

# Otimização Meta-Heurística

Fundamentos, Ferramentas e Caminhos para a Resolução de Problemas

Victor Guerreiro Pimenta



**UFG**

UNIVERSIDADE  
FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)  
INSTITUTO DE INFORMÁTICA (INF)

VICTOR GUERREIRO PIMENTA

## **Otimização Meta-Heurística**

Fundamentos, Ferramentas e Caminhos para a Resolução de Problemas

Goiânia  
2025



UNIVERSIDADE FEDERAL DE GOIÁS  
INSTITUTO DE INFORMÁTICA

## TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

### 1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): VICTOR GUERREIRO PIMENTA

Título do trabalho: Otimização Meta-Heurística

Fundamentos, Ferramentas e Caminhos para a Resolução de Problemas

### 2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento SIM NÃO<sup>1</sup>

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

#### Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro.

**Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.**



Documento assinado eletronicamente por **Victor Guerreiro Pimenta, Discente**, em 17/01/2025, às 12:48, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Fernando Marques Federson, Professor do Magistério Superior**, em 17/01/2025, às 17:17, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

---



A autenticidade deste documento pode ser conferida no site [https://sei.ufg.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **5089811** e o código CRC **FCA3DA58**.

---

Referência: Processo nº 23070.001602/2025-12

SEI nº 5089811

VICTOR GUERREIRO PIMENTA

## **Otimização Meta-Heurística**

Fundamentos, Ferramentas e Caminhos para a Resolução de Problemas

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Orientador: Prof. Dr. Fernando Marques Federson

Goiânia

2025

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

PIMENTA, VICTOR GUERREIRO

Otimização Meta-Heurística [manuscrito] : Fundamentos, Ferramentas e Caminhos para a Resolução de Problemas / VICTOR GUERREIRO PIMENTA. - 2025.

229 f.

Orientador: Prof. Dr. Fernando Marques Federson.

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2025.

1. inteligência artificial. 2. otimização. 3. metaheurísticas. I. Federson, Fernando Marques , orient. II. Título.

CDU 004

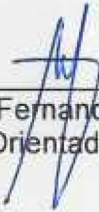
VICTOR GUERREIRO PIMENTA

## Otimização Meta-Heurística

Fundamentos, Ferramentas e Caminhos para a Resolução de Problemas

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 17 de dezembro de 2024.



---

Prof. Dr. Fernando Marques Federson  
Orientador (INF-UFG)



---

Prof. Dr. Aldo André Díaz Salazar  
Coordenador de TCC do BIA (INF-UFG)



---

Prof. Dr. Anderson da Silva Soares  
Coordenador do BIA (INF-UFG)



---

Profa. Dra. Telma Woerle de Lima Soares  
(INF-UFG)

VICTOR GUERREIRO PIMENTA

## **Otimização Meta-Heurística**

Fundamentos, Ferramentas e Caminhos para a Resolução de Problemas

### **RESUMO**

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Otimização Meta-Heurística**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: inteligência artificial, modelos grandes de linguagem, geração automática de datasets.

### **ABSTRACT**

This Course Completion Report aims to bring together the results of my journey to become an expert in **Meta-Heuristic Optimization**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: artificial intelligence, large language models, automatic dataset generation.

Goiânia

2025

# Minha Jornada



Victor Guerreiro Pimenta  
Especialista em: Otimização Meta-Heurística

## MINHA JORNADA

**Nome:** Victor Guerreiro Pimenta

**Especialidade:** Otimização Meta-Heurística

### Objetivo deste documento

Durante o processo da disciplina Residência em IA<sup>1</sup>, foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

### Minha Jornada

Minha jornada teve início na **Semana 1**, com a leitura e compreensão dos temas principais do ICAI 2024. Durante esse período, defini como área de especialização "**Inteligência Artificial e Algoritmos Evolucionários**". Esse processo envolveu uma busca inicial por livros e literatura científica que abordassem a história, fundamentos e aplicações dos algoritmos evolutivos. Também analisei possíveis temas de interseção, como **Stochastic Optimization, Simulated Annealing e Heuristic Optimization Techniques**. Essa etapa culminou na **Relação de conteúdos sobre Algoritmos Evolucionários, no Apêndice 1**. Na **Semana 2**, um **resumo técnico sobre metaheurísticas** foi produzido e também está documentado no **Apêndice 1**. Este material serviu como base para compreender os desafios intrínsecos dessas técnicas, incluindo definições, exemplos, classificações detalhadas e estudos de caso relevantes. A análise destacou a aplicabilidade das metaheurísticas em diferentes cenários, consolidando-as como uma escolha estratégica para a especialização em otimização para diversos cenários.

A decisão de priorizar técnicas de otimização heurística foi fundamentada em sua flexibilidade e aplicabilidade em problemas que demandam soluções rápidas e eficazes. Metodologias como **Simulated Annealing, Algoritmos Evolutivos e Swarm Intelligence** se destacam por balancear simplicidade e desempenho, sendo ideais para cenários onde soluções exatas não são prioridade. Essas heurísticas englobam abordagens inspiradas pela natureza, como otimização por enxame e algoritmos biológicos, oferecendo robustez e

---

<sup>1</sup> Dez semanas, entre setembro de 2024 e dezembro de 2024.

versatilidade na resolução de problemas complexos sem a necessidade de garantias formais. Uma análise sobre essa mudança está no documento **Análise e Escolha do Tema**, no **Apêndice 1**.

Na **Semana 3**, aprofundi os estudos nos **Capítulos 1 (Introdução)**, **8 (Tópicos em Estratégias Evolutivas)** e **23 (Otimização Multiobjetivo)** do livro *Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature*. Esses capítulos proporcionaram uma visão abrangente desde os conceitos introdutórios das metaheurísticas até tópicos avançados, como estratégias evolutivas e desafios associados à otimização multiobjetivo. Em paralelo, explorei a **Parte 3 (Case Studies)** do livro *Metaheuristics for Hard Optimization*, que apresenta estudos de caso detalhados sobre a aplicação de metaheurísticas em problemas complexos do mundo real.

Com base nesses estudos, dei continuidade ao **Resumo Técnico sobre Metaheurísticas**, criando uma nova versão (v0.1), onde ampliei definições e incorporei tópicos mais avançados sobre estratégias multiobjetivo e desafios técnicos, que está anexada no **Apêndice 2**. Além disso, elaborei dois documentos complementares: o primeiro sobre os principais **desafios das técnicas de otimização heurística**, incluindo tópicos como otimização multiobjetivo e paralelismo; e o segundo focado no entendimento dos **problemas e campos de aplicação** de cada técnica, detalhando as áreas onde essas metodologias podem ser implementadas e seus respectivos impactos. Essas atividades fortaleceram meu domínio sobre os conceitos e aplicações das metaheurísticas, consolidando a base para a continuação do trabalho.

Durante a **Semana 4**, realizei a **busca e análise de referências adicionais** extraídas do livro *"Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature"*. Essa investigação teve como foco estabelecer uma **relação entre problemas e os métodos utilizados** em algoritmos de otimização heurística, contribuindo para uma visão mais aprofundada sobre como essas técnicas são aplicadas em diferentes contextos. Além disso, o processo permitiu identificar avanços teóricos e práticos que sustentam a eficiência das metaheurísticas. Os resultados dessa análise foram organizados em dois documentos principais, que se encontram no **Apêndice 2**. O primeiro, **Referências Fundamentos**, reúne citações e conceitos essenciais que explicam as bases teóricas e as inspirações naturais que moldaram as metaheurísticas. Já o segundo, **Relação Problemas/Métodos**, explora a correspondência entre desafios práticos e as técnicas mais apropriadas de otimização heurística, demonstrando como cada abordagem pode ser aplicada estrategicamente. Esses documentos são projetados para servir como uma **ferramenta de consulta fundamental**, oferecendo suporte a estudantes, pesquisadores e profissionais interessados em compreender o campo de forma mais abrangente.

Na **Semana 5**, a pesquisa foi focada na busca de **artigos recentes** de 2023 a 2024, utilizando as **strings de busca** documentadas no **Relatório de Busca de Artigos** no

**Apêndice 3.** Esse processo permitiu a atualização da **Relação Problemas/Métodos**, refletindo sobre a aplicação de diferentes **metaheurísticas** para problemas específicos. Além disso, a pesquisa aprofundou a **etimologia** de conceitos chave, como **heurística**, **metaheurística** e **hiper-heurística**. A origem histórica das heurísticas foi explorada, ligando o termo à ideia de **descoberta** associada à famosa expressão "Eureka!" de Arquimedes, e sua definição como estratégias simples baseadas em **experiência**. O estudo também apresentou as **metaheurísticas** como um nível superior de abstração e discutiu as **hiper-heurísticas** como uma abordagem ainda mais avançada, que combina múltiplas **metaheurísticas** para otimização. Todos esses conceitos estão detalhados nos arquivos **Etimologia de Meta-Heurística** e **Relação Problemas/Métodos v0.1**, ambos disponíveis no **Apêndice 3**.

Na **Semana 6**, o foco foi no aprofundamento do estudo de **técnicas específicas de metaheurísticas**, como **Simulated Annealing**, **Genetic Algorithms**, **Particle Swarm Optimization**, entre outras. Essas técnicas foram analisadas detalhadamente, com uma ênfase particular nas suas aplicações práticas e teóricas, conforme descrito no arquivo **Técnicas Específicas de Metaheurísticas**, também disponível no **Apêndice 3**. Este trabalho visou entender as vantagens e desvantagens de cada abordagem, promovendo uma compreensão mais rica sobre como essas técnicas podem ser aplicadas a problemas complexos de otimização.

Na **Semana 7**, foi realizado o **screening de códigos, frameworks e implementações práticas** de metaheurísticas disponíveis em repositórios, como o **GitHub**. Foram estabelecidos critérios de avaliação como **recência**, **número de contribuições**, **qualidade da documentação**, **número de estrelas** atribuídas pelos usuários, **licença**, e a presença de **testes unitários**. A partir dessa análise, foi possível compilar uma lista dos projetos mais relevantes, que estão documentados no arquivo **Busca de códigos heurísticas** no **Apêndice 4**, com links para os repositórios e suas respectivas categorizações. Esses repositórios foram selecionados com base em sua atualidade e qualidade, permitindo uma visão abrangente das ferramentas mais eficazes e acessíveis para implementar algoritmos de metaheurísticas.

Na **Semana 8**, a pesquisa avançou para a **avaliação prática** dessas ferramentas. Foram realizados **testes e avaliações** de frameworks como **MealPy**, **Simanneal**, e **PyGAD**, entre outros, com foco nas funcionalidades oferecidas por cada um e sua compatibilidade com os algoritmos estudados. O framework **MealPy** foi destacado por permitir a definição de uma **classe de problema personalizada** com tipos de variáveis da solução, função objetivo e parâmetros de otimização, mas não inclui **Simulated Annealing**, operando somente com métodos baseados em população. Já o **Simanneal** exige a definição explícita de **movimento** e **energia**, sendo mais focado em problemas específicos de otimização. O **PyGAD** se destacou pelo suporte a **Deep Learning** com **Keras** e **PyTorch**, enquanto o

**gplearn** e **ellyn** são voltados para **regressão simbólica** e oferecem suporte para integração com a **API do sklearn**. Também foram analisados outros frameworks, como o **openai/evolution-strategies-starter**, que implementa o código para o artigo sobre **estratégias evolutivas** aplicadas ao aprendizado por reforço. Todos esses detalhes estão registrados no arquivo **Mapeamento de códigos, frameworks, e implementações**, no **Apêndice 4**, que reúne informações detalhadas sobre as funcionalidades e a aplicabilidade de cada framework testado.

Na **Semana 9**, o trabalho foi focado na **modelagem de problemas de otimização** utilizando o framework **MealPy**. Foi possível desenvolver **classes personalizadas** para definir **variáveis**, **funções objetivo** e **parâmetros de otimização**, aplicando esses conceitos a problemas reais. Para testar e validar as modelagens, explorei diversas **bases de dados** e **simuladores específicos**, como **OpenCellID**, utilizado para localização de torres de celular e cobertura de sinal, **BlueSky** para simulações de cenários de tráfego aéreo, e **TSPLIB** para problemas combinatórios. Outros recursos, como **OpenStreetMap** e **Geonames**, forneceram dados geográficos valiosos, enquanto fontes especializadas em **dados financeiros** e **biologia estrutural**, como **Yahoo Finance** e **Protein Data Bank**, foram exploradas para ampliar as aplicações. O progresso desta modelagem foi registrado no arquivo **"Bases de Dados e Simuladores"**, documentado no **Apêndice 5**.

