Universidade Estadual de Londrina
Centro de Tecnologia e Urbanismo
Departamento de Engenharia Elétrica

Vinícius Dourado Silva

Otimização da Eficiência de Entregas: Uma Análise Comparativa de Algoritmos para o Problema do Caixeiro Viajante

Londrina
2024

Universidade Estadual de Londrina

Centro de Tecnologia e Urbanismo
Departamento de Engenharia Elétrica

Vinícius Dourado Silva

Otimização da Eficiência de Entregas: Uma Análise Comparativa de Algoritmos para o Problema do Caixeiro Viajante

Trabalho de Conclusão de Curso orientado pelo Prof. Dr. Ernesto F. Ferreyra Ramirez intitulado "Otimização da Eficiência de Entregas: Uma Análise Comparativa de Algoritmos para o Problema do Caixeiro Viajante" e apresentado à Universidade Estadual de Londrina, como parte dos requisitos necessários para a obtenção do Título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Ernesto F. Ferreyra Ramirez

Londrina
2024

Ficha Catalográfica

Vinícius Dourado Silva
Otimização da Eficiência de Entregas: Uma Análise Comparativa de Algoritmos para o Problema do Caixeiro Viajante - Londrina, 2024 - 53 p., 30 cm.
Orientador: Prof. Dr. Ernesto F. Ferreyra Ramirez
1. Programação. 2. Inteligência Artificial. 3. Otimização. 4. Algoritmos.
I. Universidade Estadual de Londrina. Curso de Engenharia Elétrica. II. Otimização da Eficiência de Entregas: Uma Análise Comparativa de Algoritmos para o Problema do Caixeiro Viajante.

Vinícius Dourado Silva

Otimização da Eficiência de Entregas: Uma Análise Comparativa de Algoritmos para o Problema do Caixeiro Viajante

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica da Universidade Estadual de Londrina, como requisito parcial para a obtenção do título de Bacharel em Engenharia Elétrica.

Comissão Examinadora

Prof. Dr. Ernesto F. Ferreyra Ramirez
Universidade Estadual de Londrina
Orientador

Prof. Dr. Francisco Granziera Junior
Universidade Estadual de Londrina

Prof. Dr. Leonimer Flávio de Melo
Universidade Estadual de Londrina

Londrina, 27 de fevereiro de 2024

Dedico este trabalho a todos aqueles que, de alguma forma,
auxiliaram para a concretização desta etapa.

Agradecimentos

Adasdasdadasdsadasddasda.

"One day, you'll leave this world behind So live a life you will remember"
(Avicii)

Vinícius Dourado Silva. **Otimização da Eficiência de Entregas: Uma Análise Comparativa de Algoritmos para o Problema do Caixeiro Viajante**. 2024. 53 p. Trabalho de Conclusão de Curso em Engenharia Elétrica - Universidade Estadual de Londrina, Londrina.

Resumo

dasdasdasdasdsadasd

Palavras-Chave: 1. Programação. 2. Inteligência Artificial. 3. Otimização. 4. Algoritmos.

Vinícius Dourado Silva. **There won't be coup**. 2024. 53 p. Monograph in Electrical Engineering - Londrina State University, Londrina.

Abstract

dasdasdasdasdadadasdadadasd

Key-words: 1.Programming. 2. Artificial Intelligence. 3. Optimization. 4. Algorithms.

Lista de ilustrações

Lista de tabelas

Lista de quadros

Lista de Siglas e Abreviaturas

ABRASEL	Associação Brasileira de Bares e Restaurantes
PCV	Problema do Caixeiro Viajante
IA	Inteligência Artificial
AG	Algoritmo Genético
Fig.	Figura
RNA	Redes Neurais Artificias
SA	*Simulated Annealing*
Eq.	Equação

Sumário

1 Introdução

1.1 Motivação e Justificativa

1.2 Objetivos

1.2.1 Objetivo Geral

1.2.2 Objetivos Específicos

1.3 Organização do Trabalho

2 Fundamentação Teórica

2.1 Logística na era Pós Pandemia

Com a chegada da era pós-pandemia, a logística enfrenta uma série de desafios e oportunidades que moldam o futuro do setor. A crise econômica mundial foi a principal responsável por acelerar o uso de ferramentas digitais na logística, visando o benefício e simplificação das relações de comércio, bem como diferentes estratégias de otimização, visando o aprimoramento nos diversos setores de transporte (LMXlogistica, 2022).

Atualmente com a concorrência entre as empresas, se torna necessário estar em constante mudança para se destacar no mercado. Pois devido à grande velocidade de inovação e novas tecnologias, as empresas que não conseguem acompanhar este ritmo tendem a se tornar obsoletas e podem até fechar as portas (HUGO et al., 2017).

Carvalho et al. (1980) argumentam que a gestão de transportes, no que tange à redução de custos, é um impasse para as empresas nos dias de hoje. Ballou (2004) complementa que a logística participa de pelo menos um terço total das despesas de uma empresa. Considerando, o custo e o tempo, é inevitável que o planejamento da distribuição, não apenas de transporte de produtos, mas também de pessoas seja cada vez mais complexo.

Conforme destacado por Wang (2016)Wang (2016), a logística acompanhou o desenvolvimento da revolução industrial, alcançando marcos significativos ao longo do tempo. Um desses marcos foi a terceira inovação, que consistiu na sistematização da gestão logística com a introdução de computadores e Tecnologia da Informação (TI). Além disso, a quarta inovação, impulsionada principalmente pela Internet das Coisas (IoT) e Big Data, visa economizar mão de obra e estabelecer padrões no gerenciamento da cadeia de suprimentos.

o uso do seu próprio software, denominado de YMS Trackage(itálico), onde o sistema de gestão de pátio possibilitou uma redução de

A implementação destes sistemas logísticos otimizados gera um grande impacto no lucro dos empreendimentos, por meio da diminuição de tempo ou de custos nas atividades rotineiras, como organizar um armazém ou a frota de entregas de uma grande empresa (Mobilidade Sampa, 2023). Temos como exemplo a *YMS Trackage Maestro*, onde os números mostram que o uso do seu próprio software de gestão de pátio, denominado de *YMS Trackage*, onde o sistema de gestão de pátio possibilitou uma redução de até 60% das filas de veículos na portaria, além de um aumento de até 13% na produtividade (DIAS, 2022).

A crise econômica foi fator fundamental para acelerar o uso de ferramentas digitais na logística, com o benefício da simplificação das operações de comércio. Abordando o segmento alimentício, o serviço de *delivery* é uma prática que está presente no cotidiano de

inúmeras pessoas. Segundo a Associação Brasileira de Bares e Restaurantes (ABRASEL), o sistema de *delivery* movimentou cerca de R$11 bilhões em 2019.

Nesse sentido, destaca-se o Problema do Caixeiro Viajante (PCV), que propõe um algoritmo para otimização de rotas e que abrange várias adaptações na atualidade (BARBOSA; JR.; KASHIWABARA, 2015).

