralelo e processamento em memória na qualidade dos resultados obtidos, assim como no tempo de processamento do algoritmo.

1.3 Contribuições

A contribuição principal deste trabalho é a proposição de um método paralelo para seleção de SNPs representativos baseado em algoritmos genéticos com controle de pressão seletiva. Esta contribuição permite a seleção de um conjunto de SNPs mais representativo em relação aos demais e com tempo de processamento reduzido. Os resultados experimentais indicam que o algoritmo desenvolvido foi capaz de encontrar soluções até 11% melhores, em média, e apresentou crescimento do tempo de processamento 3,7 vezes menor do que um abordagem sequencial.

Adicionalmente, este trabalho contribui com a validação da hipótese de que os algoritmos genéticos, quando executados de maneira paralela e na presença de mecanismos de controle de pressão seletiva, possibilitam maior exploração do espaço de busca e maior velocidade de convergência para as soluções, além de apresentar desempenho superior no que diz respeito à qualidade das soluções.

1.4 Organização da dissertação

O conteúdo deste trabalho é apresentado tal como segue: na Seção 1 foi introduzido o problema de seleção de SNPs representativos, assim como foi apresentada a motivação, os objetivos e as contribuições do trabalho; na Seção 2 é apresentada a fundamentação teórica e o estado da arte relacionados aos conceitos biológicos relevantes para o correto entendimento do trabalho, à computação paralela e aos algoritmos genéticos, assim como uma discussão a respeito da necessidade de aplicação de mecanismos de pressão seletiva nestes algoritmos; nesta seção também é apresentada uma revisão da literatura científica; na Seção 3 o trabalho desenvolvido é apresentado em detalhes; na Seção 4 são apresentados os procedimentos realizados para validar o trabalho, os resultados da avaliação experimental realizada e as discussões a respeito dos resultados experimentais; por fim, na Seção 5 são apresentadas as conclusões do trabalho, as contribuições científicas são enfatizadas e os trabalhos futuros são direcionados.

2 Fundamentação teórica

A seleção de polimorfismos de nucleotídeo único representativos é uma técnica multidisciplinar, no qual estão presentes conceitos da biologia molecular e da ciência da computação. Nesta seção são apresentados os principais conceitos biológicos referentes aos estudos de variação gênica, com foco na análise de polimorfismos. Posteriormente, é definido o problema de seleção de polimorfismos de nucleotídeo único representativos. Na sequência, é descrito o funcionamento dos Algoritmos Genéticos, a saber: são abordados, desde os conceitos básicos, até algoritmos mais complexos no qual há considerável preocupação com desempenho e controle de qualidade. Também são abordados os principais modelos de programação e computação paralela atuais.

Por fim, o levantamento bibliográfico a respeito de cada tema é apresentado.

2.1 Genética e biologia molecular

Esta subseção dedica-se à apresentação dos conceitos elementares em Genética e Biologia Molecular. Os conceitos são descritos com base nas obras *Biologia Molecular e Celular* (LODISH et al., 2014) e *Genética Molecular Humana: Mecanismos das Doenças Hereditárias* (PASTERNAK, 2002).

2.1.1 Cromossomos, DNA e nucleotídeos

O genoma humano representa toda a informação genética que um organismo possui. O material genético em humanos está presente em cada célula do corpo, contido em cromossomos localizados no núcleo das células. O DNA, por sua vez, é um composto orgânico no qual suas moléculas contêm instruções genéticas que coordenam o

desenvolvimento e funcionamento dos seres vivos, tal como descrito Watson e Crick (1953). É a partir dessas informações que todas as células se reproduzem e que todas as características herdadas dos pais e outros ascendentes se manifestam. Em sua formação, há a união de compostos químicos chamados nucleotídeos, que são divididos em três partes: uma base nitrogenada, um ácido fosfórico e um açúcar desoxirribose.

2.1.2 Polimorfismos genéticos e SNPs

Quando são comparadas as sequências genéticas de diferentes indivíduos de uma população no mesmo cromossomo, pode-se notar que grande parte da sequência genética é completamente similar para todos: seres humanos do mesmo sexo compartilham em torno de 99% de sua sequência de DNA. Este fato permite que os pesquisadores trabalhem com uma única sequência de referência, no qual apenas pequenas variações genômicas fundamentam boa parte da variabilidade fenotípica entre os indivíduos. Tais variações são denominadas polimorfismos genéticos.

Um SNP é um tipo de polimorfismo genético que ocorre quando há substituição de um único nucleotídeo em certa posição do DNA (por exemplo, uma Adenina na sequência de pares de bases do cromossomo é substituída por uma Citosina). Neste caso, o nucleotídeo é chamado de alelo. Como exemplo, seja os quatro cromossomos listados na sequência:

1. A T C A **A** G C **C** A
2. A T C A **A** G C **A** A
3. A T C A **T** G C **C** A
4. A T C A **A** G C **C** A

É possível observar que nas posições 5 e 8 (em negrito) destas sequências há SNPs. Nas sequências 1, 2 e 4 existes adeninas (A) na posição 5, enquanto que na sequência 3 existe uma timina (T); de modo similar, nas sequências 1, 3 e 4 existem citosinas (C), enquanto que na sequência 2 existe uma adenina (A).

2.1.3 Haplótipos

Para que novos indivíduos sejam gerados, é importante que os cromossomos sejam passados de pais para filhos de forma aleatória, podendo haver várias combinações diferentes logo na primeira geração. Caraterísticas codificadas por um mesmo cromossomo não são sempre passadas em bloco de pai para filho, ou seja, existe mistura entre o material genético proveniente da mãe e do pai; neste processo, chamado de recombinação, os cromossomos do pai e da mãe sofrem troca de material genético.

Neste cenário, um haplótipo é um conjunto de alelos presentes em uma determinada região do cromossomo que não foram separados durante o processo de recombinação. Como a variabilidade genética do genoma humano é muito alta, a probabilidade de que indivíduos não relacionados apresentem um mesmo haplótipo é praticamente nula, logo os haplótipos se apresentam como uma ferramenta útil na determinação da relação gênica entre indivíduos, assim como no estudo da origem de mutações causadoras de doenças.

2.1.4 Desequilíbrio de ligação e blocos haplotípicos

Uma característica dos haplótipos é a associação não aleatória entre os SNPs que os compõem, conhecida como desequilíbrio de ligação (LD, do inglês *Linkage Disequibilibrium*). Como a recombinação pode ocorrer em qualquer posição ao longo dos cromossomos do pai e da mãe, e em qualquer quantidade, há uma característica de independência entre os SNPs, o qual é conhecida como equilíbrio de ligação.

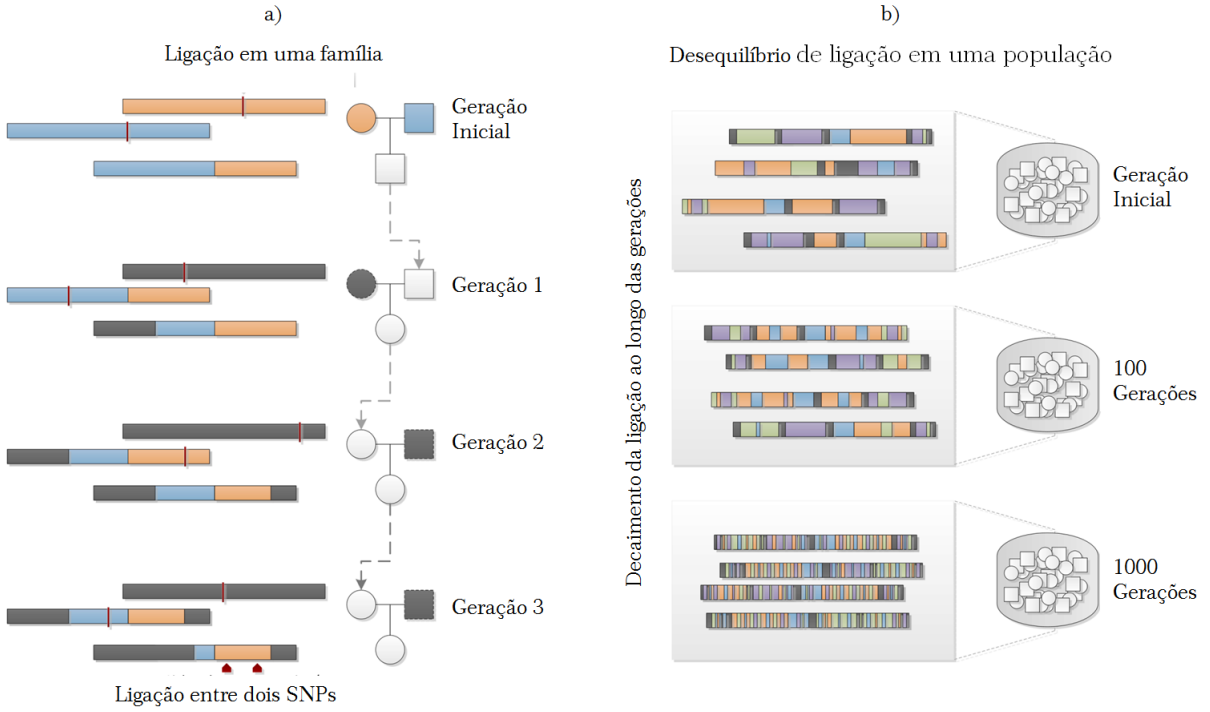
Contudo, quando dois alelos de diferentes SNPs não são independentes, ou seja, quando a ocorrência de um alelo em um SNP está relacionada à ocorrência de outro, em outro SNP, estes são caracterizados como em estado de desequilíbrio de ligação. Assim, estes SNPs tendem a serem passados juntos para os descendentes e, como resultado, os alelos do descendente serão altamente correlacionados uns com os outros.

Por meio da Figura 1a é possível observar uma sequência de eventos de recombinação em uma família, nos quais dois cromossomos pais são recombinados em um ponto de recombinação (linha vermelha), o que, no decorrer das gerações, quebra os seguimentos destes cromossomos pais. Essa quebra dos seguimentos é amplificada a cada geração, então, quando se considera uma população inteira, os eventos de recombinação irão quebrar os seguimentos de cromossomos até que, eventualmente, todos os alelos na população estejam em equilíbrio de ligação, ou seja, sejam independentes (Figura 1b).

2.2 Seleção de SNPs representativos

A seleção de SNPs representativos, ou tag SNPs, é motivada pelo desequilíbrio de ligação entre SNPs. Tal como apresentado na Seção 2.1.4, quando um alto valor de desequilíbrio de ligação existe entre SNPs, a informação dos alelos é altamente correlacionada e cada SNP pode atuar como representativo para os demais. Assim, dado um grande conjunto de SNPs, a seleção de tag SNPs visa encontrar um subconjunto de SNPs no qual a informação dos seus alelos possa reter a informação dos alelos de todos os demais

Figura 1— Ligação e Desequilíbrio de Ligação



Fonte: Extraído de Bush e Moore (2012)

SNPs (WANG et al., 2017). Neste cenário, os SNPs selecionados são conhecidos como tag SNPs.

Formalmente, o problema de seleção de SNPs representativos é definido como: seja $V = \{SNP_1, SNP_2, \dots, SNP_n\}$ um conjunto de n SNPs, $D = \{h_1, h_2, \dots, h_m\}$ um conjunto de m haplótipos, no qual cada haplótipo h_i consiste na informação dos alelos dos n SNPs, e k a quantidade máxima de SNPs que podem ser selecionados. A fim de que seja selecionado um subconjunto $V' \subset V$de k SNPs que contenha a informação dos alelos de todos os $SNP_i \in V$, dentre todos os subconjuntos possíveis, se faz necessária a presença de uma função $f(V'|D)$ que seja capaz de avaliar quão bem o subconjunto V' representa o conjunto V. Logo, um problema de seleção de tag SNPs terá como objetivo maximizar o valor retornado pela função f, assim como representado pela Equação 1.

$$\max_{V' tal\ que\ V' \subset V\ e\ |V'| \leq k} f(V'|D) \quad (1)$$

Os esforços, nesta área, estão voltados à identificação de uma função objetivo $f(V'|D)$ que melhor represente a informação dos alelos de D por meio dos SNPs em V', assim como à identificação de um subconjunto de SNPs que otimize a função f, logo trata-se de um problema de otimização.

No geral, problemas de otimização são caracterizados por situações em que se deseja maximizar ou minimizar uma função numérica de várias variáveis, em um contexto no qual

podem existir restrições (CHONG; ZAK, 2013). Formalmente, um problema de otimização pode ser representado como:

$$
\begin{aligned}
&Minimizar \\
&\qquad F(X), X = [X_1, X_2, \dots, X_N]^T, X \in \mathbb{R}^n \\
&sujeito\ a \\
&\qquad g_j \geq 0, \qquad\qquad j = 1, 2, \dots, K \\
&\qquad h_k = 0, \qquad\qquad k = 1, 2, \dots, K \\
&\qquad {X_i}^{(L)} \leq X \leq {X_i}^{(U)}, \quad i = 1,2, \dots, n
\end{aligned}
\tag{2}
$$

no qual, $F(X)$ representa a função objetivo a ser otimizada e g_j, h_k, ${X_i}^{(L)}$ e ${X_i}^{(U)}$ são restrições ao problema no qual a otimização está sendo aplicada.

2.3 Algoritmos genéticos

Tal como introduzido na Subseção 1.1, a análise de todos os SNPs do genoma humano é uma tarefa custosa do ponto de vista computacional. Além disso, sabe-se que é possível selecionar um pequeno número de SNPs que seja capaz de representar o conjunto de SNPs completo. A generalização deste conceito é conhecida na literatura científica como seleção de atributos (NOVAKOVIĆ, 2016), que é uma das técnicas empregadas para reduzir a dimensionalidade de grandes conjuntos de dados.

Neste contexto, a fim de que seja escolhido o subconjunto de atributos que melhor represente os demais, estratégias de busca são empregadas para percorrer o conjunto de atributos e analisá-lo. Este procedimento pode ser realizado de duas maneiras: exaustiva ou heurística. Nos métodos exaustivos, todos os subconjuntos $V' \subset V$ possíveis são analisados, a fim de que seja encontrada uma solução ótima, ou seja, aquela que melhor represente o conjunto de atributos V original de acordo com uma determinada função de avaliação. Já na busca heurística, o objetivo nem sempre é encontrar a solução ótima do problema, mas sim uma solução viável a partir de sucessivas aproximações direcionadas a um ponto ótimo, que não envolvem a busca exaustiva.

Esta subjetividade, ou falta de precisão, não se trata de uma deficiência dos métodos heurísticos, uma vez que tais métodos permitem a redução do custo computacional e do tempo de processamento ao se evitar a busca exaustiva, além de oferecer soluções suficientemente boas dentro de determinado cenário. No contexto da seleção de SNPs representativos, a análise exaustiva se torna inviável devido ao alto volume.

Diversos algoritmos heurísticos foram propostos, contudo destacam-se na literatura os métodos inspirados por fenômenos da natureza, como os algoritmos genéticos

(HOLLAND, 1975), de colônia de formigas (MANIEZZO, 1992) e de enxame de partículas (KENNEDY; EBERHART, 1995). Dentre eles, os Algoritmos Genéticos se destacam, superando-os pelas suas boas propriedades de convergência (DAS; MULLICK; SUGANTHAN, 2016).

Os Algoritmos Genéticos (AG) são técnicas de busca inspiradas em mecanismos de seleção e genética natural (HOLLAND, 1975). Estes algoritmos podem ser vistos como uma representação matemática das teorias de Darwin e da genética, cujos principais postulados são:

a) a evolução é um processo que opera sobre os cromossomos do organismo e não sobre o organismo que os carrega. Desta maneira, o que ocorrer com o organismo, durante sua vida, não irá se refletir sobre seus cromossomos. Entretanto, o inverso não é verdadeiro, pois os cromossomos do organismo terão reflexo direto sobre todas as características desse organismo, uma vez que o indivíduo é a decodificação de seus cromossomos;
b) a seleção natural é o elo entre os cromossomos e o desempenho que suas estruturas decodificam, ou seja, o próprio organismo. O processo de seleção natural faz com que os cromossomos que decodificaram organismos mais bem adaptados ao seu meio ambiente, sobrevivam e reproduzam mais do que aqueles que decodificam organismos menos adaptados;
c) o processo de reprodução é o ponto por meio do qual a evolução se caracteriza. Mutações podem causar diferenças entre os cromossomos dos pais e o de seus filhos. Além disso, processos de recombinação podem fazer com que os cromossomos dos filhos sejam bastante diferentes dos de seus pais, uma vez que eles combinam materiais cromossômicos de dois genitores.

De posse dessas definições, os AGs foram desenvolvidos. Um AG é um algoritmo de busca baseado em processos genéticos e de seleção natural (GOLDBERG, 1989). A partir de uma população inicial de t_p soluções, a aplicação de operadores genéticos de seleção, cruzamento e mutação é realizada de modo que esta população inicial possa evoluir até convergir para uma solução factível.

Ao se considerar um problema de otimização, os AGs inicializam a busca da melhor solução a partir de um conjunto inicial de soluções aleatórias. Cada elemento do conjunto inicial de soluções, denominados indivíduos, é representado por uma cadeia de símbolos, chamada cromossomo. Além disso, cada elemento da população deve ser capaz de representar completamente uma possível solução do problema tratado.

Em seguida, os indivíduos da população inicial, de acordo com uma determinada função de avaliação, são selecionados para dar origem a uma nova população, por meio dos operadores genéticos de cruzamento e mutação. Este processo de geração de novas populações, que é representado no Algoritmo 1, é repetido iterativamente até que o AG satisfaça algum critério de parada.

Algoritmo 1 – Algoritmo Genético

```
p ← geraPopulacaoInicial()
avaliação de cada indivíduo de p para determinar sua aptidão
repita
    p ← aplica operador de seleção
    p ← aplica operador de cruzamento com probabilidade p_c
    p ← aplica operador de mutação com probabilidade p_m
    avalia cada indivíduo de p para determinar sua aptidão
até condição de parada ser atingida
```

Na resolução de problemas por meio da utilização de um AG, duas etapas iniciais importantes precisam ser realizadas, de acordo com Mazzucco (1999):

a) determinar uma forma adequada de representar as possíveis soluções do problema considerado, por meio de uma codificação em "cromossomos";
b) determinar uma função de avaliação das possíveis soluções, de modo que seja possível analisar quão boa determinada solução é para o problema considerado.

Nas duas subseções seguintes, os conceitos de codificação das soluções e função de avaliação são detalhados com base nas obras *Genetic Algorithm Essentials* (KRAMER, 2017) e *Introduction to Genetic Algorithm* (SIVANANDAM; DEEPA, 2008).

2.3.1 Codificação das soluções

A representação do problema diz respeito ao mapeamento das possíveis soluções presentes no espaço de busca em uma estrutura de dados que possa ser manipulada computacionalmente. A forma de representação das soluções possíveis em cromossomos varia de acordo com o problema. Nos trabalhos originais de Holland (1975), essas representações eram feitas somente por meio de codificação utilizando cadeias de bits.

Como exemplo, considere o problema de otimização abaixo.

$$\begin{aligned} &Maximizar \\ &\quad f(x) = 3x^3 + 9x^2 - 5x + 6, x \in \mathbb{Z} \\ &sujeito\ a \\ &\quad -6 \leq x \leq 6 \end{aligned} \tag{3}$$

Para este problema, se faz necessário criar uma representação que contemple todas as possíveis soluções: números inteiros no intervalo $[-6, 6]$. Assim, as soluções para este problema podem ser codificadas conforme representado abaixo, no qual as posições $p_i \in \{0,1\}$, a posição p_1 indica o sinal da solução (1 para positivo e 0 para negativo) e as posições p_2, p_3 e p_4 indicam a representação binária do número inteiro, seguindo a regra $p_2.2^2 + p_3.2^1 + p_4.2^0$. Logo, um indivíduo com a configuração $0|1|1|0$ representa, nesta codificação, o número -6.

$$\underbrace{p_1 \mid p_2 \mid p_3 \mid p_4}_{x}$$

Analogamente, as soluções para um problema que envolva duas variáveis, x e y, podem ser representadas conforme a descrição abaixo, no qual as quatro primeiras posições representam x e as quatro últimas, y.

$$\underbrace{p_1 \mid p_2 \mid p_3 \mid p_4}_{x} \mid \underbrace{p_5 \mid p_6 \mid p_7 \mid p_8}_{y}$$

Uma vez definida a codificação das soluções, se faz necessário definir como estas soluções serão avaliadas.

2.3.2 Função de avaliação

A função de avaliação é a maneira utilizada pelos AGs para determinar a qualidade de um indivíduo como solução do problema em questão. Para isso, esta função recebe um indivíduo da população, ou seja, uma possível solução para o problema, e retorna um escalar como medida de desempenho do indivíduo, o qual é comumente chamado de valor de aptidão.

Dada a generalidade dos AGs, a função de avaliação é a única representação real do problema em questão, logo deve ser escolhida com cuidado (LINDEN, 2008). Esta função deve embutir todo o conhecimento que se possui sobre o problema a ser resolvido, desde suas restrições até seus objetivos de qualidade. Por exemplo, no problema de otimização descrito na subseção anterior, há uma restrição a respeito do valor de x, que deve pertencer ao intervalo $[-6,6]$. Apesar disso, a codificação das soluções proposta possibilita a criação dos indivíduos $0|1|1|1$ e $1|1|1|1$, que representam as soluções -7 e 7, respectivamente. Logo, a

função de avaliação deveria retornar uma medida de qualidade ruim para estes casos, a fim de que a probabilidade de gerarem descendentes fosse diminuída no decorrer do algoritmo.

