

---

# Algoritmo Evolutivo aplicado ao Problema do Percurso do Cavalo.

---

Murielly Oliveira Nascimento



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE COMPUTAÇÃO  
BACHALERADO EM SISTEMAS DE INFORMAÇÃO

Uberlândia  
2023



**Murielly Oliveira Nascimento**

**Algoritmo Evolutivo aplicado ao Problema do  
Percurso do Cavalo.**

Monografia apresentada à Faculdade de  
Computação da Universidade Federal de  
Uberlândia, Minas Gerais, como requisito  
exigido parcial à obtenção do grau de Bacha-  
rel em Sistemas de Informação.

Orientadora: Christiane Regina Soares Brasil.

Área de concentração: Algoritmos Evolutivos

Uberlândia

2023



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE COMPUTAÇÃO  
PROGRAMA DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Computação a aceitação da dissertação intitulada "**Algoritmo Evolutivo aplicado ao Problema do Percorso do Cavalo**" por **Murielly Oliveira Nascimento** como parte dos requisitos exigidos para a obtenção do título de **Bacharel em Sistemas de Informação**.

Uberlândia, \_\_\_de\_\_\_\_\_ de\_\_\_\_\_

Orientador: \_\_\_\_\_  
Prof.<sup>a</sup>. Dr.<sup>a</sup>. Christiane Regina Soares Brasil  
Universidade Federal de Uberlândia

Banca Examinadora:

\_\_\_\_\_  
Prof. Dr. Wendel Alexandre Xavier  
de Melo  
Universidade Federal de Uberlândia

\_\_\_\_\_  
Prof. Dr. Paulo Henrique Ribeiro  
Gabriel  
Universidade Federal de Uberlândia



*A Deus que sempre esteve ao meu lado  
e nunca largou a minha mão.*





---

# Agradecimentos

Primeiramente, meus agradecimentos aos responsáveis por criar o curso de Sistemas de Informação em 2009. É graças aos seus esforços que eu e meus colegas pudemos nos formar como profissionais da área de tecnologia.

Agradeço, também à Universidade Federal de Uberlândia pela oportunidade de acesso a educação superior e sem a qual a Faculdade de Computação não existiria.

À minha orientadora, Prof<sup>a</sup>. Christiane Regina Soares Brasil, sem a sua orientação este trabalho não seria possível. Sou uma pesquisadora melhor graças aos seus conselhos.

Ao grupo PET-SI, programa responsável por me apresentar a Iniciação Científica, da qual este trabalho é fruto. E ao Prof. Wendel Alexandre Xavier de Melo, por inspirar todos os membros do grupo PET a continuarem seus estudos e seguirem carreira na área de acadêmica.

À toda equipe docente da FACOM, não seria possível nomear as contribuições individuais que cada um teve na minha formação, deixo, no lugar, meu profundo agradecimento por cada conselho, conhecimento e motivação passado. Segundo o provérbio africano, é preciso uma aldeia inteira para criar uma criança, pois a profissional que me tornei é produto de todos os seus esforços.

Ao meu pai, por sempre colocar a minha educação em primeiro lugar e não medir esforços para que o meu futuro fosse melhor.

À minha mãe por ouvir, em detalhes, cada erro de código e artigo científico que eu estivesse lendo. E por sempre me apoiar em meus estudos.

Ao meu irmão, sem você eu ainda estaria cursando Relações Internacionais. Obrigado por me dar coragem e forças para trocar de cursos e enfrentar uma área na qual era completamente leiga.

A Deus que em tudo me guiou.



*“Através de muitos perigos, labutas e armadilhas,  
Eu já me aproximava  
A esta graça que me trouxe a salvo até agora,  
E esta graça me levará para casa.”  
(John Newton - Amazing Grace)*



---

# Resumo

A Computação Bioinspirada é uma área de pesquisa focada no desenvolvimento de técnicas inspiradas em fenômenos da natureza para a solução de problemas intratáveis em tempo polinomial (NP). Neste trabalho, o Algoritmo Evolutivo (AE) é implementado para a solução do Problema do Percurso do Cavalo (PPC). Este algoritmo é fortemente baseado na Teoria Evolucionista de Darwin, em especial, a Seleção Natural. O PPC, por sua vez, é um problema combinatório amplamente utilizado como base para o aprimoramento ou desenvolvimento de algoritmos e para a solução de problemas reais, como a criptografia de imagens. Leonhard Euler foi o primeiro a estudá-lo formalmente. O PPC consiste em encontrar uma sequência de movimentos — realizados pela peça de xadrez correspondente ao cavalo — que percorra todo o tabuleiro sem visitar uma casa mais de uma vez.

Portanto, este trabalho teve como objetivo implementar o AE para a solução do PPC, melhorando os resultados encontrados na literatura, por meio da implementação de um novo operador de seleção e mutação. Sendo o primeiro, baseado na exploração de um campo de busca maior através do cruzamento de pais dissimilares; e o segundo na troca de genes (casas do percurso) por vizinhos válidos. Os experimentos mostraram que o AE implementado foi capaz de resolver o PPC para tabuleiros  $n \times n$ , com  $5 \leq n \leq 20$ .

**Palavras-chave:** Algoritmos Evolutivos. Métodos de Otimização. Problema do Percurso do Cavalo.



---

# Abstract

Bioinspired Computing is an area of research focused on developing techniques inspired by natural phenomena to solve intractable problems in polynomial time (NP). In this work, *Evolutionary Algorithm* (EA) is implemented to solve the Knight's Tour. This algorithm is strongly based on Darwin's Evolutionary Theory, in particular, Natural Selection. The Knight's Tour, in turn, is a combinatorial problem widely used as a basis for improving or developing algorithms and for solving real problems, such as image encryption. Leonhard Euler was the first to formally study it. The Knight's Tour consists of finding a sequence of moves — made by the chess piece corresponding to the knight — that travels across the entire board without visiting a square more than once.

Therefore, this work aimed to implement EA for the Knight's Tour solution, improving the results found in the literature, through the implementation of a new selection and mutation operator. The first is based on the exploration of a larger search field through the crossing of dissimilar parents, and the second is the exchange of genes (pathways) for valid neighbours. The experiments showed that the implemented EA was able to solve Knight's Tour for  $n \times n$  boards, with  $5 \leq n \leq 20$ .

**Keywords:** Evolutionary Algorithms. Optimization Methods. Knight's Tour.





---

## Lista de ilustrações

Figura 1 – Número de recursos indexados pela <i>Library Search</i> sobre EA. . . . .	27
Figura 2 – Seleção Natural. . . . .	31
Figura 3 – Cadeia de DNA. . . . .	32
Figura 4 – Operador de Mutação. . . . .	34
Figura 5 – Recombinação 1-ponto e n-pontos. . . . .	35
Figura 6 – Recombinação Uniforme. . . . .	35
Figura 7 – Fluxograma de AEs. . . . .	36
Figura 8 – Métodos de Seleção. . . . .	37
Figura 9 – Amostragem Universal Estocástica. . . . .	37
Figura 10 – Seleção Elitista. . . . .	38
Figura 11 – Percurso do Cavalo. . . . .	38
Figura 12 – Percurso do Cavalo por Euler. . . . .	39
Figura 13 – Situações características do PPC. . . . .	40
Figura 14 – Ciclo Hamiltoniano. . . . .	41
Figura 15 – Representação do PPC. . . . .	46
Figura 16 – Função de adaptação no PPC. . . . .	47
Figura 17 – Recombinação uniforme no PPC. . . . .	49
Figura 18 – Mutação 1-ponto no PPC. . . . .	50



---

## Lista de algoritmos

1	Pseudocódigo do Algoritmo Evolutivo. . . . .	32
2	Pseudocódigo do AE aplicado ao PPC. . . . .	46
3	Pseudocódigo da Torneio dos Dissimilares. . . . .	49
4	Pseudocódigo da Mutação dos Vizinhos. . . . .	50



---

## Lista de tabelas

Tabela 1 – Tabuleiro (5×5); população (60); número máximo de gerações (180); mutação 1-ponto (15%); elitismo (10%); torneio (3). . . . .	53
Tabela 2 – Medidas de Avaliação. . . . .	54
Tabela 3 – Tabuleiro (8×8); população (80); número máximo de gerações (200); mutação 1-ponto (15%); elitismo (10%); torneio (3). . . . .	54
Tabela 4 – Medidas de Avaliação. . . . .	55
Tabela 5 – Tabuleiro (10×10); população (200); número máximo de gerações (400); mutação 1-ponto (15%); elitismo (10%); torneio (3). . . . .	55
Tabela 6 – Medidas de Avaliação. . . . .	55
Tabela 7 – Tabuleiro (10×10); população (200); número máximo de gerações (400); mutação 1-ponto (15%); elitismo (10%); torneio (3). . . . .	56
Tabela 8 – Medidas de Avaliação. . . . .	56
Tabela 9 – Tabuleiro (16×16); população (400); número máximo de gerações (1600); mutação 1-ponto (15%); elitismo (10%); torneio (3). . . . .	57
Tabela 10 – Medidas de Avaliação. . . . .	57
Tabela 11 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação 1-ponto (15%); elitismo (10%); torneio (3). . . . .	58
Tabela 12 – Medidas de Avaliação. . . . .	58
Tabela 13 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação dos vizinhos (15%); elitismo (10%); torneio (3). . . . .	59
Tabela 14 – Medidas de Avaliação. . . . .	59
Tabela 15 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação 1-ponto (15%); elitismo (10%); torneio dos dissi- milares (3). . . . .	60
Tabela 16 – Medidas de Avaliação. . . . .	60
Tabela 17 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação dos vizinhos (15%); elitismo (10%); torneio dos dis- similares (3). . . . .	61

Tabela 18 – Medidas de Avaliação. . . . .	61
Tabela 19 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação dos vizinhos (15%); elitismo (10%); torneio (3). . . .	62
Tabela 20 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação dos vizinhos (15%); elitismo (10%); torneio dos dissimilares (3). . . . .	62
Tabela 21 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação dos vizinhos (15%); elitismo (10%); torneio dos dissimilares (3). . . . .	63
Tabela 22 – Medidas de Avaliação. . . . .	63

---

# Lista de siglas

**AE** Algoritmo Evolutivo

**AEMO** Algoritmo Evolutivo Multiobjetivo

**ABC** Algoritmo de Colônia de Abelhas

**COIN** *Coincidence Algorithm*

**EA** *Evolutionary Algorithm*

**ORSA** Sociedade Americana de Pesquisa Operacional

**PPC** Problema do Percurso do Cavalo

**PSP** Problema de Predição de Estruturas de Proteínas

**PV** Fotovoltaica

**SCI** *Science Citation Index*

**SOBRAPO** Sociedade Brasileira de Pesquisa Operacional





---

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>25</b>
1.1	Objetivo(s) . . . . .	26
1.2	Justificativa(s) . . . . .	26
1.3	Estrutura do texto . . . . .	27
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>29</b>
2.1	Conceitos de Otimização . . . . .	29
2.1.1	Classificação de Problemas de Otimização por Complexidade . . . . .	30
2.2	Algoritmo Evolutivo . . . . .	30
2.2.1	Conceitos Biológicos . . . . .	32
2.2.2	Operadores de Reprodução . . . . .	33
2.2.3	Métodos de Seleção . . . . .	35
2.3	Problema do Percurso do Cavalo . . . . .	38
2.4	Exemplos de aplicações do AE . . . . .	41
2.5	Considerações Finais . . . . .	42
<b>3</b>	<b>IMPLEMENTAÇÃO DO ALGORITMO EVOLUTIVO</b> . . . . .	<b>45</b>
3.1	Desenvolvimento do Algoritmo Evolutivo . . . . .	45
3.2	Representação . . . . .	45
3.3	Inicialização . . . . .	46
3.4	Funções de Avaliação . . . . .	47
3.5	Operadores de Seleção . . . . .	48
3.6	Recombinação . . . . .	49
3.7	Operadores de Mutação . . . . .	50
3.8	Considerações Finais . . . . .	50
<b>4</b>	<b>EXPERIMENTOS</b> . . . . .	<b>53</b>
4.1	Tabuleiro 5x5 . . . . .	53

4.2	Tabuleiro 8x8 . . . . .	54
4.3	Tabuleiro 10x10 . . . . .	55
4.4	Tabuleiro 16x16 . . . . .	56
4.5	Tabuleiro 20x20 . . . . .	57
4.6	Mutação dos Vizinhos . . . . .	58
4.7	Torneio dos Dissimilares . . . . .	59
4.8	Mutação dos Vizinhos e Torneio dos Dissimilares . . . . .	60
4.8.1	Comparação entre o Torneio e o Torneio dos Dissimilares . . . . .	61
4.9	Inicialização Central . . . . .	62
4.10	Considerações Finais . . . . .	63
5	CONCLUSÃO . . . . .	65
	REFERÊNCIAS . . . . .	67

---

## Introdução

A Computação Bioinspirada é uma área de pesquisa que se baseia em conceitos biológicos para o desenvolvimento de algoritmos, sendo geralmente aplicados na solução de problemas de alta complexidade. Estes problemas podem ser resolvidos por meio de métodos determinísticos, porém, à medida que o tamanho da instância cresce, o custo computacional pode aumentar drasticamente. Neste sentido, os Algoritmos Evolutivos (AEs) são técnicas de otimização que têm se destacado.

AEs se baseiam fortemente nas teorias evolucionistas de Darwin, por exemplo, a Seleção Natural, para otimização de problemas. Os primeiros trabalhos sobre eles datam da década de 1960, quando (HOLLAND, 1992) começou suas pesquisas sobre o assunto e publicou posteriormente o livro *Adaptation in Natural and Artificial Systems*, definindo as bases para a Computação Evolutiva. (GOLDBERG, 1987), aluno de Holland, foi o primeiro a obter sucesso na aplicação industrial de AEs.

Esses algoritmos oferecem a vantagem de trabalharem com múltiplas soluções para um problema, diferente de métodos determinísticos que buscam uma única solução ótima. Ainda que haja diversas abordagens de AEs, o princípio básico é o mesmo: dado uma população de indivíduos (i.e. um conjunto de soluções), pressões do ambiente desencadeiam a seleção natural, em que indivíduos mais aptos são privilegiados (neste caso, as soluções mais adequadas) que constituirão uma nova população. O processo se repete até que um número de iterações seja atingido ou a solução mais qualificada seja encontrada.

Neste trabalho de Conclusão de Curso, o Algoritmo Evolutivo será aplicado a um famoso problema combinatório denominado o Problema do Percorso do Cavalo – PPC, em inglês *Knight's Tour*, no tabuleiro de xadrez. Esse problema pode ser definido da seguinte forma:

**Dado um tabuleiro  $n \times m$  qualquer, determine uma sequência legal de movimentos do cavalo de modo que esta peça passe por todas as casas uma única vez, a partir de qualquer casa do tabuleiro.**

A implementação do Algoritmo Evolutivo foi baseada nos trabalhos encontrados na literatura, com a inicialização da população seguindo a implementação de (PINTO, 2013), a avaliação de (GORDON; SLOCUM, 2004) e (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007), a seleção dos indivíduos pelo torneio, o cruzamento pela recombinação uniforme e a mutação 1-ponto. Em concordância com os objetivos propostos, os operadores torneio dos dissimilares, mutação dos vizinhos e inicialização central foram desenvolvidos, melhorando os resultados encontrados em um tabuleiro de dimensões 20x20.