Na **Semana 10**, a atenção se voltou para o **estudo comparativo de algoritmos de otimização**, aplicados aos problemas modelados anteriormente. Testei diversos algoritmos de **metaheurísticas** em problemas clássicos como o **Problema do Caixeiro Viajante (TSP)** e o **Problema de Roteamento de Veículos (VRP)**, avaliando seu desempenho, especialmente em relação ao **tempo de convergência** e **eficiência**. Além disso, a análise foi complementada com a criação de um gráfico do **panorama geral** das ferramentas e técnicas utilizadas ao longo do estudo, sintetizando as fundamentações e as aplicações abordadas. Esses resultados estão descritos no **documento "Estudo Comparativo"** e no **"Panorama da Otimização Meta-Heurística"**, ambos localizados no **Apêndice 6**.

Essa jornada não só ampliou meu entendimento técnico sobre **otimização meta-heurística**, mas também reforçou a importância de uma abordagem estruturada na pesquisa. Cada etapa, desde a escolha do tema até a análise de frameworks e algoritmos, contribuiu para uma compreensão mais profunda e aplicada dos desafios e soluções em otimização. Todos os materiais gerados ao longo do trabalho estão organizados em **Apêndices**, servindo como uma base valiosa para futuras consultas e reflexões. Minha trajetória, que começou com **algoritmos evolutivos**, evoluiu para uma visão abrangente e detalhada sobre o campo da **otimização meta-heurística**, destacando a aplicabilidade e eficiência das técnicas em cenários reais e desafiadores.

## APÊNDICE 1

## Termo de Aceite de Entrega

### Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

**Data da Reunião (“gate”) de aprovação:** 18 de set. de 2024

**Participantes da Entrega** [matriculados em Residência em IA]:

Victor Guerreiro Pimenta

**Entrega:** [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Atividades realizadas:

- Leitura e conhecimento dos temas do ICAI 2024.
- Decisão do tema de especialização na residência:  
**Inteligência Artificial E Algoritmos Evolucionários**
- Busca de livros e de literatura científica sobre história, fundamentos e aplicações dos AE.
  - [https://docs.google.com/document/d/1cwk8\\_IlyROnVNgn5qUcCRTsdyTIY\\_heJoQ2UDJUVug/edit?usp=sharing](https://docs.google.com/document/d/1cwk8_IlyROnVNgn5qUcCRTsdyTIY_heJoQ2UDJUVug/edit?usp=sharing)
- Consideração de possíveis temas relacionados / interseccionados em outros níveis de organização:
  - - Stochastic optimization
  - - Simulated annealing
  - - Heuristic optimization techniques

**Planejamento:** [descrever o que pretende fazer para realizar a próxima ENTREGA]

Atividades Previstas:

- Catalogação e Resumos Técnicos dos Conteúdos.
- Consolidação da escolha do tema em relação aos níveis de organização.
- Busca sobre problemas e desafios intrínsecos aos AE.
- Entendimento maior de problemas e campos onde se aplicam os AE.

**Observação:** [caso precise fazer alguma observação, de qualquer “natureza”]

## ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go!](#)

[Relação de conteúdos sobre Algoritmos Evolucionários citado no Termo de Aceite de Entrega de 18 de setembro]

Residência em IA  
Discente: Victor Guerreiro Pimenta

## Relação de conteúdos sobre Algoritmos Evolucionários

### **Conceituais:**

**Introduction to Evolutionary Algorithms**

<https://link.springer.com/book/10.1007/978-1-84996-129-5>

**Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature**

<https://link.springer.com/book/10.1007/978-3-319-41192-7>

**2019 Evolutionary Algorithms Review**

<https://arxiv.org/pdf/1906.08870>

**Data-Driven Evolutionary Optimization**

<https://link.springer.com/book/10.1007/978-3-030-74640-7>

**Metaheuristics for Hard Optimization : Simulated Annealing, Tabu Search, Evolutionary and Genetic Algorithms, Ant Colonies ... Methods and Case Studies**

<https://link.springer.com/book/10.1007/3-540-30966-7>

**Engineering Optimization: Theory and Practice, Fourth Edition**

<https://industri.fatek.unpatti.ac.id/wp-content/uploads/2019/03/018-Engineering-Optimization-Theory-and-Practice-Singiresu-S.-Rao-Edisi-4-2009.pdf>

**Evolutionary and Swarm Intelligence Algorithms**

<https://link.springer.com/book/10.1007/978-3-319-91341-4>

**Evolutionary Optimization Algorithms**

<https://www.wiley.com/en-mx/Evolutionary+Optimization+Algorithms-p-9780470937419>

---

**Hierarchical Bayesian Optimization Algorithm: Toward a new Generation of Evolutionary Algorithms**

<https://link.springer.com/book/10.1007/b10910>

**Recent Advances in Swarm Intelligence and Evolutionary Computation**

<https://link.springer.com/book/10.1007/978-3-319-13826-8>

**Intelligent Optimisation Techniques. Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks**

<https://link.springer.com/book/10.1007/978-1-4471-0721-7>

### ***Frameworks e código:***

**Genetic Algorithms and Machine Learning for Programmers: Create AI Models and Evolve Solutions**

<https://www.amazon.com/Genetic-Algorithms-Machine-Learning-Programmers/dp/168050620X>

**Genetic Algorithms with Python**

<https://www.amazon.com/Genetic-Algorithms-Python-Clinton-Sheppard/dp/1540324001>

### ***Aplicações:***

**Genetic Algorithms and Engineering Optimization (Engineering Design and Automation)**

<https://www.amazon.com/Genetic-Algorithms-Engineering-Design-Automation/dp/0471127418>

**Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming**

<https://dl.acm.org/doi/book/10.5555/554112>

**Application of Evolutionary Algorithms for Multi-objective Optimization in VLSI and Embedded Systems**

<https://link.springer.com/book/10.1007/978-81-322-1958-3>

**Multi-objective optimization of solar resource allocation in radial distribution systems using a refined slime mold algorithm**

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11200297/>

## Termo de Aceite de Entrega

### Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

**Data da Reunião (“gate”) de aprovação:** 25 de set. de 2024

**Participantes da Entrega** [matriculados em Residência em IA]:

Victor Guerreiro Pimenta

**Entrega:** [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Atividades Realizadas:

- Catalogação e Resumos Técnicos dos Conteúdos selecionados sobre Metaheurísticas.  
    📄 [Resumo Técnico Sobre Metaheurísticas](#)
- Consolidação da escolha do tema em relação aos níveis de organização.  
Mudança para **Técnicas de Otimização Heurística**  
[Análise e Escolha do Tema](#)

**Planejamento:** [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Continuar e Ampliar Catalogação e Resumos
- Busca sobre problemas e desafios intrínsecos às **Técnicas de Otimização Heurística**.
- Entendimento maior de problemas e campos onde se aplicam as **Técnicas de Otimização Heurística**.

**Observação:** [caso precise fazer alguma observação, de qualquer “natureza”]

## ACEITE DA ENTREGA:

LEONARDO ALVES: Go! ▾

[Análise e Escolha do Tema citado no Termo de Aceite de Entrega de 25 de setembro]

## **Análise e Escolha do Tema**

Ao considerar as várias abordagens para a otimização, é importante entender como cada técnica se posiciona dentro desse campo e quais são suas aplicações práticas. A escolha de técnicas de heurística pode ser justificada pela sua flexibilidade, eficiência e adequação a problemas específicos que demandam soluções rápidas e suficientemente boas, sem a necessidade de um rigor matemático intenso.

### **1. IA e Algoritmos Evolucionários (EA)**

Os algoritmos evolucionários são inspirados nos processos biológicos de evolução, como seleção natural e mutação. Eles utilizam populações de soluções e operam iterativamente para encontrar boas respostas a problemas complexos. Embora sejam eficientes para resolver problemas onde a busca exaustiva seria impraticável, esses algoritmos muitas vezes são considerados heurísticos, pois oferecem soluções aproximadas e são ajustados conforme o problema a ser resolvido.

Os algoritmos evolucionários fazem parte de uma classe maior de heurísticas, pois se concentram em encontrar boas soluções através de uma abordagem prática e adaptativa, sem se comprometer com uma solução exata.

### **2. Otimização Estocástica (Stochastic Optimization)**

A otimização estocástica utiliza processos aleatórios como parte integrante de sua busca pela solução ótima. Aqui, a ênfase está no uso de métodos probabilísticos para garantir que o espaço de soluções seja explorado de maneira eficaz. Embora esses métodos sejam poderosos para problemas com incertezas ou com ruído (como em aprendizado de máquina e controle de sistemas), o foco é mais em garantias matemáticas e análise de convergência, o que nem sempre é necessário em aplicações práticas onde boas soluções rápidas são preferíveis.

Ao optar por heurísticas, a otimização estocástica não se torna um foco principal, já que envolve um grau mais alto de rigor e um foco na teoria que pode não ser adequado para aplicações onde a rapidez e a flexibilidade são mais importantes do que a otimização perfeita.

### **3. Simulated Annealing**

O Simulated Annealing é uma técnica inspirada no processo de recozimento em metalurgia, onde a temperatura de um material é lentamente reduzida para minimizar a energia interna. Esse método é considerado uma heurística, já que usa aleatoriedade controlada para escapar de mínimos locais e procurar soluções melhores. Embora compartilhe

características com a otimização estocástica, sua natureza prática e simples o torna uma excelente escolha para resolver problemas complexos de forma eficiente.

O Simulated Annealing é um subconjunto das heurísticas, pois prioriza a busca por uma solução boa o suficiente sem a necessidade de garantias formais sobre a qualidade da solução. Ele equilibra bem a simplicidade e a eficiência, sendo uma técnica bastante versátil em muitos contextos.

#### **4. Técnicas de Otimização Heurística**

As heurísticas englobam uma variedade de técnicas práticas para a resolução de problemas de otimização. Elas são ideais quando é necessário encontrar soluções rápidas e eficazes, sem o rigor matemático exigido por outras abordagens. As heurísticas podem ser adaptadas para problemas específicos, o que as torna muito úteis em áreas como roteamento, escalonamento, design de engenharia e inteligência artificial.

Ao escolher heurísticas, há um foco claro em desenvolver soluções específicas e rápidas para problemas reais. Técnicas como algoritmos evolucionários e Simulated Annealing são subconjuntos dessas abordagens mais amplas, sendo que ambos compartilham a ideia central de encontrar boas soluções aproximadas, mas com uma aplicação prática e orientada para o desempenho.

Ao optar, portanto, pelas técnicas heurísticas, a escolha se alinha com a necessidade de soluções práticas e eficazes, especialmente para problemas onde o tempo de resolução e a simplicidade de implementação são fatores críticos. Diferentemente da otimização estocástica, que requer um rigor matemático maior, as heurísticas como os algoritmos evolucionários e o Simulated Annealing permitem um desenvolvimento mais rápido e adaptável, sem a necessidade de garantir uma otimização exata.

[**Resumo Técnico Sobre Metaheurísticas** citado no Termo de Aceite de Entrega de 25 de setembro]

### **Resumo Técnico Sobre Metaheurísticas**

Análise da organização e descrição dos conteúdos presentes nos livros "**Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature**" e "**Metaheuristics for Hard Optimization**", ambos disponíveis na plataforma Springer. Cada seção foi estruturada com base nos capítulos e tópicos abordados nas obras, fornecendo um panorama detalhado sobre metaheurísticas e suas aplicações. As descrições refletem os principais conceitos, técnicas e estudos de caso discutidos, oferecendo uma visão abrangente e didática sobre as abordagens modernas de otimização inspiradas pela natureza e métodos computacionais avançados.

### **Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature**

<https://link.springer.com/book/10.1007/978-3-319-41192-7>

---

## **Nature-Inspired Computation**

### **Introduction**

- **Definição e exemplos de computação inspirada pela natureza.**
- **Comparação entre processos biológicos e aprendizado.**
- **Swarm Intelligence: comportamento de grupo e a teoria do forrageamento.**

### **Heuristics, Metaheuristics, and Hyper-Heuristics**

- **Diferenciação entre heurísticas, metaheurísticas e hiper-heurísticas.**
- **Métodos de otimização.**
- **Lagrange Multiplier Method, busca baseada em direção, e busca Simplex.**

### **Optimization**

- **Tipos de problemas de otimização, como problemas discretos e contínuos.**
  - **Discussão sobre P, NP, NP-Hard, e NP-Complete.**
  - **Multiobjective Optimization Problem e otimização robusta.**
  - **Teorema No Free Lunch e seus impactos na otimização.**
-

---

## Metaheurísticas Individuais

### Simulated Annealing

- Introdução ao Simulated Annealing básico e suas variantes.

### Genetic Algorithms

- Introdução à computação evolutiva.
- Comparação entre algoritmos evolutivos e Simulated Annealing.
- Terminologias de computação evolutiva (codificação, seleção, reprodução, etc.).
- Exploitation vs Exploration e algoritmos genéticos para otimização de sequência.