2.2 Problema de Otimização do Caixeiro Viajante

O Problema do Caixeiro Viajante (PCV) teve seus registros iniciais a pouco menos de 100 anos atrás, que foi citado pela primeira vez por Júlia Robinson em um relatório para *RAND Corporation* (ROBINSON, 1949), quando o poder computacional não existia ou era uma fração do que é hoje e continua amplamente abordado no mundo atual.

Esse desafio clássico de otimização em que o objetivo é encontrar a rota mais eficiente para um vendedor (o caixeiro viajante) pode realizar para visitar uma série de cidades apenas uma vez, antes de retornar ao ponto de partida. O problema tem implicações em diversas áreas incluindo logística, transporte, comunicação e até mesmo biologia.

Para Souza e Romero (2014), o PCV pode ser retratado como a formação de uma rota que se inicia e termina no mesmo ponto (vértice), após passar por vários locais, a fim de reduzir custos, tempo e extensão da viagem.

Uma abordagem do PCV estabelece uma única rota que passe por cada nó de um grafo, apenas uma vez, retornando ao nó inicial no final do percurso. Este roteiro deve ser feito de modo que a distância total percorrida seja mínima.

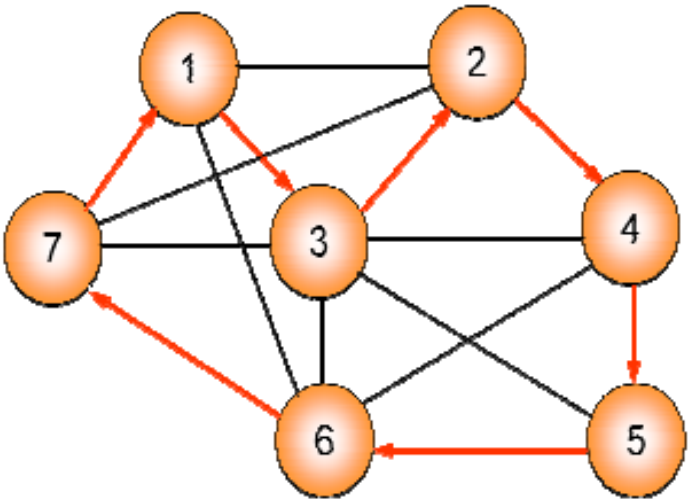


Figura 1 – Exemplo do PCV (BENEVIDES et al., 2011).

À medida que o número de nós (pontos) aumenta, a complexidade da resolução do PCV também aumenta. Para um problema com "n" nós e uma ligação entre cada par de

nós, o número de rotas possíveis pode ser calculado pela fórmula:

$$\frac{(n-1)!}{2} \tag{2.1}$$

onde cada nó possui (n – 1) possibilidades de conexão, e a divisão por dois considera que cada rota tem uma rota reversa equivalente com a mesma distância. Dessa forma, um problema com 10 nós resulta em aproximadamente 181 mil soluções, enquanto um com 20 nós pode ter até $6,08 \times 10^{16}$ soluções, e assim sucessivamente (LIEBERMAN, 2010).

De acordo com Prestes (2006), como os tempos para otimizar rotas com vários vértices por algoritmos exatos são inviáveis, a opção nesses casos seria o uso de heurísticas, onde os métodos utilizados são baseados em experiências e regras práticas que encontram soluções mais rapidamente mas não necessariamente exatas. A implementação da Heurística de construção de rotas para o PCV consiste em algoritmos que geram um circuito viável partindo de conjunto inicial de vértices, e modificando esse conjunto a cada iteração utilizando um critério de escolha (SILVA et al., 2013).

Há também diversas abordagens para o problema do caixeiro viajante utilizando algoritmos genéticos. Elas diferem entre si não apenas na questão dos parâmetros, mas também na forma de representar as soluções viáveis, de selecionar os indivíduos para reprodução e na maneira de definir os operadores genéticos.

2.3 Inteligência Artificial

A Inteligência Artificial (IA) é um campo da ciência da computação que se concentra no desenvolvimento de sistemas e algoritmos capazes de realizar tarefas que normalmente exigiriam inteligência humana (MCCARTHY, 2007). Essas tarefas incluem o reconhecimento de padrões, a tomada de decisões, o aprendizado de novos conceitos e a resolução de problemas complexos. A IA busca replicar o funcionamento do cérebro humano em máquinas, permitindo que elas aprendam com dados, se adaptem a novas situações e executem tarefas de forma autônoma.

Os avanços na IA têm impulsionado diversas áreas, desde a automação industrial até a medicina, passando pela análise de dados, reconhecimento de voz, jogos e entre outros (NADA, a). Com algoritmos sofisticados e o poder computacional cada vez maior, a IA tem transformando a maneira como interagimos com a tecnologia e como o mundo funciona.

Seja em assistentes virtuais, carros autônomos, sistemas de recomendação ou diagnósticos médicos, a Inteligência Artificial está presente em nosso cotidiano, e seu potencial para revolucionar diversos setores é vasto e promissor (PINHEIRO; OLIVEIRA, 2022).

Um dos exemplos em que mais se faz necessário o uso da IA, é o setor de otimização das entregas, na qual, é uma questão crucial na atualidade para muitas empresas,

principalmente naquelas que operam em setores altamente competitivos, como o comércio eletrônico. Para competir com eficácia, os empreendimentos precisam entregar seus produtos rapidamente e com eficiência, justamente onde a otimização de rotas se torna fundamental, especialmente quando focada para pequenas empresas, tornando assim a competição com grandes empresas mais justa (ecommercebrasil, 2023).

Para aplicar esse conceito nas empresas, é necessário uma preparação prévia, pois segundo Ailton Oliveira, cientista de dados da Trackage (DIAS, 2022):

> *"Um dos maiores desafios da inteligência artificial dentro da logística é a disponibilidade de dados. Isso acontece porque muitas vezes as empresas confundem quantidade de dados, ou seja, volume de dados, com dados de qualidade."*

Ailton (DIAS, 2022) complementa estas informações ao explicar que os dados fornecidos precisam ter um significado relevante para o aprendizado da IA e assim melhorar o trabalho realizado.

2.3.1 História

A história da inteligência artificial remete aos anos 50, com os trabalhos pioneiros de Alan Mathison Turing, conhecido popularmente como o pai da computação. Foi o principal responsável pela criação da Máquina de Turing, a qual foi utilizada para quebrar a criptografia da Enigma, causando uma verdadeira guinada na guerra para os aliados, mas a pesquisa em IA teve altos e baixos, com períodos de entusiasmo e desilusão, após essa grande descoberta(Roberto N. Onody, 2021).

Turing também foi além, afim de responder a principal dúvida sobre IAs, *afinal elas podem pensar?* Para aferir isso, Turing propôs o teste de Turing, onde o desempenho de uma máquina perante a tarefa de simular o comportamento humano e assim possa enganar a inteligência humana (SILVA; ARRUDA, 2016).

Posteriormente, surgiram os primeiros estudos sobre redes neurais para construção de redes neurais artificiais foram publicados (EBERHART, 1990), onde as redes neurais artificiais eram estabelecidas, como o perceptron, um modelo fundamental no campo. As redes neurais artificiais foram estabelecidas como uma abordagem promissora para a IA, onde as decisões são baseadas na multiplicação dos pesos dos neurônios, Figura 2.