Uma vez definidos como as possíveis soluções do problema serão representadas e como serão avaliadas, são descritas, na sequência, as etapas necessárias para que os AGs possam explorar todo o espaço de busca a fim de analisar as possíveis soluções do problema.

2.3.3 Seleção

O processo de seleção ocorre após a aplicação da função de avaliação nos cromossomos. Ela desempenha o papel da seleção natural na evolução, de modo a selecionar, para sobreviver e reproduzir, os organismos melhor adaptados ao meio, no caso, os cromossomos com melhor valor na função de avaliação.

A maneira pela qual os cromossomos são selecionados para reprodução pode variar. Entretanto, é certo que os cromossomos melhor adaptados terão, necessariamente, uma probabilidade maior de sobrevivência e reprodução do que os de baixa função de adequação.

É importante ressaltar o termo utilizado na frase anterior: probabilidade maior. Indivíduos com valores ruins de avaliação também podem gerar descendentes. Isto por que até indivíduos com avaliação ruim podem ter características genéticas que sejam favoráveis à criação de um "superindivíduo". Estas características podem não estar presentes em nenhum outro cromossomo de nossa população.

Existe um grande número de estratégias de seleção. As mais comuns são: seleção proporcional e seleção por torneio.

No método de seleção proporcional cada indivíduo tem uma chance de sorteio proporcional ao seu respectivo valor de aptidão. Para tal, calcula-se a soma das aptidões de todos os indivíduos da população e, em seguida, calcula-se a aptidão relativa de cada indivíduo, que é o resultado da divisão da aptidão deste indivíduo pela soma das aptidões calculada. Apesar de muito utilizado, o método de seleção proporcional apresenta problemas. No início da execução do algoritmo, é comum encontrar grandes diferenças nas avaliações dos indivíduos, com indivíduos muito bem avaliados, enquanto a média da aptidão da população é baixa. Nestes casos, o AG irá produzir populações com baixa variedade de indivíduos, o que poderá resultar em uma convergência prematura ou incorreta.

Para lidar com esse problema, Goldberg (1989) contemplou também a possibilidade de que a função de avaliação fosse recalculada, de modo a normalizar as aptidões dos indivíduos. Desta forma, os indivíduos com maior aptidão permanecem com maiores chances de sorteio, ao passo que os indivíduos com aptidão inferior têm suas chances

aumentadas. Este procedimento ajuda a manter a diversidade das populações e é conhecido como método de escalonamento da função de avaliação.

Na seleção por torneio, $n \geq 2$ indivíduos da população são sorteados aleatoriamente. Dentre estes n indivíduos, o com melhor aptidão é selecionado. Uma variação deste método considera, também, a geração aleatória de um número $x \in [0,1]$. Caso x seja menor que uma constante c_t previamente definida, o indivíduo com maior valor de aptidão é selecionado. Caso contrário, o menos apto é escolhido.

2.3.4 Cruzamento e mutação

A reprodução é a maneira pela qual dois novos cromossomos são gerados, a partir de dois cromossomos pais, obtidos durante a etapa de seleção. É por meio desta etapa que os AGs exploram regiões desconhecidas do espaço de busca a fim de analisar outras possíveis soluções do problema em questão. Os operadores mais utilizados durante a criação de novos indivíduos são os operadores de cruzamento e de mutação.

O operador de cruzamento executa uma troca de informações entre dois indivíduos, a partir da miscigenação do conteúdo de seus respectivos cromossomos. O cruzamento de dois indivíduos produz dois descendentes híbridos, os quais apresentam material genético dos dois progenitores. Para tal, o AG troca arbitrariamente trechos dos dois cromossomos selecionados para a reprodução, o que origina dois novos cromossomos.

Existem diferentes variações do operador de cruzamento, contudo a forma mais simples, definida na proposta original dos AGs, é o cruzamento em um ponto, no qual uma posição aleatória é escolhida e as cadeias parciais dos cromossomos definidos a partir desse ponto são utilizados na criação de novos indivíduos. De maneira análoga, o cruzamento multiponto envolve a divisão dos cromossomos pais em $k \geq 2$ seguimentos.

Uma forma mais extrema de recombinação é o cruzamento uniforme, no qual os cromossomos pais são analisados *bit* a *bit*. Caso os valores em determinada posição coincidam, ambos os descendentes recebem este valor. Caso estes valores sejam diferentes, uma determinada probabilidade p é utilizada para decidir se os bits serão trocados ou não.

Independentemente do método de cruzamento escolhido, o cruzamento em si é aplicado com uma dada probabilidade X_c a cada par de cromossomos selecionados. Assim, o cruzamento só é aplicado se o número gerado for menor que a taxa de cruzamento. Não ocorrendo o cruzamento, os filhos irão preservar as características dos pais.

Após a aplicação do operador de cruzamento, a nova geração de indivíduos passa pelo processo de mutação. O operador de mutação implica na modificação do valor de um ou mais genes de um indivíduo e visa restaurar o material genético perdido ou não explorado

em uma população, a fim de prevenir a convergência prematura do AG para soluções inadequadas.

Esta operação pode ser realizada em cada cromossomo da nova população. Dentre os métodos disponíveis para realização desta operação, o mais simples definido na proposta original dos AGs consiste em percorrer todos os *bits* dos cromossomos e, para cada um deles, gerar um valor $0 \leq m \leq 1$ aleatoriamente. Caso m seja menor que uma determinada probabilidade X_m, o valor do *bit* é invertido. Caso contrário, é mantido.

Os operadores de cruzamento e mutação desempenham papel fundamental para os AGs, uma vez que permitem uma boa avaliação do espaço de soluções possíveis. Segundo Miki et al. (1999), o desempenho de cada AG é dependente da escolha do operador de cruzamento e da taxa de aplicação do operador de mutação, e, além disso, a escolha desses operadores deve considerar o escopo do problema a ser solucionado e sua respectiva representação genética. Normalmente, a probabilidade de cruzamento dos indivíduos deve ser alta, próxima a 1, pois nesse operador reside o "poder" dos AGs, tal como expressado por Goldberg (1989). Em contrapartida, a probabilidade de mutação deve ser baixa, visto que além de aumentar a diversidade da população, este operador também pode destruir informações úteis contidas nos cromossomos. Apesar disso, a escolha destas probabilidades não é trivial, tal como será apresentado na Seção 2.5.

Como será descrito na seção seguinte, a definição de todos esses elementos tem impacto na qualidade do AG, logo se faz necessária uma escolha adequada dos elementos que compõem este algoritmo.

2.4 Pressão seletiva

O termo pressão seletiva designa a influência que o meio ambiente tem na seleção dos genes, uma vez que a variação das condições impostas pelo meio ambiente no qual os organismos estão inseridos fará com que alguns tenham maior ou menor probabilidade de sobreviveram para as próximas gerações (BÄCK, 1994). Esta influência representa um conjunto de características do meio ambiente imposta à população e que irá direcionar a evolução de determinadas características para se adaptarem a esse meio ambiente.

Nos algoritmos genéticos, pode-se classificar a pressão seletiva imposta à população por meio da função de avaliação, que é a única informação que o algoritmo utiliza para guiar o processo evolutivo. Quando os valores de aptidão dos indivíduos são próximos, a probabilidade de sobrevivência entre eles é similar, o que caracteriza uma pressão seletiva baixa. Por outro lado, quando os valores de aptidão são muito distintos, a probabilidade de os

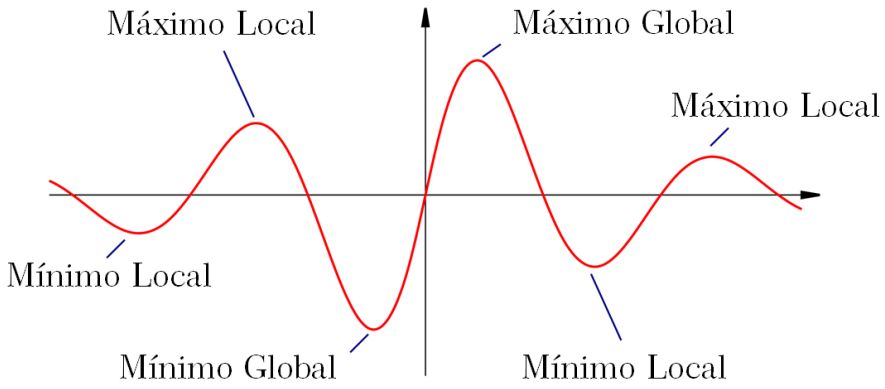
melhores indivíduos serem selecionados é muito superior àqueles com piores aptidões, o que caracteriza uma alta pressão seletiva (ALETI; MOSER, 2016).

A pressão seletiva tem relação direta com três características típicas dos AGs: tempo de convergência, direção da conversão e diversidade da população, tal como será apresentado nas três subseções seguintes. A fim de entender estas características, contudo, se faz necessária a apresentação dos conceitos de soluções locais e globais.

Para tal, será considerado um problema de otimização no qual é necessário minimizar, ou maximizar, uma função polinomial, a qual está representada na Figura 2. Este problema, ao ser analisado dentro de determinado domínio, possui pontos extremos, no qual a função atinge seus valores máximos ou mínimos. Tais pontos são considerados globais quando, para todo o restante do domínio, a função não assuma outro valor maior ou menor. Por outro lado, os pontos são considerados locais quando, em uma pequena vizinhança do ponto contida no domínio, a função não assuma outro valor maior ou menor. Como este exemplo assume um problema de otimização, então as soluções podem ser chamadas de pontos de máximo/mínimo locais ou globais.

No geral, podemos nos referir às possíveis soluções do problema como soluções ótimas locais ou globais.

Figura 2— Soluções locais e globais



Fonte: Elaborado pelo autor

2.4.1 Velocidade de convergência

O termo velocidade de convergência designa o tempo que o AG utiliza para tender a um resultado do problema em questão. Na presença de uma baixa pressão seletiva, o algoritmo pode levar muito tempo para convergir para uma solução, uma vez que o fato de os valores de aptidão dos indivíduos serem muito próximos, faz com que não existam indivíduos que se destaquem para guiar o processo evolutivo (PACHECO, 1999). Em

contrapartida, sob alta pressão seletiva, o algoritmo irá convergir rapidamente para uma solução, porém esta solução não necessariamente será um ponto ótimo global, mas possivelmente um ótimo local. Isso ocorre, pois os indivíduos mais aptos dominam as probabilidades de seleção e "contaminam" toda a população (PACHECO, 1999).

Logo, faz-se necessário estabelecer um mecanismo de pressão seletiva em relação à convergência, a fim de otimizar o tempo de execução do algoritmo, ao mesmo tempo que seja permitido analisar grande parte do espaço de soluções para evitar pontos de ótimo local.

Uma maneira de aferir a convergência do AG é por meio da diferença entre o melhor valor de avaliação e o valor de avaliação médio da população, tal como apresentado na Equação 4.

$$S_c = \begin{cases} \dfrac{f_{max} - f_{méd}}{f_{max} - f_{min}} & f_{max} > f_{min} \\ 1 & f_{max} = f_{méd} \end{cases} \tag{4}$$

2.4.2 Direção da evolução

De acordo com Linden (2008), direção da evolução é o termo que designa o caminho que o AG irá percorrer dentro do espaço de soluções possíveis. Durante a aplicação do AG, o cenário ideal seria iniciar a execução com uma população com representantes de diversos pontos do espaço de soluções para que, durante o processo evolutivo, o algoritmo possa seguir em direção à solução global. Contudo, a pressão seletiva também tem influência neste conceito, pois, caso seja muito baixa, a busca por soluções se comportará de forma quase aleatória, no qual nenhum indivíduo irá impor vantagem sobre o restante para que possa guiar o processo evolutivo. Caso seja muito alta, poucos indivíduos irão dominar a população, o que restringirá a direção da evolução a uma solução que poderá não ser a ótima global.

2.4.3 Diversidade populacional

A diversidade populacional, por sua vez, representa o quão distintas são as características dos indivíduos dentro de uma população (LINDEN, 2008). Quanto maior a diversidade populacional, maior será a área explorada dentro do universo de possíveis soluções do problema considerado.

Em problemas no qual há presença de baixa pressão seletiva, o ambiente permite que toda a população tenha boa probabilidade de sobreviver, o que, com a ajuda do operador de mutação, aumenta a diversidade populacional. Já em problemas com alta pressão seletiva, somente os indivíduos mais aptos serão beneficiados, o que resultará em queda da

diversidade populacional. Desta forma, quando a população se encontrar em uma região ótima local, a queda da diversidade irá dificultar o AG a sair desse ponto.

Por isso, pode-se dizer que a pressão seletiva atua em sentido oposto à diversidade populacional. Logo, um balanceamento adequado entre estas forças é benéfico para a obtenção de resultados satisfatórios (BÄCK, 1994).

Uma maneira de se aferir a diversidade populacional em um AG é por meio do conceito de espaçamento (SCHOTT, 1995). Formalmente, a diversidade populacional S_d é definida pela Equação 5, no qual Q representa o conjunto de soluções e f_i representa a aptidão da i-ésima solução.

$$S_d = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}\left(f_i - \left(\frac{\sum_{j=1}^{|Q|} f_j}{|Q|}\right)\right)^2} \tag{5}$$

2.4.4 Controle de pressão seletiva

Conforme visto, há então necessidade de se realizar um controle da pressão seletiva. Este controle pode ser feito de diferentes maneiras: é possível variar o método de seleção de indivíduos para reprodução, alterar as taxas dos operadores de cruzamento e mutação, escolher um tamanho de população adequado, entre outros. Contudo, tal como observado por Pacheco (1999) é importante observar que os AGs são muito sensíveis em relação a estas alterações, por exemplo:

a) o tamanho n da população afeta o desempenho global e a eficiência dos AGs. Uma população suficientemente grande fornece uma melhor cobertura do domínio do problema e previne a convergência prematura para ótimos locais. Contudo, populações com esta característica demandam um esforço computacional maior e, consequentemente, requerem um maior tempo de processamento. Logo, na definição do tamanho da população deve-se considerar tanto a diversidade populacional desejada, quanto o tempo de processamento tolerável, a fim de alcançar uma relação satisfatória;

b) quanto maior for a taxa de cruzamento, mais rapidamente novas soluções serão introduzidas na população. Porém, taxas muito altas podem gerar efeitos indesejáveis, pois a maior parte da população será substituída, a cada nova geração, o que implicará em perda de variedade genética e, consequentemente, convergência a uma população com indivíduos extremamente parecidos. Por outro lado, taxas de cruzamento baixas fazem

com que o algoritmo se torne muito lento, aumentando o tempo de convergência para uma solução aceitável;

c) a taxa de mutação contribui para a diversidade populacional e previne a estagnação do processo evolutivo. Porém, deve-se evitar o emprego de taxas de mutação muito altas, pois nestes casos a busca pelo ponto ótimo se tornará majoritariamente aleatória e prejudicará a convergência para uma solução ótima;

d) o cruzamento uniforme apresenta um poder de "destruição" maior que os operadores tradicionais de cruzamento, como o cruzamento de um e de dois pontos. Desta forma, sua utilização deve ser acompanhada de um critério de substituição de indivíduos altamente elitista, que garante a permanência dos melhores indivíduos.

Fica claro, então, que a escolha adequada dos operadores genéticos e de suas probabilidades, pode ajudar a encontrar o balanceamento entre a pressão seletiva e os conceitos descritos nas subseções anteriores. Contudo, esta escolha precisa ser feita de maneira cautelosa a fim de não prejudicar o desempenho dos AGs. Na Subseção 2.5 é apresentada uma definição detalhada de como esta escolha pode ser realizada.

2.5 Algoritmos genéticos adaptativos

Um algoritmo genético envolve, no mínimo, a definição de três parâmetros numéricos: a probabilidade de cruzamento p_c, a probabilidade de mutação p_m e o tamanho da população t_p. Além disso, é preciso realizar a escolha adequada do tipo de cruzamento e mutação. Muitos pesquisadores procuram descobrir qual o conjunto de operadores e seus parâmetros mais adequados para o problema no qual se pretende solucionar. Normalmente os parâmetros são analisados por experimentação e são apresentados sem justificação teórica.

Este tipo de ajuste, conhecido como refinamento, pode até ser efetivo para o problema exato no qual o AG está sendo aplicado, porém qualquer variação no problema ou nos dados envolvidos pode levar a uma perturbação da pressão seletiva, o que, tal como apresentado previamente, pode resultar na obtenção de soluções ótimas não globais. Além disso, tal como constatado por Linden (2008), "os algoritmos genéticos constituem um processo dinâmico, que evolui no tempo e no espaço das soluções", então a utilização de parâmetros definidos manualmente antes do início da execução do algoritmo pode não ser adequado para resolver um determinado problema em todos os seus estágios, uma vez que,

em cada estágio, a população apresenta diferentes características em relação ao espaço de soluções. Adicionalmente, o tempo gasto com experimentação de valores também é um problema, assim como afirmado por Gendreau, Hertz e Laporte (1994, apud MARQUES, 2011, p. 15): “Este algoritmo contém vários parâmetros (...). Diversos testes e um considerável esforço em refinamento foram realizados para chegar aos valores atuais”.

Neste cenário, as pesquisas na literatura científica, tanto para AG, quanto para outras meta-heurísticas, passaram a considerar a possibilidade de ajuste dos tipos de operadores a serem utilizados e seus parâmetros, no qual modificações são realizadas de acordo com o progresso do algoritmo (EIBEN; HINTERDING; MICHALEWICZ, 2007). Para tal, três técnicas de adaptação podem ser utilizadas (LINDEN, 2008), a saber:

a) determinística, no qual a mudança dos operadores e seus valores ocorre com base em uma regra determinística, sem que seja considerada nenhum tipo de informação a respeito do desempenho atual do algoritmo, como, por exemplo, considerar um aumento progressivo da probabilidade de mutação a cada nova geração;
b) adaptativa, no qual são consideradas informações a respeito do desempenho atual do algoritmo para determinar os operadores e seus valores na próxima geração, como, por exemplo, em relação ao operador de mutação - caso uma determinada porcentagem da mutação seja bem-sucedida, deve-se aumentar sua probabilidade na próxima geração, ou reduzi-la, caso contrário;
c) auto-adaptativa, no qual a própria estratégia de evolução é utilizada para alterar o valor dos parâmetros, como, por exemplo, codificar os parâmetros dentro dos próprios cromossomos.

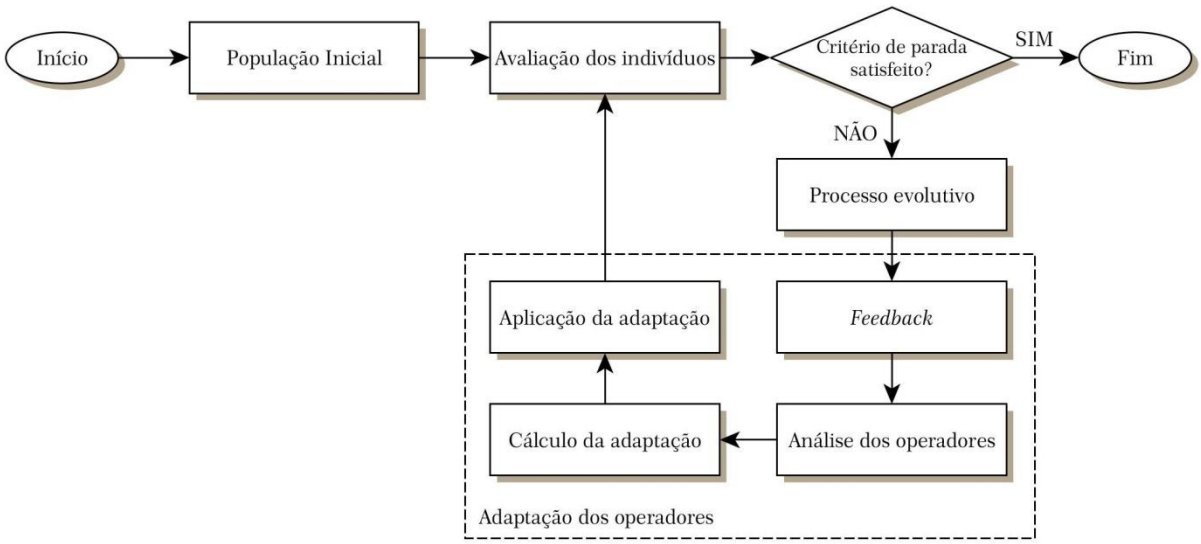
O ajuste de parâmetros dentro da computação remete ao início dos algoritmos genéticos. Rosenberg (1970) propôs adaptar probabilidades de cruzamento. Bageley (1967) considerou incorporar o controle de parâmetros à representação do indivíduo. Apesar deste início de auto-adaptação em AGs, abordagens efetivas só surgiram mais tarde. Schaffer e Morishima (1987) introduziram “cruzamento pontual auto-adaptativo” que ajustava o número de pontos de cruzamento. Poucos anos depois, Bäck (1992) propôs o primeiro método para adaptação da taxa de mutação em algoritmos genéticos. Já os trabalhos de Weinberg (1970) e Mercer & Sampson (1978) sugeriram a utilização de um AG adicional, que busca adaptar os parâmetros de controle, enquanto o AG principal resolve o problema original.