### Genetic Programming

- Sintaxe de árvores, controle de crescimento desordenado (bloat), controle de tamanho de programas, e design de operadores genéticos.

### Gene Expression Programming

- Introdução e principais aspectos de programação genética baseada em expressão.

### Evolutionary Strategies

- Busca evolutiva, estratégias evolutivas com base em gradiente e otimização por distribuição de covariância.

### Differential Evolution (DE)

- Introdução, variantes do algoritmo DE, e análise teórica do DE.

### Estimation of Distribution Algorithms (EDA)

- Introdução aos EDAs, variantes, e métodos como Compact Genetic Algorithms e Bayesian Optimization Algorithm.

### Parallel Evolutionary Algorithms

- Modelos paralelos como Master-Slave, Island Model, e Cellular EAs.

#### **Coevolution**

- Abordagens coevolutivas, com foco em otimização minimax e Interactive Evolutionary Computation.

#### **Other Heredity-Based Algorithms**

- Aplicações em otimização de redes neurais.
- 

### **Swarm Intelligence-Based Methods**

#### **Particle Swarm Optimization (PSO)**

- Introdução e variantes do PSO, como Bare-Bones PSO e Gaussian/Cauchy PSO.

#### **Artificial Immune Systems**

- Algoritmos baseados na seleção clonal e algoritmos de seleção negativa.

#### **Ant Colony Optimization (ACO)**

- Introdução ao algoritmo básico de ACO e suas aplicações em otimização contínua.

#### **Bee Metaheuristics**

- Discussão sobre Artificial Bee Colony Algorithm e algoritmos como Marriage in Honeybees Optimization.

#### **Bacterial Foraging Algorithm**

- Forrageamento bacteriano como inspiração para algoritmos de otimização.

#### **Other Swarm Intelligence Algorithms**

- Glowworm Swarm Optimization, Firefly Algorithm, Cuckoo Search, Bat Algorithm, e Krill Herd Algorithm.
-

---

## **Biomolecular and Quantum Computing**

### **Biomolecular Computing**

- **DNA Computing, embutimento de dados, e redes bioquímicas.**

### **Quantum Computing**

- **Introdução ao Grover's Search Algorithm e metaheurísticas híbridas inspiradas pela computação quântica.**

---

## **Metaheuristics Based on Sciences**

- **Otimização baseada nas leis de Newton, princípios eletromagnéticos, fenômenos térmicos, e cosmologia.**
- **Discussão sobre algoritmos como Black Hole-Based Optimization e Chemical Reaction Optimization.**

---

## **Memetic Algorithms**

- **Introdução aos Memetic Algorithms, abordando tópicos como Cultural Algorithms e otimização simplex.**

---

## **Tabu Search and Scatter Search**

- **Iterações de Tabu Search e variantes de Scatter Search.**

---

## **Search Based on Human Behaviors**

- **Algoritmos como Seeker Optimization Algorithm e Teaching–Learning-Based Optimization.**

---

## Dynamic, Multimodal, and Constrained Optimizations

- Métodos para Dynamic Optimization, mantendo a diversidade e abordagens para Multimodal Optimization como Crowding e Fitness Sharing.

---

## Multiobjective Optimization

- Discussão de algoritmos como Nondominated Sorting Genetic Algorithm II (NSGA-II), Strength Pareto Evolutionary Algorithm 2 (SPEA2), e Multiobjective Immune Algorithms.
- Técnicas para otimização multiobjetivo usando Particle Swarm Optimization e EDAs.

---

## Outros Tópicos

- Performance Metrics para algoritmos evolutivos.
- Many-Objective Optimization: Desafios e abordagens.

---

## Conclusão

O livro *Search and Optimization by Metaheuristics* cobre um extenso conjunto de algoritmos de otimização e metaheurísticas inspiradas na natureza. A obra examina metodologias clássicas como Simulated Annealing, Genetic Algorithms, e Particle Swarm Optimization, além de algoritmos inovadores como Cuckoo Search e métodos baseados em fenômenos científicos. O texto também explora computação biomolecular e quântica, além de algoritmos meméticos e baseados em comportamentos humanos, fornecendo uma perspectiva abrangente sobre as abordagens de otimização modernas.

**Metaheuristics for Hard Optimization : Simulated Annealing, Tabu Search, Evolutionary and Genetic Algorithms, Ant Colonies ... Methods and Case Studies**  
<https://link.springer.com/book/10.1007/3-540-30966-7>

## **Classificação por Livro e Capítulos:**

### **Livro: Metaheuristics for Hard Optimization**

- **Parte I: Apresentação das Metaheurísticas Principais**
  - **Simulated Annealing:**
    - Introdução e apresentação do método
    - Convergência teórica e espaço de configuração
    - Esquema de resfriamento e paralelização
    - Aplicações práticas: otimização combinatória, circuitos eletrônicos
    - Vantagens e desvantagens, sugestões práticas
  - **Tabu Search:**
    - Problema de atribuição quadrática
    - Conceitos de memória de curto e longo prazo
    - Oscilações estratégicas e convergência
  - **Evolutionary Algorithms:**
    - Operadores de seleção e variação
    - Algoritmos genéticos: representação binária, real, árvores
    - Convergência dos algoritmos evolutivos
  - **Ant Colony Algorithms:**
    - Otimização por colônias de formigas
    - Comportamento coletivo e autorganização
    - Otimização combinatória, problema do caixeiro-viajante
    - Formalização, intensificação/diversificação, busca local
- **Parte II: Variantes, Extensões e Conselhos Metodológicos**
  - **Variantes de Metaheurísticas:**
    - Variantes do Simulated Annealing: difusão simulada, microcanônica
    - Métodos alternativos: GRASP, Cross-Entropy, Particle Swarm, Noising
  - **Extensões:**
    - Otimização contínua e multimodal
    - Algoritmos evolutivos multiobjetivo
    - Otimização com restrições (métodos de penalização, reparo)

- **Metodologia:**
  - Modelagem de problemas e escolha de vizinhança
  - Programação adaptativa de memória (Ant colonies, Scatter Search, etc.)
  - Comparação de metaheurísticas iterativas
- **Parte III: Estudos de Caso**
  - **Otimização de Redes de Acesso UMTS com Algoritmos Genéticos**
    - Planejamento de redes e codificação genética
    - Otimização da capacidade, interferência e cobertura
  - **Algoritmos Genéticos no Gerenciamento de Tráfego Aéreo**
    - Resolução de conflitos de rota e otimização de tráfego terrestre
  - **Programação de Restrições e Colônias de Formigas em Problemas de Roteamento de Veículos**
    - Programação de restrições aplicada ao roteamento
    - Algoritmos de formigas com retrocesso

## Classificação por Tópico:

### Metaheurísticas Principais:

- **Simulated Annealing:**
  - Algoritmo, paralelização, esquemas de resfriamento, aplicações em circuitos eletrônicos e otimização combinatória.
- **Tabu Search:**
  - Problemas de atribuição quadrática, memória de curto/longo prazo, movimentos e vizinhanças, convergência.
- **Evolutionary Algorithms:**
  - Operadores de seleção e variação, representação genética, aplicação em problemas discretos e contínuos.
- **Ant Colony Algorithms:**
  - Otimização por comportamento coletivo de formigas, intensificação/diversificação, formalização e convergência.

### Variantes e Extensões:

- **Variantes de Simulated Annealing:**
  - Difusão simulada, microcanonic annealing, Great Deluge, Threshold Method.
- **Outras Metaheurísticas:**
  - Particle Swarm, GRASP, Cross-Entropy, Algoritmos inspirados por insetos sociais.

- **Otimização Contínua e Multiobjetivo:**
  - Métodos para variáveis contínuas, niching para problemas multimodais, algoritmos evolutivos multiobjetivo.
- **Otimização com Restrições:**
  - Métodos de penalização, reparação e operadores de variação.

#### **Estudos de Caso:**

- **Redes UMTS com Algoritmos Genéticos:**
  - Planejamento e otimização de capacidade, cobertura e interferência.
- **Tráfego Aéreo:**
  - Resolução de conflitos de rota e otimização de tráfego terrestre com algoritmos genéticos.
- **Problemas de Roteamento de Veículos:**
  - Uso de programação de restrições e algoritmos de formigas para otimização de roteamento.

## **APÊNDICE 2**

## Termo de Aceite de Entrega

### Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

**Data da Reunião (“gate”) de aprovação:** 2 de out. de 2024

**Participantes da Entrega** [matriculados em Residência em IA]:

Victor Guerreiro Pimenta

**Entrega:** [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Atividades Realizadas:

- Continuar e Ampliar Catalogação e Resumos:  
Aprofundamento no cap 1 Introdução e cap 8 Tópicos em ES cap 23 Otimização Multi-Objetivo no livro I.  
Parte 3 Case Studies no livro II.  
[Resumo Técnico Sobre Metaheurísticas v0.1](#)
- Busca sobre problemas e desafios intrínsecos às **Técnicas de Otimização Heurística**:  
Desafios adicionados: Multi-objetivo e Paralelismo  
[Desafios Técnicas de Otimização Heurística](#)
- Entendimento maior de problemas e campos onde se aplicam as **Técnicas de Otimização Heurística**.  
[Problemas em Técnicas de Otimização Heurística](#)

**Planejamento:** [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Aprofundar fundamentos a partir das referências no livro I e produzir um documento sobre eles.
- Buscar mais aplicações específicas para cada algoritmo e encontrar/produzir uma relação problemas/métodos.

**Observação:** [caso precise fazer alguma observação, de qualquer “natureza”]

## ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go!](#)

[Resumo Técnico Sobre Metaheurísticas v0.1 citado no Termo de Aceite de Entrega de 2 de outubro]

### **Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature**

<https://link.springer.com/book/10.1007/978-3-319-41192-7>

---

#### **Introduction**

Este capítulo aborda os fundamentos da otimização global e o conceito de metaheurísticas, oferecendo uma base teórica essencial para o restante do livro. Ele explora definições básicas de otimização, inteligência de enxame, processos biológicos, a diferença entre evolução e aprendizado, além do teorema “no free lunch”. A seguir, apresenta uma visão detalhada sobre como a computação inspirada na natureza desempenha um papel crucial em paradigmas de otimização e busca.

#### **Computação Inspirada na Natureza**

A inteligência artificial (IA) é uma disciplina consolidada que busca construir máquinas capazes de realizar tarefas inteligentes. Um conceito central na IA é a busca, utilizada em diversas disciplinas para encontrar soluções em espaços de busca que muitas vezes são muito grandes para serem enumerados completamente. Métodos tradicionais de cálculo e enumeração são insuficientes para resolver problemas práticos, o que motivou o desenvolvimento de paradigmas de inteligência computacional, os quais se baseiam na cooperação de agentes.

#### **Mecanismos de Busca**

A busca por uma solução ótima, ou otimização, pode ser realizada por três mecanismos principais: busca analítica, enumeração e busca heurística. A busca analítica se baseia no cálculo e utiliza gradientes ou a Hessiana da função para orientar-se, tendendo a encontrar soluções de mínimo local. Já a busca por enumeração realiza uma pesquisa exaustiva e não guiada no espaço de soluções. A busca heurística, por outro lado, é guiada e frequentemente gera soluções de alta qualidade, mesmo sem garantir a optimalidade global.

#### **Inteligência Computacional e Paradigmas Inspirados na Natureza**

A inteligência computacional, um ramo da IA, investiga mecanismos adaptativos que permitem o comportamento inteligente em ambientes complexos. Diferente da IA clássica,

que se baseia no conhecimento humano, a inteligência computacional utiliza dados numéricos coletados para gerar soluções. Os principais paradigmas incluem redes neurais artificiais, sistemas fuzzy e computação evolucionária. Esses paradigmas são fortemente inspirados por processos naturais, como sistemas biológicos e comportamentos sociais de populações biológicas.

### Exemplos de Paradigmas Naturais

Os exemplos de paradigmas de computação inspirados na natureza são diversos e incluem:

- **Redes Neurais Artificiais (ANNs):** Inspiradas no cérebro humano, são usadas para reconhecimento de padrões.
- **Algoritmos Genéticos:** Modelados a partir da evolução biológica, utilizam mecanismos de seleção, cruzamento e mutação para otimização.
- **Inteligência de Enxame:** Baseia-se no comportamento cooperativo de populações biológicas, como colônias de formigas e enxames de abelhas, para resolver problemas complexos.
- **Computação Quântica:** Explora os princípios da mecânica quântica para acelerar a resolução de problemas que vão além das capacidades dos computadores clássicos.

### Processos Biológicos e Computação Evolucionária

A estrutura molecular da vida, em especial o DNA, é abordada como fonte de inspiração para paradigmas computacionais, como a computação por DNA e a computação por membranas. O DNA, composto por nucleotídeos organizados em uma dupla hélice, codifica informações genéticas que orientam a formação de proteínas, as quais são fundamentais para todos os processos biológicos. A computação evolucionária, por exemplo, baseia-se em conceitos de evolução biológica para encontrar soluções ótimas por meio de seleção natural e adaptação.