## 1.1 Objetivo(s)

O objetivo principal deste trabalho foi aplicar o Algoritmo Evolutivo ao Problema do Percurso do Cavalo, almejando alcançar dimensões do tabuleiro  $n \times n$ , sendo  $5 \leq n \leq 20$ , uma vez que poucos trabalhos foram encontrados na literatura com AE aplicado ao PPC para tabuleiros maiores que 8.

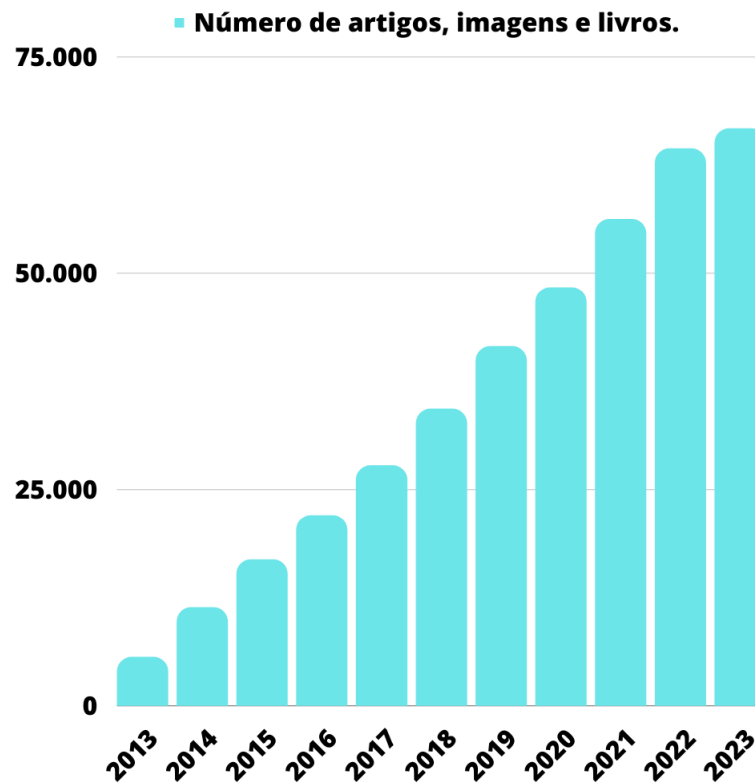
O objetivo secundário foi implementar adaptações nos operadores de seleção (torneio) e mutação, a fim de melhorar os resultados que foram obtidos por meio dos operadores tradicionais do AE.

## 1.2 Justificativa(s)

A Computação Bioinspirada ganhou destaque pelo fato de desenvolver algoritmos capazes de encontrar soluções adequadas para problemas complexos, ainda não resolvidos por outras técnicas computacionais. Além disso, o Algoritmo Evolutivo é uma técnica que se destaca por sua implementação, utilizando instruções básicas de programação, podendo adaptar-se com notável versatilidade para problemas das mais diversas áreas de pesquisa (GABRIEL; DELBEM, 2008).

De acordo, com o *Science Citation Index* (SCI) — um índice de citação criado por Eugene Garfield em 1964 e, atualmente, detido pela *Clarivate Analytics* — o tópico AE, em inglês Evolutionary Algorithm, foi mencionado em 112 artigos até o ano de 2023. A ferramenta *Library Search* da Universidade de Manchester, por sua vez, indexou 66.717 artigos, livros e imagens, com menções sobre o tópico, desde 2013. A Figura 1 ilustra o número de recursos indexados pela *Library Search* sobre AE por ano.

Vale destacar que, no âmbito deste Trabalho de Conclusão de Curso, a maioria dos trabalhos encontrados apresenta o AE aplicado a tabuleiros de dimensão até  $8 \times 8$ . Deste modo, neste trabalho almejou-se alcançar dimensões do tabuleiro  $n \times n$ , sendo  $5 \leq n \leq 20$ , a partir de variações de operadores genéticos, inclusive propondo novas abordagens de mutação e seleção, que posteriormente poderão ser aplicadas a outras áreas.

Figura 1 – Número de recursos indexados pela *Library Search* sobre EA.

### 1.3 Estrutura do texto

Este trabalho foi organizado da seguinte forma:

O Capítulo 2 apresenta a fundamentação teórica, sendo dividido em: Conceitos de Otimização, Algoritmo Evolutivo, Problema do Percuro do Cavalo, Trabalhos Correlatos e Considerações Finais. A primeira seção introduz o leitor a problemas de otimização, característicos da área de computação, e a sua classificação por complexidade (P, NP, NP Completo).

Na segunda seção são explicadas as bases biológicas sobre as quais os AEs se sustentam, bem como a sua aplicação na solução de problemas reais e científicos como o PPC; o qual é apresentado na terceira, juntamente com o referencial teórico usado para seu embasamento nesta pesquisa. A seção de trabalhos correlatos descreve várias pesquisas do AE em PPC desenvolvidas para resolução de diversos problemas. Por fim, a seção de considerações finais apresenta um breve resumo deste capítulo.

O Capítulo 3 - Implementação do Algoritmo Evolutivo — é dividido nas seções: Representação, Inicialização, Funções de Avaliação, Operadores de Seleção, Recombinação e Operadores de Mutação. A linguagem de programação, sistema operacional e características da máquina usada para os testes são descritos, bem como os novos operadores de

seleção e mutação, propostos com o intuito de melhorar os resultados dos experimentos.

O Capítulo 4 - Experimentos — apresenta os testes e resultados obtidos com o AE implementado neste trabalho. A comparação entre diferentes técnicas também é feita.

O Capítulo 5 - Conclusões — descreve as conclusões, as limitações encontradas e os trabalhos futuros sugeridos.

---

## Fundamentação Teórica

O Algoritmo Evolutivo é uma técnica da Computação Bioinspirada, caracterizado pela habilidade de evoluir a cada geração dada uma população de indivíduos iniciais. Ele é largamente aplicado na solução de problemas combinatórios como o Problema do Percorso do Cavalo, o qual também possui suas aplicações em problemas como a criptografia de imagens. A seguir, a fundamentação teórica da área de otimização, a qual o PPC pertence, e os conceitos fundamentais do AE são descritos.

### 2.1 Conceitos de Otimização

A otimização é uma área da computação que trata de diversos problemas matemáticos. Ela visa encontrar o mínimo ou máximo de uma dada função com variáveis de decisão e sujeito a restrições (TAHA, 2006). Esta área de estudo encontra múltiplas aplicações no mundo real, alguns temas pesquisados pela Sociedade Americana de Pesquisa Operacional (ORSA) são: teoria de estoques e filas, reposição de equipamentos, fluxo de redes, entre outras.

Embora não possua raízes definidas, suas origens podem ser traçadas por volta de 1934 com a invenção do radar na Inglaterra. A ORSA foi fundada em 1952 e no Brasil a Sociedade Brasileira de Pesquisa Operacional (SOBRAPO) começou em 1969.

A primeira fase para a resolução de um problema é a sua formulação. É preciso definir os tomadores de decisão, o objetivo, os aspectos sobre o controle dos decisores (variáveis de decisão) e sob quais limitações (restrições), e quais aspectos fogem do controle deles. Feito isto, pode-se construir um modelo que permitirá analisar o problema, indicando relações entre variáveis e dados relevantes.

A terceira fase é a obtenção da solução seguida pelo seu teste e sua implementação. Neste sentido, a função objetivo é uma função matemática que define a qualidade da solução em relação às variáveis de decisão; é um critério de escolha das variáveis de decisão representado por uma função (TAHA, 2006).

### 2.1.1 Classificação de Problemas de Otimização por Complexidade

Nesta Subseção são definidas as principais classes de complexidade para problemas de decisão, relativamente à performance de seus algoritmos com respeito a complexidade de tempo. As definições foram retiradas de (GAREY; JOHNSON, 1990).

A classe P é definida como o conjunto de todos os problemas de decisão resolvíveis por um algoritmo determinístico em tempo polinomial. A sua importância deriva do fato de conter todos os problemas “simples”, isto é, aqueles que são computacionalmente tratáveis.

A classe NP é definida como o conjunto de todos os problemas de decisão resolvíveis por um algoritmo não-determinístico em tempo polinomial, podendo ser vista, informalmente, como a classe dos problemas de decisão para os quais a verificação de uma solução estimada para uma dada entrada satisfaz todos os requerimentos do problema e pode ser verificada rapidamente.

Informalmente, um problema de decisão  $p$  é NP-completo se  $p \in NP$  e, para todo problema de decisão  $p' \in NP$ , tem-se que  $p' \leq p$ . Portanto, os problemas NP-completos são identificados como os problemas mais difíceis em NP.

Por fim, as técnicas conhecidas para provar a NP-completude podem também serem usadas para provar a dificuldade de problemas não restritos à classe NP. Qualquer problema de decisão  $p$ , pertencente ou não à classe NP, que seja redutível a um problema NP-completo, terá a propriedade de não ser resolvido em tempo polinomial, a menos que  $P=NP$ . Neste caso, tal problema  $p$  é chamado de NP-difícil, já que ele apresenta dificuldade não menor que qualquer problema NP-completo.

## 2.2 Algoritmo Evolutivo

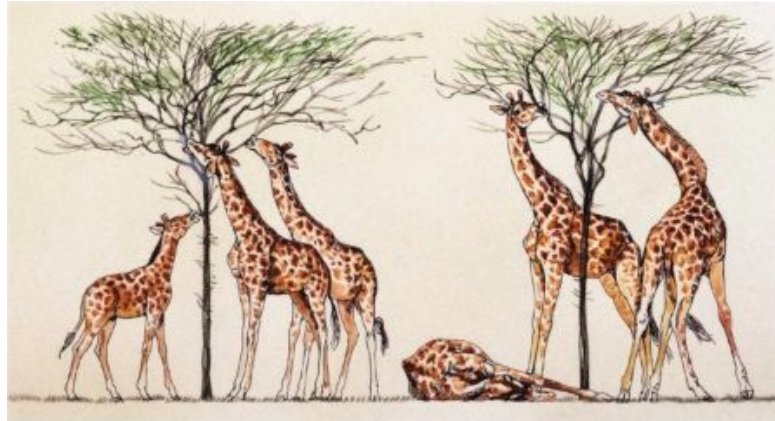
Em 24 de novembro de 1859, Charles Darwin publicou sua obra mais famosa, *A Origem das Espécies*. Nela Darwin discute que as mudanças em grupos de seres vivos ao longo do tempo são resultados de dois fatores: a ancestralidade comum — a ideia de que um ancestral em comum sofreu modificações ao longo do tempo, dando origem a outras espécies, como mostrado na Figura 2 - e a seleção natural — a principal teoria de Darwin e a mais difundida. Ela propõe que o meio seleciona os indivíduos mais aptos a sobreviver, portanto, estes se reproduzem e passam suas características adiante.

Como muitos mais indivíduos de cada espécie nascem do que podem sobreviver; e como, conseqüentemente, há uma luta frequentemente recorrente pela existência, segue-se que qualquer ser, se variar ligeiramente de alguma maneira lucrativa para si, sob as condições complexas e às vezes variáveis de vida, terá uma melhor chance de sobreviver, portanto, ser selecionado naturalmente. A partir do forte princípio da herança,



qualquer variedade selecionada tenderá a propagar sua forma nova e modificada. (DARWIN, 1859)

Figura 2 – Seleção Natural.



Fonte: Imagem retirada de (BENNETOT, 2022).

Neste contexto, os Algoritmos Evolutivos também se destacaram por utilizarem métodos relativamente simples, inspirados em princípios da Teoria da Evolução de Darwin, os quais podem ser modelados com instruções básicas de programação, além de serem bem adaptados para problemas das mais diversas áreas (GABRIEL; DELBEM, 2008).

Algoritmos Evolutivos são eficientes métodos heurísticos de busca baseados na Teoria da Evolução de Darwin com poderosas características de robustez e flexibilidade para capturar soluções globais de complexos problemas de otimização (GALVAN et al., 2003).

Os AEs se baseiam fortemente nas teorias evolucionistas de Darwin, como a Seleção Natural, para a otimização de problemas. Embora, diversos trabalhos sobre o tema tenham sido publicados durante as décadas de 1950, foi por meados de 1960 que foram desenvolvidas as bases do que hoje é identificado como as três principais áreas na computação evolutiva: a programação evolutiva, os algoritmos genéticos e as estratégias evolutivas (JONG, 2000). A Programação Evolutiva foi proposta por Fogel (1966) em San Diego, Califórnia. Os Algoritmos Genéticos foram desenvolvidos por Holland (1967) e seus alunos na Universidade de Michigan em Ann Arbor. Por fim, as Estratégias Evolutivas foram um desenvolvimento em conjunto de três estudantes, Bienert, Rechenberg e Schwefel, em Berlin 1965.

Os AEs oferecem a vantagem de trabalharem com múltiplas soluções para um problema, diferente de métodos determinísticos que buscam uma única solução ótima. Ainda que hajam diversas abordagens de AEs, o princípio básico é o mesmo: dado uma população de indivíduos (i.e. um conjunto de soluções), pressões do ambiente desencadeiam

---

**Algoritmo 1** Pseudocódigo do Algoritmo Evolutivo.

---

Gerar população inicial.

Avaliar população segundo função objetivo.

**Enquanto** número de gerações não atingido.

    Selecionar indivíduos para reprodução.

    Realizar cruzamento entre os indivíduos selecionados.

    Realizar mutação em indivíduos gerados, se for o caso.

    Inserir filhos na população.

    Avaliar população segundo função objetivo.

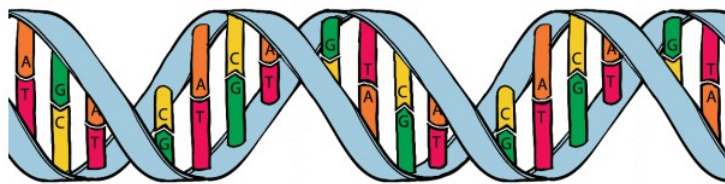
---

a seleção natural em que indivíduos mais aptos são privilegiados (nesse caso, as soluções mais adequadas) que constituirão uma nova população. Este processo é repetido até que um número de iterações seja atingido ou a solução mais qualificada seja encontrada (Algoritmo 2).

### 2.2.1 Conceitos Biológicos

A estrutura que codifica como os organismos são construídos é o cromossomo, que utiliza um conjunto de símbolos chamados gene cujo número varia de uma espécie para outra (AMABIS; MARTHO, 1985). O conjunto completo de cromossomos de um ser vivo é chamado genótipo e as características associadas a ele constituem o fenótipo.

Figura 3 – Cadeia de DNA.



Fonte: Imagem retirada de (CECIERJ, 2016).

Um gene é um segmento de DNA, como exemplificado na Figura 3, que contém uma informação codificada para determinada característica ou processo que a célula tem ou executa (AMABIS; MARTHO, 1985). Os diferentes valores de um gene são chamados alelos. A posição do gene em um cromossomo é denominada locus (COELLO; LAMONT; VELDHUIZEN, 2002).

No contexto de AEs, a representação das soluções candidatas (ou seja, dos indivíduos) é o primeiro estágio da elaboração de um Algoritmo Evolutivo e é crucial para o seu desempenho. Essa etapa consiste em

definir o genótipo e a forma com que esse é mapeado no fenótipo. O valor do fitness de um indivíduo (seja um genótipo ou um cromossomo) é um número positivo que mede o quão adequada é a solução. Em problemas de otimização, o fitness pode ser o custo da solução. Se o problema for de minimização, as soluções de maior fitness são as de menor custo (GABRIEL; DELBEM, 2008).