Já as pesquisas atuais, tal como analisado por Aleti e Moser (2016), indicam que a abordagem adaptativa é a que apresenta melhores resultados, uma vez que é capaz de captar

o impacto dos operadores durante a execução do AG e não necessita de um grande volume de processamento para ser calculada. Nos trabalhos desenvolvidos para esta abordagem, passou-se a considerar não somente a alteração dos valores de probabilidades dos operadores, mas também outras informações: a troca dos próprios operadores genéticos de cruzamento e mutação foi proposta nos trabalhos de Karafotias, Eiben e Hoogendoorn (2014) e Gomes, Antunes e Martins (2008), respectivamente; Groşan (2006) propôs uma alteração adaptativa da codificação das soluções; Bielza et al. (2013) e Montero e Riff (2011) propuseram alteração do tamanho da população e De Giovanni, Massi e Pezzella (2013) propuseram a utilização de uma função de avaliação adaptativa.

Na Figura 3 é ilustrado um modelo conceitual de um AG com características de adaptação. Nesta figura tem-se a etapa de *feedback* que se refere ao processo de obtenção de informações a respeito do desempenho atual do AG. A etapa de análise dos operadores visa identificar, dentre os operadores presentes, a contribuição de cada um para que o AG obtivesse o desempenho encontrado na etapa anterior. Já a etapa de cálculo da adaptação visa analisar o quanto e como os operadores serão modificados para, por fim, aplicar a modificação.

Figura 3— Modelo conceitual de um algoritmo genético adaptativo



Fonte: Adaptado de (ALETI; MOSER, 2016)

2.6 Computação paralela

A crescente popularidade da internet, associada a fatores como alta disponibilidade de computadores com grande capacidade de processamento e redes de alta velocidade, tem impulsionado o crescimento da quantidade de dados gerados diariamente nas mais diversas

atividades (GRAMA et al., 2003). Este fato tem mudado a maneira de se processar grandes volumes de informações, por meio da utilização de plataformas computacionais e modelos de programação que possibilitem processamento em tempo de execução viável (TROBEC; VAJTERŠIC; ZINTERHOF, 2009).

Neste cenário, o desenvolvimento e o uso de aplicações paralelas e distribuídas para o processamento de grandes volumes de dados têm crescido consideravelmente. Com o objetivo de maximizar a eficiência e a competitividade no mercado, grandes organizações têm investido em pesquisas voltadas à análise de dados. Já no meio científico, a análise de grandes conjuntos de dados, gerados a partir do avanço de novas tecnologias derivadas do avanço científico e tecnológico, tem sido suportada por meio de aplicações paralelas e distribuídas (TROBEC; VAJTERŠIC; ZINTERHOF, 2009).

A utilização de múltiplos processadores foi, então, a solução encontrada para suprir a necessidade de melhorar o desempenho dos sistemas, quando os limites físicos impostos pela tecnologia atual começaram a impedir o aumento de desempenho contínuos dos processadores (GRAMA et al, 2003; FOSTER, 1995). A computação paralela permitiu aos sistemas a execução de tarefas de forma concorrente, no qual cada processador pode executar uma tarefa. Neste tipo de arquitetura ocorre o compartilhamento de vários recursos, como por exemplo, a memória principal, o que contribui para um melhor desempenho (GRAMA et al., 2003).

Neste cenário, o paralelismo pode ser alcançado por meio de dois níveis: paralelismo de tarefas e paralelismo de dados. No paralelismo de tarefa, aplicações estão divididas em tarefas exclusivas e independentes que podem ser executadas simultaneamente em processadores diferentes. Por outro lado, no paralelismo de dados ocorre o particionamento de uma determinada quantidade de dados em partes menores, que serão processadas de maneira independente em diferentes processadores. Em cada uma dessas partições são operadas as mesmas tarefas (TROBEC; VAJTERŠIC; ZINTERHOF, 2009).

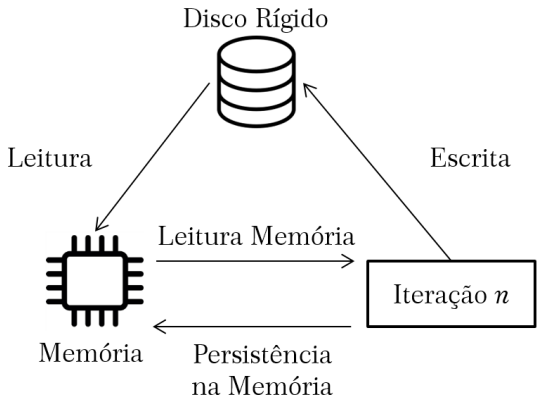
2.6.1 Computação em memória e Apache Spark

A computação em memória (*in-memory computing* – termo consolidado em inglês) refere-se à uma arquitetura capaz de carregar e armazenar dados na memória de acesso aleatório (RAM, do inglês *Random Acess Memory*), ao invés dos bancos de dados persistidos em disco. Desta forma, os requisitos de I/O (*input* e *output* – leitura e gravação de dados em disco rígido) e das transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade) dos bancos de dados são eliminados, o que acelera o acesso aos dados e, consequentemente, o processamento.

Um representante importante das tecnologias existentes que possibilitam o processamento em memória, é o Apache Spark, que é formado por um conjunto de sistemas para processamento paralelo de grandes quantidades de dados e que opera sob uma arquitetura baseada no paradigma mestre e escravo. O Apache Spark foi projetado com foco na velocidade de processamento e não necessita persistir dados em disco a cada iteração, tal como apresentado na Figura 4, no qual observa-se a possibilidade de que os dados sejam armazenados e lidos da memória RAM.

Isso é possível devido ao fato de a manipulação dos dados ocorrer por meio dos *Resilient Distributed Dataset* (RDD) (ZAHARIA et al., 2012), uma nova abstração de como os dados são armazenados em memória ou em disco. Internamente, o Apache Spark distribui os dados de um RDD em diferentes unidades de processamento a fim de atingir paralelismo (APACHE SPARK, 2019).

Figura 4— Estratégias de leitura e escrita no Apache Spark



Fonte: Elaborador pelo autor.

Um RDD é uma coleção de elementos particionados por meio dos nós do cluster e que podem ser operados em paralelo. Estas coleções contêm registros somente para leitura e apenas podem ser criadas por meio de operações de transformação nos dados armazenados ou em outros RDDs. Existem dois tipos de operações com RDDs: ações e transformações (APACHE SPARK, 2019). As transformações produzem um novo RDD a partir do RDD corrente e as ações são operações que, a partir de um RDD, retornam apenas um resultado para o programa principal ou persistem no disco.

Por definição, cada transformação em um RDD será executada toda vez que for executada uma ação, entretanto é possível persistir o RDD em memória ou em disco, o que permite manter os elementos no cluster para um acesso muito mais rápido na próxima consulta. Além disso, é possível persistir os RDDs no disco ou replicá-los em vários outros

nós o que é um dos maiores diferenciais em relação ao modelo MapReduce (APACHE SPARK, 2019).

2.7 Algoritmos genéticos paralelos

Tal como apresentado na Subseção 2.6, é possível que problemas complexos sejam solucionados de maneira mais rápida por meio da computação paralela, do que se estivessem sendo solucionados sequencialmente. No caso dos AGs, a utilização de paralelismo pode ser justificada por três fatores:

a) a função que calcula a aptidão as possíveis soluções do problema em questão, assim como os operadores de cruzamento e mutação, podem exigir um grande esforço computacional;
b) em alguns problemas a população de possíveis soluções precisa ser muito grande, o que requer muito processamento para avaliação de cada indivíduo;
c) o algoritmo pode ficar preso em pontos de solução local devido à falta de diversidade das populações.

Neste cenário, AGs paralelos têm sido investigados para lidar com esses problemas. De acordo com Linden (2008), podem ser citados três principais abordagens: padrão, celular e distribuído.

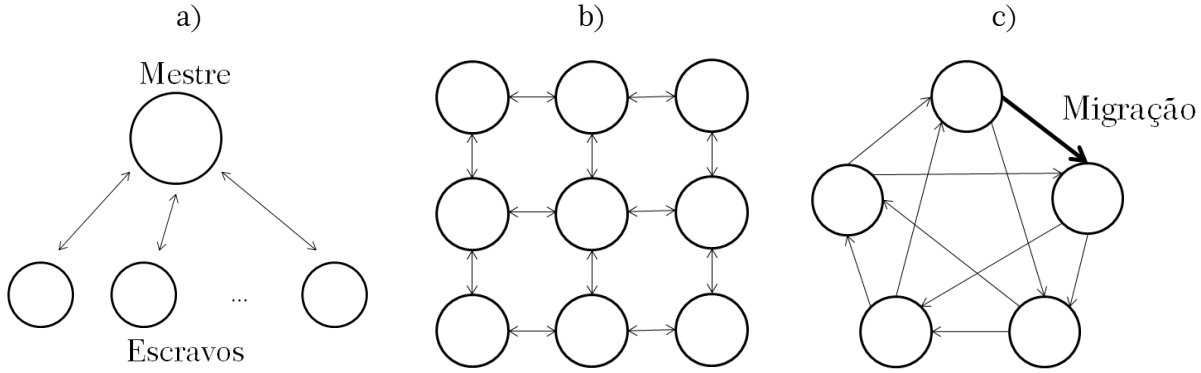
Nos AGs paralelos padrão, representados na Figura 5a, existe uma única população, tal como para os AGs tradicionais. Contudo, a função de avaliação é calculada em paralelo por diversos processadores. Nesta abordagem, que também é conhecida como mestre-escravo, o mestre armazena toda a população, realiza todos os operadores genéticos que, consequentemente, não são paralelizados, e envia os indivíduos para os processadores escravos calcularem a aptidão (LINDEN, 2008).

Nos AGs celulares, por sua vez, também existe uma única população. Cada indivíduo pertencente a esta população é colocado em uma estrutura espacial que limita a interação com outros indivíduos, tal como representado na Figura 5b, no qual apenas é possível competir ou sofrer cruzamento apenas com indivíduos vizinhos. Nesta abordagem, a população é dividida entre os processadores, que podem ser responsáveis por um ou mais indivíduos (LINDEN, 2008).

Já nos AGs distribuídos, múltiplas populações são consideradas. Cada população é localizada em um processador diferente, no qual é considerado um processo evolutivo independente. A fim de promover cooperação entre os processadores, os quais receberam o nome de ilhas, é criado um novo operador, chamado de migração. Por meio deste operador,

indivíduos podem migrar de uma ilha para outra (LINDEN, 2008). Na Figura 5c é apresentada esta abordagem.

Figura 5— Algoritmos genéticos paralelos: a) padrão, b) celular e c) distribuído



Fonte: Elaborado pelo autor

Dentre estas abordagens, a distribuída tem demonstrado maior eficácia na resolução de diversos problemas. Um dos motivos é o fato de que cada ilha inicia um processo de busca isolado a partir de sua própria população inicial e, durante algumas gerações, resulta na exploração de alguma parte do espaço de busca. Com as migrações, informações sobre diferentes regiões do espaço de busca são trocadas entre as ilhas, o que promove maior diversidade das populações.

2.8 Revisão bibliográfica

A fim de possibilitar uma análise do estado atual de pesquisas relacionadas aos conceitos apresentados, nas subseções seguintes é apresentada a revisão bibliográfica da literatura. A análise foi conduzida individualmente para cada conceito, devido à multidisciplinaridade do problema.

2.8.1 Seleção de SNPs representativos

Estudos recentes para seleção de SNPs representativos estão divididos em quatro grupos, com base na abordagem utilizada: diferença haplotípica, associação fenotípica, capacidade de predição do conjunto de SNPs original e coeficiente de associação entre os SNPs (MOUAWAD; MANSOUR, 2014).

Os métodos que se baseiam na diferença haplotípica partem do princípio que o genoma humano é particionado em blocos com base nas medidas de desequilíbrio de ligação entres os SNPs, tal como descrito na Subseção 2.4.1. Trabalhos como os de Mahdevar

(2010) e Phuong, Lin e Altman (2005) têm como objetivo encontrar um subconjunto de SNPs capaz de representar os demais SNPs de um bloco. Estas abordagens, porém, são limitadas, uma vez que não há consenso a respeito de como os blocos devem ser definidos (o que, por si só, é uma área de pesquisa) e são desconsideradas as relações inter-blocos.

Abordagens com base na associação fenotípica consideram que há uma informação fenotípica presente e tentam determinar o conjunto de SNPs que podem separar indivíduos doentes dos saudáveis em relação à determinada doença. Neste caso, técnicas tradicionais de seleção de atributos são utilizadas considerando-se um problema de classificação. Esta abordagem também é limitada, visto que a obrigatoriedade da presença da informação fenotípica restringe sua aplicação aos estudos de associação entre doenças.

Os métodos baseados em predição consideram a seleção de SNPs apenas como um problema de reconstrução dos dados originais a partir do conjunto de tag SNPs, assim, esses métodos tem como objetivo selecionar um subconjunto de tag SNPs que seja capaz de predizer os demais com o menor erro (MOUAWAD; MANSOUR, 2014). Para tal, esses métodos fazem uso de estratégias de predição para guiar o processo de avaliação dos subconjuntos de SNPs, o que pode acarretar problemas de desempenho e, também, qualidade.

Por fim, o valor de desequilíbrio de ligação é utilizado como coeficiente de associação entre SNPs para encontrar tag SNPs que sejam capazes de representar o conjunto de dados originais. Tais métodos consideram que todos os SNPs em um haplótipo são altamente relacionados a pelo menos um SNP do conjunto de tag SNPs. Esta relação é denominada como cobertura de ligação, no qual quanto maior seu valor, mais representativo é o conjunto de tag SNPs (LODISH et al., 2014).

Os primeiros métodos de seleção de tag SNPs visavam encontrar um subconjunto ótimo que fosse capaz de capturar a diversidade de haplótipos dos dados originais. Neste sentido, diversas métricas de avaliação da diversidade de haplótipos foram propostas na literatura.

Ke e Cardon (2003) utilizaram a quantidade de haplótipos exclusivamente distinguíveis em um subconjunto candidato V' como métrica de avaliação da diversidade capturada por V'.

Johnson et al. (2001) definiram a diversidade de haplótipos em relação à quantidade de alelos diferentes entre os haplótipos de um mesmo grupo, ao qual se deu o nome de diversidade residual. Logo, se um subconjunto candidato V' agrupou corretamente todos os haplótipos, sua diversidade residual é 0, senão é maior que 0. Já os trabalhos desenvolvidos por Ackerman et al. (2003), Avi-Itzhak, Su e De La Vega (2003) e Hampe, Schreiber e

Krawczak (2003), por exemplo, utilizam como métrica o conceito de entropia, no qual considera-se a frequência relativa de um haplótipo em relação aos demais.

Ainda que as abordagens propostas inicialmente, tal como as citadas anteriormente, tenham sido úteis e relevantes às análises de SNPs, a evolução das ferramentas de processamento de sequência genéticas fez com que suas aplicações se tornassem inviáveis. Tais abordagens examinavam exaustivamente todos os subconjuntos possíveis formados a partir do conjunto de SNPs original, o que impõe uma limitação em relação à quantidade de SNPs analisadas. Este problema, tal como feito em outros contextos que envolvem busca, foi solucionado por meio de abordagens que utilizam métodos heurísticas e métodos de busca eficientes, como algoritmo guloso, algoritmo genético, entre outros. Adicionalmente, outros trabalhos propuseram a utilização de técnicas de otimização para melhorar os resultados do processo.

Mouawad e Mansour (2014) propuseram uma representação do conjunto de SNPs original em uma estrutura de grafos, no qual a seleção de tag SNPs foi realizada por meio de três variações de algoritmos genéticos. Cada variação corresponde a uma métrica diferente de avaliação do coeficiente de associação entre SNPs: maximização do desequilíbrio de ligação entre pares de SNPs, maximização do desequilíbrio de ligação entre múltiplos SNPs e maximização do poder de predição entre pares de SNPs. Os autores concluíram que avaliar o conjunto de SNPs levando em consideração a maximização do desequilíbrio de ligação entre SNPs aumenta o poder de predição.

Adicionalmente, esta representação em grafos permite a utilização de outras abordagens de busca para percorrer o conjunto de SNPs, tal como o algoritmo de seleção de atributos proposto por Tabakhi, Moradi e Akhlaghian (2014) que utiliza a otimização por colônia de formigas para percorrer o espaço de busca e selecionar os atributos mais relevantes (no caso, os SNPs representativos).

Prathibha e Chandran (2016) desenvolveram um algoritmo de seleção de tag SNPs com base na capacidade de predição do conjunto original de SNPs. Por meio da utilização dos conceitos de algoritmo genético e de otimização por colônia de abelhas, foi possível desenvolver um mecanismo de agrupamento das SNPs eficiente, capaz de reduzir em 80% a quantidade de tag SNPs selecionadas. A otimização por colônia de abelhas foi utilizada na execução do agrupamento em si, enquanto que o algoritmo genético foi utilizado para otimizar os parâmetros de entrada do método.

Liao et al. (2012) e Li et al. (2014) deram foco à etapa inicial de agrupamento do conjunto de SNPs nos blocos definidos pela estrutura em bloco do genoma humano; para encontrar as tag SNPs em cada bloco, fizeram uso de algoritmos inspirados pelo

comportamento de formigas em suas colônias. Liao et al. (2012) desenvolveram uma implementação recursiva do algoritmo de colônia de formigas, que foi capaz de diminuir o tempo de processamento para obtenção do conjunto final de tag SNPs. Li et al. (2014) utilizaram o algoritmo clássico de colônia de formigas, porém deram foco à criação de uma função objetivo capaz de selecionar o melhor conjunto final de tag SNPs, com base nos tag SNPs encontrados em cada bloco.

Chuang, Huang e Yang (2012) utilizaram estratégia de otimização por enxame de partículas para guiar o processo de busca e encontrar tag SNPs altamente correlacionadas de acordo com o valor de desequilíbrio de ligação. O trabalho fez, ainda, uso da teoria do caos para otimizar os parâmetros de entrada do algoritmo e proporcionar melhores resultados.

Li et al. (2015) fizeram uso de rede neural artificial, otimização por colônia de formigas e algoritmo guloso para selecionar um conjunto de SNPs representativos: a otimização por colônia de formigas é utilizada, inicialmente, para remover redundância nos dados de SNPs; a rede neural artificial é utilizada na construção de um modelo de classificação dos SNPs, a fim de encontrar blocos com base na medida de desequilíbrio de ligação. O algoritmo guloso foi utilizado para guiar o processo de busca. O trabalho apresentou redução significativa de tempo de processamento quando comparado a outras abordagens similares da literatura.

Outros trabalhos presentes na literatura científica fazem uso das técnicas de otimização, porém aplicadas aos parâmetros de entrada dos algoritmos utilizados. Este procedimento é realizado devido ao fato de os algoritmos heurísticos serem muito sensíveis em relação aos parâmetros iniciais configurados pelo usuário, no qual a solução encontrada pode não ser adequada caso os parâmetros iniciais não sejam configurados adequadamente. Neste contexto, destaca-se o trabalho de Ilhan e Tezel (2013), que utilizaram um algoritmo de otimização por enxame de partículas para otimizar os parâmetros de entrada de um algoritmo de *Support Vector Machine* (SVM), antes da aplicação à seleção de SNPs.

Na literatura científica foram encontrados poucos trabalhos que fazem uso do processamento paralelo em problemas de seleção de SNPs representativos. Hung et al. (2015) e Chen et al. (2014) propuseram a utilização do processamento distribuído nas etapas de agrupamento dos SNPs em blocos. Apesar dos três trabalhos apresentarem resultados satisfatórios (Chen et al. relataram um tempo de processamento quase dez vezes menor quando comparado a outras abordagens não distribuídas, por exemplo), eles fazem uso da busca exaustiva para analisar o conjunto de SNPs.

Não foram encontrados trabalhos que propuseram a paralelização de algoritmos de seleção de SNPs representativos baseados em métodos heurísticos de busca, contudo muitos

destes métodos heurísticos já possuem versões distribuídas aplicadas aos mais variados contextos (GONG et al., 2015; ESLAMI; VAHIDI; ASKARZADEH, 2014; ILIE; BĂDICĂ, 2013; DI MARIO; MARTINOLI, 2014; KARABOGA; ASLAN, 2015).

2.8.2 Algoritmos genéticos

Algoritmos genéticos são uma técnica de otimização bem consolidada na literatura científica. De acordo com o portal ScienceDirect[1], apenas no primeiro semestre de 2019 foram publicados mais de 1600 artigos científicos que fizeram uso de AGs como estratégia de resolução de problemas nas mais diversas áreas, como engenharias, agronomia, meio ambiente, ciência da computação, bioinformática, entre outras. Essa quantidade expressiva de trabalhos científicos enfatiza a importância e efetividade destes algoritmos.

Poucos trabalhos desenvolvidos nos últimos anos levam em consideração o controle de pressão seletiva como estratégia para melhorar os resultados obtidos pelo processo evolutivo, apesar das pesquisas nesta área terem evoluído consideravelmente e mostrarem resultados satisfatórios (ALETI; MOSER, 2016). Grande parte dos trabalhos presentes na literatura faz uso da estratégia de refinamento, a fim de definir, previamente, os valores dos parâmetros e métodos internos a serem utilizados. Apesar disso, não se pode considerar que estes trabalhos conduziram uma avaliação adequada destes parâmetros, uma vez que a maioria apenas apresenta as escolhas realizadas, sem grande preocupação em discuti-las.

Contudo, aqueles que fazem uso de estratégias de controle de pressão seletiva, apresentam resultados superiores em relação aos demais. Por exemplo, Wang et al. (2014) relataram que o AG com controle de pressão seletiva desenvolvido, aplicado ao problema de segmentação de imagens, apresentou convergência mais rápida quando comparada aos métodos tradicionais, assim como melhor qualidade dos resultados obtidos.