### Comparação: Evolução versus Aprendizado

A adaptação dos organismos ao meio ambiente ocorre por meio de dois processos principais: evolução e aprendizado. A evolução é um processo estocástico lento que atua sobre populações, enquanto o aprendizado é um processo que ajusta indivíduos em resposta ao ambiente. Ambos os processos podem ser combinados para criar redes neurais evolutivas, que apresentam uma melhor capacidade de adaptação a ambientes dinâmicos.

- **Evolução Darwiniana:** Baseada na seleção natural, a evolução envolve a sobrevivência dos mais aptos e não permite que os conhecimentos adquiridos durante a vida de um indivíduo sejam transferidos geneticamente para as próximas gerações.

- **Aprendizado:** Refere-se à capacidade de um indivíduo ajustar-se ao ambiente ao longo de sua vida. O aprendizado conecta-se fortemente a modelos conexionistas, como as redes neurais artificiais, que ajustam seus pesos sinápticos para melhorar o desempenho.

Duas abordagens combinam evolução e aprendizado: **Lamarckismo** e **Efeito Baldwin**. No **Lamarckismo**, as características adquiridas durante a vida de um indivíduo são passadas para sua descendência, o que é biologicamente implausível, mas útil em sistemas artificiais. O **Efeito Baldwin**, por outro lado, sugere que o aprendizado acelera a evolução ao suavizar o ambiente de busca, facilitando a adaptação da espécie.

### **Teorema No-Free-Lunch e Limitações da Otimização**

Um conceito importante descrito no capítulo é o **teorema “no free lunch”**, que afirma que nenhum algoritmo de otimização é superior a todos os outros em todos os tipos de problemas. Isso significa que a eficácia de um algoritmo depende fortemente das características específicas do problema que está sendo resolvido. Esse teorema ressalta a importância de selecionar cuidadosamente a abordagem de otimização apropriada para cada caso.

### **Conclusão**

Este capítulo estabelece a importância da otimização e das metaheurísticas em diversos campos da computação e das ciências naturais. Ao destacar a relação entre evolução, aprendizado e computação inspirada na natureza, ele fornece uma base sólida para a compreensão dos métodos avançados que serão apresentados nos capítulos subsequentes.

- **Swarm Intelligence: comportamento de grupo e a teoria do forrageamento.**

### **1.4 Inteligência de Enxame**

O conceito de inteligência de enxame foi introduzido em 1989, no contexto de sistemas robóticos celulares. Inteligência de enxame refere-se à inteligência coletiva de grupos de agentes simples. Ela lida com os comportamentos coletivos de enxames descentralizados e auto-organizados, que resultam das interações locais dos componentes individuais entre si e com o ambiente. Embora normalmente não exista uma estrutura de controle centralizada que dite como os agentes devem se comportar, as interações locais entre eles frequentemente levam ao surgimento de comportamentos globais.

Na natureza, diversas espécies de animais exibem comportamentos sociais. Esses comportamentos coletivos podem variar desde a cooperação altruísta até conflitos abertos. Exemplos bem conhecidos de enxames incluem bandos de pássaros, rebanhos de quadrúpedes, colônias de insetos sociais, como formigas e abelhas, e cardumes de peixes. Através da formação de grupos, os indivíduos adquirem vantagens, como a redução do risco de serem capturados por predadores, a eficiência energética durante o deslocamento e a precisão ao seguir rotas de migração.

O princípio de "organização individual" tem sido amplamente utilizado para entender o comportamento coletivo de animais. Esse princípio indica que interações simples e repetidas entre indivíduos podem gerar padrões comportamentais complexos no nível do grupo. Os agentes de enxames comportam-se sem supervisão centralizada, sendo que cada agente possui um comportamento estocástico baseado na sua percepção do ambiente e nas interações com os vizinhos.

### **1.4.1 Comportamentos de Grupo**

Viver em grupo é um fenômeno comum na ecologia animal. Animais em busca de recursos, como comida ou parceiros, podem adotar diferentes estratégias de forrageamento, seja individualmente ou em grupo. Dentro de um grupo, as habilidades de busca e competição variam, sendo que indivíduos subordinados, menos eficientes, podem ser dispersos do grupo e explorar novos habitats.

Dois modelos de estratégia de busca em grupo são amplamente estudados: o modelo de compartilhamento de informações e o modelo produtor-explorador. No modelo de compartilhamento, os indivíduos procuram seus próprios recursos enquanto buscam oportunidades de se juntar a outros. Já no modelo produtor-explorador, os indivíduos adotam exclusivamente uma das duas estratégias: produzir (buscar recursos) ou explorar (aproveitar-se dos recursos encontrados por outros).

Em aves que se alimentam no solo, o modelo produtor-explorador é considerado mais plausível do que o modelo de compartilhamento de informações. Nesse contexto, foram observadas três estratégias básicas de exploração: copiar a área, seguir outro animal sem buscar ativamente, ou tomar diretamente o recurso do produtor.

### **1.4.2 Teoria do Forrageamento**

A seleção natural tende a favorecer animais com estratégias de forrageamento bem-sucedidas, eliminando aquelas menos eficientes. Com o tempo, essas estratégias evoluem para maximizar a eficiência energética dos animais durante a busca por alimentos.

O forrageamento pode ser modelado como um processo de otimização, no qual o animal busca maximizar a energia obtida por unidade de tempo investida, ou maximizar a taxa média de obtenção de energia no longo prazo. Em forrageamento social, grupos de animais cooperam para buscar alimentos, aproveitando a inteligência coletiva do grupo, o que pode aumentar suas chances de sucesso ao capturar presas maiores e se protegerem de predadores.

Animais que adotam diferentes estratégias de forrageamento, como procurar ativamente por alimentos (cruzeiro), emboscar presas (emboscada), ou alternar entre movimentos rápidos e pausas (saltatório), exibem comportamentos de busca otimizados para suas condições ambientais e físicas. Esses comportamentos podem ser analisados e modelados com base em cenários de decisão e otimização, fornecendo insights tanto para a biologia quanto para a aplicação em sistemas computacionais inspirados na natureza.

## Heuristics, Metaheuristics, and Hyper-Heuristics

- **Diferenciação entre heurísticas, metaheurísticas e hiper-heurísticas.**

### 1.5 Heurísticas, Meta-heurísticas e Hiper-heurísticas

Muitos problemas de otimização do mundo real são difíceis de resolver por métodos de otimização exatos, devido a características como alta dimensionalidade, multimodalidade, epistasia (interação entre parâmetros) e não-diferenciabilidade. Portanto, algoritmos aproximados são uma abordagem alternativa para esses problemas. Esses algoritmos podem ser divididos em heurísticas e meta-heurísticas. As palavras "meta" e "heurística" têm sua origem no grego antigo: "meta" significa nível superior, e "heurística" denota a arte de descobrir novas estratégias [58].

**Heurística** refere-se a técnicas baseadas em experiência para resolver problemas e aprender. Ela fornece uma solução satisfatória em um tempo computacional razoável, que pode não ser ótima. Heurísticas específicas são dependentes do problema e projetadas para resolver apenas um problema em particular. Exemplos incluem o uso de regras práticas, suposições fundamentadas, julgamentos intuitivos ou até mesmo o senso comum. Muitos algoritmos, sejam exatos ou aproximados, são heurísticos.

O termo **meta-heurística** foi cunhado por Glover em 1986 [25] para se referir a um conjunto de metodologias conceitualmente classificadas acima das heurísticas, no sentido de que elas orientam o design de heurísticas. Uma meta-heurística é um procedimento ou heurística

de nível superior, projetada para encontrar, gerar ou selecionar uma heurística de nível inferior (algoritmo de busca parcial) que possa fornecer uma solução suficientemente boa para um problema de otimização. Ao buscar sobre um grande conjunto de soluções viáveis, as meta-heurísticas podem frequentemente encontrar boas soluções com menos esforço computacional do que métodos baseados em cálculo ou heurísticas simples.

As meta-heurísticas podem ser baseadas em uma única solução ou em uma população. Meta-heurísticas baseadas em uma única solução se concentram em uma única solução por vez e incluem métodos baseados em busca local, como Simulated Annealing (SA), busca tabu, busca local iterada [40,42], busca local guiada [61], busca por padrões ou busca aleatória [31], algoritmo de Solis-Wets [54] e busca em vizinhança variável [45]. Em meta-heurísticas baseadas em população, um número de soluções é atualizado iterativamente até que a condição de término seja satisfeita. Meta-heurísticas baseadas em população são geralmente classificadas em algoritmos evolutivos (EAs) e algoritmos baseados em enxames. As meta-heurísticas baseadas em uma única solução são consideradas mais orientadas à exploração, enquanto as baseadas em população tendem a ser mais voltadas à exploração.

A ideia das **hiper-heurísticas** remonta ao início dos anos 1960 [23]. As hiper-heurísticas podem ser entendidas como heurísticas para escolher heurísticas ou como algoritmos de busca que exploram o espaço de solucionadores de problemas. Uma hiper-heurística é um método de busca heurística que busca automatizar o processo de seleção, combinação, geração ou adaptação de várias heurísticas mais simples para resolver eficientemente problemas de busca complexos. As heurísticas de nível inferior são operadores de busca local simples ou heurísticas dependentes do domínio, que operam diretamente no espaço de soluções para uma instância de problema específica. Diferentemente das meta-heurísticas, que buscam em um espaço de soluções do problema, as hiper-heurísticas sempre buscam em um espaço de heurísticas de nível inferior.

Atualmente, as principais metodologias em hiper-heurísticas são a seleção e a geração de heurísticas. No primeiro método, a hiper-heurística escolhe heurísticas de um conjunto de heurísticas de nível inferior conhecidas e dependentes do domínio. No segundo método, a hiper-heurística evolui novas heurísticas de nível inferior utilizando componentes das existentes. Hiper-heurísticas podem ser baseadas em programação genética [11] ou evolução gramatical [10], tornando-se uma excelente candidata para a geração de heurísticas.

### **Várias Meta-Heurísticas Baseadas em Solução Única**

Estratégias de busca que geram aleatoriamente soluções iniciais e realizam uma busca local também são chamadas de métodos de busca descendente com múltiplos inícios. No entanto, criar aleatoriamente uma solução inicial e realizar uma busca local frequentemente

resulta em baixa qualidade de solução, pois o espaço de busca completo é explorado uniformemente, e a busca não pode se concentrar em áreas promissoras do espaço de busca.

A **busca em vizinhança variável** [45] combina estratégias de busca local com estruturas de vizinhança dinâmicas sujeitas ao progresso da busca. A busca local é um passo de intensificação, focando a busca na direção de soluções de alta qualidade. A diversificação resulta da mudança das vizinhanças. Ao mudar vizinhanças, o método pode facilmente escapar de ótimos locais. Com o aumento da cardinalidade das vizinhanças, a diversificação se torna mais forte, já que os passos de perturbação podem escolher entre um conjunto maior de soluções e a busca local cobre uma área maior do espaço de busca.

A **busca local guiada** [61] usa um princípio semelhante e altera dinamicamente a paisagem de fitness, sujeita ao progresso feito durante a busca, permitindo que a busca local escape de ótimos locais. A estrutura de vizinhança permanece constante. Ela começa com uma solução aleatória  $x_0$  e realiza uma busca local retornando o ótimo local  $x_1$ . Para escapar do ótimo local, é adicionada uma penalidade à função de fitness  $f$ , de modo que a função de fitness resultante  $h$  permita à busca local escapar. Uma nova busca local é iniciada a partir de  $x_1$  usando a função de fitness modificada  $h$ . A busca continua até que um critério de término seja atendido.

A **busca local iterada** [40,42] conecta as fases de busca local não relacionadas, pois cria soluções iniciais não aleatoriamente, mas com base em soluções encontradas em execuções anteriores de busca local. Se os passos de perturbação forem muito pequenos, a busca não poderá escapar de um ótimo local. Se a perturbação for muito forte, a busca terá o mesmo comportamento que os métodos de busca descendente com múltiplos inícios. O passo de modificação, assim como o critério de aceitação, podem depender do histórico da busca.

- **Métodos de otimização.**

## 1.6 Otimização

A otimização pode ser classificada em duas categorias principais: **otimização discreta** ou **contínua**, dependendo se as variáveis são discretas ou contínuas. Pode haver **limites ou restrições** nas variáveis. A otimização pode ser um problema **estático ou dinâmico**, dependendo de o resultado ser ou não uma função do tempo. Tradicionalmente, a otimização é resolvida por métodos baseados em cálculo, ou por busca aleatória ou enumerativa. A **otimização baseada em heurísticas** é o tópico abordado neste livro.