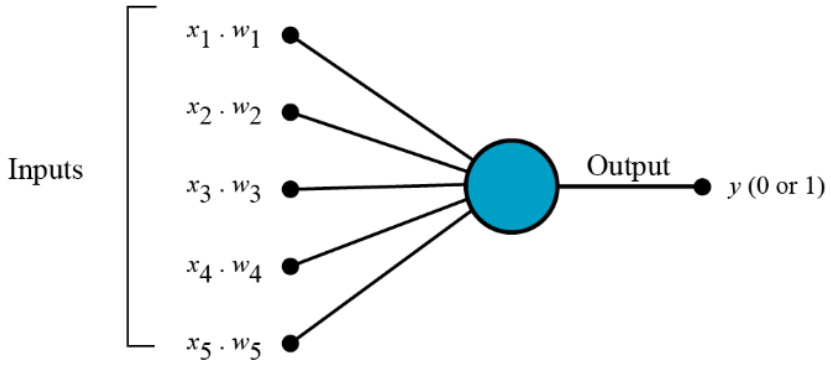


Figura 2 – Exemplo de Perceptron (Deep Learning Book, 2021).

No final da década de 70, a IA passou por um período de estagnação e redução de investimentos, conhecido como "Inverno da IA". Esse período foi caracterizado pelo declínio do interesse e dos recursos financeiros destinados a IA. Diversos fatores contribuíram para essa situação, incluindo expectativas exageradas em relação às capacidades da IA e a ausência de avanços significativos. Além disso, o alto critério dos pesquisadores e as limitações dos dispositivos existentes em termos de poder computacional também foram fatores determinantes.

Esse conjunção de elementos resultou em uma diminuição tanto do financiamento quanto do interesse acadêmico e industrial na área, evidenciando a necessidade de um realinhamento das expectativas e do desenvolvimento tecnológico para impulsionar o campo da IA.

A segunda era de desenvolvimento da IA, foi marcada pelo surgimento do *perceptron* multicamadas e os algoritmos de retro propagação (*backpropagation*), que permitiram a modelagem de problemas mais complexos e aprimorou a capacidade de aprendizado das redes neurais, ajustando os pesos das conexões entre os neurônios (PINHEIRO et al., 2021).

Esses avanços pavimentaram o caminho para a popularização e aplicação generalizada da IA em uma variedade de campos, contribuindo para o surgimento de tecnologias que utilizamos no dia a dia, como sistemas de recomendação, reconhecimento de voz e visão computacional.

2.3.2 Tipos

A IA pode ser dividida em duas classes, como IA fraca e IA forte ou IA geral. A IA fraca, em inglês *Artificial Narrow Intelligence*, é a teoria de que um sistema de inteligência artificial não teria a capacidade de raciocinar verdadeiramente e resolver os problemas (SALMEN; WACHOWICZ, 2021). De acordo com Quaresma (2021), a IA fraca ainda segue evoluindo com sistemas mais aprimorados e vem atuando nas demais frentes da

sociedade atual, como um aliado do ser humano. Salmen e Wachowicz (2021) ainda citam exemplos de IA fraca, como reconhecimento facial, de lugar, de objetos, de animais ou também na forma de *chatbot* (Software que simula uma conversa humana programado previamente).

A IA forte ou geral, em inglês *Artificial General Intelligence*, demonstra a forma de inteligência baseada em um computador, na qual, é capaz de raciocinar e solucionar qualquer problema (SALMEN; WACHOWICZ, 2021). A IA forte também pode ser definida como a hipótese de que as máquinas podem apresentar consciência, como a dos humanos (CAMPOS, 2021).

Em meados dos anos 80, John Searle afirmou que os computadores seriam autônomos e que teriam capacidade de raciocinar e de terem emoções, ainda, o autor cita que mentes artificiais seria um insucesso, devido a nenhum programa ser o suficiente para fornecer um sistema inteligível (SEARLE, 1987).

Nesse sentido, há os subtipos de IA: Simbólica, RNAs, Lógica *Fuzzy* e Algoritmo Genético. A IA simbólica ou então Sistemas Especialistas, são aqueles que podem resolver problemas a partir de regras já definidas, não generalizando suas respostas. Na implementação desse sistema, um especialista (humano) irá reger o sistemas com as regras necessárias, e posteriormente, o sistema retribuirá com respostas coerentes para certo problema (MARTINS, 2010).

Segundo Martins (2010), na prática do sistema simbólico, normalmente se baseia em um tipo de diálogo entre a máquina e o ser humano e assim, o sistema pode chegar ou não em uma conclusão.

Outro tipo de IA, a Redes Neurais Artificiais ou RNAs, são sistemas que geram classificações de modo automático ou semi automático, elas são configuradas para classificar ou reconhecer informações (MARTINS, 2010). Ela está inserida na área de *machine learning*, é como um modelo matemático que tenta simular a estrutura e funcionalidades de uma rede neural biológica, o uso do RNAs é como uma ferramenta potencial que auxilia na redução dos problemas observados nos modelos tradicionais de gerenciamento de resultados (SILVEIRA; SOUZA; BRITTO, 2023).

Em outras palavras Moreira et al. (2021) definem RNAs como sistemas paralelos que são compostos por unidades de processamento simples, que são capazes de calcular funções matemáticas. As unidades de processamento simples estão organizadas em uma ou mais camadas e interligadas por conexões, na maioria, unidirecionais.

No entanto, O pai da lógica *fuzzy*, Lofti Zadeh, divulgou em 1965 o artigo *"Fuzzy Sets"*, onde ele propôs uma teoria de conjuntos onde não existe descontinuidade, ou seja, que não existe uma interrupção abrupta entre os pertencentes e não pertencentes de um conjunto (ZADEH, 1965).

Segundo Zadeh (1965), a lógica *fuzzy* pode ser definido como:

> Com **X** sendo um espaço de objetos, e um elemento qualquer de

> **X**, sendo denotado por **x**. Assim: **X** = **x**. Um conjunto *fuzzy* **A** em **X** é caracterizado por uma função de pertencimento **fA**(**x**), que associa cada ponto em **X** a um número real no intervalo [0, 1], com o valor de **fA**(**x**) em **x** representando o "grau de pertencimento de **x** em **A**. Assim, quanto mais próximo de 1 for o valor de **fA**(**x**), maior o pertencimento de **x** em **A**.

Esse tipo de IA, propõe tratar de problemas dentro da IA, reduzindo as dificuldades em diversos setores, como os problemas de imprecisão nos dados e a incerteza do conhecimento. Em 2004, Russell e Norvig (2004) afirmaram que a lógica *fuzzy* era considerada como uma solução eficiente. Com suas numerosas possibilidades práticas, o sistema *fuzzy* pode ser considerado como um avanço nas metodologias de aplicação em produtos industriais, facilitando os processos matemáticos (VÁZQUEZ-GUZMÁN; MARTÍNEZ-RODRÍGUEZ; SOSA-ZÚÑIGA, 2015). Além disso, o último dos principais tipos de IA, é o Algoritmo Genético (AG), retratado na subseção 2.3.3.