Em relação aos AG paralelos, muitos trabalhos foram propostos. Inicialmente, estes trabalhos focavam-se em explorar a capacidade adaptativa dos algoritmos por meio de estratégias de paralelização diferentes, como paralelização das funções de avaliação, dos operadores genéticos, entre outros. Em relação aos *frameworks* específicos de programação paralela, o MapReduce foi muito utilizado na paralelização de AGs, como feito por Mckenna et al. (2010) para análise de sequencias de DNA e por Triguero et al. (2015) para redução de protótipos em problemas de classificação.

Mais recentemente, propôs-se a paralelização dos AGs por meio da utilização do *framework* Apache Spark, que se mostrou mais eficiente, principalmente em problemas que

[1] <https://www.sciencedirect.com> - Busca realizada em 19 de julho de 2019, com o termo “genetic algorithm” presente no título, resumo ou palavras-chave dos artigos publicados a partir de 2019.

envolviam grandes espaços de busca: Qi, Wang e Li (2016) propuseram uma paralelização de AG para resolução do problema de geração do conjunto mínimo de teste, em testes *pairwise* de *software*; Hu et al. (2017) aplicaram um AG paralelo com base em Spark a um problema de distribuição de sensores no espaço; Gaifang et al. (2017) propuseram a paralelização de um algoritmo híbrido, que utiliza recursos dos AGs e da otimização por colônia de formigas.

No contexto do problema de seleção de SNPs representativos, o primeiro trabalho a utilizar AG foi proposto por Mahdevar et al. (2010). Este trabalho teve como objetivo maximizar a acurácia da predição do conjunto de SNPs em relação aos tag SNPs, logo se refere aos métodos que se baseiam na abordagem de maximização da capacidade de predição. Já Mouawad e Mansour (2014) realizaram a seleção de tag SNPs por meio de um AG convencional com base na abordagem de coeficiente de associação, que foi executado em três funções de avaliação diferentes. Técnicas de controle de pressão seletiva e de processamento paralelo não foram consideradas em nenhum destes trabalhos.

2.8.3 Computação paralela e em memória

O processamento paralelo e em memória tem sido amplamente utilizado na literatura científica para lidar com problemas que requerem grande capacidade de processamento, a fim de torná-los viáveis. Nos contextos atuais, com o crescimento da quantidade de dados produzidos nas mais diferentes áreas, sua aplicação tem se tornado ainda mais utilizada, uma vez que o processamento de grandes volumes de dados implica, muitas vezes, em ineficiência do algoritmo utilizado (TROBEC; VAJTERŠIC; ZINTERHOF, 2009).

Ao considerar este recurso, os diversos algoritmos e problemas clássicos da literatura foram adaptados para suportar paralelização por meio dos paradigmas MapReduce e Spark. Liu, Xu e Li (2017) propuseram duas estratégias de paralelizações das redes neurais artificiais tradicionais por meio do MapReduce e Spark, a fim de lidar com tarefas complexas de classificação, e Maillo et al. (2017) propuseram uma paralelização iterativa para o classificador *k-Nearest-Neighbors* por meio do Spark. Bhuiyan e Al Hasan (2015) utilizaram uma abordagem iterativa do MapReduce para lidar com o problema de mineração de subgrafos mínimos frequentes, e Sethi e Ramesh (2017) lidaram com o problema de mineração de *itemsets* frequentes das regras de associação por meio do Spark.

Observa-se, ainda, uma crescente utilização do MapReduce e Spark na área de bioinformática, uma vez que esta área envolve a construção de modelos matemáticos que comumente requerem a utilização de uma grande quantidade de recursos computacionais para executar experimentos em larga escala, tal como observado por Guo (2012). Em

relação ao problema de seleção de tag SNPs da bioinformática, pode-se citar o trabalho de Hung et al. (2015), que propõe uma abordagem paralela com base na estrutura em blocos do genoma humana, no qual as tarefas de geração dos blocos haplotípicos é realizada de forma paralela por meio do MapReduce para, posteriormente, permitir a busca por tag SNPs.

2.9 Considerações finais

Nesta seção foram apresentados os principais conceitos relacionados ao problema de selação de SNPs representativos e uma análise do estado da arte. Percebe-se que, com a evolução da tecnologia e com o crescimento do volume de dados de SNPs, a utilização de métodos exaustivos para análise de todas as soluções possíveis em busca de uma solução ótima foi substituída pela aplicação de métodos heurísticos, capazes de encontrar soluções satisfatórias, com menor tempo de processamento, tal como sintetizado na Tabela 1. Apesar dos avanços, percebe-se também uma área não muito explorada, relacionada ao processamento paralelo e em memória, neste contexto. Os trabalhos encontrados fazem uso apenas da busca exaustiva e não abordam melhorias de desempenho no processo de seleção de SNPs representativos, tampouco de melhorias relacionadas à utilização de estratégias de adaptação, logo ainda é possível obter melhorias nesta área.

Tabela 1— Principais características das implementações encontradas na literatura

Proposta	Estratégia	Método Busca	Controle de Pressão Seletiva	Proc. Paralelo	Proc. em Memória
Mahdevar et al. (2010)	Predição	AG	Não	Não	Não
Mouawad e Mansour (2014)	Coeficiente de Associação	AG	Não	Não	Não
Prathibha e Chandran (2016)	Predição	ACO	Parâmetros	Não	Não
Ilhan e Tezel (2013)	Predição	AG	Parâmetros	Não	Não
Li et al. (2014)	Bloco Haplotípico	ACO	Não	Não	Não
Chuang, Huang e Yang (2012)	Coeficiente de Associação	PSO	Parâmetros	Não	Não
Hung et al. (2015)	Bloco Haplotípico	Exaustivo	Não	MR	Não

Fonte: Elaborada pelo autor

3 Algoritmo genético adaptativo e paralelo para seleção de tag SNPs

Após a introdução dos conceitos relacionados à seleção de SNP representativos, assim como dos principais trabalhos desenvolvidos nesta área, é apresentado, nesta seção, o algoritmo genético adaptativo e paralelo desenvolvido para o problema de seleção de tag SNPs. Inicialmente o problema é definido formalmente. Na sequência, a configuração do algoritmo genético desenvolvido é apresentada, no qual está incluída a estratégia de adaptação utilizada. Por fim, a estratégia de paralelização é definida.

3.1 Formulação do problema

Seja $L = \{SNP_1, SNP_2, \dots, SNP_n\}$ um conjunto de n SNPs ($n = |L|$), para o qual deseja-se encontrar um subconjunto $L' \subset L$ de k SNPs que contenha a informação dos alelos de todos os $SNP_i \in L$.

Seja $G = \langle V, E \rangle$ um grafo completo valorado não-dirigido, no qual $V = \{v_1, v_2, \dots, v_n\}$ é um conjunto não vazio de objetos denominados vértices, $V = L$, logo $\forall SNP_i \in L: \exists\, v_i \in V$, e $E = \{(v_i, v_j), \forall v_i, v_j \in V\}$ é um conjunto de pares não ordenados de V, denominados arestas. Seja, ainda, $w(v_i, v_j)$ o peso da aresta composta pelos vértices $(v_i, v_j) \in E$, que representa o valor de desequilíbrio de ligação entre os SNPs.

Então, o problema de seleção de SNPs representativos consiste em determinar um subconjunto L', no qual $|L'| \ll |L|$ e a soma de $w(u, v)$ para todo $u, v \in L'$ seja máximo, ou seja, um subconjunto menor de SNPs com alta cobertura de ligação.

3.2 Codificação das soluções

A codificação das soluções para o problema de representação de SNPs representativos se dá por meio de vetores binários.

Seja $|L|$ a quantidade inicial de SNPs. Uma possível solução para o problema é, então, representada por $C = (c_1, c_2, \dots, c_{|L|})$. Cada $c_i \in \{0,1 \; \forall i \mid 1 \leq i \leq |L|\}$, no qual $c_i = 0$ indica que o SNP na posição i não faz parte do conjunto de tag SNPs, ou seja, $SNP_i \notin L'$, ao passo que um valor igual a 1 indica a presença do SNP no conjunto de tag SNPs.

3.3 Seleção

Tal como apresentado na Subseção 2.4, é preciso controlar a pressão seletiva durante a execução do AG, uma vez que é preciso direcionar a solução a um ponto ótimo global. Neste cenário, a efetividade do operador de seleção está diretamente relacionada à diversidade mantida dentro da população.

Sabe-se que um AG consiste, basicamente, na alternância entre exploração do espaço de soluções e a busca do ponto ótimo de uma região explorada anteriormente (GIGER; KELLER; ERMANNI, 2007). Desta forma, o algoritmo desenvolvido, primeiramente, observa a situação atual da população para então decidir se as novas populações devem ser criadas no sentido de i) explorar novas soluções ou ii) buscar a melhor solução em uma área promissora recentemente descoberta. Este controle é feito por meio da escolha entre os dois métodos de seleção: proporcional e torneio, tal como descritos na Subseção 2.3.3.

Para que seja possível decidir qual o método de seleção a ser utilizado em cada geração, a avaliação da população atual se dá por meio da verificação da distribuição dos indivíduos com base em seus respectivos valores de aptidão. Para tal, é utilizada a Equação 5.

A partir desta análise da população atual, é possível decidir o método de seleção a ser utilizado. Seja o valor de diversidade da população S_d:

a) se S_d for alto, a população apresenta características de exploração, no qual os indivíduos estão mais distribuídos em relação aos seus valores de aptidão. Desta forma, na próxima iteração do algoritmo será utilizado o método de seleção proporcional, pois, como o espaço de solução é bem distribuído, as melhores soluções têm grande chance de representar a solução ótima global;
b) Se S_d for baixo, a população está estagnada no espaço de soluções, logo, na próxima iteração será utilizado o método de seleção por torneio que é mais efetivo para manter a diversidade das populações, uma vez que proporciona

boas chances de reprodução também para os indivíduos com menores valores de aptidão.

Esta escolha do método de seleção é realizada com base na métrica de diversidade, uma vez que há diferença na configuração da população no início do algoritmo e no decorrer da execução. No início, a discrepância entre a aptidão dos indivíduos pode causar perda de diversidade caso o método de seleção não dê as chances adequadas aos indivíduos menos aptos. Já quando o processamento se estende por um período de tempo grande, os indivíduos tendem a ter níveis de aptidão muito próximos um dos outros, o que faz com que os melhores indivíduos não se destaquem para reproduzir e, consequentemente, diminui a velocidade de convergência.

3.4 Operadores de cruzamento e mutação

O AG para seleção de tag SNPs implementa o operador de cruzamento em dois pontos. Sejam dois cromossomos pais C_i e C_j que serão recombinados para gerar dois novos cromossomos filhos C_{f1} e C_{f2}. Seja ainda X_c um valor definido como taxa de cruzamento.

Primeiramente, um valor i é gerado aleatoriamente no intervalo $[0,1]$. Caso $i < X_c$, o operador de cruzamento não é aplicado. Caso $i \geq X_c$, dois valores r_1 e r_2 são gerados aleatoriamente no intervalo $[0, |L|]$, no qual $r_1, r_2 \in \mathbb{Z}$ e $r_1 \neq r_2$. Então, os genes dos cromossomos pais no intervalo $[r_1, r_2]$ são trocados.

Por exemplo, seja $C_i = 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1$, $C_j = 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0$, $r_1 = 2$ e $r_2 = 4$, então os cromossomos filhos serão $C_{f1} = 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1$ e $C_{f2} = 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0$.

Adicionalmente, o AG proposto implementa ainda a técnica de elitismo, de modo que o indivíduo com melhor valor de aptidão é sempre reinserido na nova população, independentemente de ter sido selecionado ou não para reproduzir.

O operador de cruzamento é realizado continuamente até que a nova população P atinja a quantidade de indivíduos previamente definida.

Após a criação da nova população, a operação de mutação é então realizada em todos os indivíduos da nova população P. Seja C um indivíduo da população atual e X_m um valor definido como taxa de mutação.

Para cada gene $c \in C$, um valor i é gerado aleatoriamente no intervalo $[0,1]$. Caso $i < X_m$, o operador de mutação não é aplicado. Caso $i \geq X_m$, o valor do gene é alterado de 0 para 1 ou de 1 para 0.

As taxas de cruzamento e mutação, tal como visto na Subseção 5.1.4, influenciam na direção da busca no espaço de soluções e, consequentemente, afetam o desempenho do AG.

Logo, é interessante que o algoritmo utilize mecanismos para adaptação destas taxas, de modo a possibilitar a criação de melhores indivíduos a cada geração.

Para realizar esta adaptação, o AG desenvolvido implementa o conceito de produtividade, proposto inicialmente por Lin, Lee e Hong (2003) e utilizado como base em outras abordagens futuras.

Ao final de cada geração, é calculado um valor de produtividade média Φ para cada um dos operadores de cruzamento e mutação, o que é feito por meio da Equação 6, no qual n_a representa a quantidade de aplicações do operador em questão e φ_i representa a produtividade do operador em cada uma das n_a aplicações.

$$\Phi = \frac{1}{n_a}\sum_{i=1}^{n_a}\varphi_i \tag{6}$$

Para o operador de cruzamento, a produtividade φ^c é calculada como a diferença entre a soma dos valores de avaliação dos indivíduos pai e filhos (Equação 7), no qual C_{pai} representa os indivíduos pai, C_{filho} representa os indivíduos filhos e $fit()$ representa os respectivos valores de avaliação.

$$\varphi^c = fit\left(C_{pai1}\right) + fit(C_{pai2}) - fit\left(C_{filho1}\right) + fit(C_{filho2}) \tag{7}$$

Para o operador de mutação, a produtividade φ^m é calculada como a diferença entre os valores de avaliação do indivíduo depois e antes da aplicação do operador (Equação 8).

$$\varphi^m = fit(C_{novo}) - fit(C_{antigo}) \tag{8}$$

Após o cálculo da produtividade média de cada operador, as taxas de cada um deles são atualizadas. O operador que cruzamento recebe uma taxa X_c de acordo com a Equação 9 e o operador de mutação recebe uma taxa X_m de acordo com a Equação 10.

$$X_c = \begin{cases} X_c + \theta & se\ \Phi^c \geq \Phi^m \\ X_c - \theta & se\ \Phi^c < \Phi^m \end{cases} \tag{9}$$

$$X_m = \begin{cases} X_m + \theta & se\ \Phi^c < \Phi^m \\ X_m - \theta & se\ \Phi^c \geq \Phi^m \end{cases} \tag{10}$$

Nas equações, θ representa o tamanho do ajuste a ser aplicado nos operadores e é definido de maneira adaptativa de acordo com a convergência da população atual (Equação 11), no qual S_c é o valor de convergência definido na Subseção 2.4.1 pela Equação 4. Uma vez que o valor de convergência S_c é definido no intervalo (0,1] e que as probabilidades X_c e X_m são definidas no intervalo (0,1), o tamanho do ajuste precisa ser normalizado. Lin, Lee e Hong (2003) mostraram que, ao normaliza-lo no intervalo (0, 0,01], ocorre melhoria no valor de aptidão geral das soluções quando comparado à outras abordagens.

$$\theta = 0{,}01.S_c \tag{11}$$

Desta forma, quando a população converge, o tamanho do ajuste a ser aplicado nos operadores é maior, a fim de reduzir a probabilidade de o AG ficar parado em um ponto de solução ótima local. Apesar disso, a convergência não é prejudicada, pois a característica elitista faz com que não haja perda de boas soluções já encontradas.

As taxas iniciais de cruzamento e mutação são $X_c = X_m = 0{,}5$, de acordo com instrução dos autores (LIN; LEE; HONG, 2003).

3.5 Função de avaliação

A função de avaliação utilizada no AG proposto atua de forma a maximizar a cobertura de ligação entre os tag SNPs selecionados. Assim, a aptidão de determinado cromossomo C é definida pela função $fit(C)$ (Equação 12).

$$fit(C) = LC(L', L - L') - \psi \tag{12}$$

no qual L' é o conjunto de tag SNPs representado pelo indivíduo, $(L - L')$ é o conjunto de SNPs não selecionados como tag SNPs e LC é a função de mensuração da cobertura de ligação. O valor de ψ é explicado na Subseção 3.5.1.

A função LC é definida como

$$LC(T,S) = \frac{1}{|S|}. \sum_{SNP_j \in S} \max_{SNP_i \in T} LD\left(SNP_i, SNP_j\right) \tag{13}$$

no qual LD é a função de mensuração do desequilíbrio de ligação entre os pares de SNPs com base na métrica r^2 (DEVLIN; RISCH, 1995).

A função LD é definida como

$$LD\left(SNP_i, SNP_j\right) = LD(i,j) = r^2 = \frac{(P_{AB} - P_A P_B)^2}{P_A P_a P_B P_b} \tag{14}$$

no qual A/a são os dois alelos possíveis para o SNP_i, B/b são os dois alelos possíveis para o SNP_j, P_A e P_a são as frequências dos alelos A e a para o SNP_i, P_B e P_b são as frequências dos alelos B e b para o SNP_j e P_{AB}, P_{Ab}, P_{aB}, P_{ab} são as frequências das combinações dos alelos entre os SNPs.

Quando $LD(i,j) = 1$, os SNPs i e j estão em total desequilíbrio de ligação. Neste caso, observar os alelos de um SNP permite encontrar informações completas de todos os alelos do outro SNP.

O AG aplicado ao problema de seleção de tag SNPs tem o objetivo de maximizar a função $fit(C)$.

3.5.1 Problema de minimização da quantidade de SNPs selecionados

É importante ressaltar que a maximização da função de avaliação não garante que seja escolhido o menor subconjunto de tag SNPs possível. Em experimentos preliminares conduzidos neste trabalho e, ainda, de acordo com os resultados apontados por outros trabalhos de seleção de tag SNPs, mostrou-se que o valor da cobertura de ligação entre os indivíduos selecionados aumenta conforme a quantidade de tag SNPs selecionadas também aumenta. Logo, o problema precisa lidar com dois objetivos conflitantes.

Muitas técnicas de otimização multi-objetivo têm sido propostas na literatura, inclusive por meio da utilização de AGs. Contudo, para aplicação destas técnicas é necessário conhecimento prévio a respeito do conjunto de SNPs para que seja considerado um valor de ponderação entre os dois objetivos. Os problemas de seleção de tag SNPs comumente recorrem a uma parametrização da quantidade de SNPs a serem selecionados, no qual o algoritmo busca um subconjunto de tamanho igual ao informado pelo usuário.

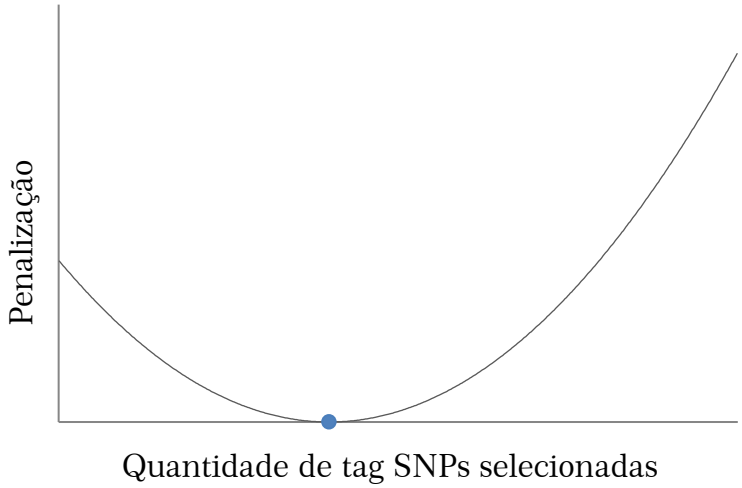
Mouawad e Mansour (2014), que também utilizam um AG para resolução do problema, seguem esta estratégia: após a aplicação dos operadores de cruzamento e mutação, cada novo indivíduo é avaliado e, caso possua uma quantidade de tag SNPs diferente da informada pelo usuário, novos SNPs são incluídos ou removidos do indivíduo aleatoriamente, até que a quantidade seja satisfeita. Porém, a utilização desta estratégia nos AGs fere princípios relacionados ao funcionamento básico destes algoritmos, além de apresentar problemas de desempenho: sejam dois cromossomos $C_1 = 1,1,1,0,0,0$ e $C_2 = 0,0,0,1,1,1$, definidos de acordo com a codificação descrita na Subseção 3.2, selecionados como pais para gerar dois novos filhos. Para este problema deseja-se uma quantidade de 3 SNPs no conjunto de tag SNP. Suponha-se que, durante a aplicação do cruzamento de um ponto, foi selecionado o ponto 3 para corte. Logo, os novos filhos serão $Cf_1 = 1,1,1,1,1,1$ e $Cf_2 = 0,0,0,0,0,0$. De acordo com a estratégia de reparação de Mouawad e Mansour (2014), 3 genes do indivíduo Cf_1 serão alterados aleatoriamente de 1 para 0 e 3 genes do indivíduo Cf_2 serão alterados aleatoriamente de 0 para 1. Neste caso, o processamento dispendido na realização da operação de cruzamento poderia ter sido evitado e somente um processo de geração aleatória de modificações poderia ter sido realizado. Logo, esta abordagem não se mostra adequada.