As técnicas de otimização podem ser divididas em métodos que utilizam ou não derivadas, dependendo da necessidade de derivadas da função objetivo para o cálculo do ótimo. Os métodos baseados em derivadas são fundamentados no cálculo, podendo ser métodos de **busca por gradiente** ou métodos de **segunda ordem**. Esses métodos são **otimizadores locais**. A descida do gradiente, também chamada de **descida mais íngreme**, busca um mínimo local seguindo o **negativo da direção do gradiente** da função. Exemplos de métodos de segunda ordem incluem o método de **Newton**, o método de **Gauss-Newton**, métodos **quase-Newton**, o método da **região de confiança** e o método de **Levenberg-Marquardt**. Métodos de gradiente conjugado e gradiente natural também podem ser vistos como formas reduzidas do método quase-Newton.

Os métodos baseados em derivadas podem ser classificados em **métodos baseados em modelo** e **métodos baseados em métrica**. Os métodos baseados em modelo melhoram o ponto atual por meio de um modelo de aproximação local. Métodos de Newton e quase-Newton pertencem a essa categoria. Métodos baseados em métrica realizam uma transformação nas variáveis e depois aplicam um método de busca por gradiente para melhorar o ponto. Os métodos de **descida mais íngreme**, quase-Newton e gradiente conjugado pertencem a essa última categoria.

Métodos que não requerem informação de gradiente para realizar uma busca e exploram sequencialmente o espaço de soluções são chamados de **métodos de busca direta**. Eles mantêm um grupo de pontos e utilizam algum tipo de método de exploração determinística para pesquisar o espaço, quase sempre utilizando um **método ganancioso** para atualizar os pontos mantidos. **Busca simplex** e **busca por padrões** são dois exemplos eficazes de métodos de busca direta.

Métodos típicos sem derivadas para funções multivariáveis incluem a busca colina com reinício aleatório, busca aleatória, muitos métodos heurísticos e metaheurísticos, e seus híbridos. A busca colina tenta otimizar uma função discreta ou contínua para encontrar um ótimo local. Quando opera no espaço contínuo, é chamada de **ascensão por gradiente**. Outros métodos de busca sem derivadas incluem a busca univariada paralela a um eixo (ou seja, o método de busca coordenada), o método simplex sequencial e métodos de aceleração na busca direta, como o método de **Hooke-Jeeves**, o método de **Powell** e o método de **Rosenbrock**. Os métodos de ponto interior representam técnicas de ponta para resolver programas de otimização lineares, quadráticos e não lineares.

## 1.7 Indicadores de Desempenho

Ao avaliar diferentes Algoritmos Evolutivos (EAs) ou algoritmos iterativos, podem ser implementados indicadores de desempenho globais e indicadores de desempenho evolutivos.

## Indicadores de Desempenho Globais

Os indicadores de desempenho globais fornecem uma descrição geral do desempenho. Eles podem ser comparados com base em **eficácia**, **eficiência** e **confiabilidade**, em um problema de benchmark com várias execuções.

- **Eficácia:** Avalia a qualidade dos resultados, independentemente da velocidade do algoritmo. O **MBF (Mean Best Fitness)** é definido como a média do melhor fitness na última população ao longo de todas as execuções. Valores de fitness até o momento podem ser usados como uma medida mais absoluta da eficácia.
- **Confiabilidade:** Indica até que ponto o algoritmo pode fornecer resultados aceitáveis. A **Taxa de Sucesso (SR - Success Rate)** é definida como o percentual de execuções bem-sucedidas. Uma execução bem-sucedida ocorre quando a diferença entre o melhor valor de fitness na última geração ( $f^*$ ) e um valor pré-definido ( $f_0$ ) está abaixo de um limite ( $\epsilon$ ).
- **Eficiência:** Refere-se à capacidade de encontrar a solução ótima global rapidamente. O **AES (Average number of Evaluations to a Solution)** é o número médio de avaliações necessárias para alcançar uma solução nas execuções bem-sucedidas. Se não houver execuções bem-sucedidas, o AES é indefinido.

Ao analisar esses indicadores, uma **baixa SR** combinada com um **alto MBF** pode indicar que o algoritmo converge lentamente. Já uma **alta SR** com um **baixo MBF** pode significar que o algoritmo é confiável, mas fornece resultados ruins ocasionalmente. O ideal é ter um **AES baixo** e uma **SR alta**, sendo que a métrica **AES/SR** considera a confiabilidade e eficiência simultaneamente.

## Indicadores de Desempenho Evolutivos

Os indicadores de desempenho evolutivos fornecem informações mais detalhadas, geração por geração.

- **BSF (Best-So-Far):** Registra a melhor solução encontrada até o momento para cada geração.
- **BCP (Best of Current Population):** Registra a melhor solução de cada geração.
- **ACP (Average of Current Population):** Registra a solução média de cada geração.
- **WCP (Worst of Current Population):** Registra a pior solução de cada geração.

Com múltiplas execuções e configurações iniciais aleatórias, é possível tirar conclusões sobre o algoritmo aplicando **visualizações estatísticas** e **inferência estatística**. O **box plot**

é amplamente utilizado para comparar algoritmos visualmente. Ele exibe a mediana, quartis e valores atípicos, permitindo uma análise rápida da distribuição dos dados.

## 1.8 Teorema No Free Lunch

O **Teorema No Free Lunch (NFL)**, proposto em 1995, estabelece que não existe um algoritmo de busca universalmente melhor para todos os problemas. Antes de seu surgimento, acreditava-se que era possível projetar algoritmos que fossem universalmente eficazes.

O teorema afirma que, ao considerar o conjunto de todas as funções possíveis, nenhum algoritmo de busca supera outro em média na localização de um extremo de uma função. Em outras palavras, qualquer algoritmo de busca, quando avaliado em todas as funções possíveis, tem o mesmo desempenho que uma busca aleatória.

- **Teorema 1.1 (No Free Lunch):** Dado o conjunto de todas as funções ( $F$ ) e um subconjunto de funções de benchmark ( $F_1$ ), se o algoritmo  $A_1$  é melhor que o algoritmo  $A_2$  em  $F_1$ , então  $A_2$  será melhor em  $F \setminus F_1$ .

Esse teorema mostra que, sem conhecimento estrutural sobre o problema, todos os algoritmos têm o mesmo desempenho. No entanto, em aplicações práticas, sempre existe algum conhecimento sobre o problema (como suavidade, simetria, etc.), o que torna o NFL menos aplicável na prática. Por isso, o desenvolvimento de algoritmos de busca é, na verdade, uma construção de métodos especializados para resolver problemas específicos.

O teorema foi estendido para incluir métodos de codificação, validação cruzada, e outros cenários, mostrando que não há uma única abordagem melhor para todas as situações, incluindo a combinação de estratégias em **hiper-heurísticas**.

## Impacto na Otimização

O teorema NFL tem um impacto direto no campo da otimização, pois desafia a ideia de encontrar algoritmos que funcionem bem para qualquer problema de otimização. Ele sugere que é mais eficiente projetar algoritmos adaptados a classes específicas de problemas. Por exemplo, algoritmos como **busca evolutiva**, **algoritmos de colônia de formigas**, e **inteligência de enxame** tendem a ser aplicados em problemas que possuem características específicas que favorecem suas abordagens.

---

## Metaheurísticas Individuais

### Simulated Annealing

- Introdução ao Simulated Annealing básico e suas variantes.

### Genetic Algorithms

- Introdução à computação evolutiva.
- Comparação entre algoritmos evolutivos e Simulated Annealing.
- Terminologias de computação evolutiva (codificação, seleção, reprodução, etc.).
- Exploitation vs Exploration e algoritmos genéticos para otimização de sequência.

### Genetic Programming

- Sintaxe de árvores, controle de crescimento desordenado (bloat), controle de tamanho de programas, e design de operadores genéticos.

### Gene Expression Programming

- Introdução e principais aspectos de programação genética baseada em expressão.

### Evolutionary Strategies

- Busca evolutiva, estratégias evolutivas com base em gradiente e otimização por distribuição de covariância.

### Differential Evolution (DE)

- Introdução, variantes do algoritmo DE, e análise teórica do DE.

### Estimation of Distribution Algorithms (EDA)

- Introdução aos EDAs, variantes, e métodos como Compact Genetic Algorithms e Bayesian Optimization Algorithm.

### Parallel Evolutionary Algorithms

- Modelos paralelos como Master-Slave, Island Model, e Cellular EAs.

## 8.3 Algoritmos Evolutivos Paralelos

### 8.3.1 Introdução

Os algoritmos evolutivos paralelos (EAs) são projetados para operar em ambientes distribuídos, onde as tarefas de evolução são processadas em múltiplos processadores simultaneamente. As propriedades dos algoritmos distribuídos podem ser caracterizadas por dois pares de dimensões: **sincronismo/assíncronismo** e **homogeneidade/heterogeneidade**. Esses fatores influenciam diretamente o desempenho e a eficiência dos EAs.

### 8.3.2 Sincronismo e Assíncronismo

- **Sincronismo:** Todos os processadores comunicam-se de acordo com um sinal de relógio. A execução é controlada por um tempo fixo, onde todas as operações devem ser concluídas antes que uma nova geração de indivíduos seja criada.
- **Assíncronismo:** A comunicação é impulsionada por dados, onde os processadores podem operar de forma independente e enviar informações assim que suas operações estiverem concluídas.

### 8.3.3 Homogeneidade e Heterogeneidade

- **Homogeneidade:** Todos os processadores utilizam as mesmas configurações algorítmicas. Isso resulta em um comportamento previsível e controlado.
- **Heterogeneidade:** As configurações algorítmicas variam entre os processadores. Essa diversidade pode levar a um aumento na qualidade das soluções, uma vez que diferentes abordagens são exploradas.

### 8.3.4 Métricas de Desempenho

As métricas de desempenho dos EAs distribuídos incluem:

- **Aceleração:** Medida da relação entre o tempo de execução sequencial e o tempo de execução paralelo. Idealmente, a aceleração deve ser igual ao número de processadores utilizados.
- **Eficiência Distribuída:** Proporção da aceleração em relação ao número de processadores. O valor ideal é de 100%, mas na prática é limitado por sobrecarga computacional e velocidade de comunicação.

- **Tolerância a Falhas:** Capacidade do algoritmo continuar a otimização em caso de falha de componentes físicos.
- **Escalabilidade:** Envolve escalabilidade de tamanho (aumento proporcional da performance com mais processadores) e escalabilidade de tarefa (capacidade de adaptar-se a mudanças na escala do problema).

### 8.3.5 Classificação de EAs Distribuídos

Os EAs distribuídos podem ser classificados em dois grupos principais com base em seus mecanismos de divisão de tarefas:

- **Modelos distribuídos por população:**
  - **Master-Slave:** A população inteira é mantida em um processador mestre que seleciona indivíduos para reprodução e os envia para processadores escravos.
  - **Coarse-Grained Parallel:** A população é dividida em subpopulações isoladas, onde indivíduos podem migrar entre elas.
  - **Fine-Grained Parallel:** A população é dividida em muitas subpopulações pequenas, onde os indivíduos interagem apenas com vizinhos locais.
  - **Pool-based Methods:** Utilizam computação em nuvem, permitindo a alocação dinâmica de tarefas em processadores.
- **Modelos distribuídos por dimensão:** Envolvem coevolução e modelos multiagente, focando na redução de dimensões.

### 8.3.6 Computação em Nuvem

A computação em nuvem emergiu como uma plataforma popular para a implementação de EAs distribuídos. Ela representa um conjunto de recursos computacionais virtualizados que podem ser alocados dinamicamente. O modelo **MapReduce** é utilizado para acessar e processar dados de forma escalável, permitindo que os usuários se concentrem apenas no problema e no algoritmo.

### 8.3.7 Computação em GPU

A computação em GPU é uma estratégia para acelerar a execução de EAs paralelizando operações. GPUs têm uma arquitetura de múltiplos processadores que permite a execução simultânea de várias instâncias de EAs. As abordagens de paralelização podem ser classificadas em:

- **Fitness Parallel:** Avalia todos os casos de fitness em paralelo, um indivíduo de cada vez.
- **Population Parallel:** Avalia múltiplos indivíduos simultaneamente, executando a mesma instrução em diferentes dados.

### 8.3.8 Comparação de Modelos Distribuídos

Uma tabela comparativa pode ser utilizada para resumir os níveis de paralelismo, operação, custo de comunicação, escalabilidade e tolerância a falhas de diferentes modelos de EAs distribuídos, incluindo **Master-Slave**, **Island**, **Cellular**, **Hierarchical**, **Pool** e **Coevolution**.

## 8.4 Coevolução

### 8.4.1 Introdução

A coevolução melhora a eficácia dos EAs ao permitir que várias populações coexistam e evoluam simultaneamente. Essa abordagem possibilita a definição de componentes separados para representar um problema, utilizando uma estratégia de dividir para conquistar.