2.3.3 Algoritmos Genéticos

Algoritmos Genéticos (AG), são algoritmos gerais e normalmente são aplicados em problemas complexos, com grandes espaços de busca e difícil modelagem (GOUVEIA et al., 2021). Os AG são considerados como métodos de otimização e se inspira nos mecanismos de evolução de populações de seres vivos (LACERDA; CARVALHO, 1999).

Em meados dos anos 70, os AG foi proposto pelo cientista e professor John Holland e popularizado por seu aluno, David Goldberg. Holland teve como objetivo de modelar uma implementação computacional de seleção baseado na adaptação da natureza, onde foi descrita pela teoria da evolução das espécies de Charles Darwin (ARAÚJO, 2021). Devido a isso, a linguagem do método é ligada à área biológica e possui nomenclaturas como apresenta-se no Quadro 1.

Termo biológico	AG
Alelo	Valor da característica
Cromossomo	Palavra binária, vetor, matriz, etc
Fenótipo	Estrutura submetida ao problema
Gene	Característica do problema
Genótipo	Estrutura
Geração	Ciclo
Indivíduo	Solução
Loco	Posição na palavra, vetor

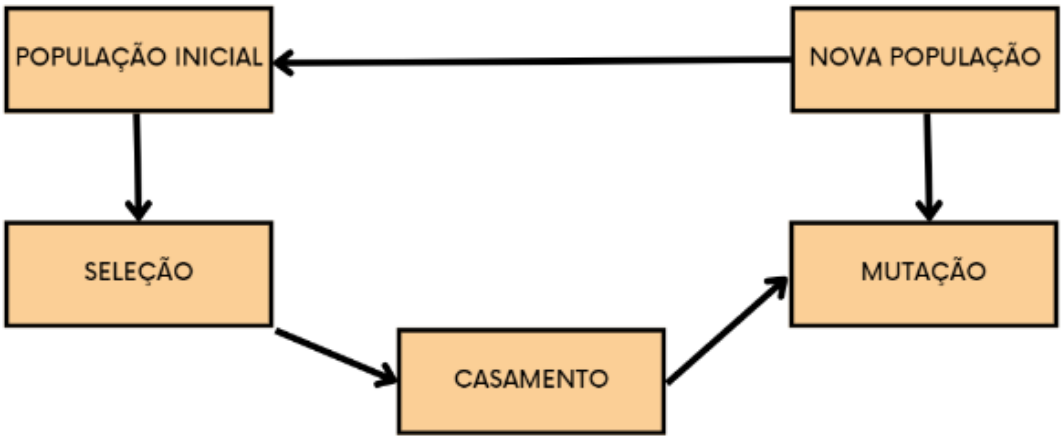
Tabela 1 – Adaptado de Pacheco et al. (1999).

Segundo Barreto (1999), os AG são métodos tradicionais de busca e otimização que se baseiam em quatro critérios, que são:

a. Trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros do problema;
b. Operam em uma população e não em pontos isolados, reduzindo o risco de busca;
c. Utilizam informações de custo ou recompensa (informações da função objetiva – payoff) e não derivadas ou outro conhecimento auxiliar;
d. Procedem a busca utilizando operadores escolásticos e não determinísticos.

A busca pela melhor solução se dá a partir da população inicial, que quando combinado os melhores desta população, obtém uma nova população, substituindo a inicial. A cada mutação ou então, cruzamento, é formada uma nova população que apresenta novas e melhores soluções para o problema, assim, atingindo os melhores resultados (FILITTO, 2008), conforme é apresentado na Figura 3.

Figura 3 – Fluxograma básico dos algoritmos genéticos



Fonte: Adaptado de (GOUVEIA et al., 2021)

A principal ideia do processo de seleção é permitir que os indivíduos mais adaptados tenham maior chance de se reproduzir. Barboza e Olandoski (2005) afirmam que a seleção elitista consiste em copiar ou reproduzir os melhores indivíduos da população atual para a próxima geração, garantindo que estes cromossomos não sejam destruídos nas etapas de recombinação e mutação. Sua vantagem é que se no caso ótimo global for descoberto durante o processo de busca, o algoritmo deve convergir para tal solução

Portanto, existe uma tendência de convergência rápida para uma região de mínimos locais ao invés de mínimos globais. Para que isso não ocorra, impõe-se uma rotina para explorar outras áreas do espaço de busca por meio de alterações nos genes por meio da mutação. A probabilidade de se efetuar uma mutação deve ser relativamente baixa, caso contrário o algoritmo se comportará fazendo uma busca aleatória

2.4 Simulated Annealing

Outra abordagem para solução do PVC se baseia da técnica de otimização conhecida por *Simulated Annealing* (SA), técnica que foi usada para simular em um computador o processo de *"annealing"* de cristais, processo este de resfriamento gradativo de materiais a partir de uma alta temperatura levando-os as estados mínimos de energia.

A ideia de aplicar este método para resolver problemas de otimização combinatória foi proposta por Kirkpatrick et al. (1983).

No recozimento, os metais são aquecidos a altas temperaturas e depois resfriados lentamente, permitindo que os átomos se reorganizem em uma estrutura mais estável. Da mesma forma, o SA começa com uma solução inicial e, em seguida, perturba essa solução de maneira controlada para explorar o espaço de busca (KIRKPATRICK et al., 1983).

O fato do método *Simulated Annealing* permitir a aceitação de configurações intermediárias do problema em que cresce o valor da função objetivo que se deseja minimizar é crucial. Essa aceitação temporária de soluções "piores", significa que o método admite caminhar "morro acima", na esperança de encontrar "vales" mais profundos (BENEVIDES et al., 2011).