A fim de garantir que a quantidade de SNPs selecionada não seja alta, a função de avaliação $fit(C)$ considera também uma penalidade às soluções com quantidades de tag SNPs diferentes da definida pelo usuário. Para tal, durante a avaliação, cada indivíduo é penalizado de maneira proporcional à diferença entre a quantidade de SNPs presente e a

desejada. Seja $0 < k < |L|$ a quantidade desejada, definida no início da execução do algoritmo, $|L|$ o tamanho do conjunto original e $|C'|$ a quantidade de tag SNPs representada pelo indivíduo C. A penalidade ψ é, então, definida de acordo com a Equação 15. Na Figura 6 é ilustrada a intensidade desta penalização.

$$\psi = \left(\frac{|C'| - k}{|L| - k}\right)^2 \tag{15}$$

Figura 6— Intensidade da penalização em relação à diferença entre a quantidade de SNPs desejada e a contida na solução – O ponto azul representa a quantidade desejada



Fonte: Elaborado pelo autor

3.6 Critério de parada

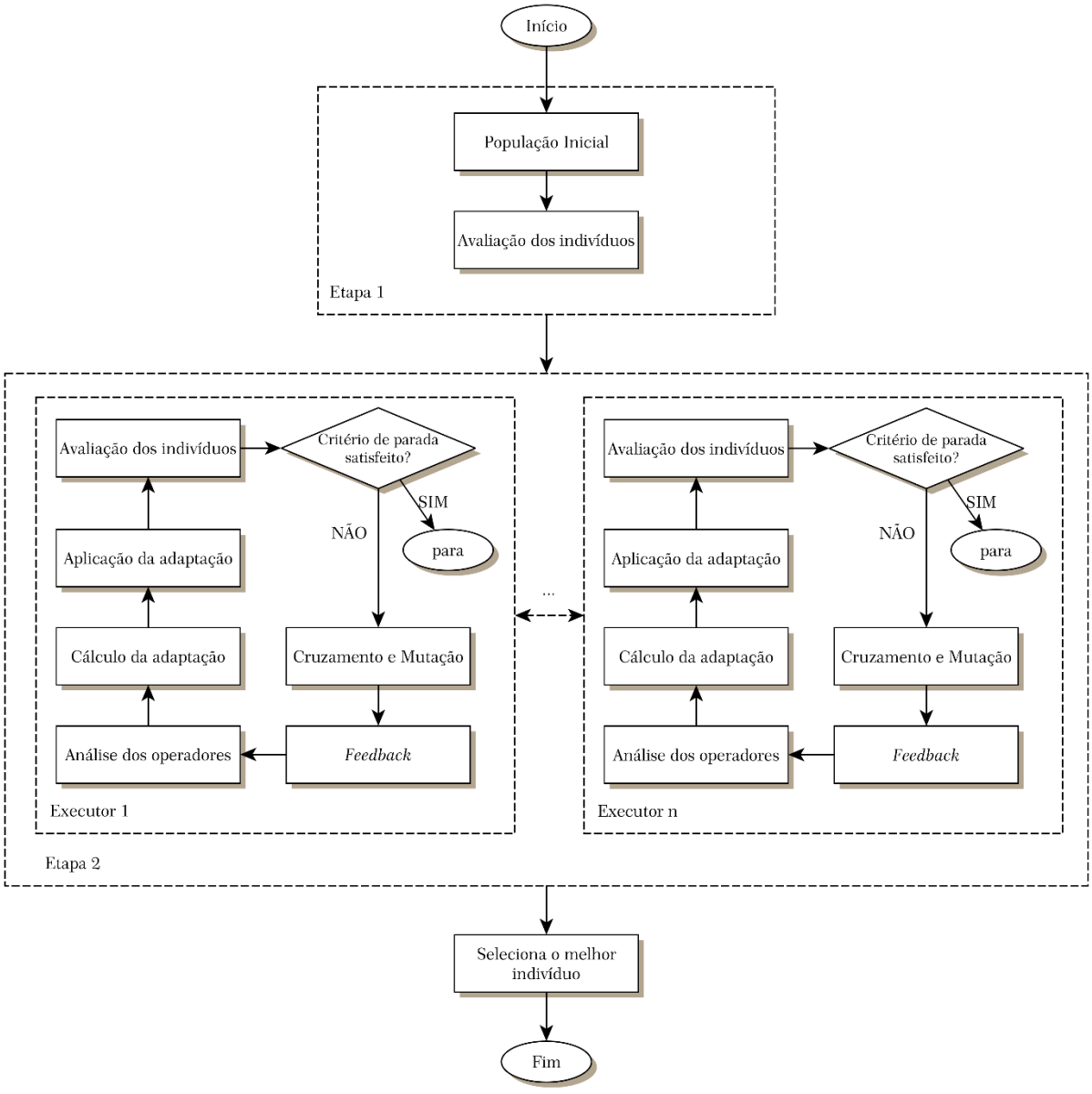
Como a solução ótima global para o problema de seleção de tag SNPs não é conhecida previamente, é preciso que o AG considere algum critério de parada para interromper a geração de novas populações e retorne o resultado atual. Neste trabalho, foram considerados dois critérios:

a) caso o melhor indivíduo da população não apresente melhora no decorrer de 500 gerações sucessivas, considera-se que o AG convergiu para uma solução ótima e o processo é interrompido. Caso contrário, aplica-se o segundo critério;
b) caso tenham sido decorridas 100.000 gerações, considera-se que o AG convergiu para uma solução ótima e o processo é interrompido.

3.7 Estratégia de paralelização

Tal como visto na Subseção 2.7, algoritmos genéticos paralelizados por meio da abordagem distribuída apresentam maior eficiência. Por outro lado, a paralelização de algoritmos genéticos por meio da abordagem padrão também é relevante, uma vez que é possível calcular de maneira paralela operações complexas, como o cálculo das funções de aptidão (LINDEN, 2008). Neste cenário, a estratégia de paralelização considerada neste trabalho é implementada em duas fases, tal como representado na Figura 7: avaliação dos indivíduos da população inicial e aplicação dos operadores genéticos, de modo que a primeira etapa corresponde à abordagem padrão e a segunda à abordagem distribuída.

Figura 7— Algoritmo Genético desenvolvido



Fonte: Elaborado pelo autor

A avaliação dos indivíduos da população inicial é a primeira fase da estratégia de paralelização e é feita com base no processamento *in-memory* possibilitado pelo *framework* Apache Spark. Primeiramente, a população inicial, gerada aleatoriamente, é recebida e, posteriormente, paralelizada em diferentes partições de um RDD. Então, uma transformação é aplicada para transformar cada elemento de RDD de origem em um RDD que contenha os pares `(indivíduo, valor_ avaliação)`. Uma função `.calculaAvaliacao()` é executada nos núcleos de processamento disponíveis.

Esta operação é realizada somente para a primeira geração, criada aleatoriamente, uma vez que todos os indivíduos são novos e precisam ser avaliados. Nas gerações seguintes, considera-se que indivíduos que sobreviveram à geração anterior já foram avaliados e, assim, o cálculo é realizado após as operações de cruzamento e mutação.

Após a geração e avaliação da população inicial, a estratégia de paralelização faz uso de n executores para conduzir a realização de processos evolutivos independentes (Etapa 2). A população inicial com os respectivos valores de avaliação de cada indivíduo é paralelizada em um RDD. O processo evolutivo é, então, iniciado em cada executor por meio de uma transformação, no qual uma operação `.solve()` é executada, de modo a representar todo o processo evolutivo descrito nas seções anteriores.

A troca de material genético entre as populações evoluídas nestes processos evolutivos é realizada por meio de indivíduos que migram de uma população para outra, aos quais atribui-se o nome de indivíduos migrantes. Este mecanismo de troca de indivíduos busca acelerar a convergência, uma vez que dissemina as características evolutivas no conjunto de populações, ao mesmo tempo em que tenta evitar a convergência para um mínimo local. Esta operação é realizada por meio do conceito de particionamento do Apache Spark, no qual as transformações `.repartition()` e `.coalesce()` são responsáveis por selecionar os indivíduos de cada instância evolutiva e os transferirem para outras partições.

O processo evolutivo da origem a um RDD que contém pares `(melhor_indivíduo, valor_avaliação)` de cada executor. Por fim, os valores são agregados de modo a ser possível recuperar o melhor indivíduo.

3.8 Considerações finais

Nesta seção foram apresentados os detalhes de implementação do AG desenvolvido. Inicialmente o problema foi formulado formalmente e os detalhes do algoritmo genético foram expostos. A estratégia de adaptação é apresentada na definição do método de seleção

de indivíduos para reprodução e na descrição dos operadores de cruzamento e mutação. Por fim, a estratégia de paralelização foi exposta, o que contempla a versão final do algoritmo.

Na seção seguinte são apresentados resultados e discussões a respeito da avaliação experimental do trabalho.

4 Avaliação experimental

Nesta seção é apresentada a metodologia aplicada aos experimentos de avaliação do algoritmo genético adaptativo e paralelo para seleção de polimorfismos de nucleotídeo único representativos, assim como os resultados dos experimentos obtidos a partir da implementação do trabalho desenvolvido.

4.1 Metodologia de experimentação

Os experimentos realizados tiveram como objetivo validar a efetividade das duas características do algoritmo genético desenvolvido quando aplicado ao problema de seleção de tag SNPs: estratégia de adaptação e paralelismo.

Todos os algoritmos descritos foram executados cinco vezes e os resultados relatados correspondem à média das cinco execuções, a fim de garantir consistência estatística. O desvio padrão é apresentado entre parênteses à frente dos valores reportados.

Na Tabela 2 é apresentada a configuração do ambiente de experimentação.

Tabela 2— Ambiente de experimentação

Variável controlada	Valor
Processador	Inter Core i5-4440S 4 núcleos a 2.80GHz
Memória	8Gb a 1333MHz
Disco rígido	500Gb
Sistema Operacional	Ubuntu Desktop 16.12 LTS

Fonte: Elaborado pelo autor.

A fim de avaliar as características qualitativas do algoritmo, no que diz respeito à qualidade dos subconjuntos de SNPs produzidos, foram utilizados conjuntos de dados reais,

disponíveis por meio dos Projetos ENCODE (fase final 1, julho 2005), tal como feito em trabalhos relacionados da literatura. Na Tabela 3 são apresentados todos os conjuntos de dados e as respectivas quantidades de SNPs em cada. Os conjuntos ENr113 e ENm014 consistem em conjuntos de haplótipos pertencentes à indivíduos da população Yoruba, da Nigéria (conjuntos de dados YRI) e as bases de dados ENr112 e ENm013 consistem em conjuntos de haplótipos pertencentes à indivíduos da população de ancestrais europeus em Utah (conjuntos de dados CEU).

Tabela 3— Conjuntos de dados utilizados nos experimentos

Conjunto de dados	População	Quantidade de SNPs
ENr112	CEU	1134
ENr113	YRI	1525
ENm013	CEU	1001
ENm014	YRI	1224

Fonte: Elaborado pelo autor

Primeiramente, na Subseção 4.2 é apresentada uma análise detalhada a respeito da característica adaptativa do algoritmo genético desenvolvido. Na Subseção 4.3 são apresentados os resultados dos experimentos realizados com o objetivo de comparar a qualidade dos subconjuntos de tag SNPs obtidos por meio da execução de três variações de algoritmo genético: convencional, apenas adaptativo, e adaptativo e paralelo. Por fim, na Subseção 4.4 são apresentados testes de desempenho.

4.2 Avaliação da estratégia de adaptação

A fim de validar a característica adaptativa dos parâmetros, foram realizados experimentos numéricos fora do contexto do problema de seleção de SNPs representativos. Estes experimentos foram conduzidos com o objetivo de que fosse possível analisar o comportamento dos AGs na ausência e presença de estratégias de adaptação de parâmetros e operadores. Para análise, foram considerados valores relacionados à:

a) quantidade de iterações necessárias para conversão até a solução ótima;
b) diversidade das populações no decorrer das execuções e;
c) qualidade da solução obtida.

Nestes experimentos, a codificação das soluções é a representação binária dos valores considerados em cada problema, com precisão de 14 casas decimais, as funções de avaliação das soluções são as próprias funções $f(x)$, o tamanho da população $P = 10$ e foi estabelecido como critério de parada a execução de 300 iterações sem melhoria. Estes valores foram

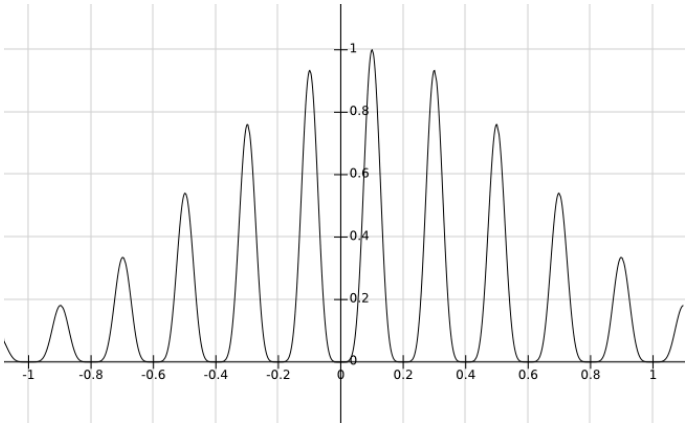
definidos empiricamente por meio de experimentos preliminares. Além disso, o algoritmo sem estratégia de adaptação apresenta os seguintes valores: taxa de cruzamento $p_c = 0{,}8$ e taxa de mutação $p_m = 0{,}01$. Os valores de diversidade foram normalizados no intervalo [0,1]. O algoritmo com estas configurações foi, então, comparado com o algoritmo desenvolvido, descrito na Seção 3, com exceção da estratégia de paralelização.

Primeiramente, foi considerado o problema numérico de otimização definido na Equação 16:

$$\begin{aligned} &Maximizar \\ &f(x) = 2^{-2\left(\frac{x-0{,}1}{0.9}\right)^2}.\sin(5\pi x)^6 \\ &sujeito\ a \\ &-1 < x < 1 \end{aligned} \tag{16}$$

Este problema de maximização tem sua solução ótima conhecida em $x = 0{,}1$, quando a função atinge seu valor máximo $f(x) = 1$, tal como pode ser visto por meio do Gráfico 1.

Gráfico 1— Função f(x) para o problema de maximização

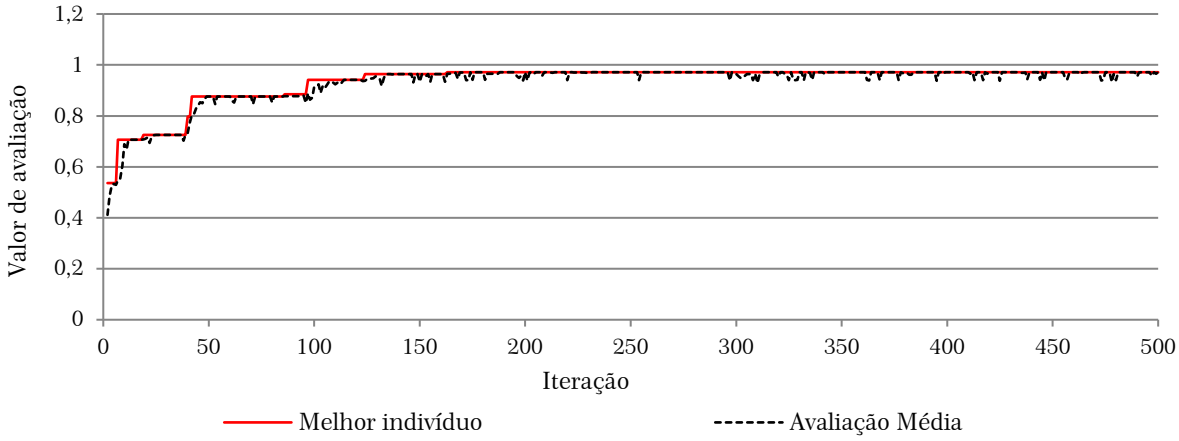


Fonte: Elaborado pelo autor

No Gráfico 2 está representado o melhor indivíduo, assim como o valor médio de avaliação da população, em cada iteração do AG sem estratégia de adaptação e, no Gráfico 3, as mesmas informações são apresentadas, porém considerando-se um AG com estratégia de adaptação.

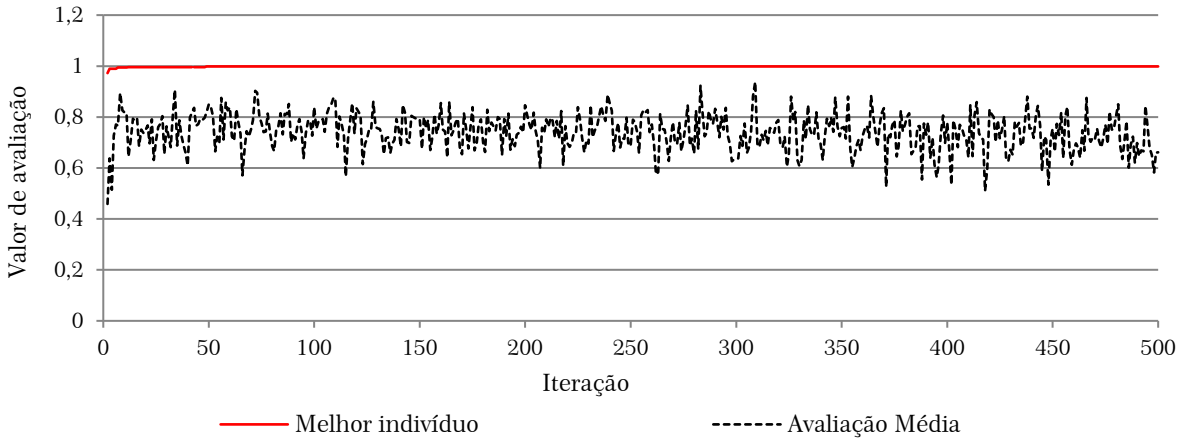
Por meio da análise destes gráficos é possível verificar que o valor de avaliação médio das populações é maior para os experimentos no qual não há presença de estratégia de adaptação. Por se tratar de um problema de maximização, seria possível, então, inferir que esta abordagem apresenta melhores resultados. Contudo, o fato de o valor de avaliação médio ser maior para esta abordagem permite apenas inferir que seus indivíduos apresentam maiores valores de aptidão no decorrer das iterações o que, por si só, não é uma informação necessariamente positiva, tampouco negativa.

Gráfico 2— Desempenho do AG sem estratégia de adaptação para o problema de maximização



Fonte: Elaborado pelo autor

Gráfico 3— Desempenho do AG com estratégia de adaptação para o problema de maximização



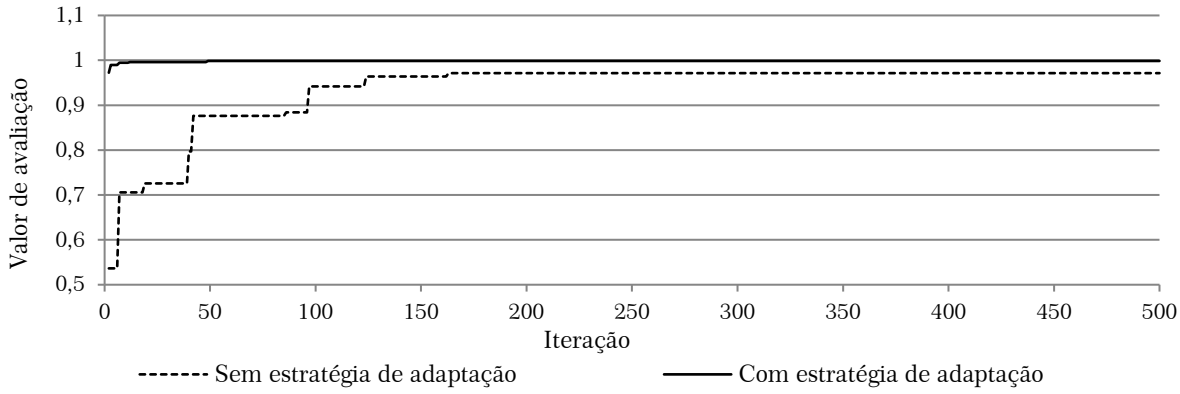
Fonte: Elaborado pelo autor

A fim de permitir melhor análise das diferentes abordagens, no Gráfico 4 é apresentada uma comparação entre o desempenho dos AGs na presença e ausência da estratégia de adaptação, em relação ao melhor indivíduo de cada iteração e, no Gráfico 5, é apresentado uma comparação em relação ao valor de diversidade populacional em cada iteração com base na Equação 5.

Ao se analisar os resultados apresentados no Gráfico 4, percebe-se que a abordagem com estratégia de adaptação apresenta melhores resultados, visto que o melhor indivíduo de cada população – aquele a ser eleito como solução final do problema – é melhor nesta abordagem do que na abordagem sem esta estratégia.