Os AEs possuem três componentes importantes: operadores de busca (mutação e recombinação), um controle de fluxo e uma representação que mapeia variáveis adequadas para implementação de soluções candidatas (BARTZ-BEIELSTEIN et al., 2014).

A recombinação genética é um evento decorrente da divisão celular, no qual uma dupla fita de DNA troca material genético com outra dupla fita. Este evento ocorre durante a reprodução sexuada pela combinação dos genes provenientes de diferentes indivíduos. As diferentes formas de recombinação genética fornecem mecanismos para o reparo do DNA, permitindo a manutenção da vida e da informação genética das espécies (BERNARDELI, 2022). A mutação gênica, na biologia, é o processo pelo qual o gene sofre alterações do código das bases nitrogenadas do DNA, que origina novas versões do gene.

A geração é uma iteração do Algoritmo Evolutivo, que é onde se realiza o controle do fluxo, na qual os indivíduos da população atual são selecionados e passam pela recombinação e/ou mutação, gerando descendentes. Devido à criação de novos descendentes, a população cresce em diversidade genética; então um mecanismo de seleção controla a qualidade do material genético gerado (GABRIEL; DELBEM, 2008).

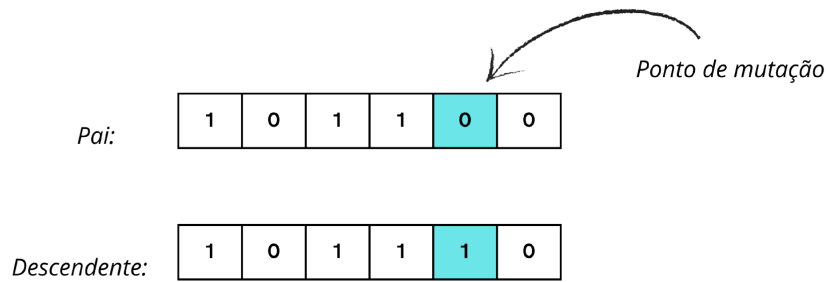
Por fim, a representação de um problema num AE é a escolha da estrutura de dados usada para as soluções. Dependendo da dificuldade do problema, a escolha de uma boa representação pode ter um forte impacto na performance do algoritmo (ASHLOCK; MCGUINNESS; ASHLOCK, 2012). Vale destacar que grande parte dos estudos empíricos são baseados nas formas canônicas, como sequências de caracteres binárias ou vetores com números reais, enquanto a maioria dos estudos em aplicações reais exige representações especializadas do problema (BARTZ-BEIELSTEIN et al., 2014).

## 2.2.2 Operadores de Reprodução

### 2.2.2.1 Mutação

A mutação, como mencionado na Seção 1.3, modifica aleatoriamente um ou mais genes de um cromossomo, sendo necessária para introdução e manutenção da diversidade genética da população. No contexto de AEs, um cromossomo pode ser representado por uma sequência de bits (0s e 1s) e cada posição dela um alelo. O operador de mutação de 1-ponto simplesmente troca o valor de um gene em um cromossomo (GOLDBERG, 1989). Por exemplo, selecionando aleatoriamente um alelo de valor 0 e aplicando o operador de mutação, ele passa a ser 1. A Figura 4 mostra como esse processo pode ocorrer.

Figura 4 – Operador de Mutação.



Além da representação dos cromossomos como vetores binários, há a representação em ponto flutuante. Essa representação é usualmente empregada quando os genes são distribuídos em um intervalo contínuo, em vez de um conjunto de valores discretos (EIBEN; SMITH, 2003). Neste caso, geralmente, são usados os operadores de mutação uniforme e mutação Gaussiana.

O primeiro seleciona aleatoriamente um alelo  $k \in \{1, 2, \dots, n\}$  do cromossomo  $x = \{x_1, \dots, x_k, \dots, x_n\}$  e gera um descendente  $x' = \{x_1, x_2, \dots, x_n\}$ , em que  $x'_k$  é um número aleatório (com distribuição de probabilidade uniforme) amostrado no intervalo  $[LI, LS]$ , sendo  $LI$  e  $LS$ , respectivamente, os limites inferior e superior para o valor do alelo  $x_k$ . No caso da mutação Gaussiana, todos os componentes de um cromossomo  $x$  são modificados pela seguinte expressão:

$$x_0 = x + N(0, \sigma)$$

Na qual  $N(0, \sigma)$  é um vetor de variáveis aleatórias Gaussianas independentes, com média zero e desvio padrão  $\sigma$ .

### 2.2.2.2 Recombinação

O operador de recombinação é responsável pela troca das características dos pais durante a reprodução que serão passadas aos descendentes, sendo o principal operador de reprodução do AE (GOLDBERG, 1989). Pode ser usado de diversas formas, algumas das tradicionalmente usadas na literatura são: 1-ponto, n-pontos, uniforme (GABRIEL; DELBEM, 2008). Na 1-ponto, é selecionado aleatoriamente um ponto de cruzamento e a partir dele as informações genéticas dos pais são combinadas. As aplicações 1 e n-pontos do operador de recombinação são exemplificadas na Figura 5.

A recombinação n-pontos é uma generalização da forma anterior, permitindo que mais de um ponto de cruzamento seja selecionado. Por fim, a uniforme não usa pontos de cruzamento, mas calcula a probabilidade de cada gene ser trocado entre os pais. A Figura 6 mostra a aplicação desse operador em sequências de números binários.

Figura 5 – Recombinação 1-ponto e n-pontos.

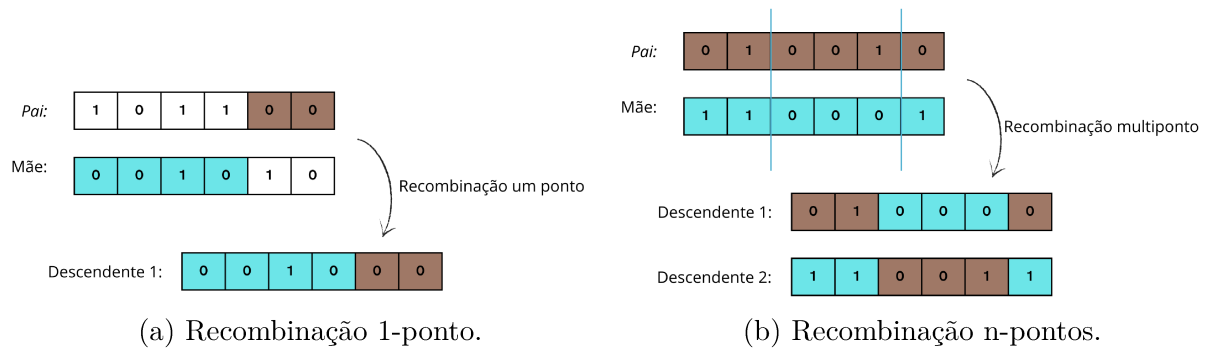
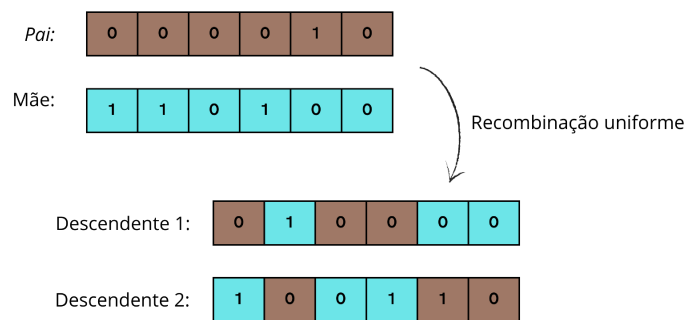


Figura 6 – Recombinação Uniforme.



### 2.2.3 Métodos de Seleção

Os Algoritmos Evolutivos seguem uma sequência de passos que garante a geração de populações compostas pelas soluções mais adequadas. Na Figura 7 esses passos são ilustrados com o auxílio de um fluxograma. Uma das etapas abordadas nesta seção é a seleção dos indivíduos (soluções) para reprodução.

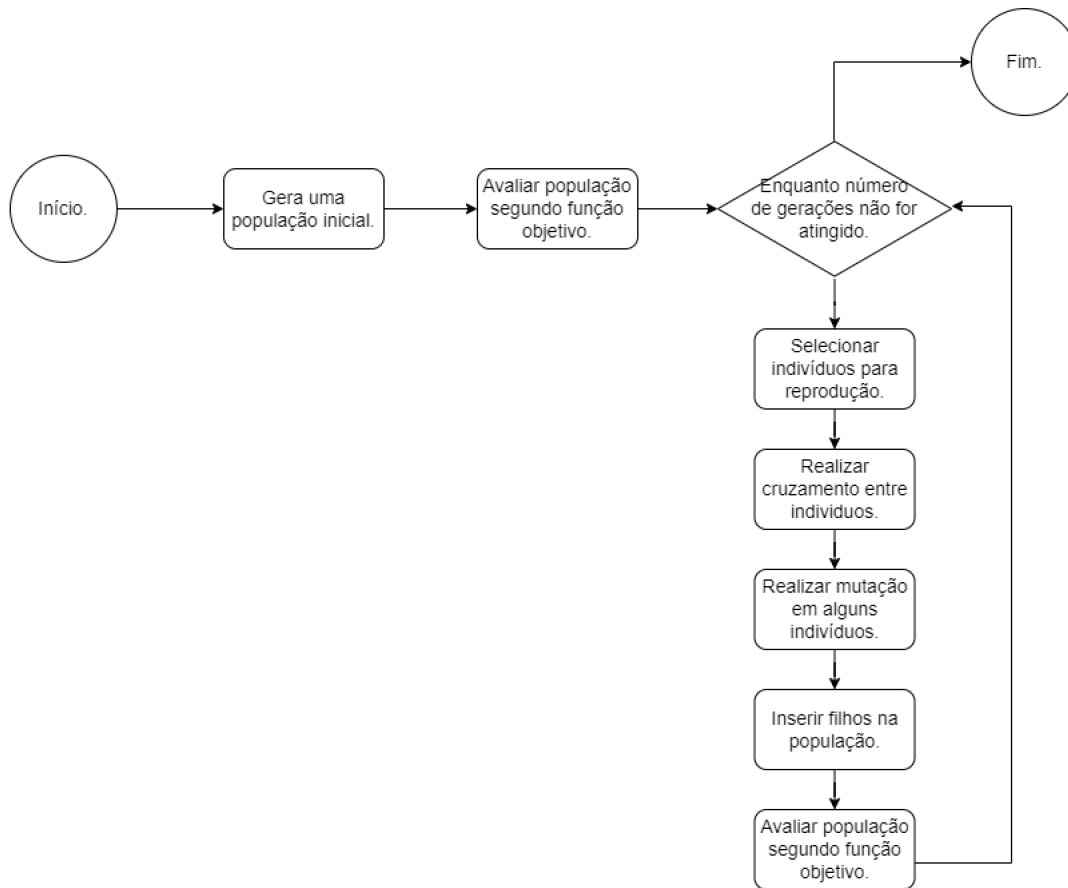
O princípio básico da Seleção Natural é: dado uma população de indivíduos e pressões do ambiente, aqueles a perpetuarem as próximas gerações são os mais adaptados ao meio. Há diversos métodos de seleção descritos na literatura: (GOLDBERG, 1989) aplicou em seu trabalho o método da roleta, (MILLER; GOLDBERG, 1995) descreveu os benefícios da seleção por torneio para a convergência dos resultados, (BAKER, 1987) discute em seu trabalho o método da amostragem universal estocástica, (MICHALEWICZ, 1992) discutiu os benefícios de preservar os melhores indivíduos por meio da seleção elitista.

No contexto de Algoritmos Evolutivos, existem três métodos de seleção, comumente usados: roleta, torneio e amostragem universal estocástica.

Como nascem muitos mais indivíduos de cada espécie, que não podem subsistir; como, por consequência, a luta pela existência se renova a cada instante, segue-se que todo o ser que varia, ainda que pouco, de maneira a torná-lo aproveitável tal variação, tem maior probabilidade de sobreviver, este ser é também objeto de uma seleção natural. Em virtude

do princípio tão poderoso da hereditariedade, toda a variedade objeto da seleção tenderá a propagar a sua nova forma modificada.(DARWIN, 1859)

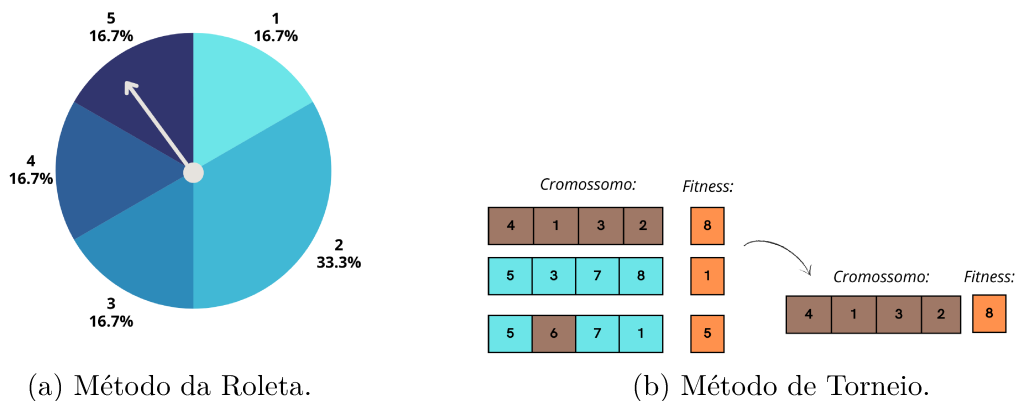
Figura 7 – Fluxograma de AEs.



No Método da Roleta, os indivíduos de uma população são colocados em uma roleta em que cada um recebe uma fatia equivalente a sua aptidão relativa. A roleta é girada e o indivíduo no qual a agulha da roleta parar é selecionado. O sorteio é repetido  $n$  vezes, até que o número de indivíduos necessários seja selecionado. Uma implementação comum deste método calcula uma lista de valores  $[a_1, a_2, a_3, \dots, a_n]$ , de modo que  $a_i = \sum_1^i P(i)$ , onde  $P(i)$  é a probabilidade proporcional ao *fitness* de um indivíduo  $i$  passar para a próxima geração.(GABRIEL; DELBEM, 2008). Este método de seleção é exemplificado na Figura 8a.

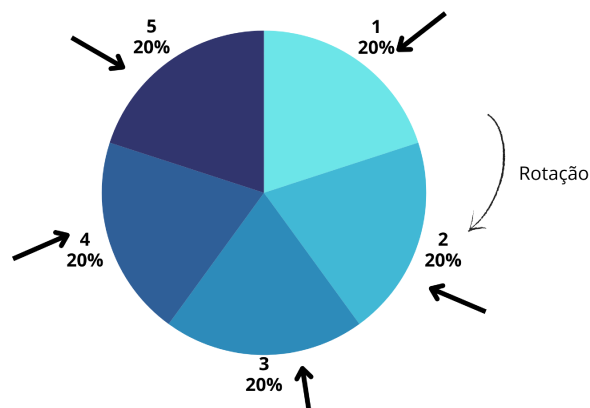
No método de Torneio, mostrado na Figura 8b, a pressão seletiva ocorre por meio de torneios entre  $N$  competidores, sendo  $N$  o tamanho do torneio. O vencedor é aquele com o melhor *fitness* dentre os  $N$  competidores e será um pai para a reprodução. Em seguida, repete-se o processo para escolher outro pai. A pressão seletiva é o grau pelo qual indivíduos mais adaptados são favorecidos: quanto mais alta a pressão seletiva, maior

Figura 8 – Métodos de Seleção.



o número de indivíduos aptos escolhidos (MILLER; GOLDBERG, 1995). Deste modo, quanto maior o  $N$ , maior a pressão seletiva.