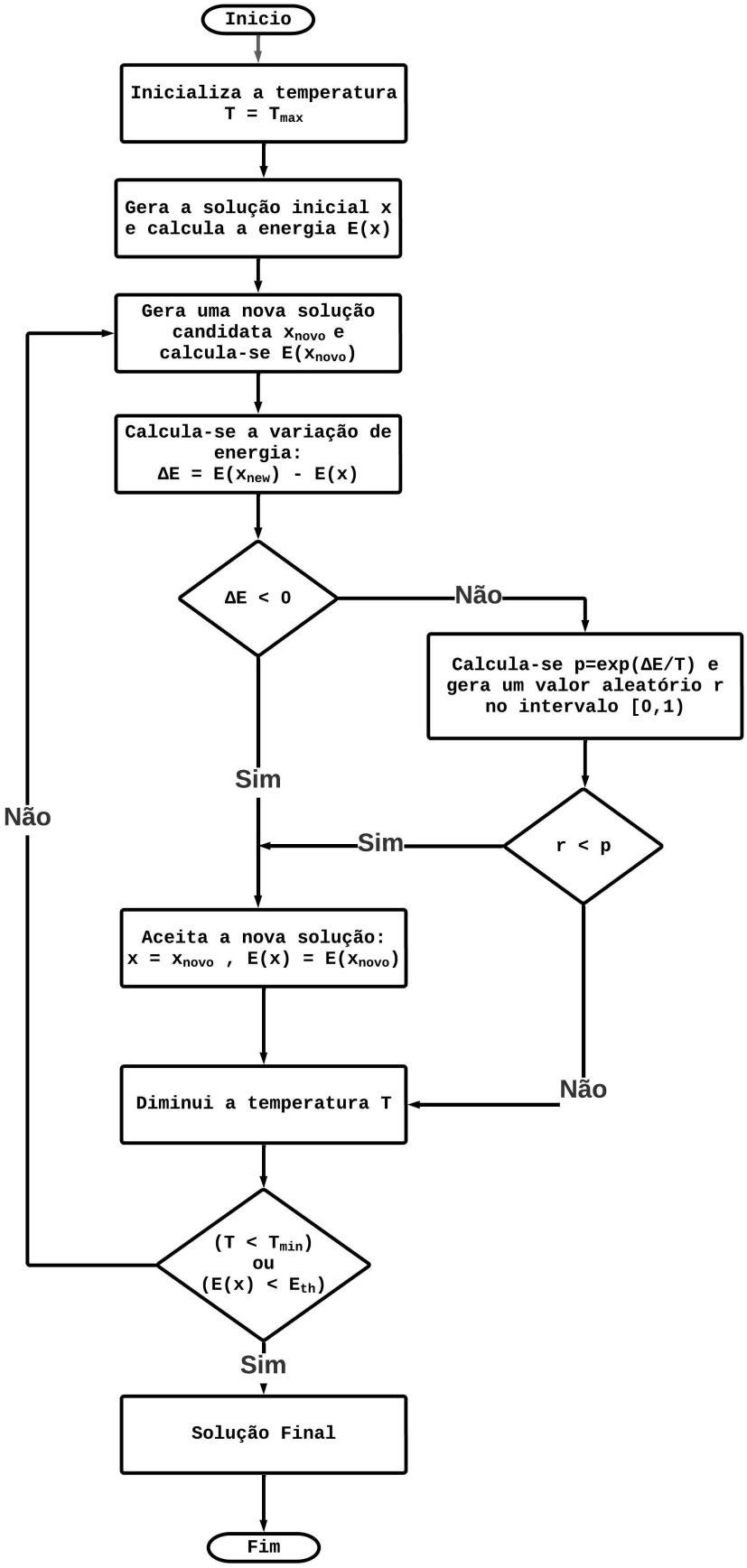
O SA é especialmente útil para problemas de otimização complexos, onde métodos de busca local podem ficar presos em mínimos locais. Ao permitir movimentos que aumentam a função objetivo (ou seja, pioram a solução), o SA pode escapar desses mínimos locais e explorar mais completamente o espaço de busca. A probabilidade de aceitar uma solução pior diminui gradualmente à medida que o "resfriamento" do algoritmo prossegue, de forma análoga ao processo de recozimento (CRISTINA et al., 2023).

O SA tem sido aplicado com sucesso a uma variedade de problemas complexos. Por exemplo, tem sido utilizado na criptografia e no projeto de quantizadores vetoriais. Dentre as aplicações práticas mais conhecidas do Problema do Caixeiro Viajante, destacam-se a sequencia das operações de máquinas em manufatura, otimização de perfurações de furos em placas de circuitos impressos e a maioria dos problemas de roteamento de veículos (NASCIMENTO et al., 2004).

Podendo convergir para aplicações mais específica, como a otimização rotas de entregas de materiais em uma rede hospitalar (FONSECA et al., 2020), otimização de rotas em uma concessionária de energia elétrica (FERREIRA, 2020), até o planejador de roteiros turísticos (BISPO, 2018).

No entanto, apesar de suas vantagens, o SA também tem suas desvantagens. Por ser um método estocástico, o SA pode ser mais lento do que outros métodos de otimização. Além disso, o desempenho do SA pode ser sensível à escolha dos parâmetros, como a temperatura inicial e a taxa de resfriamento. Na Figura 4 demonstra um fluxograma genérico do SA.

Figura 4 – Exemplo genérico Simulated Annealing.



Fonte: O Autor.

2.5 Python

O Python é uma linguagem de programação de alto nível e de propósito geral. Criada por Guido van Rossum em 1982 (SILVA; SILVA, 2019), para evitar as complexidades e os problemas da linguagem C. Lançada pela primeira vez em 1991. Desde então, tornou-se uma das linguagens de programação mais populares do mundo devido à sua simplicidade, legibilidade e vasta gama de aplicações.

Como os principais destaques do Python, temos sua sintaxe limpa e facilmente legível, tornando o código mais amigável, facilitando suas manutenções futura. Isso faz com que seja uma ótima escolha para iniciantes em programação, mas também é utilizada por desenvolvedores experientes em várias áreas, como desenvolvimento web, ciência de dados, automação, inteligência artificial, entre outros.

Além da legibilidade, Python é conhecido por sua extensa biblioteca padrão, que oferece uma vasta gama de módulos e ferramentas para facilitar o desenvolvimento de aplicativos em diversas áreas. Esses módulos cobrem desde manipulação de arquivos até desenvolvimento de interfaces gráficas de usuário.

Outro aspecto importante do Python é sua comunidade ativa e engajada. Existem milhares de pacotes de terceiros disponíveis por meio do PyPI (Python Package Index), o que permite aos desenvolvedores acessar uma ampla gama de funcionalidades adicionais para seus projetos, além de redes sociais focadas em desenvolvedores, para assistência e aprimoramento dos códigos.

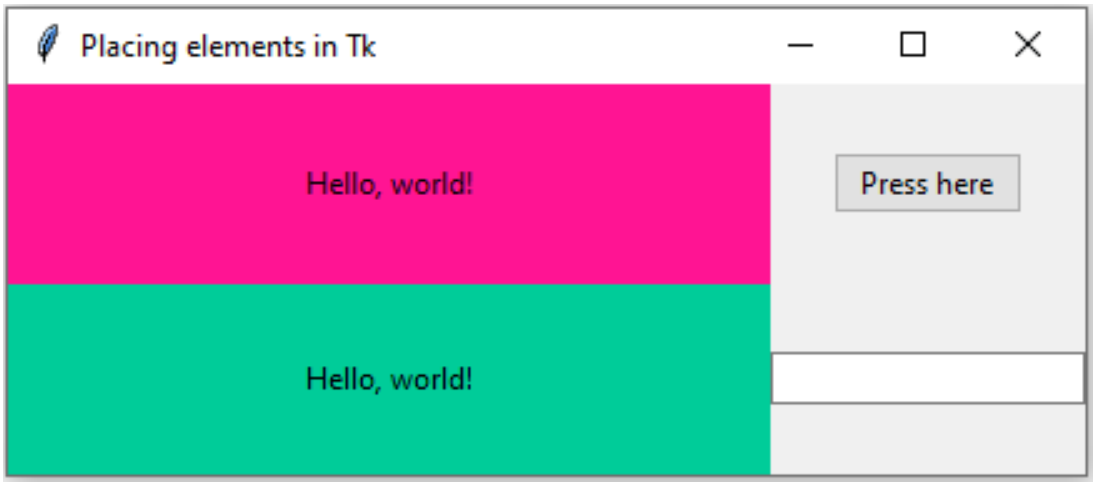
2.5.1 Interface gráfica em Python

O Python oferece diversas bibliotecas para criação de interfaces gráficas, como Tkinter, PyQt, wxPython e Kivy. Cada uma dessas bibliotecas tem suas próprias características e vantagens, sendo escolhida de acordo com as necessidades específicas do projeto.