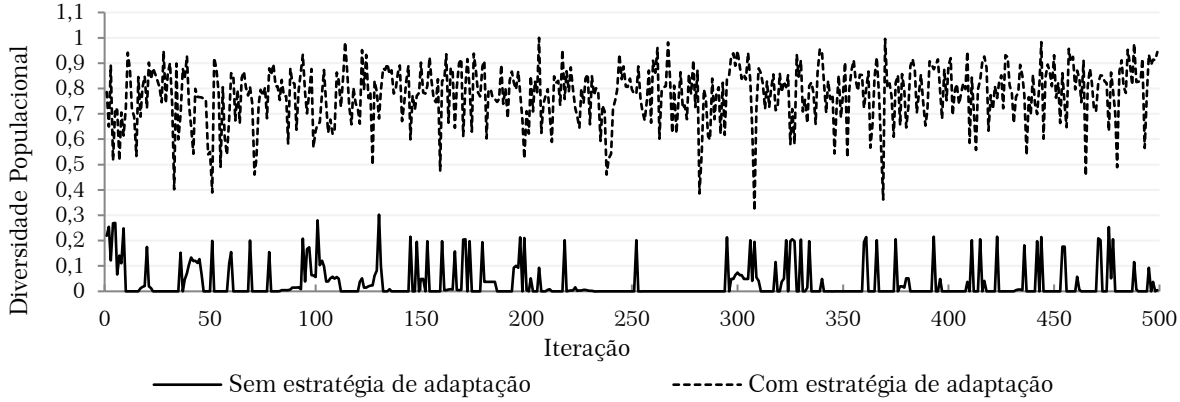
Adicionalmente, por meio do Gráfico 5, é possível perceber que diversidade populacional da abordagem com estratégia de adaptação também é melhor, o que, de acordo com o que foi apresentado na Seção 2.4, ajuda a explorar melhor o espaço de possíveis soluções do problema. Esta afirmação se mostra verdadeira ao analisarmos o comportamento

Gráfico 4— Comparação do desempenho dos AGs na presença e ausência da estratégia de adaptação em relação aos melhores indivíduos para o problema de maximização



Fonte: Elaborado pelo autor

Gráfico 5— Comparação do desempenho dos AGs na presença e ausência da estratégia de adaptação em relação aos melhores indivíduos para o problema de maximização



Fonte: Elaborado pelo autor

da função $f(x)$, por meio Gráfico 1: percebe-se que esta função possui muitos pontos de solução ótima local. Deste modo, observa-se que o AG sem estratégia de adaptação se mostrou estagnado no decorrer do processo evolutivo em soluções locais, uma vez que o espaço de soluções não foi explorado adequadamente. Consequentemente, a diversidade populacional entre os indivíduos considerados no AG sem estratégia de adaptação manteve-se muito baixa, ao passo que a estratégia de adaptação, ainda assim, foi eficaz no sentido de promover a diversidade populacional.

A quantidade de iterações necessárias até a conversão também é comparada no Gráfico 6, no qual os valores relatados representam a 300ª iteração sem melhoria e o valor entre parênteses representa o desvio padrão. Por meio da análise, é possível constatar que a abordagem com estratégia de adaptação possui velocidade de convergência maior, uma vez que apresentou redução de 32,6% na quantidade de iterações em relação à abordagem sem estratégia de adaptação. Adicionalmente, pode-se verificar a maior velocidade de convergência ao se analisar as primeiras iterações representadas no Gráfico 4.

Gráfico 6— Comparação do desempenho dos AGs na presença e ausência da estratégia de adaptação em relação à quantidade de iterações para o problema de maximização

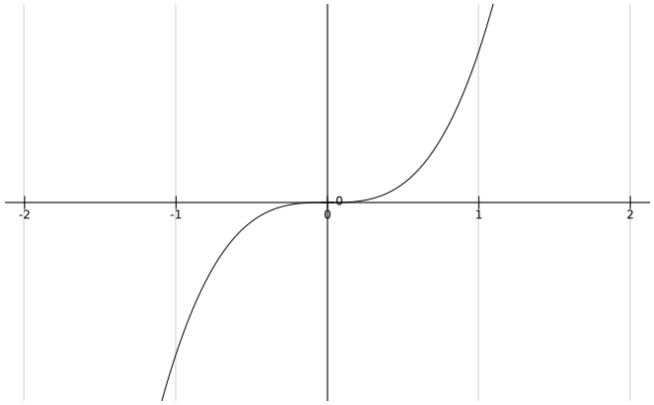
Fonte: Elaborado pelo autor

O segundo problema de otimização considerado é definido como:

$$\begin{aligned} &Minimizar \\ &\qquad f(x) = x^3 \\ &sujeito\ a \\ &\qquad -8 \leq x \leq 8 \end{aligned} \tag{17}$$

A função $f(x) = x^3$ não possui solução ótima global, tal como pode ser visto por meio do Gráfico 7. Porém, quando é considerado a restrição do problema de otimização, no qual $-8 \leq x \leq 8$, a solução ótima deste problema passa a ser $x = -8$, quando a função atinge seu valor mínimo $f(x) = -512$.

Gráfico 7— Função f(x) para o problema de minimização



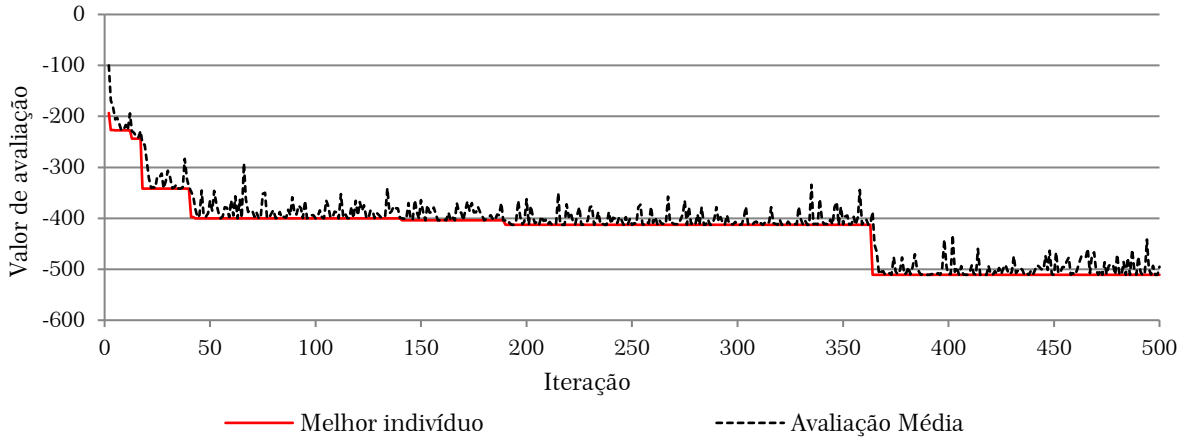
Fonte: Elaborado pelo autor

No Gráfico 8 está representado o melhor indivíduo, assim como o valor médio de avaliação da população, em cada iteração do AG sem a presença de estratégia de adaptação e, na Gráfico 9, as mesmas informações são apresentadas, porém considerando-se um AG com estratégia de adaptação.

Além disso, a fim de permitir melhor comparação entre as abordagens, é realizada comparação entre o desempenho dos AGs na presença e ausência da estratégia de adaptação, em relação ao melhor indivíduo de cada iteração (Gráfico 10) e em relação ao valor de diversidade populacional em cada iteração com base na Equação 5 (Gráfico 11). A quantidade de iterações necessárias até a conversão também é comparada no Gráfico 12, no qual os valores relatados representam a 300ª iteração sem melhoria.

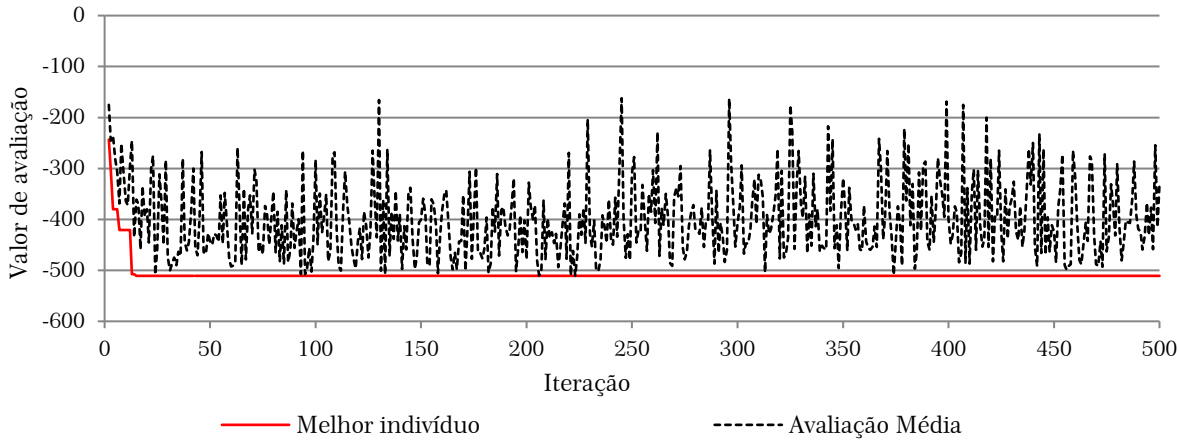
Ao analisar os resultados apresentados no Gráfico 8 e Gráfico 9, também poderia ser inferido que, para o problema de minimização da função $f(x)$, a abordagem sem estratégia

Gráfico 8— Desempenho do AG sem estratégia de adaptação para o problema de minimização



Fonte: Elaborado pelo autor

Gráfico 9— Desempenho do AG com estratégia de adaptação para o problema de minimização

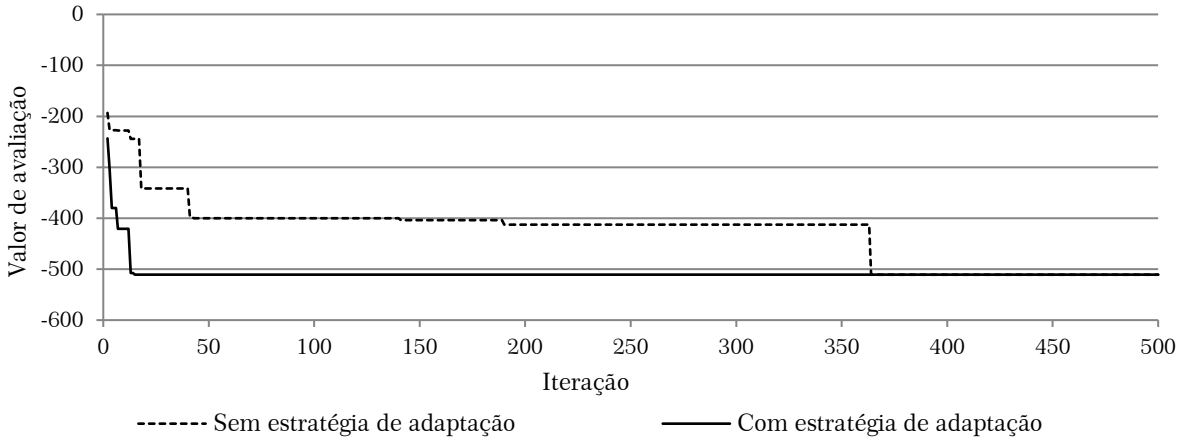


Fonte: Elaborado pelo autor

de adaptação é melhor, visto que os valores de avaliação médio das populações são menores. Contudo, por meio do Gráfico 10, é possível verificar que ambas as abordagens convergem corretamente para o mesmo ponto de solução ótima, ainda que a diversidade populacional da abordagem com estratégia de adaptação seja melhor (Gráfico 11). Este fato pode ser explicado pela característica da função $f(x)$, que não possui pontos de solução ótima local.

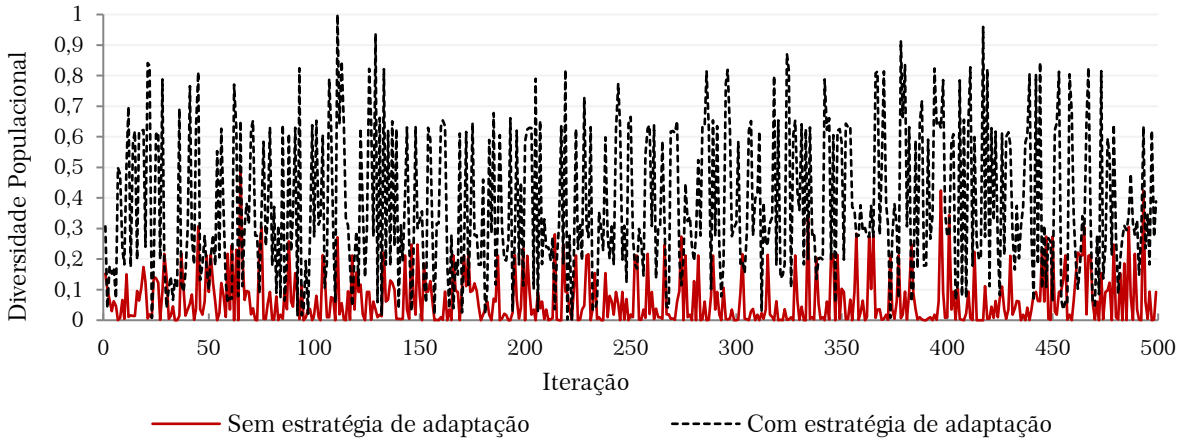
Apesar disso, não se pode inferir que houve desperdício de processamento com a aplicação do controle de pressão seletiva para a resolução deste problema, uma vez que houve maior velocidade de convergência para esta abordagem (Gráfico 12), no qual foi constatada redução de 52,2% na quantidade de iterações em relação à abordagem sem estratégia de adaptação. Esta característica pode tornar viável a resolução de problemas complexos que demandam muito processamento computacional.

Gráfico 10— Comparação do desempenho dos AGs na presença e ausência da estratégia de adaptação em relação aos melhores indivíduos para o problema de minimização



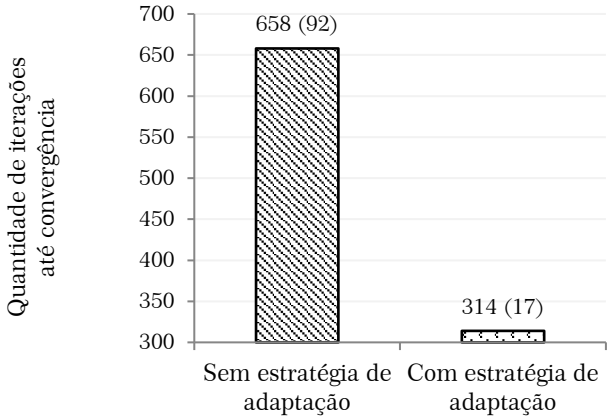
Fonte: Elaborado pelo autor

Gráfico 11— Comparação do desempenho dos AGs na presença e ausência da estratégia de adaptação em relação à métrica de diversidade populacional para o problema de minimização



Fonte: Elaborado pelo autor

Gráfico 12— Comparação do desempenho dos AGs na presença e ausência da estratégia de adaptação em relação à quantidade de iterações para o problema de minimização



Fonte: Elaborado pelo autor

4.3 Testes de qualidade

A fim de validar a eficácia das estratégias de adaptação e paralelização desenvolvidas na identificação de SNPs representativos, foram comparados os resultados da qualidade do subconjunto de SNPs gerados considerando-se três variações de algoritmos genéticos:

a) algoritmo genético convencional;
b) algoritmo genético adaptativo e;
c) algoritmo genético adaptativo e paralelo.

O algoritmo genético convencional considerado faz uso das mesmas configurações definidas na Seção 3, porém não são considerados os valores de convergência e diversidade para alterar o mecanismo de seleção durante a execução, tampouco para alterar as probabilidades de ocorrência dos operadores de cruzamento e mutação. Também não há estratégia de paralelização envolvida, logo é executado tal como definido pelo Algoritmo 1 (página 22). Neste caso, foram considerados os seguintes parâmetros, com base em correlatos da literatura: taxa de cruzamento $p_c = 0{,}8$ e taxa de mutação $p_m = 0{,}01$. O algoritmo genético adaptativo faz uso das mesmas configurações definidas na Seção 3, com exceção da estratégia de paralelização. O algoritmo genético adaptativo e paralelo representa o trabalho desenvolvido, de acordo com o relatado na Seção 3, e considera a presença de 4 processos evolutivos simultâneos. Nos experimentos, foi considerado o tamanho da população $P = 200$ e a quantidade de tag SNPs $k = 100$.

Os resultados da execução de cada algoritmo, apresentados por meio da Tabela 4, foram comparados em relação as métricas:

a) quantidade de iterações até a convergência;
b) aptidão do conjunto de tag SNPs gerado.

Tabela 4— Comparação de resultados entre três variações de algoritmos genéticos

	Conjunto de Dados	AG Convencional *	AG Adaptativo *	AG Adaptativo e Paralelo *
Quantidade de iterações até convergência	ENr112	2178,4 (227,40 – 10,4%)	1413,8 (129,55 – 9,1%)	1338,8 (106,30 – 7,9%)
	ENr113	2326,6 (427,60 – 18,3%)	1573,8 (153,76 – 9,7%)	1579,2 (76,83 – 4,8%)
	ENm013	1589,4 (543,46 – 34,1%)	1175,2 (132,06 – 11,2%)	1070,6 (115,20 – 10,7%)
	ENm014	1996,2 (327,31 – 16,3%)	1831,2 (125,09 – 6,8%)	1827,6 (80,46 – 4,4%)
Aptidão solução	ENr112	0,7479 (0,0367 – 4,9%)	0,8027 (0,0128 – 1,59%)	0,8105 (0,0074 – 0,9%)
	ENr113	0,7166 (0,0340 – 4,7%)	0,7746 (0,0078 – 1%)	0,7780 (0,0097 – 1,25%)
	ENm013	0,5840 (0,0786 – 13,4%)	0,6954 (0,0123 – 1,7%)	0,6976 (0,0098 – 1,4%)
	ENm014	0,6647 (0,0461 – 6,9%)	0,7106 (0,0112 – 1,57)	0,7170 (0,0109 – 1,52%)

** Dados apresentados no formato X (Y – Z), no qual X representa a média, Y o desvio padrão e Z o coeficiente de variação.*

Fonte: Elaborador pelo autor.

Por meio da análise dos dados da Tabela 4, é possível, de modo geral, observar melhoria das soluções encontradas para o problema de seleção de SNPs representativos por meio da utilização da estratégia de adaptação desenvolvida: em todas as bases de dados analisadas, houve aumento do valor de aptidão das soluções encontradas em relação ao algoritmo genético convencional. Os aumentos foram de 7,27%, 8,09%, 19,08% e 6,92% em relação às bases de dados ENr112, ENr113, ENm013 e ENm014, respectivamente.

Observa-se, ainda, que o AG com estratégia de adaptação foi capaz de convergir para uma solução em uma quantidade menor de iterações em relação ao AG convencional. A diminuição foi de 35,10%, 32,36%, 26,06% e 08,27% em relação às bases de dados ENr112, ENr113, ENm013 e ENm014, respectivamente.

Esses resultados vão de encontro à analise descrita na Subseção 4.2, no qual concluiu-se que, na presença de mecanismos de controle da pressão seletiva, é possível garantir maior diversidade de indivíduos nas populações, aumentar a velocidade de convergência e, consequentemente, melhorar o resultado das soluções encontradas pelo algoritmo.

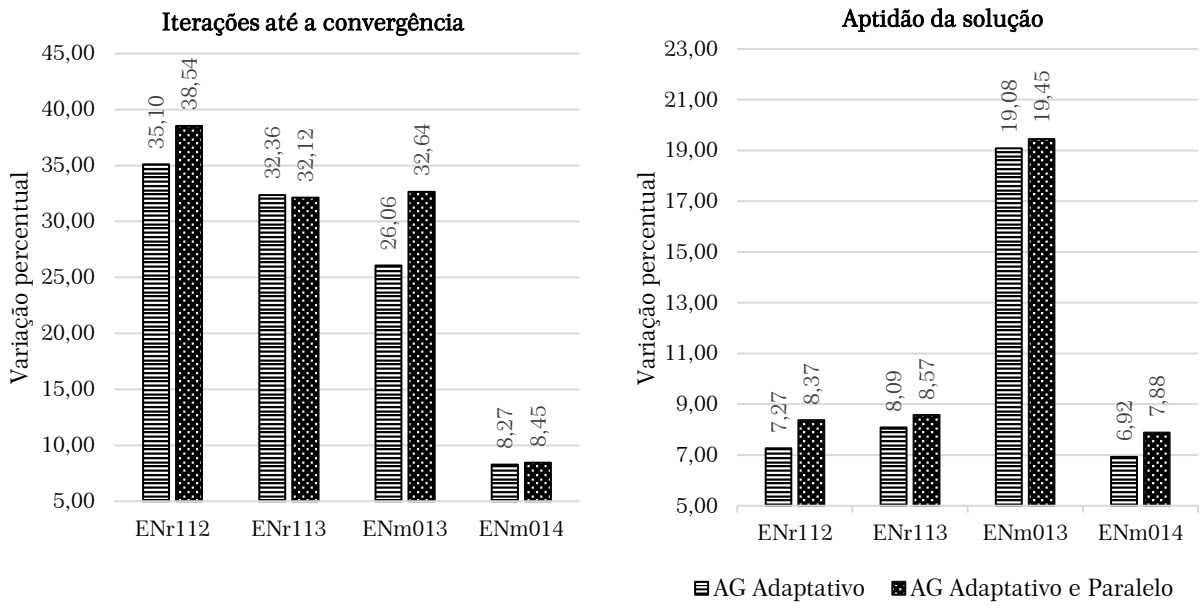
Foi possível observar resultados semelhantes quando se considerou, além da estratégia de adaptação, a estratégia de paralelização. Os valores de aptidão também foram melhores quando comparados aos da abordagem convencional: os aumentos foram de

8,37%, 8,57%, 19,45% e 7,88% em relação às bases de dados ENr112, ENr113, ENm013 e ENm014, respectivamente.

O AG adaptativo e paralelo desenvolvido também apresentou melhores resultados em relação à quantidade de iterações até a convergência quando comparados aos da abordagem convencional: a diminuição foi de 38,54%, 32,12%, 32,64% e 8,45% em relação às bases de dados ENr112, ENr113, ENm013 e ENm014, respectivamente.

No **Erro! Fonte de referência não encontrada.** há uma comparação das melhorias p ercentuais observadas entre os AGs adaptativo e adaptativo/paralelo em relação ao AG convencional. É possível observar que em 7 das 8 comparações, a variação percentual em relação ao AG convencional é maior para o AG adaptativo e paralelo desenvolvido. Assim, conclui-se que a estratégia de paralelização desenvolvida, apesar de o controle de pressão seletiva ter se mostrado o principal responsável pelas melhorias observadas, também contribui com a melhoria dos resultados por meio da maior exploração do espaço de soluções do problema.