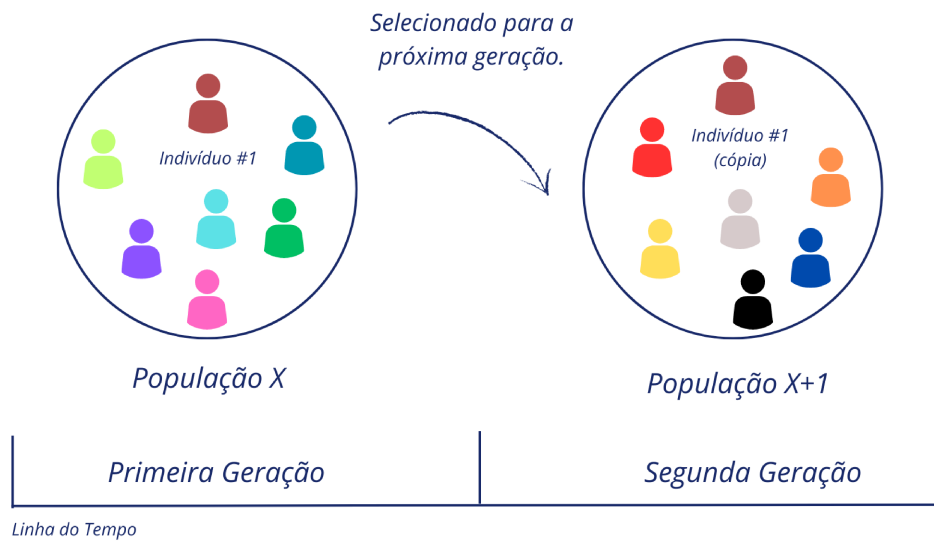
Figura 9 – Amostragem Universal Estocástica.



A Amostragem Universal Estocástica, exemplificada na Figura 9, funciona similarmente à Seleção por Roleta. Contudo, neste método são usadas  $k$  agulhas, correspondentes ao número de indivíduos, igualmente espaçadas, sendo giradas uma só vez. A Amostragem Universal Estocástica traz resultados menos variantes em comparação com o Método da Roleta.

Por fim, o elitismo tem como objetivo manter os indivíduos com melhor *fitness*. Esta técnica pode ser utilizada simultaneamente a outros métodos de seleção, como os explicados anteriormente. Com a seleção elitista, dada uma população inicial com  $k$  indivíduos, gerada aleatoriamente,  $n$  cópias daqueles com melhor *fitness* são guardadas nas posições iniciais e ficarão preservadas para a próxima geração (SOBRINHO, 2014). A porcentagem de indivíduos a serem selecionados é variável e pode ser definida pelo especialista, no caso da Figura 10, aproximadamente 14% da população (1 indivíduo) foi selecionado.

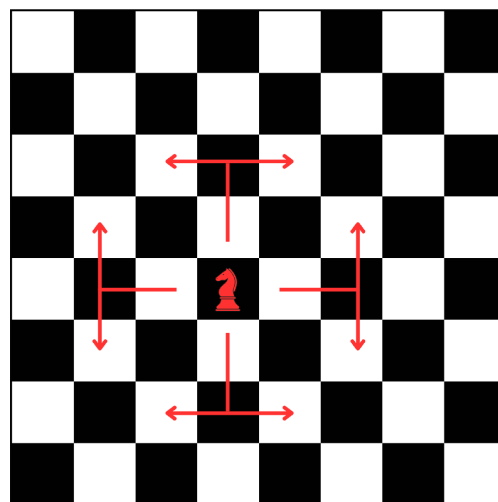
Figura 10 – Seleção Elitista.



## 2.3 Problema do Percurso do Cavalo

O Percurso do Cavalo (em inglês, *Knight Tour*), num tabuleiro de xadrez, consiste em uma sequência de movimentos feitos pela peça de xadrez correspondente ao cavalo, de tal maneira que cada quadrado do tabuleiro seja visitado exatamente uma vez (PINTO, 2013). Esses movimentos são realizados em forma de 'L', dois no eixo vertical e um no horizontal, ou dois na horizontal e um na vertical, como mostra a Figura 11.

Figura 11 – Percurso do Cavalo.

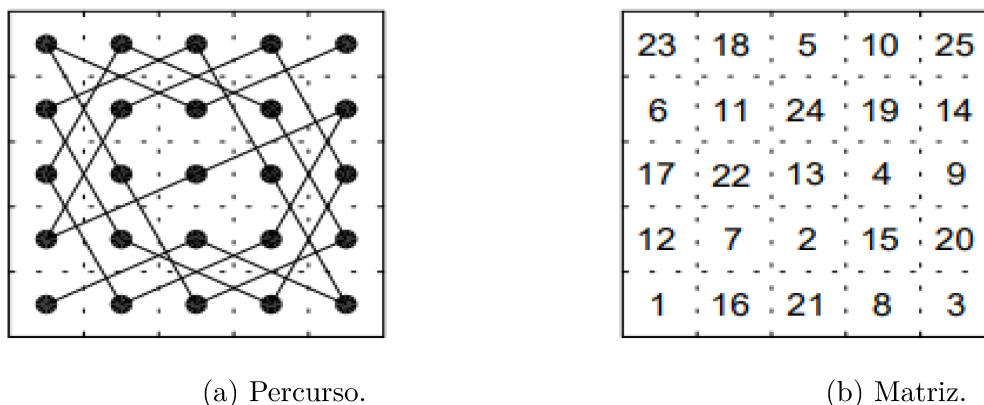


Leonhard Euler é considerado o primeiro a ter estudado formalmente o PPC (EULER, 1759). Na Figura 12 a solução de Euler para o PPC é ilustrada. O percurso é feito em um tabuleiro 5×5, representado na Figura 12a, com início na casa (5,5). O final desse



percurso é a casa de número 1, na Figura 12b. Diversas versões do problema e soluções para cada uma delas foram propostas desde então. Na versão *fechada*, o cavalo deve conseguir retornar para a casa inicial em um único movimento após ter visitado as demais (COSTA; Sá, 2013). A versão aberta não exige essa condição.

Figura 12 – Percurso do Cavalo por Euler.



Fonte: Imagem retirada de (PINTO, 2013).

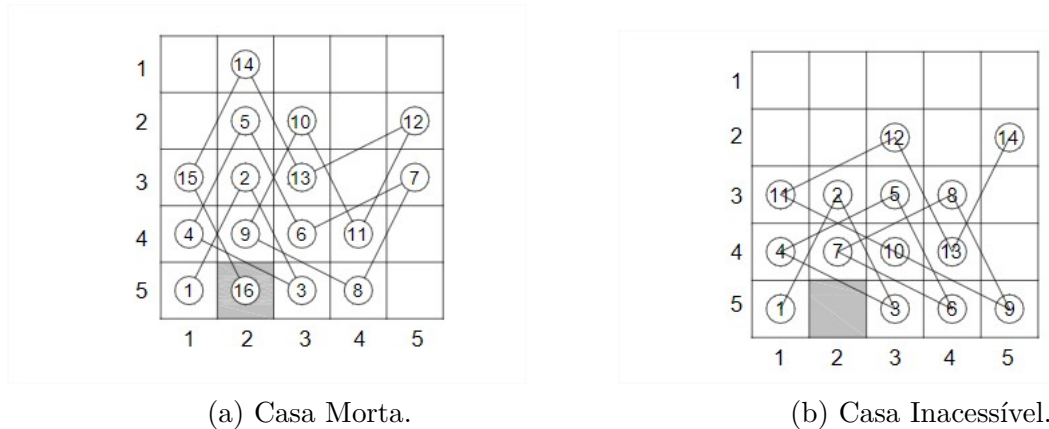
Não há algoritmo exato eficiente considerando tabuleiros  $n \times n$ , com  $n$  suficientemente grande. No entanto, existem diversos trabalhos encontrados na literatura que lidam com o PPC, alcançando resultados relevantes. A versão para percursos fechados foi resolvida em (SCHWENK, 1991) para tabuleiros quadrados e retangulares. Neste artigo, Schwenk define os tabuleiros que aceitam esta versão do passeio, fechada, e delinea um método eficiente para encontrá-lo. É claro, se um tabuleiro admite o passeio fechado, então o mesmo tabuleiro aceita o passeio aberto a partir de todas as possíveis casas iniciais (COSTA; Sá, 2013). Contudo, há uma imensa quantidade de tabuleiros que não o admitem.

Em 1823, Warnsdorff, propôs um algoritmo eficiente para o PPC aberto. Segundo ele, “a cada passo, escolha a casa com o menor número de casas acessíveis; se um empate ocorrer, o mesmo pode ser resolvido arbitrariamente” (WARNSDORF, 1823). Esse algoritmo, leva as casas com tendência a ficarem isoladas, as quais geram situações problemáticas como a casa inacessível ou a casa morta, a serem visitadas antes (PINTO, 2013).

As situações, casa morta e casa inacessível, são características do PPC e devem ser evitadas na solução do problema (PARIS, 2004). Na Figura 13a a situação da casa morta ocorre após o 15º movimento uma vez que todas as casas adjacentes já foram visitadas. Na Figura 13b a situação da casa inacessível ocorre na casa(2,5), ainda não visitada, que se torna inacessível, pois os seus vizinhos já foram.

O PPC também pode ser enunciado como um problema em grafos (PINTO, 2013). Um grafo  $G$  é um par de conjuntos  $(V, E)$ , tal que  $V = V(G) = \{v_1, \dots, v_n\}$  é um conjunto

Figura 13 – Situações características do PPC.



Fonte: Imagem retirada de (PINTO, 2013).

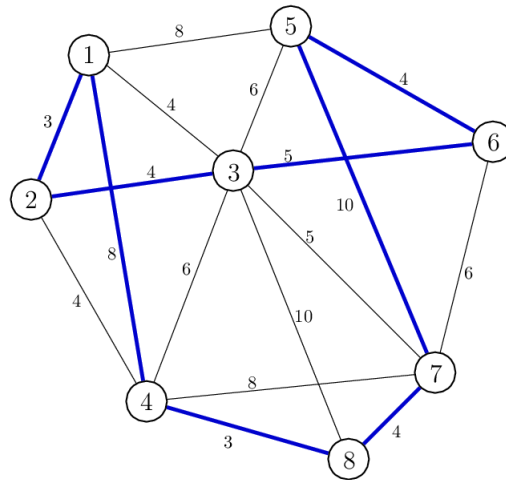
de vértices e  $E = E(G)$  é o conjunto das arestas, a cada uma das quais corresponde um subconjunto de  $V(G)$  de cardinalidade 2, i.e.,  $E(G) = \{e_1, \dots, e_m\}$ , com  $e_k = \{v_{k_i}, v_{k_j}\}$ , para  $k \in \{1, \dots, m\}$  (CARDOSO, 2005).

Neste sentido, um caminho entre um vértice  $x$  e  $y$  é um trajeto entre eles. Trajetos fechados, nos quais o vértice final coincide com o inicial, são chamados de circuitos, e os trajetos fechados nos quais o vértice inicial e final são os únicos a coincidirem são chamados de ciclos (CARDOSO, 2005). Portanto, um caminho e um ciclo que percorram todos os vértices de um grafo (sem repeti-los) são denominados, respectivamente, de caminho hamiltoniano e ciclo hamiltoniano de um grafo (SANTOS, 2016), como ilustrado na Figura 14. No contexto de grafos, o PPC torna-se a busca de um caminho (ou ciclo) hamiltoniano no grafo subjacente (PINTO, 2013).

Os algoritmos utilizados para solução de problemas de otimização podem ser determinísticos, em que se usa um ponto de referência como partida para busca de soluções ótimas, ou estocásticos, nestes o caráter aleatório de vários processos é simulado. Diferente dos determinísticos, os métodos aleatórios trazem uma resposta diferente a cada execução do código.

Os métodos de otimização natural possuem um forte caráter aleatório, pois são algoritmos que tentam simular fenômenos biológicos e, em sua maioria, são empregados na solução de funções e processos que envolvem análise combinatória. Portanto, os AEs são classificados neste tipo de otimização (YU; GEN, 2010), sendo a sua implementação para a solução do PPC muito proveitosa (PINTO, 2013).

Figura 14 – Ciclo Hamiltoniano.



Fonte: Imagem retirada de (PONCE et al., 2014).

## 2.4 Exemplos de aplicações do AE

Há diversos trabalhos que abordam o tema Algoritmos Evolutivos, como mostra a Figura 1, sendo o seu precursor (HOLLAND, 1992) e seus alunos da Universidade de Michigan. Posteriormente (GOLDBERG, 1989) contribuiu para a área, introduzindo o conceito de Pareto a AEs multiobjetivos.

Os AEs são aplicados na solução de inúmeros problemas, sendo algumas das áreas: engenharia elétrica, automação de sistemas de controle e pesquisa operacional (SLOWIK; KWASNICKA, 2020). (LV et al., 2017) apresenta o problema de otimização do *layout* de painéis solares o qual é resolvido pelo AE, além disso, ele demonstra que a potência de saída do painel solar é significativamente melhorada devido a essa otimização. (CARVALHO, 2021), por sua vez, analisa os indicadores de convergência de Algoritmo Evolutivo Multiobjetivo (AEMO), dando ênfase à Taxa de Concentração e à Taxa de Diversidade.

Uma importante área de atuação do AE é o problema de predição da estrutura tridimensional (nativa) de uma proteína, dada a sua sequência de aminoácidos (CUTELLO; NARZISI; NICOSIA, 2006). (BRASIL, 2012) investiga o Algoritmo Evolutivo Multiobjetivo baseado em Tabelas aplicado ao Problema de Predição de Estruturas de Proteínas (PSP) ab initio. Enquanto (SILVA, 2023) analisa os resultados obtidos pelo uso de AEs no problema PSP com o modelo de energia 2D HP utilizando energia simplificada.

O Problema do Percorso do Cavalo é trabalhado em vários artigos e múltiplas soluções já foram propostas a ele. (EULER, 1759) foi o primeiro a propor formalmente uma solução a ele, seguido por (SCHWENK, 1991), que resolveu o PPC para percursos fechados. (PARBERRY, 1997) implementou um algoritmo, baseado na estratégia de divisão e conquista, que pôde resolver o PPC em tabuleiros de dimensões: 8x8, 10x10, 16x16,

20x20, 27x27.

(LEE, 2000) propôs o uso do AE para a solução do PPC, demonstrando em suas pesquisas que o AE tradicional não conseguia encontrar uma solução para o PPC em tabuleiros de dimensões  $N \times N$  com  $N \geq 8$ . (GORDON; SLOCUM, 2004), por sua vez, conseguiu resolver o PPC em tabuleiros  $8 \times 8$  propondo a adaptação de indivíduos durante a sua avaliação. (PINTO, 2013) estudou diferentes adaptações nos operadores de reprodução e seleção que melhoraram os resultados do AE para o PPC.

(AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007), baseando-se no trabalho de (LEE, 2000) e (GORDON; SLOCUM, 2004), introduziu a ideia do uso da Regra de Warnsdorff - que prioriza casas que minimizem o número de próximos movimentos possíveis na seleção — a função de adaptação de (GORDON; SLOCUM, 2004) obtendo bons resultados para o tabuleiro  $8 \times 8$ . (SIROVETNUKUL et al., 2011) propôs o uso do *Coincidence Algorithm* (COIN) + AE para a solução do PPC, porém não obteve resultados bons como os de (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007).

(LONG-FU; YING-LONG; SEN, 2019) realiza a comparação do uso de AE com o Algoritmo de Colônia de Formigas e Busca em Profundidade para a solução do PPC, considerando-o como um ciclo hamiltoniano. (BANHARNSAKUN, 2019) propõe o uso do Algoritmo de Colônia de Abelhas (ABC), como uma abordagem alternativa para a solução do PPC, cujos resultados demonstraram a eficiência dos métodos dessa área de pesquisa ao gerar sequência de movimentos admissíveis para a peça de xadrez em relação a outros métodos.

(NEGRAO; RAMPAZZO, 2021) aborda o PPC com algoritmos baseados em heurísticas e meta-heurísticas, como o AE, por meio de importantes estruturas matemáticas (matrizes e grafos). Neste sentido, o PPC, além de ser usado na fundamentação de AE, também pode ser aplicado na solução de problemas do mundo real. (SINGH; KAKKAR; SINGH, 2015) utiliza o PPC para a geração de um esquema para criptografia de imagens e (KUMAR; NIRMALA, 2015) estuda a eficiência do AE combinado ao PPC como abordagem para este mesmo problema. (REZAZADEH et al., 2022) aplica o PPC na extração da potência máxima de matrizes Fotovoltaica (PV) em condições de sombreamento parcial. O passeio do cavalo reconfigura as matrizes PV com base nos movimentos do cavalo no tabuleiro de xadrez.

## 2.5 Considerações Finais

Na computação, diversos problemas matemáticos são investigados utilizando algoritmos de otimização, cujas soluções podem ser aplicadas a problemas reais. Dado a dificuldade de um problema, o mesmo pode ser classificado em P — problemas de decisão resolvíveis por um algoritmo determinístico em tempo polinomial —, NP — problemas de decisão resolvíveis por um algoritmo não-determinístico em tempo polinomial —, NP-

completo e NP-difícil.

O Algoritmo Evolutivo é uma técnica da Computação Bioinspirada, que pode ser utilizada para solução de problemas combinatoriais como o PPC. O AE é caracterizado pelos operadores de reprodução (recombinação e mutação), seleção (torneio, roleta, amostragem universal), funções de inicialização (aleatória, heurística) e função de avaliação (*fitness*). Tais funções são adaptadas dependendo do problema. A aplicação destas funções, porém, vai além da solução de problemas matemáticos, sendo usada para a solução do problema de predição de proteínas, do problema de otimização do layout de painéis solares, dentre tantos outros.

O Percurso do Cavalo é um problema NP-Completo, que busca encontrar uma sequência de movimentos válidos — realizados pela peça de xadrez correspondente ao cavalo — que percorra todo o tabuleiro sem visitar uma casa mais de uma vez. Este também é aplicado na solução de problemas do mundo real, como a criptografia de imagens, e a extração da potência máxima de matrizes fotovoltaicas Fotovoltaica (PV) em condições de sombreamento parcial, por exemplo.

No Capítulo 3 a implementação do AE para a solução do PPC é descrita.



---

# Implementação do Algoritmo Evolutivo

Há diversas implementações para o PPC, por exemplo, algumas impõem restrições no formato do tabuleiro, se quadrangular ( $n \times n$ ) ou retangular ( $n \times m$ ), se par ou ímpar. Além disso, há a distinção entre percursos fechados — todas as casas do tabuleiro são visitadas e é possível voltar a primeira a partir da última — e abertos — todas as casas do tabuleiro são visitadas e **não** é possível voltar a primeira a partir da última (LIN; WEI, 2005). A abordagem do algoritmo para este trabalho restringiu o tamanho do tabuleiro para  $n \times n$  com  $5 \leq n \leq 20$  considerando, valores pares e ímpares, e buscando percursos abertos.

## 3.1 Desenvolvimento do Algoritmo Evolutivo

O desenvolvimento deste AE foi em linguagem C no sistema operacional Windows 11 de um *notebook* Acer i3-1005G1 com 8 GB de memória RAM, tendo como referência o algoritmo desenvolvido por (GORDON; SLOCUM, 2004). A codificação do AE implementado neste Trabalho de Conclusão de Curso está disponível em: <<https://github.com/Murielly-Nascimento/AE>>.

O Algoritmo 2 descreve a sequência de passos para a implementação do AE.

## 3.2 Representação

A representação do caminho percorrido pelo cavalo é feita como descrita na Figura 15b, supondo um tabuleiro  $5 \times 5$ . De modo que a estrutura responsável por armazenar o percurso do cavalo seja um vetor de inteiros, ao invés de uma matriz com as coordenadas  $x$  e  $y$ , como é mostrado na Figura 15a. Neste trabalho, a representação do *INDIVÍDUO* foi uma *struct* — em linguagem C —, com os campos *fitness* e *tour* — um *array* de inteiros do tamanho do tabuleiro.

A seguir, serão mostradas as principais funções utilizadas neste AE e uma breve descrição de cada uma.

**Algoritmo 2** Pseudocódigo do AE aplicado ao PPC.

Gerar população inicial.

Avaliar população segundo função objetivo.

**Enquanto** número de gerações não atingido.

    Aplicar o elitismo a  $p\%$  da população.

**Enquanto** o contador  $i$  é menor que o tamanho da população.

        Selecionar dois indivíduos para o cruzamento.

        Realizar cruzamento entre os indivíduos selecionados.

        Realizar mutação em indivíduos gerados, se for o caso.

        Avaliar filho segundo função objetivo.

        Inserir filho na nova população.

Figura 15 – Representação do PPC.

	1	2	3	4	5
1	1	2	3	4	5
2	6	7	8	9	10
3	11	12	13	14	15
4	16	17	18	19	20
5	21	22	23	24	25

(a) Tabuleiro para o Percorso do Cavalo.

Indivíduo com fitness 24.

1	12	21	18	25	14	5	8	19	10	3	6	17	24	15	4	7	16	23	20	9	2	13	22	11
---	----	----	----	----	----	---	---	----	----	---	---	----	----	----	---	---	----	----	----	---	---	----	----	----

(b) Representação do Percorso do Cavalo.

### 3.3 Inicialização

A estrutura *INDIVIDUO* armazena o percurso do cavalo — usando como estrutura um vetor sequencial — e o *fitness*, calculado como a maior sequência de movimentos válidos, considerando um tabuleiro  $N \times N$  com um total de  $M$  casas, esse valor é  $M - 1$ . Neste trabalho duas formas de inicialização foram estudadas: a implementação de (PINTO, 2013) — a população é inicializada preenchendo o vetor com números de 1 a  $M$  e depois permutando aleatoriamente este vetor — e a inicialização central — a população é inicializada da mesma forma que (PINTO, 2013), com a exceção da primeira posição do vetor, cujo número de casa é calculado da seguinte forma:  $\frac{N^2+1}{2}$ . Esta função determina o número da casa central de um tabuleiro ímpar e o número de uma casa próxima ao centro de um tabuleiro par. Esta inicialização é baseada na premissa de que as casas centrais de um tabuleiro possuem o maior número de movimentos possíveis.



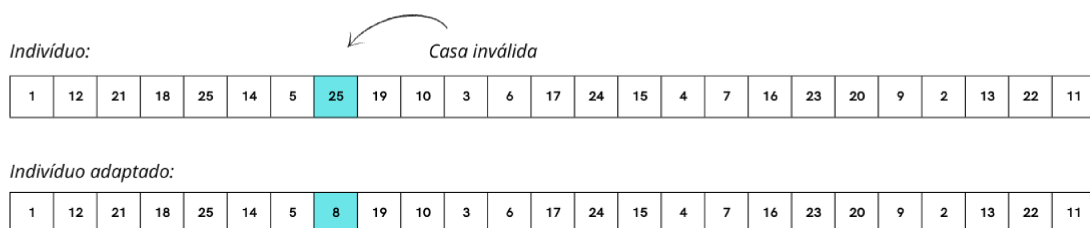
### 3.4 Funções de Avaliação

Foram implementadas duas funções de avaliação neste trabalho. A primeira realiza uma adaptação no *INDIVÍDUO*, substituindo genes (casas) inválidos do cromossomo (GORDON; SLOCUM, 2004). A segunda, além de efetuar esta adaptação, aplica a Regra de Warnsdorff, ou seja, prioriza as casas com menor número de movimentos possíveis. Em ambos os casos, o *fitness* é a soma dos movimentos válidos realizados pelo cavalo começando a partir da primeira posição do vetor. O valor de *fitness* é obtido somando a sequência de movimentos válidos, isto é, considerando um tabuleiro  $N \times N$  com um total de  $M$  casas, o valor ótimo esperado é  $M - 1$ , uma vez que todas as casas devem ser visitadas uma única vez.

(GORDON; SLOCUM, 2004) desenvolveu um método de reparação que ocorre quando a avaliação do percurso de um *INDIVÍDUO* é interrompida (causado por um vizinho inválido ou um número de casa duplicado). Neste ponto, o número de casa selecionado é substituído por um que permita o percurso continuar. Uma vez que o cavalo possui, no máximo, 8 movimentos possíveis, este cálculo não é custoso ao algoritmo. Caso a substituição não possa ser feita, a avaliação do percurso é interrompida.

As implicações da alteração de um *INDIVÍDUO* durante a sua avaliação foram estudadas por (WHITLEY; GORDON; MATHIAS, 1994). Tal processo é chamado de adaptação. Na Figura 16 observa-se que a casa número 25 é inválida, pois ela já foi visitada anteriormente, além de não ser um vizinho válido da casa 5, portanto, ela é substituída pela casa número 8 - vizinho válido da 5 e, ainda, não visitada.

Figura 16 – Função de adaptação no PPC.



De acordo com (LEE, 2000), um Algoritmo Genético tradicional não conseguiria resolver um tabuleiro  $8 \times 8$ . O mesmo elaborou este argumento após testá-lo em tabuleiros menores, com os quais conseguiu bons resultados. O mesmo problema é discutido em (GORDON; SLOCUM, 2004) e (PINTO, 2013). (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007) propõem uma solução para este problema usando a estratégia de (GORDON; SLOCUM, 2004) — a reparação do *INDIVÍDUO* durante a sua avaliação — combinada a Regra de (WARNSDORF, 1823).

Segundo (WARNSDORF, 1823), a casa com menor quantidade de movimentos possíveis deve ser visitada primeiro, em detrimento das demais. Aplicando esta regra, (AL-

GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007) obteve bons resultados. O uso desta heurística tornou 1.51% do total de cromossomos em percursos válidos, enquanto a função de reparação de (GORDON; SLOCUM, 2004) conseguiu o mesmo com apenas 0.57% dos cromossomos.

(SIROVETNUKUL et al., 2011) propôs o uso de Algoritmos de Coincidência, em inglês *Coincidence Algorithm* (COIN) + AE, como um método competitivo ao AE + heurística de (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007). A ideia deste método, COIN, é a junção do aprendizado de correlação negativa com o processo de otimização. Nele o algoritmo aprende tanto com as soluções boas quanto as ruins. Contudo, os testes feitos por (SIROVETNUKUL et al., 2011) demonstraram que o AE + heurística possui resultados melhores (1.51%) do que o COIN (1.05%). Portanto, neste trabalho a metodologia de (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007) também foi usada.

A regra de Warnsdorff consiste em selecionar a casa candidata que minimiza o número de próximos movimentos possíveis. Em outras palavras, por este critério o cavalo deve se deslocar para a casa a partir da qual ele contará com a menor quantidade de casas candidatas para dar seu passo seguinte. Intuitivamente, a ideia é a de atacar primeiro as restrições mais severas, deixando para mais tarde as casas que poderão ser mais facilmente visitadas. Do contrário, imagina-se, as casas que ora já dispõem de poucas opções poderão facilmente se transformar em becos sem saída. (COSTA; Sá, 2013)

### 3.5 Operadores de Seleção

A estratégia de elitismo ordena a população pelo *fitness*, reservando as posições iniciais para os melhores. Desta forma, independente do método de seleção aplicado, um nível de convergência dos resultados é garantido. Neste trabalho, o método de seleção por torneio foi estudado — largamente utilizado na literatura — e um novo operador proposto — o Torneio dos Dissimilares —, o qual é baseado na ideia que pais dissimilares levam explorar espaços de busca maiores.

Em relação à convergência, algumas vezes os AEs podem convergir prematuramente para um ótimo local, fato que geralmente ocorre quando há baixa diversidade genética da população. Por outro lado, se a população for muito diversificada geneticamente, há um custo computacional alto para que haja a convergência dos resultados (HUSSAIN; MUHAMMAD, 2020).

Os operadores de reprodução (mutação e recombinação) e a forma de seleção são fatores fundamentais no desenvolvimento de um algoritmo que equilibre a *exploitation* — termo em inglês, usado aqui para se referir ao uso de pontos previamente detectados para busca do ótimo — e *exploration* — exploração de novas áreas de busca (HUSSAIN; MUHAMMAD, 2020). A combinação de estratégias como elitismo (garante a convergência

dos resultados) e recombinação uniforme (aumenta a diversidade da população) é uma forma de alcançar tal equilíbrio.

(ISMKHAN, 2018) apontou que métodos de seleção, como o torneio, têm como consequência a combinação de pais similares, levando a descendentes com material genético próximo aos dos progenitores. Durante o processo evolutivo, isto pode conduzir à baixa diversidade de uma população. Uma solução para isto, descrita por ele, foi a seleção de pais dissimilares, uma vez que o material genético obtido a partir de sua recombinação seria igualmente diferente, garantindo assim a *exploration*.

O método Torneio dos Dissimilares, proposto neste trabalho de conclusão de curso, é inspirado nos estudos de (ISMKHAN, 2018). A ideia é que a recombinação seja aplicada entre um dos  $N$  *INDIVÍDUOS* de melhor *fitness* e um dos  $N$  de pior. Para isso, a seleção por torneio é modificada, de modo que o pai1 é o resultado de um torneio entre  $N$  *INDIVÍDUOS*, cujo vencedor é aquele com melhor *fitness* e a pai2 é o resultado de um torneio entre  $N$  *INDIVÍDUOS*, cujo vencedor é aquele com pior *fitness*. O Pseudocódigo 3 descreve os passos necessários para a implementação do Torneio dos Dissimilares.

---

**Algoritmo 3** Pseudocódigo da Torneio dos Dissimilares.

---

**Enquanto** o contador  $i$  for menor que o tamanho da *POPULAÇÃO*.

Realizar torneio entre  $N$  *INDIVÍDUOS*, priorizando o de melhor *fitness* (pai1).

Realizar torneio entre  $N$  *INDIVÍDUOS*, priorizando o de pior *fitness* (pai2).

Realizar cruzamento entre pai1 e pai2.

Aplicar mutação ao filho, se for o caso.

Avaliar filho segundo função objetivo.