Sendo uma das mais populares a Tkinter, que é uma biblioteca nativa do Python e proporciona uma maneira simples de construir interfaces gráficas para aplicativos *desktop*.

A Tkinter permite a criação de janelas, botões, caixas de texto, menus e outros componentes de interface de forma intuitiva e eficiente, Figura 5. Com ela, é possível desenvolver aplicações com uma variedade de recursos visuais, tornando a interação do usuário mais amigável e produtiva.

Figura 5 – Exemplo genérico Tkinter.



Fonte: O Autor.

Com a crescente popularidade de *Python* e sua vasta gama de bibliotecas para desenvolvimento de interfaces gráficas, criar aplicativos *desktop* com uma interface visual atraente e funcional tornou-se mais acessível e eficiente para desenvolvedores de todos os níveis de experiência.

3 Desenvolvimento

3.1 Utilização dos Algoritmos

A fim de realizar comparações, dois métodos para solução do Problema do Caixeiro viajante (PCV) foram montadas, sendo uma solução por meio da busca exaustiva e outra por meio do algoritmo genético. Todos os códigos abaixo foram executados em uma máquina com as seguinte configurações:

- **Processador:** Processador Intel(R) Core(TM) i3-4330 CPU @ 3.50GHz 3.50 GHz
- **Memória RAM:** 10,0 GB
- **Tipo de Sistema:** Windows 10 Pro, Sistema operacional de 64 bits, processador baseado em x64

3.2 Busca exaustiva

A maneira mais eficiente de encontrar uma solução para o Problema do Caixeiro Viajante (PCV) é através da avaliação de todas as possibilidades de rota. Portanto, o código necessário deve ser o mais simples possível. Nesse sentido, a metodologia proposta envolve a utilização de uma matriz de distâncias das cidades selecionadas para os testes, além das permutações das cidades, que representam as diferentes rotas a serem consideradas. Com essas informações disponíveis, é possível calcular o custo de cada rota e, em seguida, identificar a rota de menor e maior custo, bem como determinar o tempo necessário para a realização da simulação.

```
%clc
clear all
tic

% Dados Cidades N = 10
% 1  - UEL, Londrina
% 2  - Ibiporã
% 3  - Maringá
% 4  - Cascavel
% 5  - Paranagua
% 6  - Pérola
% 7  - Palmas
% 8  - Brasilandia do Sul
% 9  - Sengés
% 10 - Santo Antonio do Sudoeste
```

```
DistMatriz = [0      21     98     378    485    309    525    319    308    534;
              21     0      111    391    498    322    538    332    285    547;
              98     111    0      283    523    212    508    222    399    439;
              378    391    283    0      603    171    310    106    556    163;
              485    498    523    603    0      693    450    697    371    657;
              309    322    212    171    693    0      497    63     638    327;
              525    538    508    310    450    497    0      409    493    221;
              319    332    222    106    697    63     409    0      648    262;
              308    285    399    556    371    638    493    648    0      622;
              534    547    439    163    657    327    221    262    622    0];

VetorIndi = [1 2 3 4 5 6 7 8 9 10];

%permutação dos indices
Pi = perms(VetorIndi);

% Variavel da Distancia
distanciamenor  = inf;
distanciamaior = 0;

% Calculo da rota
VetorDistancias = zeros(length(Pi),1);
Indice_Rota_menor = 0;
Indice_Rota_maior = 0;

for i = 1:length(Pi)
    rota = Pi(i, :); % Armazena a rota atual
    distancia   = 0;

    for j = 1:length(rota)
        Cidade_partida = rota(j);
        Cidade_chegada = rota(mod(j, length(rota)) + 1);
        distancia = distancia + DistMatriz(Cidade_partida, Cidade_chegada);
    end

    % Armazenador de distancias
    VetorDistancias(i,1) = distancia;

    % Comparadores
    if distancia < distanciamenor
        distanciamenor = distancia;
        Indice_Rota_menor = i;
        Rota_menor = rota;
    end
    if distancia > distanciamaior
        distanciamaior = distancia;
```

```
        Indice_Rota_maior = i;
        Rota_maior = rota;
    end

end

% Exibi as distâncias
disp(['A distância total da menor rota é e a rota é a:']);
distanciamenor
Indice_Rota_menor
Rota_menor
disp(['A distância total da maior rota ée a rota é a:']);
distanciamaior
Indice_Rota_maior
Rota_maior


toc
```

3.3 Algoritmo Genético

O algoritmo genético proposto busca solucionar o Problema do Caixeiro Viajante (PCV) aplicando a técnica de otimização baseada em processos evolutivos. O código abaixo descreve a implementação dessa metodologia.

Na etapa de inicialização, é necessário definir as cidades a serem testadas, bem como os parâmetros necessários. Isso inclui determinar o tamanho da população inicial e o número máximo de gerações. Para cada cromossomo na população inicial, um vetor de 11 posições é gerado, com os índices das cidades distribuídos aleatoriamente.

Em seguida, para o processo de evolução, em cada população, o cálculo do percurso total é realizado. Os dois melhores indivíduos são selecionados como pais da próxima geração. Cada pai gera dois filhos através da permutação de duas posições. Posteriormente, ocorre a atualização da população.

Como resultado, o código proposto foi desenvolvido para fins comparativos com o método de força bruta. Ele produz observações semelhantes, como o tempo de execução e o menor percurso encontrado ao longo das gerações.