Gráfico 13— Variações percentuais dos AGs adaptativo e adaptativo/paralelo em relação ao AG convencional



Fonte: Elaborador pelo autor.

Observou-se, ainda, por meio da análise dos coeficientes de variação, que os resultados do experimento foram mais homogêneos nos experimentos com utilização da abordagem desenvolvida: as variações nos resultados foram mais acentuadas quando da utilização do AG convencional, principalmente no que se refere à quantidade de iterações até a convergência. Uma hipótese para esta observação é o fato de o AG convencional não

apresentar controle de pressão seletiva e, consequentemente, falhar em lidar com os pontos de solução local.

Esta característica foi observada de maneira mais intensa no conjunto ENm013. A quantidade de iterações até a solução do problema variou, durante os experimentos, de 907 até 2273. No primeiro caso, houve convergência prematura para uma solução com valor de aptidão de 0,4320, muito inferior aos resultados encontrados por meio da utilização da abordagem desenvolvida neste trabalho. Estes resultados também validam as observações relatadas na Subseção 4.2, no que diz respeito à utilização de mecanismos de controle de pressão seletiva.

Adicionalmente, é possível observar menores valores de coeficiente de variação entre as amostras observadas por meio da utilização do AG adaptativo e paralelo. Este resultado demonstra que o algoritmo desenvolvido, além de convergir mais rapidamente e apresentar melhores soluções, é mais eficiente no sentido de encontrar soluções com pouca variação no decorrer das execuções.

O trabalho desenvolvido foi avaliado, ainda, em relação à estratégia utilizada para lidar com a necessidade de diminuição do subconjunto de SNPs gerados, ao mesmo tempo em que se deseja maximizar o valor da função de avaliação. Considerações a respeito da estratégia utilizada neste trabalho, assim como da utilizada em trabalhos da literatura, foram realizadas oportunamente na Subseção 3.5.1.

Neste cenário, foi realizada uma comparação entre diferentes estratégias. Nesta comparação, foram utilizados dois algoritmos genéticos convencionais configurados com os mesmos parâmetros. O primeiro AG fez uso da estratégia de reparação do trabalho de Mouawad e Mansour (2014), que consiste em acrescentar ou remover tagSNPs dos indivíduos aleatoriamente para reparar indivíduos com quantidades de tag SNPs diferentes da desejada. O segundo AG, por sua vez, fez uso da estratégia de penalização definida pela Equação 15. Em ambos os algoritmos foram considerados os seguintes valores: taxa de cruzamento $p_c = 0{,}8$, taxa de mutação $p_m = 0{,}01$, tamanho da população $P = 200$ e quantidade de tag SNPs $k = 100$.

Os resultados, apresentados na Tabela 5, indicam que o algoritmo que utiliza a abordagem de penalização desenvolvida neste trabalho, embora não faça uso de mecanismos de controle de pressão seletiva, tampouco de estratégias de paralelização, convergiu mais rapidamente para uma solução em todas as bases de dados analisadas: em média, observou-se uma quantidade de iterações 18% menor. Pode-se inferir, assim, que a estratégia de reparação utilizada por Mouawad e Mansour (2014) atuou de forma negativa na direção da evolução, o que ocasionou demora na convergência.

Tabela 5— Quantidade de iterações até convergência

	AG com estratégia de reparação de Mouawad e Mansour (2014)	AG com penalização de indivíduos deste trabalho
ENr112	2824 (355,23 – 12,5%)	2178,4 (227,40 – 10,4%)
ENr113	2836,4 (508,11 – 17,9%)	2326,6 (427,60 – 18,3%)
ENm013	2116,8 (403,76 – 19%)	1589,4 (543,46 – 34,1%)
ENm014	2278,6 (340,79 – 14,9%)	1996,2 (327,31 – 16,3%)

Fonte: Elaborado pelo autor.

4.4 Testes de desempenho

Uma vez apresentados os testes de qualidade, esta subseção tem como objetivo fazer considerações a respeito do desempenho do algoritmo desenvolvido. Os experimentos descritos foram executados com os SNPs do conjunto ENr113.

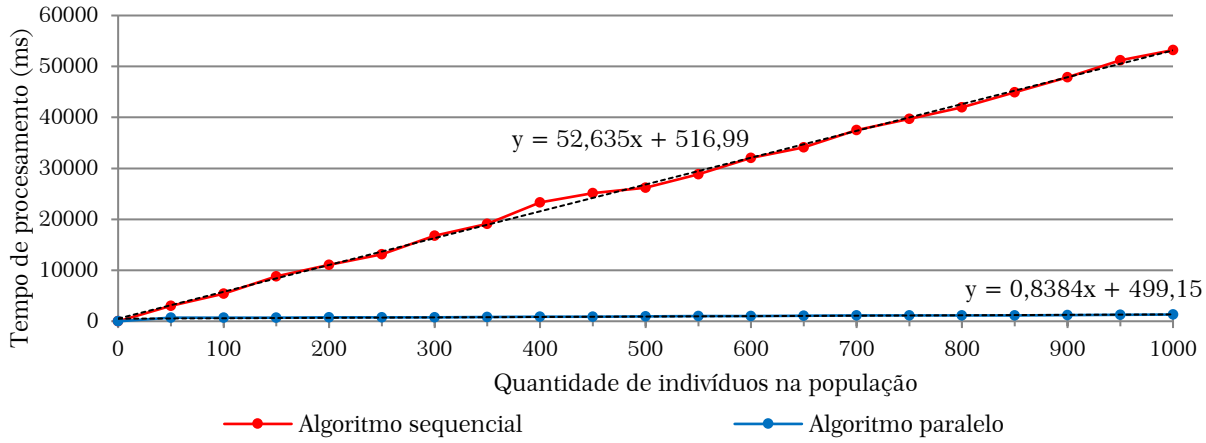
É importante ressaltar que o algoritmo contempla duas etapas de paralelização, tal como definido na Subseção 3.7. A primeira etapa faz uso do processamento paralelo para realizar a avaliação inicial de todos os indivíduos gerados no início da execução do algoritmo e corresponde à abordagem padrão de paralelização de algoritmos genéticos, descrita na Subseção 2.7. Neste caso, será analisada a capacidade do algoritmo desenvolvido de processar uma determinada quantidade de dados em tempo inferior quando comparado a um algoritmo sequencial. Na segunda etapa, por outro lado, os mecanismos de paralelização são utilizados com o objetivo de permitir a análise de uma maior quantidade de soluções, uma vez que corresponde à abordagem distribuída, também descrita na Subseção 2.7. Neste sentido, serão analisados os ganhos obtidos em relação à utilização do processamento em memória para permitir o processamento de uma maior quantidade de dados.

Inicialmente, foi analisado o desempenho do algoritmo de cálculo da função de avaliação, descrita na Subseção 3.5, para diferentes tamanhos de população. Os experimentos foram executados considerando-se apenas o tempo de processamento dispendido para avaliar todos os indivíduos da população. No Gráfico 14 são apresentados os resultados. É possível verificar que o algoritmo sequencial apresentou crescimento mais acentuado do tempo de processamento, conforme a quantidade de indivíduos da população aumentou: o crescimento do tempo de processamento é representado pelo Equação 18, que possui coeficiente angular de 52,635. O algoritmo paralelo, por sua vez, tem o crescimento do tempo de processamento representado pela Equação 19, que possui coeficiente angular de 0,8384. Logo, a abordagem paralela possui crescimento do tempo de processamento 62 vezes menor do que a abordagem sequencial.

$$y = 52{,}635x + 516{,}99 \quad (18)$$

$$y = 0{,}8384x + 499{,}15 \quad (19)$$

Gráfico 14— Comparação do tempo de processamento para diferentes tamanhos de população



Fonte: Elaborado pelo autor

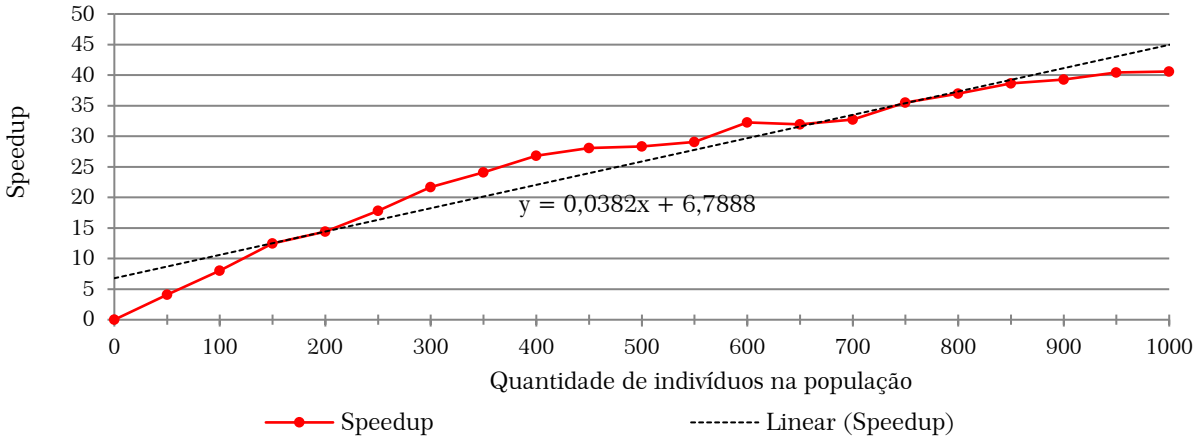
Para este experimento, considerou-se relevante, também, analisar o *speedup* em relação aos resultados do algoritmo para avaliação sequencial e paralela dos indivíduos. O *speedup* representa quão mais rápida determinada tarefa pode ser executada em um sistema que recebeu aprimoramentos, em relação ao sistema original. Seja t_{sm} o tempo de execução do sistema sem aprimoramento e t_{cm} o tempo de execução do sistema com aprimoramento, o $speedup$ é dado, então, pela Equação 20.

$$speedup = \frac{t_{sm}}{t_{cm}} \quad (20)$$

No Gráfico 15 é apresentado o *speedup* calculado para cada variação de tamanho da população. Observa-se, por meio dela, que o valor do *speedup* aumenta conforme a quantidade de indivíduos na população também aumenta, logo conclui-se que os benefícios derivados da implementação do algoritmo por meio do *framework* Apache Spark se acentuam conforme o volume de dados a ser processado aumenta.

Esta observação pode, ainda, ser reafirmada quando se analisa o tempo necessário para avaliação de cada indivíduo no decorrer dos experimentos. O algoritmo sequencial demorou, em média, 58,3 e 59,9 milissegundos para avaliar cada indivíduo, quando foi considerada uma população com 50 e 1000 indivíduos, respectivamente. Já o algoritmo paralelo demorou, em média, 14,6 e 1,3 milissegundos para avaliar cada indivíduo, quando foi considerada uma população com 50 e 1000 indivíduos, respectivamente.

Gráfico 15— Speedup para diferentes tamanhos de população

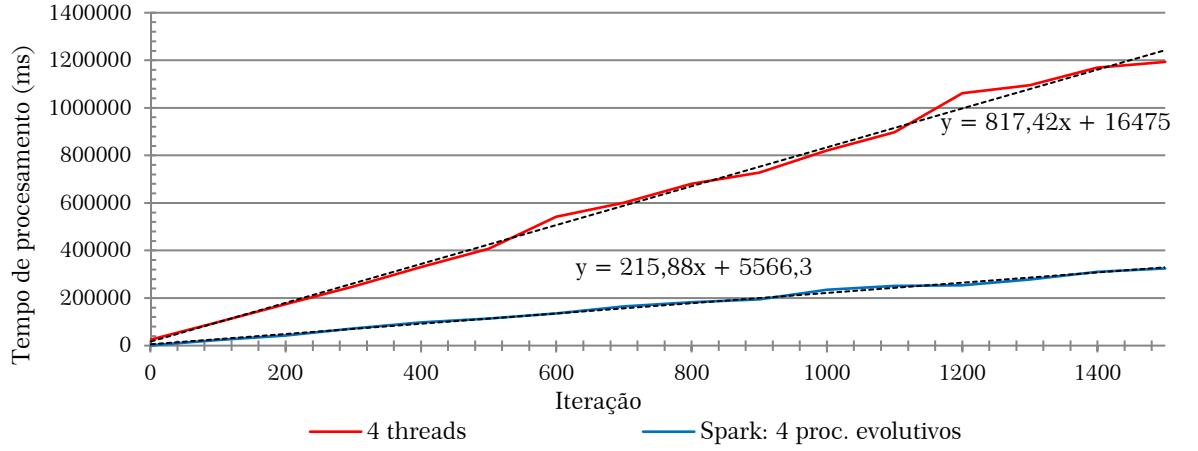


Fonte: Elaborado pelo autor

A fim de comparar os ganhos de desempenho obtidos em relação à segunda etapa da estratégia de paralelização, o algoritmo desenvolvido foi comparado à uma implementação com paralelização realizada por meio de *Threads* da linguagem JAVA. Nesta implementação, após a geração e avaliação da população inicial, cada processo evolutivo independente é executado simultaneamente em uma *thread*. Ambos os algoritmos foram executados com 4 processos evolutivos simultâneos. Foram considerados os seguintes valores: tamanho da população $P = 500$ e quantidade de tag SNPs $k = 100$.

O objetivo deste experimento foi apresentar os ganhos obtidos por meio da utilização do processamento em memória possibilitado pelo *framework* Apache Spark. Os resultados estão representados no Gráfico 16, no qual é apresentada a evolução do tempo de processamento no decorrer do processo evolutivo para cada uma das implementações consideradas.

Gráfico 16— Comparação da evolução temporal no decorrer das iterações com diferentes implementações de AG paralelo



Fonte: Elaborado pelo autor

O AG paralelizado por meio de *threads* atingiu a 1500ª iteração, considerada como critério de parada neste experimento, em 1193 (157) segundos, em média, ao passo que o AG desenvolvido neste trabalho atingiu a mesma iteração em 324 (38) segundos, uma redução de 72,8%. O experimento permitiu verificar, ainda, que após a primeira iteração, os AGs paralelizado com *thread* e o AG desenvolvido apresentaram uma média de tempo por iteração de 849 (46) milissegundos e 226 (10) milissegundos, respectivamente.

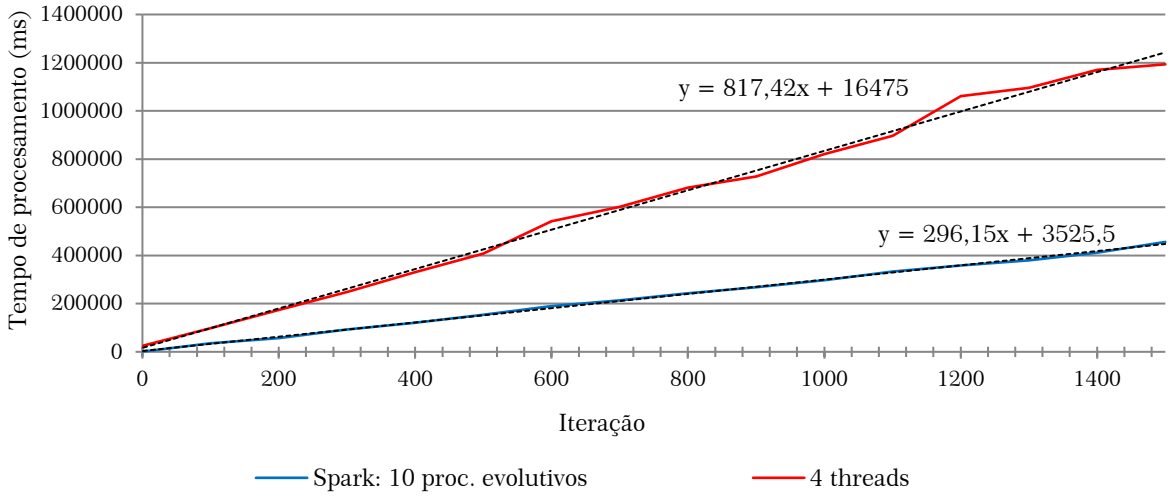
É possível, ainda, analisar o crescimento do tempo de processamento em relação à quantidade de iterações do algoritmo: o AG paralelizado com *threads* tem o crescimento do tempo de processamento representado pela Equação 21, que possui coeficiente angular de 817,42. O AG desenvolvido, por sua vez, tem o crescimento do tempo de processamento representado pela Equação 22, que possui coeficiente angular de 215,88, logo apresenta crescimento do tempo de processamento 3,7 vezes menor.

$$y = 817{,}42x + 16475 \tag{21}$$

$$y = 215{,}88x + 5566{,}3 \tag{22}$$

A fim de verificar a capacidade do algoritmo desenvolvido de analisar mais soluções do espaço de busca e, ainda assim, ser executado com tempo de processamento viável, foi analisado o comportamento de um AG no qual há a presença de 10 processos evolutivos simultâneos. O resultado está representado no Gráfico 17, no qual há, também, os resultados do experimento por meio da utilização de *threads* com 4 processos evolutivos.

Gráfico 17— Comparação da evolução temporal no decorrer das iterações com diferentes quantidades de processos evolutivos



Fonte: Elaborado pelo autor

Observa-se que o crescimento do tempo de processamento com a utilização de 10 processos evolutivos, representado por meio da Equação 23, possui coeficiente angular de 296,15. Assim, observou-se que, apesar de executar mais processos evolutivos, o algoritmo

desenvolvido foi capaz de apresentar crescimento 2,76 vezes menor do que a implementação com *threads*.

$$y = 296{,}15x + 3525{,}5 \tag{23}$$

4.5 Considerações finais

A avaliação experimental realizada permitiu, a partir da análise dos resultados, observar a efetividade das estratégias implementadas neste trabalho. Por meio do algoritmo genético adaptativo e paralelo desenvolvido, foi possível realizar maior exploração do espaço de busca, reduzir a quantidade de iterações até a convergência dos algoritmos e, consequentemente, encontrar melhores soluções para os problemas de seleção de SNPs representativos. Além disso, a utilização dos conceitos de computação em memória, possibilitados por meio da paralelização com o *framework* Apache Spark, permitiu a redução no tempo de processamento da função de avaliação de indivíduos, assim como do processo evolutivo como um todo. Esta característica, além de viabilizar o processamento em grandes conjuntos de dados, possibilita a exploração de um maior número de soluções do problema, o que pode acarretar ganhos significativos no que diz respeito à qualidade dos subconjuntos de SNPs encontrados.

5 Conclusão

Com o avanço das pesquisas nas áreas de biologia molecular e celular, a análise de SNPs se apresenta como uma ferramenta importante para o entendimento de patologias e os respectivos tratamentos. No entanto, devido à quantidade de dados de SNPs disponíveis, a seleção de SNPs representativos se tornou uma tarefa necessária. Neste cenário, a utilização de meta-heurísticas, como os algoritmos genéticos, tem se mostrado uma estratégia eficiente devido à complexidade do problema. Contudo, os trabalhos existentes na literatura falham em lidar com a pressão seletiva imposta a estes algoritmos, no qual o contexto do problema exerce influência sobre as possíveis soluções. Por outro lado, a análise de uma grande quantidade de soluções, aliada ao processamento adicional dos métodos de controle de pressão seletiva, podem acarretar em tempo de processamento elevados.

Inicialmente, foram apresentados conceitos fundamentais necessários para o entendimento do problema sob a ótica da genética e da biologia molecular. Também foram apresentadas as principais técnicas para lidar com os problemas identificados, contextualizando-as no estado da arte. Neste trabalho, então, foi dado foco ao uso de métodos de controle de pressão seletiva, da computação paralela e do processamento de dados em memória, como forma de redução do tempo de execução e aceleração da convergência de um algoritmo genético para resolução de problemas de seleção de SNPs representativos.

5.1 Contribuições

Por meio da avaliação experimental realizada, observou-se ganhos no que diz respeito à qualidade das soluções, assim como ao tempo de processamento, o que permitiu validar a hipótese inicial de que a presença de métodos de controle de pressão seletiva acelera a

convergência do AG para resolução dos problemas por meio do aumento da diversidade das populações, além de possibilitar melhoria dos resultados. Desta forma, a contribuição científica deste trabalho é a proposição de um método paralelo para seleção de SNPs representativos baseado em algoritmos genéticos com controle de pressão seletiva. Na Tabela 6 são apresentadas as principais características de trabalhos encontrados na literatura para resolução do problema de seleção de tag SNPs. Na última linha são apresentadas as características do trabalho desenvolvido. A presença de métodos de controle de pressão seletiva, assim como de processamento paralelo e em memória, são as características que permitem ao AG desenvolvido apresentar resultados superiores.

5.2 Trabalho futuros

Por fim, propõe-se como trabalhos futuros uma maior exploração dos recursos da computação em memória, assim como da utilização dos recursos da computação distribuída.

Tabela 6— Comparação das implementações encontradas na literatura e do trabalho realizado

Proposta	Estratégia	Método Busca	Controle de Pressão Seletiva	Proc. Paralelo	Proc. em Memória
Mahdevar et al. (2010)	Predição	AG	Não	Não	Não
Mouawad e Mansour (2014)	Coeficiente de Associação	AG	Não	Não	Não
Prathibha e Chandran (2016)	Predição	ACO	Parâmetros	Não	Não
Ilhan e Tezel (2013)	Predição	AG	Parâmetros	Não	Não
Li et al. (2014)	Bloco Haplotípico	ACO	Não	Não	Não
Chuang, Huang e Yang (2012)	Coeficiente de Associação	PSO	Parâmetros	Não	Não
Hung et al. (2015)	Bloco Haplotípico	Exaustivo	Não	MR	Não
Este trabalho	Coeficiente de Associação	AG	Parâmetros e Operadores	Spark	Sim

Fonte: Elaborada pelo autor

Espera-se que, assim, sejam obtidos ganhos mais expressivos no que diz respeito ao tempo de processamento, além de permitir a análise de uma maior variedade de soluções do problema de seleção de SNPs representativos. Ressalta-se ainda, que este trabalho realizou a adaptação apenas dos parâmetros mais relevantes dos AGs, de acordo com a literatura científica, como as taxas de mutação e cruzamento, e do método de seleção de indivíduos para reprodução. Assim, propõe-se, também, que sejam examinadas as implicações do uso de estratégias de adaptação em outros parâmetros não considerados.