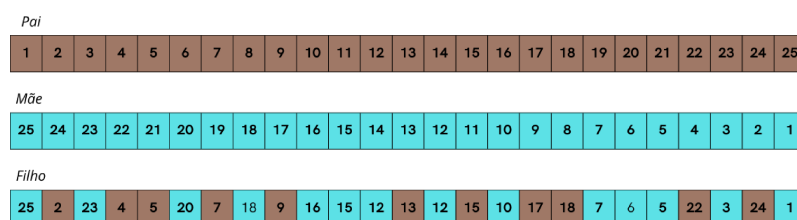
Inserir filho na nova população.

---

## 3.6 Recombinação

Neste trabalho, a recombinação uniforme foi implementada devido à variabilidade que ela acrescenta ao material genético, ao contrário de (GORDON; SLOCUM, 2004) que usou a recombinação-1-ponto. Esta função é ilustrada na Figura 17.

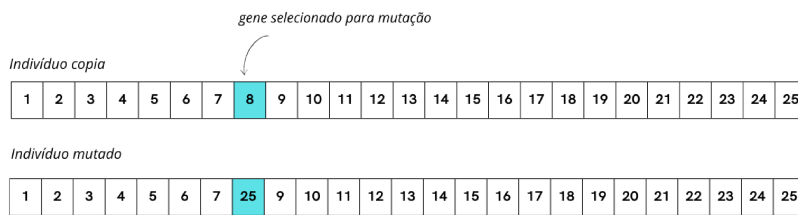
Figura 17 – Recombinação uniforme no PPC.



### 3.7 Operadores de Mutação

A mutação 1-ponto apresenta melhores resultados na convergência a um ponto ótimo do que a mutação n-pontos utilizada por (GORDON; SLOCUM, 2004). A troca de genes (número de casa) é feita selecionando uma posição aleatória do cromossomo (percurso) e a substituindo por um número entre 1 e  $M$ , sendo  $M$  o tamanho de um tabuleiro  $N \times N$ . Esta função é ilustrada na Figura 18.

Figura 18 – Mutação 1-ponto no PPC.



A Mutação dos Vizinhos, proposta neste trabalho, busca aumentar o *exploitation* do AE. Este operador é baseado no cálculo dos vizinhos válidos, próximos a uma das casas, sendo aplicado a um gene  $g$  do cromossomo, a uma probabilidade  $p$ . O operador avalia quais seriam os movimentos válidos, considerando  $g$ , e forma um vetor com esses valores. Após completar esses passos, um sorteio é aplicado a este vetor e o número sorteado irá ocupar a casa  $g + 1$ . O Pseudocódigo 4 descreve os passos necessários para a Mutação dos Vizinhos.

---

#### Algoritmo 4 Pseudocódigo da Mutação dos Vizinhos.

---

Aplicar mutação ao *INDIVÍDUO*  $I$  dado uma probabilidade  $p$  e taxa de MUTAÇÃO  $M$ .

**Se**  $p < M$

Selecionar gene  $g$  aleatório do cromosso.

Agrupar em um vetor os vizinhos válidos a  $g$ .

Sortear um dos vizinhos válidos.

Inserir vizinho válido na posição  $g + 1$ .

---

### 3.8 Considerações Finais

O AE aplicado na solução do PPC, tem como objetivo produzir um *INDIVÍDUO*, cujo cromossomo represente um percurso válido, feito pelo cavalo, por todo o tabuleiro, sem que uma casa seja visitada mais de uma vez. A implementação proposta foi baseada nos trabalhos de (GORDON; SLOCUM, 2004) — responsável por introduzir a ideia de correção do percurso do cavalo na função de avaliação — (AL-GHARAIBEH; QAWAGNEH;

AL-ZAHAWI, 2007) — que trabalhou o uso da Regra de Warnsdorff aliada à estratégia de correção do *INDIVÍDUO*, proposta por (GORDON; SLOCUM, 2004) — e (PINTO, 2013) — que descreveu os resultados com a inicialização do percurso do cavalo aleatoriamente. Os operadores de recombinação uniforme, mutação 1-ponto e seleção por torneio seguiram os exemplos tradicionais da literatura.

Além disso, três novos operadores foram propostos neste trabalho: torneio dos dissimilares — o pai de um *INDIVÍDUO* é aquele com melhor *fitness* de um torneio entre  $N$  *INDIVÍDUOS*, enquanto a mãe é o com pior *fitness* de um torneio entre  $N$  *INDIVÍDUOS* —, mutação dos vizinhos — um gene  $g$  é selecionado aleatoriamente o gene  $g + 1$  é substituído por um dos seus possíveis vizinhos válidos — e inicialização central — a primeira casa do percurso dos *INDIVÍDUOS* é inicializada com o número de uma casa no centro do tabuleiro.

No Capítulo 4 os experimentos realizados com o AE e suas respectivas funções são descritos.



## Experimentos

Neste capítulo serão apresentados os experimentos realizados com o AE desenvolvido neste trabalho, bem como os resultados obtidos. O AE foi executado 10 vezes para cada configuração de parâmetros, imprimindo em um arquivo o tempo gasto para cada execução e o *fitness* do melhor INDIVÍDUO encontrado. O cruzamento de recombinação uniforme foi utilizado em todos os testes com taxa 100%. A eficiência de um operador e a configuração de parâmetros foram avaliados pela média do *fitness*, o desvio padrão, a média do número de gerações e o tempo gasto para execução do programa. Os tabuleiros de dimensões 5x5, 8x8, 10x10, 16x16 e 20x20 foram aplicados nestes experimentos.

### 4.1 Tabuleiro 5x5

Na Tabela 1, são descritos os testes para um tabuleiro 5x5, cujo *fitness* máximo é 24. O cálculo do *fitness* é feito da mesma forma que (GORDON; SLOCUM, 2004). O *fitness* ótimo (24) foi obtido em todas as execuções na 13<sup>ª</sup> geração.

Tabela 1 – Tabuleiro (5x5); população (60); número máximo de gerações (180); mutação 1-ponto (15%); elitismo (10%); torneio (3).

Resultados dos Testes.			
Execução	Geração	Tempo em segundos	<i>Fitness</i>
1 <sup>a</sup>	13	0.002	24
2 <sup>a</sup>	13	0.002	24
3 <sup>a</sup>	13	0.003	24
4 <sup>a</sup>	13	0.003	24
5 <sup>a</sup>	13	0.002	24
6 <sup>a</sup>	13	0.003	24
7 <sup>a</sup>	13	0.003	24
8 <sup>a</sup>	13	0.002	24
9 <sup>a</sup>	13	0.003	24
10 <sup>a</sup>	13	0.003	24

A Tabela 2 mostra as medidas de avaliação para o tabuleiro 5×5 com a configuração descrita na Tabela 1. Nesta tabela, observa-se a média dos resultados é 24 e o desvio padrão foi 0.

Tabela 2 – Medidas de Avaliação.

Medidas de Avaliação.						
Tabela	População	Média de Gerações	Melhor Resultado	Tempo Médio	Média do <i>Fitness</i>	Desvio Padrão
1	60	13	24	0	24	0

## 4.2 Tabuleiro 8x8

Um tabuleiro 8×8 possui casas no intervalo de 1 a 64, com maior *fitness* de 63. Na Tabela 3 são descritos os testes com a configuração: população (100), número máximo de gerações (400), mutação 1-ponto (15%), elitismo (10%) e torneio (3). O *fitness* ótimo (63) foi obtido em todas as execuções na 39<sup>o</sup> geração.

Tabela 3 – Tabuleiro (8×8); população (80); número máximo de gerações (200); mutação 1-ponto (15%); elitismo (10%); torneio (3).

Resultados dos Testes.			
Execução	Geração	Tempo em segundos	<i>Fitness</i>
1 <sup>a</sup>	39	0.019	63
2 <sup>a</sup>	39	0.018	63
3 <sup>a</sup>	39	0.018	63
4 <sup>a</sup>	39	0.021	63
5 <sup>a</sup>	39	0.025	63
6 <sup>a</sup>	39	0.016	63
7 <sup>a</sup>	39	0.025	63
8 <sup>a</sup>	39	0.021	63
9 <sup>a</sup>	39	0.013	63
10 <sup>a</sup>	39	0.022	63

A Tabela 4 mostra as medidas de avaliação para o tabuleiro 8×8 com a configuração descrita na Tabela 3. Nesta tabela, observa-se a média dos resultados (63), o desvio padrão (0), o tempo médio para execução do programa (0.02 segundos) e a média das gerações (39).

O tabuleiro 8x8 é tradicionalmente usado para análise de algoritmos na busca por um percurso válido do cavalo na literatura, como (LEE, 2000), (GORDON; SLOCUM, 2004). Neste trabalho, buscou-se avaliar o comportamento do AE em tabuleiros de dimensões maiores. Portanto, na Seção 4.3 o tabuleiro 10x10 é avaliado.



Tabela 4 – Medidas de Avaliação.

Medidas de Avaliação.						
Tabela	População	Média de Gerações	Melhor Resultado	Tempo Médio	Média do <i>Fitness</i>	Desvio Padrão
3	80	39	63	0.02	63	0

### 4.3 Tabuleiro 10x10

Um tabuleiro 10×10 possui casas no intervalo de 1 a 100, com maior *fitness* de 99. A Tabela 5 descreve os resultados para a seguinte configuração: população (200), gerações (400), mutação (15%), elitismo (10%) e torneio (3). Com esses parâmetros, o *fitness* ótimo (99) foi obtido em 3 dentre as 10 execuções.

Tabela 5 – Tabuleiro (10×10); população (200); número máximo de gerações (400); mutação 1-ponto (15%); elitismo (10%); torneio (3).

Resultados dos Testes.			
Execução	Geração	Tempo em segundos	<i>Fitness</i>
1 <sup>a</sup>	400	0.554	98
2 <sup>a</sup>	400	0.689	98
3 <sup>a</sup>	400	0.725	97
4 <sup>a</sup>	400	0.604	98
5 <sup>a</sup>	400	0.521	98
6 <sup>a</sup>	400	0.603	97
7 <sup>a</sup>	400	0.614	97
8 <sup>a</sup>	63	0.075	99
9 <sup>a</sup>	63	0.093	99
10 <sup>a</sup>	63	0.075	99

A Tabela 6 mostra as medidas de avaliação para os testes realizados na Tabela 5. A média do *fitness* foi 98 e o desvio padrão, 0.77. Note que a média do tempo (0.081) e o número máximo de gerações (63) correspondem aos testes que alcançaram o ponto ótimo. Portanto, é interessante observar que nos testes bem sucedidos o tempo médio para a execução do programa continuou baixo (0.081) em relação aos experimentos realizados com o tabuleiro 8x8 (0.02).

Tabela 6 – Medidas de Avaliação.

Medidas de Avaliação.						
Tabela	População	Média de Gerações	Melhor Resultado	Tempo Médio	Média do <i>Fitness</i>	Desvio Padrão
5	200	63	99	0.081	98	0.77

Como mencionado no Capítulo 4, modificações no AE são necessárias a fim de obter o *fitness* ótimo. Embora, (GORDON; SLOCUM, 2004) tenha conseguido bons resultados

aplicando a adaptação — substituir casas inválidas na função de cálculo do *fitness* — ao *INDIVÍDUO*, (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007) mostrou que o uso da Regra de Warnsdorff na função de adaptação — que consiste em selecionar a casa candidata que minimiza o número de próximos movimentos possíveis — consegue produzir resultados ainda melhores.

Portanto, na Tabela 7 são descritos os testes para o tabuleiro  $10 \times 10$  — com a mesma configuração de parâmetros utilizados na Tabela 3 - aplicando a Regra de Warnsdorff. Nesta tabela, observa-se que em todas as execuções foi possível alcançar *fitness* ótimo (99) na 1ª geração.

Tabela 7 – Tabuleiro ( $10 \times 10$ ); população (200); número máximo de gerações (400); mutação 1-ponto (15%); elitismo (10%); torneio (3).

Resultados dos Testes.			
Execução	Geração	Tempo em segundos	<i>Fitness</i>
1ª	1	0.008	99
2ª	1	0.012	99
3ª	1	0.011	99
4ª	1	0.010	99
5ª	1	0.009	99
6ª	1	0.010	99
7ª	1	0.009	99
8ª	1	0.010	99
9ª	1	0.010	99
10ª	1	0.011	99

A Tabela 8 mostra as medidas de avaliação para os testes realizados na Tabela 7. A média do *fitness* foi 99 e o desvio padrão, 0. Embora o número de casas seja 100, o programa gastou, em média, 0.01 segundos e conseguiu encontrar o ponto ótimo na 1ª geração, reduzindo o tempo e o número máximo de gerações do experimento anterior.

Tabela 8 – Medidas de Avaliação.

Medidas de Avaliação.						
Tabela	População	Média de Gerações	Melhor Resultado	Tempo Médio	Média do <i>Fitness</i>	Desvio Padrão
7	200	1	99	0.01	99	0

## 4.4 Tabuleiro 16x16

Um tabuleiro  $16 \times 16$  possui casas no intervalo de 1 a 256, com maior *fitness* de 255. Neste experimento, foi implementada a ideia proposta por (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007), ou seja, o uso da Regra de Warnsdorff na função de adaptação,

a fim de buscar o *fitness* ótimo em, ao menos, uma das 10 execuções do AE. Na Tabela 9 os testes para esse tabuleiro são descritos. O *fitness* ótimo (255) foi obtido em 4 das 10 execuções com um intervalo de gerações variando entre 30 e 119.

Tabela 9 – Tabuleiro (16×16); população (400); número máximo de gerações (1600); mutação 1-ponto (15%); elitismo (10%); torneio (3).

Resultados dos Testes.			
Execução	Geração	Tempo em segundos	<i>Fitness</i>
1 <sup>a</sup>	30	0.367	255
2 <sup>a</sup>	30	0.372	255
3 <sup>a</sup>	30	0.391	255
4 <sup>a</sup>	119	0.999	255
5 <sup>a</sup>	1600	13.407	254
6 <sup>a</sup>	1600	15.220	253
7 <sup>a</sup>	1600	15.571	254
8 <sup>a</sup>	1600	15.100	254
9 <sup>a</sup>	1600	14.815	253
10 <sup>a</sup>	1600	15.610	253

A Tabela 10 mostra as medidas de avaliação para os testes realizados na Tabela 9. A média do *fitness* foi 254 e o desvio padrão, 0.83. O tempo de execução do programa continua relativamente baixo, considerando o tamanho da instância (256 casas), gastando, em média, 0.53 segundos, para encontrar o ponto ótimo, nos testes que foram bem sucedidos.

Tabela 10 – Medidas de Avaliação.

Medidas de Avaliação.						
Tabela	População	Média de Gerações	Melhor Resultado	Tempo Médio	Média do <i>Fitness</i>	Desvio Padrão
9	400	52	255	0.53	254	0.83

No entanto, com os testes do tabuleiro 16x16 foi verificada uma limitação da abordagem de (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007), mostrando que, embora o algoritmo obtenha o *fitness* ótimo, a sua eficiência decai, — considerando os testes realizados no tabuleiro 10x10 cujo *fitness* ótimo foi obtido em todas as 10 execuções.

## 4.5 Tabuleiro 20x20

Nos experimentos anteriores, verificou-se que a abordagem de (GORDON; SLOCUM, 2004) produz bons resultados em tabuleiros 5x5 e 8x8 e a de (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007) em tabuleiros 5x5, 8x8, 10x10. Apesar de não obter resultados ideais, a abordagem de (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007) ainda consegue obter o ponto ótimo em tabuleiro 16x16. Portanto, a sua aplicação em tabuleiros

20x20 — que possui casas no intervalo de 1 a 400, com maior *fitness* de 399 — é avaliada. Na Tabela 11 os testes para esse tabuleiro são descritos. O *fitness* ótimo (399) não foi obtido em nenhuma das 10 execuções.