3.4 Avaliação do Desempenho Computacional

Avaliar comparativamente o desempenho computacional e cada método

3.5 Aplicando refinamentos nos modelos

Aplicar melhorias necessárias em cada método

3.6 Desenvolvimento do Software

Explicar a linha do tempo para o desenvolvimento do software

3.6.1 Funcionamento Estrutural (back-end)

Explicar especificamente o back end

3.6.2 Interface Gráfica (Front-end)

Explicar especificamente o front end

4 Resultados

4.1 Resultados(trocar?) da avaliação Computacional

4.2 Resultados(trocar?) após os ajustes computacionais

4.3 Funcionamento do Aplicativo desenvolvido em Python

5 Discussões e Conclusões

5.1 Sugestões de Trabalhos Futuros

Referências

DA Silva, Rogério Oliveira and Silva, Igor Rodrigues Sousa. 29

ARAÚJO, W. L. d. *Projeto ideal de suspensão considerando modelo de um quarto de veículo empregando algoritmos genéticos*. Dissertação (B.S. thesis), 2021. 33

BALLOU, R. *Business Logistics/supply Chain Management: Planning, Organizing, and Controlling the Supply Chain*. Pearson/Prentice Hall, 2004. ISBN 9780130661845. Disponível em: <https://books.google.com.br/books?id=sgsdQAAACAAJ>. 27

BARBOSA, D.; JR., C.; KASHIWABARA, A. Aplicação da otimização por colônia de formigas ao problema de múltiplos caixeiros viajantes no atendimento de ordens de serviço nas empresas de distribuição de energia elétrica. In: *Anais do XI Simpósio Brasileiro de Sistemas de Informação*. Porto Alegre, RS, Brasil: SBC, 2015. p. 23–30. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/sbsi/article/view/5797>. 28

BARBOZA; OLANDOSKI, A. Simulação e técnicas da computação evolucionária aplicadas a problemas de programação linear inteira mista. Centro Federal de Educação Tecnológica do Paraná, 2005. Disponível em: <http://repositorio.utfpr.edu.br:8080/jspui/handle/1/74>. 34

BARRETO, J. M. Inteligência artificial no limiar do século xxi. *Florianópolis: PPP edições*, v. 97, 1999. 33

BENEVIDES, P. F. et al. Aplicação de algoritmos genéticos e simulated annealing para o problema do caixeiro viajante em uma situação real de distribuição de produtos. Brasil, 2011. 15, 28, 35

BISPO, R. C. Planejador de roteiros turísticos: uma aplicação do problema do caixeiro viajante na cidade do recife. Brasil, 2018. Disponível em: <http://repository.ufrpe.br/handle/123456789/722>. 35

CAMPOS, R. S. Desmistificando a inteligência artificial: Uma breve introdução conceitual ao aprendizado de máquina. *Aoristo-International Journal of Phenomenology, Hermeneutics and Metaphysics*, v. 3, n. 1, p. 106–123, 2021. 32

CARVALHO, L. et al. Redução de custo com combustível para uma frota. p. 55–62, 1980. Disponível em: <https://periodicos.set.edu.br/cadernoexatas/article/view/896/725>. 27

CRISTINA, A. et al. A TERMODINÂMICA COMO ALGORITMO DE OTIMIZAÇÃO. n. 2015, 2023. 35

Deep Learning Book. *Capítulo 6 – O Perceptron – Parte 1*. 2021. Acesso em: 20 de fevereiro de 2024. Disponível em: <https://www.deeplearningbook.com.br/o-perceptron-parte-1/>. 15, 31

DIAS. *Inteligência Artificial na logística: como funciona e seus benefícios*. 2022. Disponível em: <https://www.trackage.com.br/blog/inteligencia-artificial-na-logistica/>. Acesso em: 01 de março de 2023. 27, 30

EBERHART, R. C. *Neural network PC tools: a practical guide*. [S.l.]: Academic Press, 1990. 30

ecommercebrasil. *Entenda com o Grupo Intelipost como fazer uma roteirização inteligente*. 2023. Disponível em: <https://www.ecommercebrasil.com.br/noticias/grupo-intelipost-roteirizacao-inteligente>. Acesso em: 01 de março de 2023. 30

FERREIRA, I. A. Caixeiro viajante: aplicação da modelagem matemática na otimização de rotas em uma concessionária de energia elétrica. *Revista Produção Online*, v. 20, n. 1, p. 221–246, mar. 2020. Disponível em: <https://www.producaoonline.org.br/rpo/article/view/3491>. 35

FILITTO, D. Algoritmos genéticos: uma visão explanatória. *Revista Multidisciplinar da*, 2008. 34

FONSECA, J. D. d. O. et al. Otimização de rotas de entregas de materiais em uma rede hospitalar por meio do algoritmo do problema do caixeiro viajante. v. 9, 2020. Disponível em: <https://periodicos.uninove.br/revistargss/article/view/16570>. 35

GOUVEIA, L. B. et al. Algoritmos genéticos: aplicando a teoria a um estudo de caso. *Brazilian Journal of Development*, v. 7, n. 3, p. 21053–21077, 2021. 33, 34

HUGO, V. et al. O problema do caixeiro viajante aplicado ao grupo crítico do sistema de entregas de um restaurante. 2017. Disponível em: <http://anais.unespar.edu.br/xi_eepa/data/uploads/artigos/3/3-04.pdf>. 27

KIRKPATRICK, S. et al. Optimization by simulated annealing. *Science*, v. 220, n. 4598, p. 671–680, 1983. Disponível em: <https://www.science.org/doi/abs/10.1126/science.220.4598.671>. 35

LACERDA, E. G. de; CARVALHO, A. D. Introdução aos algoritmos genéticos. *Sistemas inteligentes: aplicaçoes a recursos hıdricos e ciências ambientais*, v. 1, p. 99–148, 1999. 33

LIEBERMAN, F. *Introdução à Pesquisa Operacional*. McGraw Hill Brasil, 2010. ISBN 9788563308283. Disponível em: <https://books.google.com.br/books?id=XM_4wAEACAAJ>. 29

LMXlogistica. *MERCADO LOGÍSTICO PÓS-PANDEMIA: O QUE ESPERAR?* 2022. Acesso em: 16 de fevereiro de 2024. Disponível em: <https://www.lmxlogistica.com.br/mercado-logistico-pos-pandemia-o-que-esperar/#:~:text=Principais%20mudanÃğas%20na%20logÃŋstica%20pÃşs-pandemia&text=A%20crise%20tem%20contribuÃŋdo%20para,transporte%20ferroviÃąrio%2C%20rodoviÃąrio%20e%20aÃľreo.> 27

MARTINS, A. L. Potenciais aplicações da inteligência artificial na ciência da informação. *Informação & Informação*, v. 15, n. 1, p. 1–16, 2010. 32

MCCARTHY, J. WHAT IS ARTIFICIAL INTELLIGENCE ? p. 1–15, 2007. 29

Mobilidade Sampa. *Carnaval 2023: como o feriado impacta nos processos de logística?* 2023. Disponível em: <https://mobilidadesampa.com.br/2023/02/carnaval-2023-como-o-feriado-impacta-nos-processos-de-logistica/>. Acesso em: 01 de março de 2023. 27

MOREIRA, I. I. P. et al. Redes neurais artificiais para previsão de capacidade de carga em estacas do tipo hélice contínua artificial neural networks for load capacity prediction in continuous flight auger piles. *Brazilian Journal of Development*, v. 7, n. 12, p. 112577–112597, 2021. 32

NASCIMENTO, D. B. et al. Análise comparativa de algoritmos heurísticos para resolução do problema do caixeiro-viajante em grafos não clusterizados. 2004. Disponível em: <https://abepro.org.br/biblioteca/ENEGEP2004_Enegep0601_0567.pdf>. 35

PACHECO, M. A. C. et al. Algoritmos genéticos: princípios e aplicações. *ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro. Fonte desconhecida*, v. 28, 1999. 17, 33