### 8.4.2 Abordagens Coevolutivas

- **Coevolução Cooperativa:** Inspirada por relações ecológicas de simbiose, onde diferentes espécies evoluem juntas, otimizando um problema em módulos.
- **Integração de Populações:** A solução completa é obtida unindo indivíduos de diferentes populações, aumentando o valor de fitness colaborativo entre elas.

A coevolução é uma estratégia poderosa para a otimização de problemas complexos, frequentemente resultando em soluções de alta qualidade que superam abordagens tradicionais.

### Coevolution

- **Abordagens coevolutivas, com foco em otimização minimax e Interactive Evolutionary Computation.**

### Other Heredity-Based Algorithms

- Aplicações em otimização de redes neurais.
- 

## Swarm Intelligence-Based Methods

### Particle Swarm Optimization (PSO)

- Introdução e variantes do PSO, como Bare-Bones PSO e Gaussian/Cauchy PSO.

### Artificial Immune Systems

- Algoritmos baseados na seleção clonal e algoritmos de seleção negativa.

### Ant Colony Optimization (ACO)

- Introdução ao algoritmo básico de ACO e suas aplicações em otimização contínua.

### Bee Metaheuristics

- Discussão sobre Artificial Bee Colony Algorithm e algoritmos como Marriage in Honeybees Optimization.

### Bacterial Foraging Algorithm

- Forrageamento bacteriano como inspiração para algoritmos de otimização.

### Other Swarm Intelligence Algorithms

- Glowworm Swarm Optimization, Firefly Algorithm, Cuckoo Search, Bat Algorithm, e Krill Herd Algorithm.
- 

## Biomolecular and Quantum Computing

### Biomolecular Computing

- DNA Computing, embutimento de dados, e redes bioquímicas.

### Quantum Computing

---

- **Introdução ao Grover's Search Algorithm e metaheurísticas híbridas inspiradas pela computação quântica.**
- 

## **Metaheuristics Based on Sciences**

- **Otimização baseada nas leis de Newton, princípios eletromagnéticos, fenômenos térmicos, e cosmologia.**
  - **Discussão sobre algoritmos como Black Hole-Based Optimization e Chemical Reaction Optimization.**
- 

## **Memetic Algorithms**

- **Introdução aos Memetic Algorithms, abordando tópicos como Cultural Algorithms e otimização simplex.**
- 

## **Tabu Search and Scatter Search**

- **Iterações de Tabu Search e variantes de Scatter Search.**
- 

## **Search Based on Human Behaviors**

- **Algoritmos como Seeker Optimization Algorithm e Teaching–Learning-Based Optimization.**
- 

## **Dynamic, Multimodal, and Constrained Optimizations**

- **Métodos para Dynamic Optimization, mantendo a diversidade e abordagens para Multimodal Optimization como Crowding e Fitness Sharing.**
-

## Multiobjective Optimization

### 23. Otimização Multiobjetivo

A otimização multiobjetivo (MOPs) envolve múltiplos objetivos conflitantes que precisam ser otimizados simultaneamente. O desafio é encontrar um conjunto de Pareto, que consiste em soluções não dominadas distribuídas de forma uniforme ao longo da Fronteira de Pareto. Metaheurísticas para otimização multiobjetivo têm se mostrado abordagens eficientes para resolver MOPs.

#### 23.1 Introdução

As metaheurísticas para otimização multiobjetivo, conhecidas como algoritmos evolutivos multiobjetivo (MOEAs), são abordagens estabelecidas e eficientes para resolver MOPs. Essas metaheurísticas podem ser não baseadas em Pareto ou baseadas em Pareto.

As técnicas não baseadas em Pareto incluem o método lexicográfico e o método de função agregada. No método lexicográfico, os objetivos são classificados em ordem decrescente, e a otimização avança de objetivos de maior prioridade para os de menor prioridade, um de cada vez. Uma vez otimizado um objetivo, busca-se melhorar os próximos sem reduzir a qualidade dos anteriores. No método de função agregada, todos os objetivos são convertidos em um único objetivo a ser otimizado, geralmente requerendo normalização devido às diferentes escalas dos objetivos. A variação dos pesos durante a execução permite gerar diferentes soluções não dominadas em uma única execução.

O GA avaliado por vetor (VEGA) foi o primeiro algoritmo evolutivo para otimização multiobjetivo. A população é dividida em subpopulações de tamanho igual, cada uma buscando o ótimo de um único objetivo, e depois as subpopulações são mescladas. Ao realizar o crossover, indivíduos que são bons em um objetivo se recombinaem com indivíduos que são bons em outro, resultando em várias soluções não dominadas em uma única execução. No entanto, esse método tende a negligenciar bons compromissos entre os objetivos.

No algoritmo de busca local genética multiobjetiva (MOGLS), o MOP é reformulado como uma otimização simultânea de todas as funções Tchebycheff ponderadas ou uma soma ponderada de múltiplos objetivos como uma função de aptidão. Um procedimento de busca local é aplicado a cada indivíduo gerado por operações genéticas, examinando um pequeno número de soluções vizinhas.

Os algoritmos evolutivos tendem a convergir para uma única solução se executados por tempo suficiente, tornando necessária uma mecânica para manter a diversidade. Todas as soluções não dominadas devem ser consideradas igualmente boas. Goldberg introduziu a ordenação não dominada para classificar uma população de busca com base na optimalidade de Pareto. O método de classificação de Pareto atribui um ranking a cada solução com base em seu domínio Pareto.

Técnicas de nicho e especiação podem ser usadas para promover a diversidade genética, garantindo que toda a fronteira de Pareto seja coberta. O GA multiobjetivo (MOGA) utiliza um método de atribuição de aptidão baseado em ranking, promovendo a diversidade entre candidatos preferíveis.

### **23.2 Algoritmos Evolutivos Multiobjetivo**

Os métodos baseados em Pareto podem ser não elitistas ou elitistas. Eles geralmente adotam a classificação de Pareto, alguma forma de elitismo e uma estratégia de manutenção de diversidade. Os MOEAs elitistas retêm as soluções não dominadas geradas em um arquivo externo ou na população principal, permitindo que soluções globalmente não dominadas sejam preservadas.

Um bom MOEA para MOPs deve satisfazer requisitos de convergência, distribuição e elitismo. Deve ter um mecanismo de convergência para encontrar a fronteira de Pareto rapidamente e distribuir os indivíduos uniformemente ao longo da fronteira para fornecer mais soluções não dominadas.

### **23.4 Muitos Objetivos**

Um problema de otimização multiobjetivo que envolve um grande número de objetivos ( $M > 4$ ) é geralmente chamado de problema de otimização de muitos objetivos. Aumentar o número de objetivos geralmente agrava o conflito entre convergência e diversidade, tornando a dominância de Pareto menos eficaz em espaços de alta dimensão.

Os MOEAs clássicos, como NSGA-II e SPEA2, enfrentam dificuldades em lidar com problemas de muitos objetivos devido à perda de pressão de seleção na avaliação de aptidão. O equilíbrio entre convergência e diversidade é um desafio, pois a proporção de soluções não dominadas em uma população aumenta rapidamente com o número de objetivos, dificultando a distinção entre indivíduos.

Soluções em um espaço objetivo podem ser distribuídas uniformemente, mas longe da fronteira de Pareto desejada. Um estudo mostrou que um algoritmo de busca aleatória pode,

---

em alguns casos, produzir melhores resultados que algoritmos baseados em Pareto em MOPs com um alto número de objetivos.

A busca por soluções em cantos da fronteira de Pareto, em vez de buscar a fronteira completa, é uma estratégia que pode render soluções mais relevantes.

- **Discussão de algoritmos como Nondominated Sorting Genetic Algorithm II (NSGA-II), Strength Pareto Evolutionary Algorithm 2 (SPEA2), e Multiobjective Immune Algorithms.**
- **Técnicas para otimização multiobjetivo usando Particle Swarm Optimization e EDAs.**

---

## Outros Tópicos

- **Performance Metrics para algoritmos evolutivos.**
- **Many-Objective Optimization: Desafios e abordagens.**

---

## Conclusão

O livro *Search and Optimization by Metaheuristics* cobre um extenso conjunto de algoritmos de otimização e metaheurísticas inspiradas na natureza. A obra examina metodologias clássicas como Simulated Annealing, Genetic Algorithms, e Particle Swarm Optimization, além de algoritmos inovadores como Cuckoo Search e métodos baseados em fenômenos científicos. O texto também explora computação biomolecular e quântica, além de algoritmos meméticos e baseados em comportamentos humanos, fornecendo uma perspectiva abrangente sobre as abordagens de otimização modernas.

**Metaheuristics for Hard Optimization : Simulated Annealing, Tabu Search, Evolutionary and Genetic Algorithms, Ant Colonies ... Methods and Case Studies**

<https://link.springer.com/book/10.1007/3-540-30966-7>

**Classificação por Livro e Capítulos:**

**Livro: Metaheuristics for Hard Optimization**

- **Parte I: Apresentação das Metaheurísticas Principais**
  - **Simulated Annealing:**
    - Introdução e apresentação do método
    - Convergência teórica e espaço de configuração
    - Esquema de resfriamento e paralelização
    - Aplicações práticas: otimização combinatória, circuitos eletrônicos
    - Vantagens e desvantagens, sugestões práticas
  - **Tabu Search:**
    - Problema de atribuição quadrática
    - Conceitos de memória de curto e longo prazo
    - Oscilações estratégicas e convergência
  - **Evolutionary Algorithms:**
    - Operadores de seleção e variação
    - Algoritmos genéticos: representação binária, real, árvores
    - Convergência dos algoritmos evolutivos
  - **Ant Colony Algorithms:**
    - Otimização por colônias de formigas
    - Comportamento coletivo e autorganização
    - Otimização combinatória, problema do caixeiro-viajante
    - Formalização, intensificação/diversificação, busca local
- **Parte II: Variantes, Extensões e Conselhos Metodológicos**
  - **Variantes de Metaheurísticas:**
    - Variantes do Simulated Annealing: difusão simulada, microcanônica
    - Métodos alternativos: GRASP, Cross-Entropy, Particle Swarm, Noising
  - **Extensões:**
    - Otimização contínua e multimodal
    - Algoritmos evolutivos multiobjetivo
    - Otimização com restrições (métodos de penalização, reparo)
  - **Metodologia:**
    - Modelagem de problemas e escolha de vizinhança

- Programação adaptativa de memória (Ant colonies, Scatter Search, etc.)
- Comparação de metaheurísticas iterativas
- **Parte III: Estudos de Caso**
  - **Otimização de Redes de Acesso UMTS com Algoritmos Genéticos**
    - Planejamento de redes e codificação genética
    - Otimização da capacidade, interferência e cobertura
  - **Algoritmos Genéticos no Gerenciamento de Tráfego Aéreo**
    - Resolução de conflitos de rota e otimização de tráfego terrestre
  - **Programação de Restrições e Colônias de Formigas em Problemas de Roteamento de Veículos**
    - Programação de restrições aplicada ao roteamento
    - Algoritmos de formigas com retrocesso

## Classificação por Tópico:

### Metaheurísticas Principais:

- **Simulated Annealing:**
  - Algoritmo, paralelização, esquemas de resfriamento, aplicações em circuitos eletrônicos e otimização combinatória.
- **Tabu Search:**
  - Problemas de atribuição quadrática, memória de curto/longo prazo, movimentos e vizinhanças, convergência.
- **Evolutionary Algorithms:**
  - Operadores de seleção e variação, representação genética, aplicação em problemas discretos e contínuos.
- **Ant Colony Algorithms:**
  - Otimização por comportamento coletivo de formigas, intensificação/diversificação, formalização e convergência.

### Variantes e Extensões:

- **Variantes de Simulated Annealing:**
  - Difusão simulada, microcanonic annealing, Great Deluge, Threshold Method.
- **Outras Metaheurísticas:**
  - Particle Swarm, GRASP, Cross-Entropy, Algoritmos inspirados por insetos sociais.
- **Otimização Contínua e Multiobjetivo:**

- Métodos para variáveis contínuas, niching para problemas multimodais, algoritmos evolutivos multiobjetivo.
- **Otimização com Restrições:**
  - Métodos de penalização, reparação e operadores de variação.

#### **Estudos de Caso:**

- **Redes UMTS com Algoritmos Genéticos:**
  - Planejamento e otimização de capacidade, cobertura e interferência.
- **Tráfego Aéreo:**
  - Resolução de conflitos de rota e otimização de tráfego terrestre com algoritmos genéticos.
- **Problemas de Roteamento de Veículos:**
  - Uso de programação de restrições e algoritmos de formigas para otimização de roteamento.