Referências

ACKERMAN, H.; USEN, S.; MOTT, R.; RICHARDSON, A.; SISAY-JOOF, F.; KATUNDU, P.; TAYLOR, T.; WARD, R.; MOLYNEUX, M.; PINDER, M.; KWIATKOWSKI, D.P. Haplotypic analysis of the TNF locus by association efficiency and entropy. **Genome biology**, v. 4, n. 4, p. R24, 2003.

ALETI, A.; MOSER, I. A systematic literature review of adaptive parameter control methods for evolutionary algorithms. **ACM Computing Surveys (CSUR)**, v. 49, n. 3, p. 56, 2016.

APACHE SPARK. RDD Programming Guide. **The Apache Software Foundation**, 2019. Disponível em <https://goo.gl/qXho2n>. Acesso em 17 ago. 2019.

AVI-ITZHAK, H. I.; SU, X.; DE LA VEGA, F. M. Selection of minimum subsets of single nucleotide polymorphisms to capture haplotype block diversity. In: **Pac Symp Biocomput**. 2003. p. 466.

BÄCK, T. Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In: **Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on**. IEEE, 1994. p. 57-62.

BÄCK, T. Self-adaptation in genetic algorithms. In: **Proceedings of the first european conference on artificial life**. MIT Press, 1992. p. 263-271.

BAFNA, V.; HALLDORSSON, B.V.; SCHWARTZ, R.; CLARK, A.G.; ISTRAIL, S. Haplotypes and informative SNP selection algorithms: don't block out information. In: **Proceedings of the seventh annual international conference on Research in computational molecular biology**. ACM, 2003. p. 19-27.

BAGELEY, J.D. **The behavior of adaptive systems which employ genetic and correlation algorithms**. 1967. 186 f. Doctoral Dissertation (Doctor of Philosophy) – University of Michigan, Bethesda, Maryland. 1967. Disponível em <https://goo.gl/sjBhea>. Acesso em 17 ago. 2019.

BHUIYAN, M. A.; AL HASAN, M. An iterative MapReduce based frequent subgraph mining algorithm. **IEEE Transactions on Knowledge & Data Engineering**, n. 1, p. 1-1, 2015.

BIELZA, C.; DEL POZO, J.A.F.; LARRAÑAGA, P. Parameter control of genetic algorithms by learning and simulation of Bayesian networks—a case study for the optimal ordering of tables. **Journal of Computer Science and Technology**, v. 28, n. 4, p. 720-731, 2013.

BUSH, W. S.; MOORE, J. H. Chapter 11: Genome-Wide Association Studies. PLoS Computational Biology, v. 8, n. 12, p. e1002822, 2012.

CHEN, W.; HUNG, C.; TSAI, S.J.; LIN, Y. Novel and efficient tag SNPs selection algorithms. **Bio-medical materials and engineering**, v. 24, n. 1, p. 1383-1389, 2014.

CHONG, E. K. P; ZAK, S. H. **An introduction to optimization**. New Jersey, John Wiley & Sons, 2013.

CHUANG, L.; HUANG, W.; YANG, C. An improved particle swarm optimization for tag single nucleotide polymorphism selection. In: **Proceedings of the International Multi Conference of Engineers & Computer Scientists (IMECS 2012)**. Hong Kong: IAENG, 2012. p. 13-18.

DAS, S.; MULLICK, S.S.; SUGANTHAN, P.N. Recent advances in differential evolution–an updated survey. **Swarm and Evolutionary Computation**, v. 27, p. 1-30, 2016.

DE GIOVANNI, L.; MASSI, G.; PEZZELLA, F. An adaptive genetic algorithm for large-size open stack problems. **International Journal of Production Research**, v. 51, n. 3, p. 682-697, 2013.

DEVLIN, B.; RISCH, N. A comparison of linkage disequilibrium measures for fine-scale mapping. **Genomics**, v. 29, n. 2, p. 311-322, 1995.

DI MARIO, E.; MARTINOLI, A. Distributed particle swarm optimization for limited-time adaptation with real robots. **Robotica**, v. 32, n. 2, p. 193-208, 2014.

EIBEN, A.E; HINTERDING, R.; MICHALEWICZ, Z. Parameter control in evolutionary algorithms. In: **Parameter setting in evolutionary algorithms. Studies in Computational Intelligence, vol 54.** Springer, Berlin, Heidelberg, 2007. p. 19-46.

ESLAMI, M.; VAHIDI, J.; ASKARZADEH, M. Designing and Implementing a Distributed Genetic Algorithm for Optimizing Work Modes in Wireless Sensor Network. **Journal of mathematics and computer Science**, v. 11, p. 291-299, 2014.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI magazine**, v. 17, n. 3, p. 37, 1996.

FOSTER, I. **Designing and building parallel programs**. Reading: Addison Wesley Publishing Company, 1995.

GAIFANG, D.; XUELIANG, F.; HONGHUI, L.; PENGFEI, X. Cooperative ant colony-genetic algorithm based on spark. **Computers & Electrical Engineering**, v. 60, p. 66-75, 2017.

GENDREAU, M.; HERTZ, A.; LAPORTE, G. A tabu search heuristic for the vehicle routing problem. **Management science**, v. 40, n. 10, p. 1276-1290, 1994.

GIGER, M.; KELLER, D.; ERMANNI, P. AORCEA–An adaptive operator rate controlled evolutionary algorithm. **Computers & Structures**, v. 85, n. 19-20, p. 1547-1561, 2007.

GOLDBERG, D.E. **Genetic algorithms in search, optimization, and machine learning**. Addison-Wesley Longman Publishing Co. Inc. Boston, MA, 1989.

GOMES, A.; ANTUNES, C.H.; MARTINS, A.G. Design of an adaptive mutation operator in an electrical load management case study. **Computers & Operations Research**, v. 35, n. 9, p. 2925-2936, 2008.

GONG, Y; CHEN, W.; ZHAN, Z.; ZHANG, J.; LI, Y.; ZHANG, Q.; LI, J. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. **Applied Soft Computing**, v. 34, p. 286-300, 2015.

GRAMA, A.; KUMAR, V.; GUPTA, A.; KARYPIS, G. **Introduction to parallel computing**. Pearson Education, 2003.

GRAY, I. C.; CAMPBELL, D. A.; SPURR, N. K. Single nucleotide polymorphisms as tools in human genetics. **Human molecular genetics**, v. 9, n. 16, p. 2403-2408, 2000.

GROŞAN, C. Multiobjective adaptive representation evolutionary algorithm (marea)-a new evolutionary algorithm for multiobjective optimization. In: **Applied Soft Computing Technologies: The Challenge of Complexity**. Springer, Berlin, Heidelberg, 2006. p. 113-121.

GUO, X. Cloud Computing and Parallel Strategy for Bioinformatics: A Review. 2012. Disponível em <https://goo.gl/rERDTL>. Acesso em 17 ago. 2019.

HAMPE, J.; SCHREIBER, S.; KRAWCZAK, M. Entropy-based SNP selection for genetic association studies. **Human genetics**, v. 114, n. 1, p. 36-43, 2003.

HOLLAND, J. H. Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. **Ann Arbor, MI: University of Michigan Press**, p. 439-444, 1975.

HU, C.; REN. G.; LIU, C.; LI, M.; JIE, W. A Spark-based genetic algorithm for sensor placement in large scale drinking water distribution systems. **Cluster Computing**, v. 20, n. 2, p. 1089-1099, 2017.

HUNG, C., CHEN, W.P.; HUA, G.J.; ZHENG. H.; TSAI, S.J. Cloud computing-based TagSNP selection algorithm for human genome data. **International journal of molecular sciences**, v. 16, n. 1, p. 1096-1110, 2015.

ILHAN, I.; TEZEL, G. A genetic algorithm–support vector machine method with parameter optimization for selecting the tag SNPs. **Journal of biomedical informatics**, v. 46, n. 2, p. 328-340, 2013.

ILIE, S.; BĂDICĂ, C. Multi-agent approach to distributed ant colony optimization. **Science of Computer Programming**, v. 78, n. 6, p. 762-774, 2013.

JOHNSON, G.C.L.; ESPOSITO, L.; BARRATT, B.J.; SMITH, A.N.; HEWARD, J.; GENOVA, G.; UEDA, H.; CORDELL, H.J.; EAVES, I.A.; DUDBRIDGE, F.; TWELLS, R.C.J.; PAYNE, F.; HUGHES, W.; NUTLAND, S.; STEVENS, H.; CARR, P.; TUOMILEHTO-WOLF, E.; TUOMILEHTO, J.; GOUGH, S.C.L.; CLAYTON, D.G.; TODD, J.A. Haplotype tagging for the identification of common disease genes. **Nature genetics**, v. 29, n. 2, p. 233-237, 2001.

KAISLER, S.; ARMOUR, F.; ESPINOSA, J.A.; MONEY, W. Big Data: Issues and Challenges Moving Forward. **2013 46th Hawaii International Conference on System Sciences**, p. 995–1004, jan. 2013.

KARABOGA, D.; ASLAN, S. A new emigrant creation strategy for parallel artificial bee colony algorithm. In: **Electrical and Electronics Engineering (ELECO), 2015 9th International Conference on**. IEEE, 2015. p. 689-694.

KARAFOTIAS, G.; EIBEN, A.E.; HOOGENDOORN, M. Generic parameter control with reinforcement learning. In: **Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation**. ACM, 2014. p. 1319-1326.

KE, X.; CARDON, L. R. Efficient selective screening of haplotype tag SNPs. **Bioinformatics**, v. 19, n. 2, p. 287-288, 2003.

KENNEDY, R. J.; EBERHART R. C. Particle swarm optimization. In: **Proceedings of IEEE International Conference on Neural Networks IV**, 1995.

KRAMER, O. **Genetic algorithm essentials**. Studies in Computational Intelligence, v. 679. Springer, 2017. 94p.

LI, X.; CAO, Z.; LI, X.; CHEN, J.; LI, G. A Tag SNP Selection Method Based on Haplotype Recognition. **Journal of Computational and Theoretical Nanoscience**, v. 11, n. 12, p. 2495-2498, 2014.

LI, Z.; CAI, L.; CHEN, M.; ZENG, L. An informative SNP selection method based on artificial neural network. **Metallurgical & Mining Industry**, n. 9, 2015.

LIAO, B.; LI, X.; ZHU, W.; SHULIN WANG, R.L. Multiple ant colony algorithm method for selecting tag SNPs. **Journal of biomedical informatics**, v. 45, n. 5, p. 931-937, 2012.

LIAO, B.; WANG. X.; ZHU, W.; LI, X.; CAI, L.; CHEN, H. New multilocus linkage disequilibrium measure for tag SNP selection. **Journal of Bioinformatics and Computational Biology**, v. 15, n. 01, p. 1750001, 2017.

LIN, W.Y.; LEE, W.Y.; HONG, T.P. Adapting crossover and mutation rates in genetic algorithms. **Journal of Information Science and Engineering**, v. 19, n. 5, p. 889-903, 2003.

LINDEN, R. **Algoritmos Genéticos**. 2. ed. Rio de Janeiro; Brasport, 2008. v. 1. 428p.

LIU, Y.; XU, L.; LI, M. The parallelization of back propagation neural network in MapReduce and Spark. **International journal of parallel programming**, v. 45, n. 4, p. 760-779, 2017.

LODISH, H.; BERK, A.; KAISER. C.A.; KRIEGER, M.; BRETSCHER, A. **Biologia celular e molecular**. Tradução de Bizarro et al. 7. ed. Porto Alegre; Artmed, 2014.

MAHDEVAR, G.; ZAHIRI, J.; SADEGHI, M. Tag SNP selection via a genetic algorithm. **Journal of Biomedical Informatics**, v. 43, n. 5, p. 800-804, 2010.

MAILLO, J.; RAMÍREZ, S.; TRIGUERO, I.; HERRERA, F. kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. **Knowledge-Based Systems**, v. 117, p. 3-15, 2017.

MANIEZZO, A.C.M.D.V. Distributed optimization by ant colonies. In: **Toward a practice of autonomous systems: proceedings of the First European Conference on Artificial Life**. Mit Press, 1992. p. 134.

MARQUES, V.H.A. **Adaptação de parâmetros em meta-heurísticas com sistemas nebulosos genéticos**. 2011. 127 f. Dissertação (Mestre em Engenharia Elétrica) – Universidade Estadual de Campinas, Campinas, 2011. Disponível em <http://repositorio.unicamp.br/handle/REPOSIP/259064>. Acesso em 17 ago. 2019.

MAZZUCCO, J. **Uma abordagem híbrida do problema da programação da produção através dos algoritmos simulated annealing e genético**. 1999. 125 f. Tese (Doutor em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis, 1999. Disponível em <https://goo.gl/Hd7UQZ>. Acesso em 17 ago. 2019.

MCKENNA, A.; HANNA, M. BANKS, E.; SIVACHENKO, A.; CIBULSKIS, K.; KERNYTSKY, A.; GARIMELLA, K.; ALTSHULER, D.; GABRIEL, S.; DALY, M.; DEPRISTO, M.A. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. **Genome research**, 2010.

MERCER, R.E.; SAMPSON, J.R. Adaptive search using a reproductive meta-plan. **Kybernetes**, v. 7, n. 3, p. 215-228, 1978.

MIKI, M.; HIROYASU, T.; KANEKO, M.; HATANAKA. K. A parallel genetic algorithm with distributed environment scheme. In: **Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on**. IEEE, 1999. p. 695-700.

MONTERO, E.; RIFF, M.C. On-the-fly calibrating strategies for evolutionary algorithms. **Information Sciences**, v. 181, n. 3, p. 552-566, 2011.

MOUAWAD, A. E.; MANSOUR, N. Multi-marker-LD based genetic algorithm for tag SNP selection. **Interdisciplinary sciences, computational life sciences**, v. 6, n. 4, p. 303, 2014.

NOVAKOVIĆ, J. Toward optimal feature selection using ranking methods and classification algorithms. **Yugoslav Journal of Operations Research**, v. 21, n. 1, 2016.

PACHECO, M.A.C. Algoritmos genéticos: princípios e aplicações. **ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro.** p. 28, 1999.

PASTERNAK, J. J. **Genética molecular humana: mecanismos das doenças hereditárias**. Tradução de Ida Cristina Gubert. 1. ed. Barueri. Manole Ltda, 2002.

PESSOA, F. **Alguma prosa**. Editora Nova Aguilar, 1976. 247p.

PHUONG, T.M.; LIN, Z.; ALTMAN, R.B. Choosing SNPs using feature selection. In: **Computational Systems Bioinformatics Conference, 2005. Proceedings. 2005 IEEE**. IEEE, 2005. p. 301-309.

PRATHIBHA, P. H.; CHANDRAN, C. P. Feature selection for mining SNP from Leukaemia cancer using Genetic Algorithm with BCO. In: **Data Mining and Advanced Computing (SAPIENCE), International Conference on**. IEEE, 2016. p. 57-63.

QI, R.; WANG, Z.; LI, S. A parallel genetic algorithm based on spark for pairwise test suite generation. **Journal of Computer Science and Technology**, v. 31, n. 2, p. 417-427, 2016.

ROSENBERG, R.S. Simulation of genetic populations with biochemical properties: II. selection of crossover probabilities. **Mathematical Biosciences**, v. 8, n. 1-2, p. 1-37, 1970.

SCHAFFER, J.D.; MORISHIMA, A. An adaptive crossover distribution mechanism for genetic algorithms. In: **Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms**. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc, 1987. p. 36-40.

SCHOTT, J.R. **Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization**. 1995. X f. Thesis (Master of Science in Aeronautics and Astronautics) – Massachusetts Institute of Technology, 1995. Disponível em <http://www.dtic.mil/dtic/tr/fulltext/u2/a296310.pdf>. Acesso em 17 ago. 2019.

SETHI, K. K.; RAMESH, D. HFIM: a Spark-based hybrid frequent itemset mining algorithm for big data processing. **The Journal of Supercomputing**, v. 73, n. 8, p. 3652-3668, 2017.

SHERRY, S.T.; WARD, M.H.; KHOLODOV, M.; BAKER, J.; PHAN. L.; SMIGIELSKI, E.M.; SIROTKIN, K. dbSNP: the NCBI database of genetic variation. **Nucleic acids research**, v. 29, n. 1, p. 308-311, 2001.

SIVANANDAM, S. N.; DEEPA, S. N. **Introduction to genetic algorithms**. Springer, 2008. v. 1. 543p.

TABAKHI, S.; MORADI, P.; AKHLAGHIAN, F. An unsupervised feature selection algorithm based on ant colony optimization. **Engineering Applications of Artificial Intelligence**, v. 32, p. 112-123, 2014.

TRIGUERO, I.; PERALTA, D.; BACARDIT, J.; GARCÍA, S.; HERRETA, F. MRPR: A MapReduce solution for prototype reduction in big data classification. **Neurocomputing**, v. 150, p. 331-345, 2015.

TROBEC, R.; VAJTERŠIC, M.; ZINTERHOF, P. **Parallel computing: numerics, applications, and trends**. Springer Science & Business Media, 2009. 531p.

ÜSTÜNKAR, G.; ÖZÖĞÜR-AKYÜZ, S.; WEBER, G.W.; FRIEDRICH, C.M.; SON, Y.A. Selection of representative SNP sets for genome-wide association studies: a metaheuristic approach. **Optimization Letters**, v. 6, n. 6, p. 1207-1218, 2012.

WANG, F.; LI, J.; LIU, S.; ZHAO, X.; ZHANG, D.; TIAN, Y. An improved adaptive genetic algorithm for image segmentation and vision alignment used in microelectronic bonding. **IEEE/ASME Transactions on Mechatronics**, v. 19, n. 3, 2014.

WANG, S.; HE, S.; ZHU, X. Tagging SNP-set selection with maximum information based on linkage disequilibrium structure in genome-wide association studies. **Bioinformatics**, p. btx151, 2017.

WATSON, J. D.; CRICK, F. HC. The structure of DNA. In: **Cold Spring Harbor symposia on quantitative biology**. Cold Spring Harbor Laboratory Press, 1953. p. 123-131.

WEINBERG, R. **Computer simulation of a living cell: interdisciplinary synergism**. 1970. 170 f. Doctoral Dissertation (Doctor of Philosophy) – University of Michigan, Bethesda, Maryland. 1967. Disponível em <https://goo.gl/hiBcgt>. Acesso em 17 ago. 2019.

ZAHARIA, M, CHOWDHURY, M.; DAS, T.; DAVE, A.; MA, J.; MCCAULEY, M.; FRANKLIN, M.J.; SHENKER, S.; STOICA, I. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In: **Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation**. USENIX Association, 2012. p. 2-2.

ZEGGINI, E.; MORRIS, A. **Analysis of complex disease association studies: a practical guide**. Academic Press, 2010.

Glossário

Alelos: Formas alternativas de um gene. Para polimorfismos de sequência os alelos se referem ao nucleotídeo específico (Adenina, Timina, Guanina, Citosina) encontrado em uma determinada posição no cromossomo. Para polimorfismos de inserção-deleção, dois alelos são possíveis, o alelo maior (L) e o alelo menor (S). Ver também Polimorfismo.

Base: ver Nucleotídeo.

Bloco haplotípico: região do cromossomo onde há vários genes, mas não há recombinação, ou seja, troca de genes na reprodução. Um bloco haplotípico é herdado inalterado por muitas gerações até que ocorra uma mutação.

Cromossomo: estrutura encontrada nos núcleo das células constituído de DNA condensado em associação com proteínas. O genoma humano consiste de um conjunto de 23 cromossomos. Cada um de nós herdou um conjunto do pai e outro da mãe. Cada cromossomo contém uma única molécula de DNA – segmentos deste DNA são os genes.

DNA: Ácido desoxirribonucleico. É a molécula que armazena a informação genética e consiste de duas cadeias de nucleotídeos unidas pela interação das bases complementares Adenina e Timina, e Citosina e Timina.

Genoma: O DNA em um conjunto haplóide (23 cromossomos) humano. Uma célula humana contém dois genomas, um paterno e um materno.

Haplótipo: conjunto de alelos presentes em uma determinada região do cromossomo que não foram separados durante o processo de recombinação. Indivíduos com haplótipos similares têm relações genealógicas.

Nucleotídeo: Subunidades informacionais que quando unidas em cadeia constituem o DNA. Existem quatro subunidades diferentes - Adenina (A), Guanina (G), Timina (T) e Citosina (C). Os nucleotídeos também são chamados de bases.

Polimorfismo: Uma diferença na sequência de DNA presente em mais de 1% dos indivíduos de uma população.

SNP: Polimorfismo caracterizado pela troca de uma base por outra na sequência do DNA.

TERMO DE REPRODUÇÃO XEROGRÁFICA

Autorizo a reprodução xerográfica do presente Trabalho de Conclusão, na íntegra ou em partes, para fins de pesquisa.

São José do Rio Preto, 04 de setembro de 2019.

William Tenório