Tabela 11 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação 1-ponto (15%); elitismo (10%); torneio (3).

Resultados dos Testes.			
Execução	Geração	Tempo em segundos	<i>Fitness</i>
1 <sup>a</sup>	10000	374.268	395
2 <sup>a</sup>	10000	405.169	397
3 <sup>a</sup>	10000	400.069	398
4 <sup>a</sup>	10000	391.910	398
5 <sup>a</sup>	10000	410.678	398
6 <sup>a</sup>	10000	373.254	398
7 <sup>a</sup>	10000	455.152	398
8 <sup>a</sup>	10000	450.955	398
9 <sup>a</sup>	10000	407.035	398
10 <sup>a</sup>	10000	410.635	397

A Tabela 12 mostra as medidas de avaliação para os testes realizados na Tabela 11. A média do *fitness* foi 397 e o desvio padrão, 1.04. Embora, a população e o número de gerações sejam relativamente altos, não foi possível obter um *fitness* de 399.

Tabela 12 – Medidas de Avaliação.

Medidas de Avaliação.						
Tabela	População	Média de Gerações	Melhor Resultado	Tempo Médio	Média do <i>Fitness</i>	Desvio Padrão
11	1000	-	398	-	397	1.04

A partir destes resultados, conclue-se que a abordagem de (AL-GHARAIBEH; QAWAG-NEH; AL-ZAHAWI, 2007) não consegue produzir um percurso válido (*fitness* 399) em tabuleiros 20x20, embora chegue próximo do ponto ótimo esperado (398). Portanto, foi crucial realizar alterações no AE para resolver o PPC.

## 4.6 Mutação dos Vizinhos

A fim de obter resultados melhores com o tabuleiro 20x20, neste trabalho três operadores foram propostos — Mutação dos Vizinhos, Torneio dos Dissimilares e Inicialização Central, todos descritos no Capítulo 3. Tais operadores buscam equilibrar a *exploration* — exploração de novas áreas de busca — e *exploitation* — uso de pontos previamente detectados para busca do ótimo — definidos por (HUSSAIN; MUHAMMAD, 2020). A Mutação dos Vizinhos, por exemplo, aumenta a *exploration* do algoritmo, uma vez, que

ela insere vizinhos válidos a um percurso previamente definido. Portanto, na Tabela 13 são descritos os resultados obtidos com a implementação da Mutação dos Vizinhos. Observa-se que, do total de 10 execuções, 4 alcançaram o ponto ótimo (399), melhorando significativamente o resultado do teste anterior.

Tabela 13 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação dos vizinhos (15%); elitismo (10%); torneio (3).

Resultados dos Testes.			
Execução	Geração	Tempo em segundos	<i>Fitness</i>
1 <sup>a</sup>	41	1.674	399
2 <sup>a</sup>	10000	397.192	395
3 <sup>a</sup>	1163	46.521	399
4 <sup>a</sup>	10000	393.091	398
5 <sup>a</sup>	10000	399.135	397
6 <sup>a</sup>	1079	42.460	399
7 <sup>a</sup>	10000	414.886	397
8 <sup>a</sup>	10000	475.759	398
9 <sup>a</sup>	1017	44.068	399
10 <sup>a</sup>	10000	521.014	397

A Tabela 14 mostra as medidas de avaliação para os testes realizados na Tabela 13. A média do *fitness* foi 397 e o desvio padrão, 1.48. Para os 4 testes, do total de 10, que obtiveram o ponto ótimo, foram necessárias, em média, 825 gerações e 33.68 segundos para a execução do programa.

Tabela 14 – Medidas de Avaliação.

Medidas de Avaliação.						
Tabela	População	Média de Gerações	Melhor Resultado	Tempo Médio	Média do <i>Fitness</i>	Desvio Padrão
13	1000	825	399	33.68	397	1.48

## 4.7 Torneio dos Dissimilares

Como descrito no Capítulo 3 a seleção por torneio leva a combinação de pais com material genético similar. Desta forma, não há uma exploração maior do espaço de busca pelo *fitness* ótimo. O objetivo do Torneio dos Dissimilares é aumentar a *exploration* do algoritmo, por meio da combinação de pais com *fitness* distintos. Para isto, são realizados dois torneios entre  $N$  *INDIVÍDUOS*, um para selecionar o pai — o vencedor sendo o de melhor *fitness* entre os  $N$  selecionados — e outro a mãe — o vencedor sendo o de pior *fitness* entre os  $N$  selecionados.

Na Tabela 15 são mostrados os resultados utilizando o Torneio dos Dissimilares sem a interferência da Mutação dos Vizinhos. Desta forma, é possível verificar como os operadores atuam individualmente antes de combiná-los. Observa-se que, do total de 10 execuções, 4 alcançaram o ponto ótimo (399). O tempo gasto para a execução do algoritmo, por sua vez, varia de 10 a 502 segundos.

Tabela 15 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação 1-ponto (15%); elitismo (10%); torneio dos dissimilares (3).

Resultados dos Testes.			
Execução	Geração	Tempo em segundos	<i>Fitness</i>
1 <sup>a</sup>	10000	501.085	397
2 <sup>a</sup>	10000	417.413	398
3 <sup>a</sup>	1519	54.436	399
4 <sup>a</sup>	359	15.114	399
5 <sup>a</sup>	10000	343.475	397
6 <sup>a</sup>	10000	391.272	397
7 <sup>a</sup>	10000	308.381	398
8 <sup>a</sup>	8670	304.173	399
9 <sup>a</sup>	4346	153.957	399
10 <sup>a</sup>	10000	398.102	398

A Tabela 16 mostra as medidas de avaliação para os testes realizados na Tabela 15. A média do *fitness* foi 398, e o desvio padrão, 0.83. Para os 4 testes, do total de 10, que obtiveram o ponto ótimo, foram necessárias, em média, 3723 gerações e 131.92 segundos para a execução do programa.

Tabela 16 – Medidas de Avaliação.

Medidas de Avaliação.						
Tabela	População	Média de Gerações	Melhor Resultado	Tempo Médio	Média do <i>Fitness</i>	Desvio Padrão
15	1000	3723	399	131.92	398	0.83

## 4.8 Mutação dos Vizinhos e Torneio dos Dissimilares

Uma vez que ambos os operadores — Torneio dos Dissimilares e Mutação dos Vizinhos — foram avaliados separadamente, a Tabela 17 verifica como ambos os operadores atuam em conjunto. Nesta tabela, observa-se que do total de 10 execuções, 5 alcançaram o *fitness* ótimo — algumas em poucas gerações (118) e outras precisaram de mais (2774) —, um pouco melhor do que quando os operadores atuam individualmente — ambos obtiveram o *fitness* ótimo em 4 do total de 10 execuções.

A Tabela 18 mostra as medidas de avaliação para os testes realizados na Tabela 17. A média do *fitness* foi 398, e o desvio padrão, 0.77. Para os 5 testes, do total de 10, que

Tabela 17 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação dos vizinhos (15%); elitismo (10%); torneio dos dissimilares (3).

Resultados dos Testes.			
Execução	Geração	Tempo em segundos	<i>Fitness</i>
1 <sup>a</sup>	2774	148.247	399
2 <sup>a</sup>	10000	417.056	398
3 <sup>a</sup>	10000	358.577	398
4 <sup>a</sup>	10000	378.221	397
5 <sup>a</sup>	2548	145.754	399
6 <sup>a</sup>	118	8.040	399
7 <sup>a</sup>	6394	319.480	399
8 <sup>a</sup>	10000	446.787	398
9 <sup>a</sup>	10000	421.592	398
10 <sup>a</sup>	1408	50.296	399

obtiveram o ponto ótimo, foram necessárias, em média, 2648 gerações e 134.36 segundos para a execução do programa.

Tabela 18 – Medidas de Avaliação.

Medidas de Avaliação.						
Tabela	População	Média de Gerações	Melhor Resultado	Tempo Médio	Média do <i>Fitness</i>	Desvio Padrão
17	1000	2648	399	134,36	398	0.77

#### 4.8.1 Comparação entre o Torneio e o Torneio dos Dissimilares

Este experimento foi realizado a fim de obter uma comparação entre os melhores e os piores indivíduos na população ao se utilizar a seleção por torneio e a seleção por torneio dos dissimilares. Portanto, as Tabelas 19 e 20 analisam o comportamento da seleção por torneio e o torneio dos dissimilares na população. Com o primeiro operador, observa-se que há uma tendência geral para a convergência dos resultados — o menor *fitness* encontrado na última geração (10000) é 77 —, embora o operador de mutação atue na manutenção da diversidade do material genético, o número de casa distintas entre o melhor *INDIVÍDUO* e pior é, apenas, 12.

O Torneio de Dissimilares, assim como o torneio tradicional, apresenta uma convergência dos resultados — o *fitness* do pior *INDIVÍDUO* é 80 na geração 10.000, porém o número de casa distintas é maior (180) entre o melhor e pior *INDIVÍDUO*. Nesse sentido, é importante destacar que o PPC é sensível a alterações feitas no percurso, em outras palavras, a troca de uma casa não afeta somente o seu vizinho anterior, mas altera completamente o percurso a partir dela, principalmente se a casa alterada for uma das iniciais. Dessa forma, pode-se concluir que o torneio tradicional, largamente aplicado na literatura,

Tabela 19 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação dos vizinhos (15%); elitismo (10%); torneio (3).

Seleção por Torneio.			
Geração	Número de casas distintas	<i>Fitness</i> do melhor	<i>Fitness</i> do pior
1	397	388	21
2	400	390	9
3	396	390	19
5000	94	398	158
5001	14	398	77
5002	60	398	165
5003	126	398	21
9999	15	398	77
10000	12	398	77

apresenta uma forte *exploitation* dos resultados, enquanto o Torneio de Dissimilares tende a *exploration* do campo de busca.

Tabela 20 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação dos vizinhos (15%); elitismo (10%); torneio dos dissimilares (3).

Torneio de Dissimilares.			
Geração	Número de casas distintas	<i>Fitness</i> do melhor	<i>Fitness</i> do pior
1	400	391	59
2	398	391	14
3	400	391	7
5000	172	398	99
5001	284	398	97
5002	197	398	81
5003	272	398	69
9999	213	398	81
10000	180	398	80

## 4.9 Inicialização Central

Por fim, o último operador proposto neste trabalho foi a Inicialização Central que reduz a *exploration* do AE, uma vez que todos os percursos começam na mesma casa. Em (PARBERRY, 2020) é descrita a implementação de métodos que calculam o percurso do cavalo a partir da casa central, sendo ela uma das casas com maior possibilidade de movimentos. Neste sentido, a Tabela 21 apresenta os resultados para o AE com inicialização da população restrita, ou seja, todos os *INDIVÍDUOS* da população são inicializados com a casa central, lembrando que os outros operadores — Mutação dos Vizinhos e Torneio



dos Dissimilares — também estão sendo aplicados. Com esta configuração, foi possível obter o *fitness* desejado em 8 das 10 execuções.

Tabela 21 – Tabuleiro (20×20); população (1000); número máximo de gerações (10000); mutação dos vizinhos (15%); elitismo (10%); torneio dos dissimilares (3).

Resultados dos Testes.			
Execução	Geração	Tempo em segundos	<i>Fitness</i>
1 <sup>a</sup>	7769	316.381	399
2 <sup>a</sup>	639	27.719	399
3 <sup>a</sup>	701	32.423	399
4 <sup>a</sup>	7360	306.372	399
5 <sup>a</sup>	3096	133.702	399
6 <sup>a</sup>	1191	53.044	399
7 <sup>a</sup>	3024	131.805	399
8 <sup>a</sup>	1758	80.331	399
9 <sup>a</sup>	10000	367.791	398
10 <sup>a</sup>	10000	365.118	398

A Tabela 22 mostra as medidas de avaliação para os testes realizados na Tabela 21. A média do *fitness* foi 398, e o desvio padrão, 0.89. Para os 8 testes, do total de 10, que obtiveram o ponto ótimo, foram necessárias, em média, 3192 gerações e 180.86 segundos para a execução do programa.

Tabela 22 – Medidas de Avaliação.

Medidas de Avaliação.						
Tabela	População	Média de Gerações	Melhor Resultado	Tempo Médio	Média do <i>Fitness</i>	Desvio Padrão
21	1000	3192	399	180.86	398	0.89

## 4.10 Considerações Finais

(LEE, 2000) argumentou em seu trabalho que o AE tradicional não conseguiria solucionar o PPC em tabuleiros de tamanho  $n \times n$  com  $n \geq 8$ . Desde o seu trabalho, diversas abordagens e operadores diferentes foram propostos, com o intuito de melhorar o desempenho do AE na solução do PPC.

(GORDON; SLOCUM, 2004) foi o primeiro a propor a adaptação dos percursos das soluções durante o cálculo do seu *fitness*. (PINTO, 2013) estudou a implementação de um operador de correção cuja função era remover as casas repetidas de um percurso resultante do cruzamento dos pais. (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007) combinou a estratégia de (GORDON; SLOCUM, 2004) a Regra de Warnsdorff — que prioriza casas com menor número de movimentos possíveis — obtendo bons resultados, (SIROVETNUKUL et al., 2011) propôs o uso do *Coincidence Algorithm* aliado ao AE.

Apesar da vasta literatura, até o momento não foram encontrados trabalhos que estudem a abordagem do AE para a solução do PPC em tabuleiros de dimensões maiores ao tradicional 8x8. Com os experimentos realizados foi possível averiguar a eficiência das propostas de (GORDON; SLOCUM, 2004), que funciona bem em tabuleiros 5x5 e 8x8, e (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007), que consegue obter o *fitness* ótimo nos tabuleiros 5x5, 8x8, 10x10, 16x16. Entretanto, com as abordagens estudadas não foi possível resolver o tabuleiro 20x20, sendo este o desafio deste trabalho de conclusão de curso.

A fim de solucionar esta limitação, neste trabalho foram propostos os seguintes operadores: mutação dos vizinhos — que considera a substituição de um gene  $g$  por um que seja um vizinho válido do gene  $g - 1$  —, torneio dos dissimilares — o pai1 é o resultado de um torneio entre  $N$  indivíduos, cujo vencedor é aquele com melhor *fitness* e a pai2 é o resultado de um torneio entre  $N$  indivíduos, cujo vencedor é aquele com pior *fitness* —, e a inicialização central — na qual a primeira casa dos percursos é uma que esteja na região central do tabuleiro. Analisando as soluções obtidas (com a média, o melhor resultado e o desvio padrão) pelo AE desenvolvido neste estudo, pode-se constatar a convergência para a solução ótima em um tabuleiro 20x20, resolvendo o problema de modo satisfatório. No Capítulo 5 são apresentadas as conclusões deste trabalho.

---

## Conclusão

Analisando os resultados obtidos no Capítulo 4 - Experimentos, conclui-se que o AE é uma técnica interessante para a solução do PPC com dimensão até  $20 \times 20$ , provando ser capaz de resolver o problema em um tempo viável. Além disso, foi possível avaliar positivamente o efeito dos três operadores propostos neste trabalho — Seleção por Dissimilares, Mutação dos Vizinhos e Inicialização Central. Por meio da revisão bibliográfica realizada, foi traçada uma evolução das implementações do AE para o PPC partindo de (LEE, 2000) até este Trabalho de Conclusão de Curso.