[Desafios Técnicas de Otimização Heurística citado no Termo de Aceite de Entrega de 2 de outubro]

#### **Desafios nos Algoritmos de Heurísticas: Multiobjetivo**

O aprendizado de máquina tradicional frequentemente busca otimizar uma única métrica de desempenho, como a precisão ou a perda. No entanto, em muitos cenários do mundo real, como na geoprocessamento e na análise de dados urbanos, é crucial lidar com múltiplos objetivos que podem ser conflitantes. Por exemplo, ao modelar a georreferência de dados urbanos, um modelo pode precisar equilibrar a precisão na localização dos objetos, a eficiência computacional e a minimização de erros de medição. A abordagem multiobjetivo propõe a otimização simultânea de várias funções de perda, permitindo que soluções diferentes sejam exploradas e que um compromisso seja encontrado entre objetivos

conflitantes. Este desafio exige métodos avançados de otimização, como algoritmos genéticos ou técnicas de aprendizado por reforço, que devem ser adaptados para considerar a complexidade e a dinamicidade dos dados. Além disso, a avaliação das soluções multiobjetivo é mais complexa, pois não se trata apenas de um único valor de desempenho, mas de um conjunto de trade-offs que podem ser analisados usando conceitos como o conjunto de Pareto. Portanto, desenvolver modelos que integrem eficientemente a otimização multiobjetivo não só amplia a aplicabilidade dos métodos de aprendizado de máquina, mas também enriquece a análise e a interpretação dos resultados em contextos multidimensionais.

### **Desafios e arquiteturas da Paralelização em Algoritmos de Heurísticas**

A paralelização de algoritmos de heurísticas apresenta desafios significativos, especialmente em contextos onde a complexidade do problema é alta e a diversidade das soluções requer abordagens adaptativas. Embora os Algoritmos Evolutivos (EAs) tenham sido amplamente explorados e aplicados em diversas áreas, com um histórico consolidado em aplicações práticas, há um espaço considerável para a exploração do paralelismo em outras classes de algoritmos de heurísticas. A dificuldade reside, em grande parte, na natureza interdependente de muitas heurísticas, onde as soluções em uma geração podem impactar diretamente as soluções subsequentes. Essa interdependência pode limitar a eficácia do paralelismo, uma vez que partes do algoritmo precisam ser executadas em sequência. No entanto, com o avanço das tecnologias de computação e a crescente disponibilidade de arquiteturas de hardware paralelo, como GPUs e clusters de computação, é possível repensar a estrutura de diversos algoritmos heurísticos. A implementação de técnicas que permitam a execução simultânea de avaliações de solução, combinações e até mesmo a geração de novos indivíduos pode abrir novos horizontes, aumentando a eficiência e a escalabilidade de abordagens que ainda não foram suficientemente exploradas sob a ótica do paralelismo. Assim, a busca por métodos que equilibrem a adaptabilidade das heurísticas com a capacidade de processamento paralelo pode resultar em avanços significativos na resolução de problemas complexos.

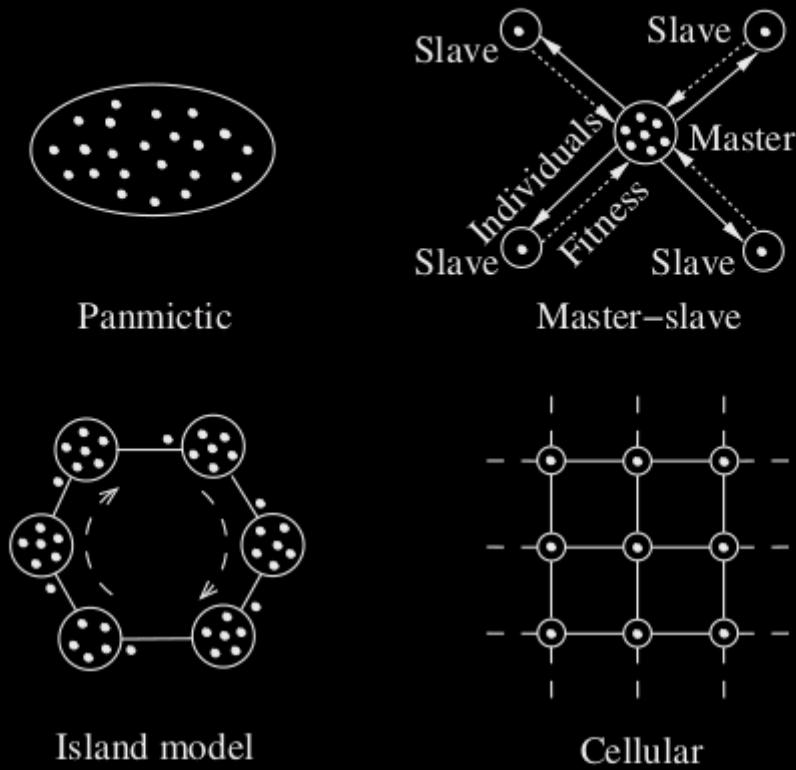


Figure 8.3 Parallel EA.

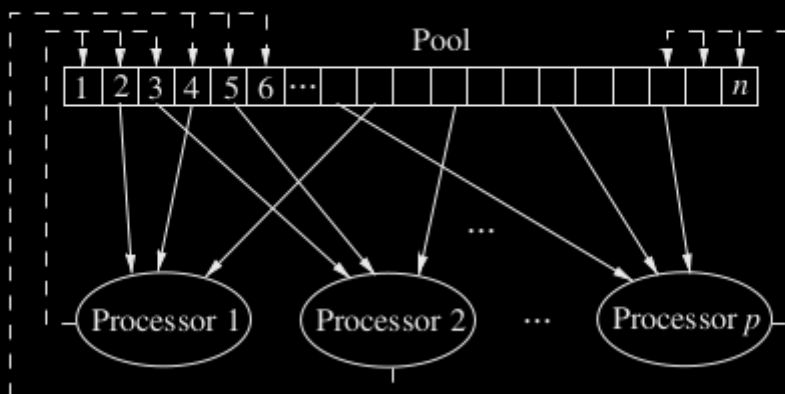


Figure 8.4 Pool-based EA.

**Table 8.1** Comparison of distributed models [42]

Model	Parallelism level	Diversity	Communication cost	Scalability	Fault-tolerance
Master-slave	Operation, evaluation	Like sequential EA	Medium to high	Medium	High
Island	Population	Good	Low to medium	Low	Medium
Cellular	Individual	Good	Medium	Medium to high	Medium to high
Hierarchical	Population, individual, operation	Good	Medium	Medium to high	Medium to high
Pool	Population, individual, operation	–	Low	High	High
Coevolution	Variable, variable-block	–	Medium	Low	Low

[Problemas em Técnicas de Otimização Heurística citado no Termo de Aceite de Entrega de 2 de outubro]

# Análise de Problemas e Aplicações em Otimização Heurística

## 1. Introdução à Otimização de Redes de Acesso Rádio UMTS com Algoritmos Genéticos

A otimização das redes de acesso rádio do Sistema Universal de Telecomunicações (UMTS) se torna crucial à medida que as redes móveis de terceira geração se expandem para suportar uma gama mais ampla de serviços, como internet, telefonia por vídeo e serviços em tempo real. O elevado custo de infraestrutura e licenciamento exigido para a implantação das redes UMTS, onde aproximadamente 80% do investimento se destina à instalação da rede, enfatiza a necessidade de otimização eficiente. Os operadores, frequentemente dependentes de métodos manuais e empíricos, buscam automatizar o processo de otimização para testar uma gama maior de configurações, melhorar o desempenho das redes e economizar tempo dos especialistas em rádio.

A abordagem de otimização adotada é conhecida como Planejamento Automático de Células (ACP), onde os Algoritmos Genéticos (GA) se destacam como uma técnica eficaz para resolver o problema de otimização. A complexidade do problema e o elevado número de parâmetros a serem otimizados tornam a utilização de métodos de otimização combinatória quase obrigatória.

### Estrutura do Capítulo

- **Parte 1:** Apresentação do funcionamento das redes UMTS e das quantidades envolvidas na análise de desempenho.
- **Parte 2:** Definição do problema de otimização para planejamento da rede UMTS.
- **Parte 3:** Aplicação do GA ao problema de planejamento de rede.
- **Parte 4:** Análise dos resultados obtidos para uma rede realista.

## 2. Algoritmos Genéticos Aplicados à Gestão do Tráfego Aéreo

O aumento constante do tráfego aéreo, desde o início da aviação comercial, trouxe desafios significativos na gestão do tráfego aéreo (ATM), especialmente em aeroportos e áreas de aproximação. Embora os aviões sejam amplamente otimizados e automatizados, o controle do tráfego aéreo ainda depende fortemente da experiência humana.

## Problemas Abordados

1. **Resolução de Conflitos em Vãos:** O gerenciamento do tráfego aéreo pode ser representado por uma série de filtros, cada um com objetivos específicos e gerenciando diferentes horizontes espaciais e temporais. O problema de resolução de conflitos em voo se destaca como uma aplicação em que um GA pode ser empregado para determinar manobras a serem ordenadas aos aviões, minimizando os conflitos e o atraso gerado.
2. **Gerenciamento do Tráfego em Plataformas de Aeroporto:** O controle do tráfego em áreas críticas de aeroportos envolve ajustes diários que consideram eventos recentes, como condições climáticas e horários de decolagem.

## Desafios de Otimização

A complexidade da resolução de conflitos decorre da necessidade de prever as trajetórias dos aviões, onde a modelagem se torna difícil devido a incertezas em velocidades de subida e descida, além das limitações operacionais de um piloto humano. O desenvolvimento do GA para este problema é detalhado, seguido pela apresentação dos resultados numéricos obtidos.

## 3. Programação por Restrições e Colônias de Formigas Aplicadas a Problemas de Roteamento de Veículos

Os problemas de roteamento de veículos são fundamentais para a eficiência das cadeias logísticas nas indústrias, onde o planejamento adequado pode levar a uma redução significativa dos custos de distribuição. Esses problemas incluem aplicações em transporte público, coleta de lixo, entrega de combustível e muito mais.

## Algoritmo de Colônia de Formigas

Inspirado pelo comportamento das colônias de formigas, o algoritmo de colônia de formigas combina abordagens construtivas e mecanismos de aprendizado baseados em memória. O

estudo de caso apresentado explora a aplicação desse método na resolução de problemas de roteamento de veículos, que incluem ordens de coleta e entrega, com a modelagem do problema feita por meio de programação por restrições.

## **Integração com Programação por Restrições**

A programação por restrições é utilizada para modelar os problemas de roteamento, permitindo a propagação de restrições e facilitando a implementação de soluções. O algoritmo de resolução, baseado em colônias de formigas, é desenvolvido para abordar a complexidade dos problemas de roteamento.

## Termo de Aceite de Entrega

### Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

**Data da Reunião (“gate”) de aprovação:** 9 de out. de 2024

**Participantes da Entrega** [matriculados em Residência em IA]:

Victor Guerreiro Pimenta

**Entrega:** [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Busca e leitura das referências externas no livro **Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature.**
  - Referências Fundamentos
- Busca sobre relação problemas/métodos nos algoritmos de otimização heurística.
  - Relação Problemas/Métodos

**Planejamento:** [descrever o que pretende fazer para realizar a próxima ENTREGA]

Busca de literatura científica realizada em 2023 e 2024 para entender o panorama dos avanços recentes e catalogar os conteúdos de acordo com as classificações que vêm sendo construídas nos documentos do processo.

**Observação:** [caso precise fazer alguma observação, de qualquer “natureza”]

## ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

[Referências Fundamentos citado no Termo de Aceite de Entrega de 9 de outubro]

Este documento foi desenvolvido com o objetivo de reunir e organizar **referências fundamentais** extraídas do livro "*Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature*". Ele serve como um **guia essencial** para estudantes, pesquisadores e profissionais interessados em aprofundar seus conhecimentos sobre **metaheurísticas, otimização e técnicas inspiradas pela natureza**.

As referências estão organizadas por capítulos, abordando temas específicos e destacando **artigos e publicações chave** que contribuíram para o desenvolvimento teórico e prático dessa área. Este esforço busca oferecer uma base sólida para **consultas, estudos e pesquisas**, promovendo uma compreensão interdisciplinar.

Ao reunir em um único documento essas contribuições valiosas, espera-se facilitar o acesso ao conhecimento e destacar a importância de reconhecer as **principais fontes de inspiração e avanços científicos** que moldaram o campo das metaheurísticas e da otimização.