PINHEIRO, M.; OLIVEIRA, H. Inteligência artificial: Estudos e usos na ciência da informação no brasil. *Revista Ibero-Americana de Ciência da Informação*, v. 15, n. 3, p. 950–968, 2022. 29

PINHEIRO, M. D. M. et al. Perceptron multicamadas: uma ferramenta de aprendizado supervisionado. Niterói, 2021. 31

PRESTES Álvaro. Uma análise experimental de abordagens heurísticas aplicadas ao problema do caixeiro viajante. 2006. Disponível em: <https://repositorio.ufrn.br/handle/123456789/17962>. 29

QUARESMA, A. Inteligência artificial fraca e força bruta computacional. *TECHNO REVIEW. International Technology, Science and Society Review/Revista Internacional De Tecnología, Ciencia Y Sociedad*, v. 10, n. 1, p. 67–78, 2021. 31

Roberto N. Onody. *Teste de Turing e Inteligência Artificial*. 2021. Acesso em: 16 de fevereiro de 2024. Disponível em: <https://www2.ifsc.usp.br/portal-ifsc/teste-de-turing-e-inteligencia-artificial/.> 30

ROBINSON. *On the Hamiltonian Game*. 1949. Disponível em: <https://web.archive.org/web/20200629071813/https://apps.dtic.mil/dtic/tr/fulltext/u2/204961.pdf>. Acesso em: 01 de março de 2023. 28

RUSSELL, S.; NORVIG, P. *Inteligência Artificial*. Rio de Janeiro: Elsevier, 2004. 33

SALMEN, C. S.; WACHOWICZ, M. A atribuição da pessoa jurídica à inteligência artificial: desafios e sua efetividade the attribution of the legal entity to artificial intelligence: challenges and its effectiveness. *Brazilian Journal of Development*, v. 7, n. 7, p. 71438–71457, 2021. 31, 32

SEARLE, J. R. *Mente, Cérebro e Ciência*. Lisboa: Edições 70, 1987. 32

SILVA, G. A. N. d. et al. Algoritmos heuristicos contrutivos aplicados ao problema do caixeiro viajante para definição de rotas otimizadas. *Colloquium Exactarum. ISSN: 2178-8332*, v. 5, n. 2, p. 30–46, nov. 2013. Disponível em: <https://journal.unoeste.br/index.php/ce/article/view/939>. 29

SILVA, G. N.; ARRUDA, N. Teste de turing: Um computador é capaz de pensar? In: *CONGRESSO NACIONAL DE PESQUISA E ENSINO EM CIÊNCIAS (CONAPESC)*. [S.l.: s.n.], 2016. 30

SILVA, R. O. da; SILVA, I. R. S. Linguagem de programação python. *TECNOLOGIAS EM PROJEÇÃO*, v. 10, n. 1, p. 55–71, 2019. 37

SILVEIRA, E. D. da; SOUZA, P. V. S. de; BRITTO, P. A. P. de. Ensaio teórico sobre o uso das redes neurais artificiais no gerenciamento de resultados. *Revista de Gestão e Secretariado (Management and Administrative Professional Review)*, v. 14, n. 1, p. 913–931, 2023. 32

SOUZA, S. S. F.; ROMERO, R. Algoritmo Imunológico Artificial CLONALG e Algoritmo Genético Aplicados ao Problema do Caixeiro Viajante Introdução. v. 2, p. 1–6, 2014. Disponível em: <https://proceedings.sbmac.org.br/sbmac/article/view/307/309>. 28

VÁZQUEZ-GUZMÁN, G.; MARTÍNEZ-RODRÍGUEZ, P. R.; SOSA-ZÚÑIGA, J. M. Inversor monofásico de alta eficiencia sin transformador para aplicaciones fotovoltaicas. *Ingeniería, investigación y tecnología*, Facultad de Ingeniería, UNAM, v. 16, n. 2, p. 173–184, 2015. 33

WANG, K. Logistics 4.0 solution-new challenges and opportunities. In: ATLANTIS PRESS. *6th international workshop of advanced manufacturing and automation*. [S.l.], 2016. p. 68–74. 27

ZADEH, L. A. Information and control. fuzzy sets. *North-Holland*, v. 8, n. 3, p. 15–25, 1965. 32

```
    %clc
clear all
tic

% Dados Cidades N = 10
% 1  - UEL, Londrina
% 2  - Ibiporã               ; Raio de 20km
% 3  - Maringá               ; Raio de 40km
% 4  - Cascavel              ; Raio de 80km
% 5  - Paranagua             ; Raio maior que 160km
% 6  - Pérola
% 7  - Palmas
% 8  - Brasilandia do Sul
% 9  - Sengés
% 10 - Santo Antonio do Sudoeste

DistMatriz = [0      21     98     378    485    309    525    319    308    534;
              21     0      111    391    498    322    538    332    285    547;
              98     111    0      283    523    212    508    222    399    439;
              378    391    283    0      603    171    310    106    556    163;
              485    498    523    603    0      693    450    697    371    657;
              309    322    212    171    693    0      497    63     638    327;
              525    538    508    310    450    497    0      409    493    221;
              319    332    222    106    697    63     409    0      648    262;
              308    285    399    556    371    638    493    648    0      622;
              534    547    439    163    657    327    221    262    622    0];


numCidades = 10;

%indices
tam_Pop_ini = 10;              %Tamanho População Inicial
tam_gera    = 100;             %Nmr de gerações

%Geracao da Populacao Inicial
for k = 1:tam_Pop_ini

    %Gera um vetor de 11 posicoes com numeros de 2 a 10 (cidades) distribuidos aleatoriamente

    [a b] = sort(rand(1,9));
    b = b + 1;

    %Cromossomo
    Populacao(k, :) = [1, b, 1];

end
```

```
for g = 1:tam_gera

    %Calcula o percurso total de cada vetor solucao
    for k = 1:size(Populacao,1),
        Percurso(k, g) = Calc_Dist(DistMatriz, Populacao(k,:));
    end

    %Escolhe os mais aptos
    clear a b;
    [a b] = sort(Percurso(:,g));

    Pai = Populacao(b(1,1), :);
    Mae = Populacao(b(2,1), :);




    %Cada Genitor vai gerar 2 filhos por permuta
    %Sorteia e permuta as duas posicoes do Pai
    for k = 1:4,
        clear a b;
        [a b] = sort(rand(1,9));
        b = b + 1;

        FilhosPai(k, :) = Pai;
        FilhosPai(k, b(1, 1)) = Pai(1, b(1, 2));
        FilhosPai(k, b(1, 2)) = Pai(1, b(1, 1));
    end

    %Sorteia e permuta as duas posicoes da Mae
    for k = 1:4,
        clear a b;
        [a b] = sort(rand(1,9));
        b = b + 1;

        FilhosMae(k, :) = Mae;
        FilhosMae(k, b(1, 1)) = Mae(1, b(1, 2));
        FilhosMae(k, b(1, 2)) = Mae(1, b(1, 1));
    end


    %Atualiza a Populacao
    Populacao = [Pai; Mae; FilhosPai; FilhosMae];

end
```

```
toc
plot(min(Percurso, [], 1));
```