Em seu trabalho, (LEE, 2000) concluiu, empiricamente, que o AE tradicional não conseguiria resolver o PPC em tabuleiro de tamanho  $8 \times 8$ , o que o fez restringir seus experimentos a tabuleiros menores. Isso também é observado em (GORDON; SLOCUM, 2004) que, por sua vez, propôs a adaptação dos *INDIVÍDUOS* durante a função de avaliação. O mesmo utilizou uma representação binária para os indivíduos da população e utilizou a seguinte configuração em um tabuleiro  $8 \times 8$ : população (50), recombinação-1-ponto (80%), mutação uniforme (1%), elitismo (1%) e número de gerações (20.000).

(PINTO, 2013), por sua vez, propôs o uso da Regra de Warnsdorff na inicialização da população, enquanto um operador de correção eliminaria valores repetidos — todos os testes foram feitos em um tabuleiro  $8 \times 8$ . Embora, esta estratégia tenha produzido valores próximos ao esperado, o algoritmo não alcançou um bom desempenho.

(AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007) observou os resultados de (LEE, 2000) e (GORDON; SLOCUM, 2004), e, ao contrário de (PINTO, 2013), propôs o uso da Regra de Warnsdorff durante a avaliação do indivíduo, e cunhou tal processo como adaptação. Assim como (GORDON; SLOCUM, 2004), (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007) usou a representação binária, usando os seguintes parâmetros em um tabuleiro  $8 \times 8$ : população (50), recombinação-1-ponto (85 – 95%), mutação (0.5%), seleção pelo método da roleta russa e número de gerações (50). Com esta configuração, o autor conseguiu obter bons resultados, sendo 1.51% do total de cromossomos avaliados completaram o percurso do cavalo.

Outros trabalhos propuseram estratégias diferentes para o AE com o intuito de me-

lhorar os resultados, porém não foram realizados testes em tabuleiros maiores que  $8 \times 8$ , dentre eles: (SIROVETNUKUL et al., 2011), (KUMAR; NIRMALA, 2015) e (NEGRAO; RAMPAZZO, 2021). Portanto, neste Trabalho de Conclusão de Curso buscou-se agregar à literatura experimentos com tabuleiros de tamanho  $n \times n$  com  $n \geq 8$ .

Como descrito no Capítulo 4, a combinação da estratégia usada por (AL-GHARAIBEH; QAWAGNEH; AL-ZAHAWI, 2007) — a substituição de movimentos inválidos por vizinhos válidos com o menor número de movimentos possíveis durante a função de avaliação — com a Seleção por Dissimilares, a Mutação dos Vizinhos, e a Inicialização Central, consegue obter o *fitness* ótimos em tabuleiros de tamanho  $n \times n$  com  $5 \leq n \leq 20$ .

Para trabalhos futuros, sugere-se testes com tabuleiro  $n \times n$  com  $n \geq 24$ , a fim de avaliar a eficiência do algoritmo proposto e buscar novas adaptações, caso o mesmo não convirja para o *fitness* ótimo. A aplicação de uma estratégia de divisão e conquista, descrita no artigo (PARBERRY, 2020), aliada ao AE é sugerida, bem como a implementação de novos operadores de reprodução como o sugerido por (ARRAM; AYOB, 2019).

---

## Referências

AL-GHARAIBEH, J.; QAWAGNEH, Z.; AL-ZAHAWI, H. Genetic algorithms with heuristic — knight's tour problem. In: . [S.l.: s.n.], 2007. p. 177–181.

AMABIS, J. M.; MARTHO, G. **Curso básico de biologia**. São Paulo: Editora Moderna Ltda, 1985. v. 3.

ARRAM, A.; AYOB, M. A novel multi-parent order crossover in genetic algorithm for combinatorial optimization problems. **Computers Industrial Engineering**, v. 133, p. 267–274, 2019. ISSN 0360-8352. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360835219302773>>.

ASHLOCK, D.; MCGUINNESS, C.; ASHLOCK, W. Representation in evolutionary computation. In: \_\_\_\_\_. **Advances in Computational Intelligence: IEEE World Congress on Computational Intelligence, WCCI 2012, Brisbane, Australia, June 10-15, 2012. Plenary/Invited Lectures**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 77–97. ISBN 978-3-642-30687-7. Disponível em: <[https://doi.org/10.1007/978-3-642-30687-7\\_5](https://doi.org/10.1007/978-3-642-30687-7_5)>.

BAKER, J. E. Reducing bias and inefficiency in the selection algorithm. In: **Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application**. USA: L. Erlbaum Associates Inc., 1987. p. 14–21. ISBN 0805801588.

BANHARNSAKUN, A. Artificial bee colony algorithm for solving the knight's tour problem. In: VASANT, P.; ZELINKA, I.; WEBER, G.-W. (Ed.). **Intelligent Computing & Optimization**. Cham: Springer International Publishing, 2019. p. 129–138. ISBN 978-3-030-00979-3.

BARTZ-BEIELSTEIN, T. et al. Evolutionary algorithms. **WIREs Data Mining Knowl Discov**, 2014.

BENNETOT, B. **Domestication des champignons du fromage**. Tese (Doutorado), 2022. Thèse de doctorat dirigée par Ropars, Jeanne Biologie université Paris-Saclay 2022. Disponível em: <<http://www.theses.fr/2022UPASB066/document>>.

BERNARDELI, J. **Recombinação gênica homóloga: o que é e sua função**. Artigo, 2022.

- BRASIL, C. R. S. **Algoritmo evolutivo de muitos objetivos para predição ab initio de estrutura de proteínas**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2012. Disponível em: <<https://www.teses.usp.br/teses/disponiveis/55/55134/tde-20072012-163056/pt-br.php>>. Acesso em: 28.3.2023.
- CARDOSO, D. M. **Teoria dos Grafos e Aplicações**. Dissertação (Mestrado) — Departamento de Matemática da Universidade de Aveiro, Aveiro, Portugal, 2005.
- CARVALHO, L. C. F. **Indicadores de Convergência e Diversidade em Algoritmos Evolutivos para Otimização Multiobjetivo**. Tese (Doutorado) — Faculdade de Computação, Universidade Federal de Uberlândia, 2021. Disponível em: <<https://repositorio.ufu.br/handle/123456789/36044>>. Acesso em: 01.6.2023.
- CECIERJ, F. **Biologia - Fascículo 2**. RJ, Brasil: Fundação CECIERJ, 2016.
- COELLO, C. A. C.; LAMONT, G. B.; VELDHUIZEN, D. A. V. **Evolutionary algorithms for solving multi-objective problems**. 2. ed. New York, NY: Springer, 2002.
- COSTA, V. S.; Sá, V. G. P. de. **Heurística eficiente para o passeio aberto do cavalo a partir de casas arbitrárias em tabuleiros quadrados**. Artigo, 2013.
- CUTELLO, V.; NARZISI, G.; NICOSIA, G. A multi-objective evolutionary approach to the protein structure prediction problem. **Journal of the Royal Society, Interface / the Royal Society**, v. 3, p. 139–51, 03 2006.
- DARWIN, C. **On the Origin of Species**. London: JohnMurray, 1859.
- EIBEN, A.; SMITH, J. Introduction to evolutionary computing. In: **Natural Computing Series**. Berlin: Springer, 2003.
- EULER, L. Solution d'une question curieuse qui ne paroît soumise à aucune analyse. p. 15:310–337, 1759.
- GABRIEL, P. H. R.; DELBEM, A. C. B. **Fundamentos de algoritmos evolutivos**. [S.l.]: ICMC-USP, 2008.
- GALVAN, B. et al. Parallel evolutionary computation for solving complex cfd optimization problems: A review and some nozzle applications. **Science Direct**, 2003.
- GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability; A Guide to the Theory of NP-Completeness**. USA: W. H. Freeman & Co., 1990. ISBN 0716710455.
- GOLDBERG, D. E. Computer-aided pipeline operation using genetic algorithms and rule learning. part ii: Rule learning control of a pipeline under normal and abnormal conditions. **Engineering with Computers**, v. 3, n. 1, p. 47–58, Mar 1987. ISSN 1435-5663. Disponível em: <<https://doi.org/10.1007/BF01198148>>.
- \_\_\_\_\_. **Genetic algorithms in search, optimization, and machine learning**. New York, NY: Addison-Wesley, 1989.
- GORDON, V.; SLOCUM, T. The knight's tour - evolutionary vs. depth-first search. In: **Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)**. [S.l.: s.n.], 2004. v. 2, p. 1435–1440 Vol.2.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence**. The MIT Press, 1992. ISBN 9780262275552. Disponível em: <<https://doi.org/10.7551/mitpress/1090.001.0001>>.

HUSSAIN, A.; MUHAMMAD, Y. S. Trade-off between exploration and exploitation with genetic algorithm using a novel selection operator. **Complex & Intelligent Systems**, v. 6, n. 1, p. 1–14, Apr 2020. ISSN 2198-6053. Disponível em: <<https://doi.org/10.1007/s40747-019-0102-7>>.

ISMKHAN, H. Black box optimization using evolutionary algorithm with novel selection and replacement strategies based on similarity between solutions. **Applied Soft Computing**, v. 64, p. 260–271, 2018. ISSN 1568-4946. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1568494617307214>>.

JONG, K. D. A history of evolutionary computation. In: FOGEL, D. B.; SCHWEFEL, H.-P. (Ed.). **Handbook of Evolutionary Computation**. Oxford: Oxford University Press, 2000. cap. 6, p. 40 a 58.

KUMAR, J.; NIRMALA, S. Securing the contents of document images using knight moves and genetic approach. In: **2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)**. [S.l.: s.n.], 2015. p. 1091–1095.

LEE, M. Finding solutions to the knight's tour problem using genetic algorithms. In: **Genetic Algorithms and Genetic Programming at Stanford 2000**. Stanford University: J Koza, 2000.

LIN, S.-S.; WEI, C.-L. Optimal algorithms for constructing knight's tours on arbitrary  $n \times m$  chessboards. **Discrete Applied Mathematics**, v. 146, n. 3, p. 219–232, 2005. ISSN 0166-218X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0166218X04003488>>.

LONG-FU, Z.; YING-LONG, W.; SEN, B. An algorithm of generalized knight's tour problem based on path joint. **IEEE**, 2019.

LV, M. et al. Solar array layout optimization for stratospheric airships using numerical method. **Energy Conversion and Management**, v. 135, p. 160–169, 2017. ISSN 0196-8904. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S019689041631175X>>.

MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. [S.l.]: Springer Berlin, Heidelberg, 1992. 252 p.

MILLER, B. L.; GOLDBERG, D. E. Genetic algorithms, tournament selection, and the effects of noise. In: **Complex Systems**. University of Illinois at Urbana-Champaign, USA: University of Illinois, 1995. cap. 9, p. 193 a 212.

NEGRAO, A. T.; RAMPAZZO, P. C. B. **A Integração entre o Xadrez, a Matemática e a Computação**. Artigo — UNICAMP, Campinas, São Paulo, 2021. Disponível em: <<https://www.prp.unicamp.br/inscricao-congresso/resumos/2021P18393A3603905426.pdf>>. Acesso em: 01.6.2023.

- PARBERRY, I. An efficient algorithm for the knight's tour problem. **Discrete Applied Mathematics**, v. 73, n. 3, p. 251–260, 1997. ISSN 0166-218X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0166218X96000108>>.
- \_\_\_\_\_. **Tourneys and the Fast Generation and Obfuscation of Closed Knight's Tours**. 2020.
- PARIS, L. Heuristic strategies for the knight tour problem. In: . [S.l.: s.n.], 2004. p. 1121–1125.
- PINTO, F. T. A. e P. E. D. Aplicação de algoritmos genéticos ao problema do percurso do cavalo. **Cadernos do IME - Série Informática**, v. 22, p. 9–25, jun. 2013. Disponível em: <<https://www.e-publicacoes.uerj.br/cadinf/article/view/6556>>.
- PONCE, J. et al. **Inteligencia Artificial**. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014. Disponível em: <[https://www.researchgate.net/publication/269466259\\_Inteligencia\\_Artificial](https://www.researchgate.net/publication/269466259_Inteligencia_Artificial)>.
- REZAZADEH, S. et al. Photovoltaic array reconfiguration under partial shading conditions for maximum power extraction via knight's tour technique. **J Ambient Intell Human Comput**, 2022. Disponível em: <<https://doi.org/10.1007/s12652-022-03723-1>>.
- SANTOS, M. de S. **Ciclos Hamiltonianos em Grafos**. Dissertação (Mestrado) — Instituto de Matemática e Estatística da Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016.
- SCHWENK, A. Schwenk, a.j.: Which rectangular chessboards have a knight's tour? *math. mag.* 64, 325-332. **Mathematics Magazine**, v. 64, 12 1991. Disponível em: <[https://www.researchgate.net/publication/265453950\\_Schwenk\\_AJ\\_Which\\_rectangular\\_chessboards\\_have\\_a\\_knight's\\_tour\\_Math\\_Mag\\_64\\_325-332](https://www.researchgate.net/publication/265453950_Schwenk_AJ_Which_rectangular_chessboards_have_a_knight's_tour_Math_Mag_64_325-332)>.
- SILVA, C. A. da. **Uma análise de parâmetros do algoritmo evolutivo voltado para o problema de predição de proteínas**. Monografia — Faculdade de Computação, Universidade Federal de Uberlândia, 2023. Disponível em: <<https://repositorio.ufu.br/handle/123456789/37547>>. Acesso em: 01.6.2023.
- SINGH, M.; KAKKAR, A.; SINGH, M. Image encryption scheme based on knight's tour problem. **Procedia Computer Science**, v. 70, p. 245–250, 2015. ISSN 1877-0509. Proceedings of the 4th International Conference on Eco-friendly Computing and Communication Systems. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050915032457>>.
- SIROVETNUKUL, R. et al. The effectiveness of hybrid negative correlation learning in evolutionary algorithm for combinatorial optimization problems. In: **2011 IEEE International Conference on Industrial Engineering and Engineering Management**. [S.l.: s.n.], 2011. p. 476–481.
- SLOWIK, A.; KWASNICKA, H. Evolutionary algorithms and their applications to engineering problems. **Springer**, 2020.



SOBRINHO, P. de S. **Algoritmos Genéticos Canônico e Elitista: uma abordagem comparativa**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, Natal, 2014. Disponível em: <<https://repositorio.ufrn.br/jspui/handle/123456789/17015>>. Acesso em: 02.5.2023.

TAHA, H. A. **Operations Research: An Introduction (8th Edition)**. USA: Prentice-Hall, Inc., 2006. ISBN 0131889230.

WARNSDORF, H. von. **Des Rösselsprunges einfachste und allgemeinste Lösung**. Verhagen, 1823. Disponível em: <<https://books.google.com.br/books?id=zvNdAAAAcAAJ>>.

WHITLEY, D.; GORDON, V. S.; MATHIAS, K. Lamarckian evolution, the baldwin effect and function optimization. In: DAVIDOR, Y.; SCHWEFEL, H.-P.; MÄNNER, R. (Ed.). **Parallel Problem Solving from Nature — PPSN III**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994. p. 5–15. ISBN 978-3-540-49001-2.

YU, X.; GEN, M. **Introduction to Evolutionary Algorithms**. Bedford, UK: Springer, 2010.