## Cap 01 introdução

### References

1. Adleman LM. Molecular computation of solutions to combinatorial problems. *Science*. 1994;266:1021–4.
2. Auger A, Teytaud O. Continuous lunches are free plus the design of optimal optimization algorithms. *Algorithmica*. 2010;57:121–46.
3. Banks A, Vincent J, Phalp K. Natural strategies for search. *Nat Comput*. 2009;8:547–70.26  
1 Introduction
4. Barnard CJ, Sibly RM. Producers and scroungers: a general model and its application to flocks of house sparrows. *Anim Behav*. 1981;29:543–50.
5. Battail G. Heredity as an encoded communication process. *IEEE Trans Inf Theory*. 2010;56(2):678–87.
6. Beni G, Wang J. Swarm intelligence in cellular robotics systems. In: *Proceedings of NATO Advanced Workshop on Robots Biological Systems, Toscana, Italy, June 1989*, p. 703–712.
7. Bishop CM. *Neural networks for pattern recognition*. New York: Oxford Press; 1995.
8. Bonabeau E, Dorigo M, Theraulaz G. *Swarm intelligence: from natural to artificial systems*. New York: Oxford University Press; 1999.
9. Broom M, Koenig A, Borries C. Variation in dominance hierarchies among group-living ani-

- mals: modeling stability and the likelihood of coalitions. *Behav Ecol.* 2009;20:844–55.
10. Burke EK, Hyde MR, Kendall G. Grammatical evolution of local search heuristics. *IEEE Trans Evol Comput.* 2012;16(3):406–17.
11. Burke EK, Hyde MR, Kendall G, Ochoa G, Ozcan E, Woodward JR. Exploring hyper-heuristic methodologies with genetic programming. In: Mumford CL, Jain LC, editors. *Computational intelligence: collaboration, fusion and emergence.* Berlin, Heidelberg: Springer; 2009. p. 177–201.
12. Cataltepe Z, Abu-Mostafa YS, Magdon-Ismael M. No free lunch for early stopping. *Neural Comput.* 1999;11:995–1009.
13. Clark CW, Mangel M. Foraging and robbing strategies: information in an uncertain environment. *Am Nat.* 1984;123(5):626–41.
14. Cook SA. The complexity of theorem-proving procedures. In: *Proceedings of the 3rd ACM symposium on theory of computing*, Shaker Heights, OH, USA, May 1971, p. 151–158.
15. Couzin ID, Krause J, James R, Ruxton GD, Franks NR. Collective memory and spatial sorting in animal groups. *J Theoret Biol.* 2002;218:1–11.
16. de Castro LN, Timmis J. *Artificial immune systems: a new computational intelligence approach.* Springer; 2002.
17. Dorigo M, Maniezzo V, Colorni A. *Ant system: an autocatalytic optimizing process.* Technical Report 91-016, Politecnico di Milano, Milan, Italy, 1991.
18. Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst, Man, Cybern Part B.* 1996;26(1):29–41.
19. Du K-L, Swamy MNS. *Neural networks and statistical learning.* London: Springer; 2014.
20. Duenez-Guzman EA, Vose MD. No free lunch and benchmarks. *Evol Comput.* 2013;21(2):293–312.
21. Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: *Proceedings of the 6th International symposium on micro machine and human science*, Nagoya, Japan, October 1995, p. 39–43.
22. Engelbrecht AP. *Fundamentals of computational swarm intelligence.* New Jersey: Wiley; 2005.
23. Fisher H, Thompson GL. Probabilistic learning combinations of local job shop scheduling rules.

- In: Muth JF, Thompson GL, editors. Industrial scheduling. New Jersey: Prentice Hall;1963. p. 225–251.
24. Fletcher R. Practical methods of optimization. New York: Wiley; 1991.
25. Glover F. Future paths for integer programming and links to artificial intelligence. *Comput Oper Res.* 1986;13:533–49.
26. Glover F, Laguna M, Marti R. Scatter search. In: Ghosh A, Tsutsui S, editors. *Advances in evolutionary computing: theory and applications.* Berlin: Springer;2003. p. 519–537.
27. Hinton GE, Nowlan SJ. How learning can guide evolution. *Complex Syst.* 1987;1:495–502.
28. Hirvensalo M. Quantum computing. Springer. 2004.
29. Hodgkin AL, Huxley AF. Quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol.* 1952;117:500.
30. Holland JH. Outline for a logical theory of adaptive systems. *J ACM.* 1962;9(3):297–314. References
- 27
31. Hooke R, Jeeves TA. “Direct search” solution of numerical and statistical problems. *J ACM.* 1961;8(2):212–29.
32. Hopfield JJ, Tank DW. Neural computation of decisions in optimization problems. *Biol Cybern.* 1985;52:141–52.
33. Hoppe W, Lohmann W, Markl H, Ziegler H. *Biophysics.* New York: Springer; 1983.
34. Igel C, Toussaint M. A no-free-lunch theorem for non-uniform distributions of target functions. *J Math Model Algorithms.* 2004;3(4):313–22.
35. Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Kayseri, Turkey. 2005.
36. Kasabov N. Integrative connectionist learning systems inspired by nature: current models, future trends and challenges. *Natural Comput.* 2009;8:199–218.
37. Kirkpatrick S, Gelatt CD Jr, Vecchi MP. Optimization by simulated annealing. *Science.* 1983;220:671–80.
38. Kleene SC. *Introduction to metamathematics.* Amsterdam: North Holland; 1952.
39. Ku KWC, Mak MW, Siu WC. Approaches to combining local and evolutionary search for training neural networks: a review and some new results. In: Ghosh A, Tsutsui S, editors. *Advances in evolutionary computing: theory and applications.* Berlin: Springer; 2003. p. 615–641.
40. Lourenco HR, Martin O, Stutzle T. Iterated local search: framework and applications. In: *Handbook of metaheuristics, 2nd ed.* New York: Springer. 2010.

41. Magdon-Ismail M. No free lunch for noise prediction. *Neural Comput.* 2000;12:547–64.
42. Martin O, Otto SW, Felten EW. Large-step Markov chains for the traveling salesman problem. *Complex Syst.* 1991;5:299–326.
43. McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys.* 1943;5:115–33.
44. Mirjalili S, Lewis A, Mostaghim S. Confidence measure: a novel metric for robust meta-heuristic optimisation algorithms. *Inf Sci.* 2015;317:114–42.
45. Mladenovic N, Hansen P. Variable neighborhood search. *Comput Oper Res.* 1997;24:1097–100.
46. Moore M, Narayanan A. Quantum-inspired computing. Technical Report, Department of Computer Science, University of Exeter, Exeter, UK. 1995.
47. Nelder JA, Mead R. A simplex method for function minimization. *Comput J.* 1965;7:308–13.
48. von Neumann J. Zur Theorie der Gesellschaftsspiele. *Ann Math.* 1928;100:295–320.
49. Ozcan E, Bilgin B, Korkmaz EE. A comprehensive analysis of hyper-heuristics. *Intell Data Anal.* 2008;12(1):3–23.
50. Passino KM. Biomimicry of bacterial foraging for distributed optimisation and control. *IEEE Control Syst Mag.* 2002;22(3):52–67.
51. Paun G. Membrane computing: an introduction. Berlin: Springer; 2002.
52. Ray T, Liew KM. Society and civilization: an optimization algorithm based on simulation of social behavior. *IEEE Trans Evol Comput.* 2003;7:386–96.
53. Reynolds RG. An introduction to cultural algorithms. In: Proceedings of the 3rd Annual conference on evolutionary programming, San Diego, CA, USA. New Jersey: World Scientific; 1994. p. 131–139
54. Solis FJ, Wets RJ. Minimization by random search techniques. *Math Oper Res.* 1981;6:19–30.
55. Stannett M. The computational status of physics: a computable formulation of quantum theory. *Nat Comput.* 2009;8:517–38.
56. Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim.* 1997;11:341–59.
57. Sumper D. The principles of collective animal behaviour. *Philos Trans R Soc B.* 2006;36(1465):5–22.
58. Talbi E-G. Metaheuristics: from design to implementation. Hoboken, NJ: Wiley; 2009. 281 Introduction

59. Turney P. Myths and legends of the Baldwin effect. In: Proceedings of the 13th international conference on machine learning, Bari, Italy, July 1996, p. 135–142.
60. Turney P. How to shift bias: lessons from the Baldwin effect. *Evol Comput.* 1997;4(3):271–95.
61. Voudouris C, Tsang E. Guided local search. Technical Report CSM-247, University of Essex, Colchester, UK. 1995.
62. Whitley D, Gordon VS, Mathias K. Lamarckian evolution, the Baldwin effect and function optimization. In: Proceedings of the 3rd Conference on parallel problem solving from nature (PPSN III), Jerusalem, Israel, October 1994. p. 6–15.
63. Wolpert DH, Macready WG. No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute, Santa Fe, NM, USA. 1995.
64. Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Trans Evol Comput.* 1997;1(1):67–82.
65. Wolpert DH, Macready WG. Coevolutionary free lunches. *IEEE Trans Evol Comput.* 2005;9(6):721–35.
66. Yao X. Evolving artificial neural networks. *Proc IEEE.* 1999;87(9):1423–47.
67. Zhu H. No free lunch for cross validation. *Neural Comput.* 1996;8(7):1421–6.
68. Zimmermann HJ, Sebastian HJ. Intelligent system design support by fuzzy-multi-criteria decision making and/or evolutionary algorithms. In: Proceedings of IEEE international conference on fuzzy systems, Yokohama, Japan, March 1995. p. 367–374.
69. Zitzler E, Thiele L, Laumanns M, Fonseca CM, da Fonseca VG. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans Evol Comput.* 2003;7:117–32.

## Cap 08 Topics in EA

### References

1. Akbarzadeh-T M-R, Davarynejad M, Pariz N. Adaptive fuzzy fitness granulation for evolutionary optimization. *Int J Approx Reason.* 2008;49:523–38.
2. Alba E. Parallel evolutionary algorithms can achieve superlinear performance. *Inf Process Lett.* 2002;82(1):7–13.
3. Alba E, Dorronsoro B. The exploration/exploitation tradeoff in dynamic cellular evolutionary algorithms. *IEEE Trans Evol Comput.* 2005;9(2):126–42.
4. Alba E, Tomassini M. Parallelism and evolutionary algorithms. *IEEE Trans Evol Comput.* 2002;6(5):443–62.

5. Al-Madi NA. De Jong's sphere model test for a human community based genetic algorithm model (HCBGA). *Int J Adv Compu Sci Appl*. 2014;5(1):166–172.
6. Al-Madi NA, Khader AT. A social based model for genetic algorithms. In: Proceedings of the 3rd international conference on information technology (ICIT), Amman, Jordan, May 2007. p. 23–27
7. Al-Naqi A, Erdogan AT, Arslan T. Adaptive three-dimensional cellular genetic algorithm for balancing exploration and exploitation processes. *Soft Comput*. 2013;17:1145–57.
8. Arora R, Tulshyan R, Deb K. Parallelization of binary and realcoded genetic algorithms on GPU using CUDA. In: Proceedings of IEEE world congress on computational intelligence, Barcelona, Spain, July 2010. p. 3680–3687.
9. Arsuaga-Rios M, Vega-Rodriguez MA. Multiobjective energy optimization in grid systems from a brain storming strategy. *Soft Comput*. 2015;19:3159–72.
10. Bai H, Ouyang D, Li X, He L, Yu H. MAX-MIN ant system on GPU with CUDA. In: Proceedings of the IEEE 4th international conference on innovative computing, information and control (ICICIC), Kaohsiung, Taiwan, Dec 2009. p. 801–204.
11. Barabasi AL, Freeh VW, Jeong H, Brockman JB. Parasitic computing. *Nature*. 2001;412(6850):894–7. References 147
12. Barbosa HJC. A genetic algorithm for min-max problems. In: Proceedings of the 1st international conference on evolutionary computation and applications, Moscow, Russia, 1996. p. 99–109.
13. Beyer H-G. An alternative explanation for the manner in which genetic algorithms operate. *Biosystems*. 1997;41(1):1–15.
14. Biles J. Genjam: a genetic algorithm for generating jazz solos. In: Proceedings of international computer music conference, Arhus, Denmark, 1994. p. 131–137.
15. Bongard J, Zykov V, Lipson H. Resilient machines through continuous self-modeling. *Science*. 2006;314(5802):1118–21.
16. Bozejko W, Smutnicki C, Uchronski M. Parallel calculating of the goal function in meta-heuristics using GPU. In: Proceedings of the 9th international conference on computational science, Baton Rouge, LA, USA, May 2009, vol. 5544 of Lecture Notes in Computer Science. Berlin: Springer; 2009. p. 1014–2023.
17. Brownlee AEI, McCall JAW, Zhang Q. Fitness modeling with Markov networks. *IEEE Trans Evol Comput*. 2013;17(6):862–79.
18. Calazan RM, Nedjah N, De Macedo Mourelle L. Parallel GPU-based implementation of high dimension particle swarm optimizations. In: Proceedings of the IEEE 4th Latin American

- symposium on circuits and systems (LASCAS), Cusco, Peru, Feb 2013. p. 1–4.
19. Caldwell C, Johnston VS. Tracking a criminal suspect through “face-space” with a genetic algorithm. In: Proceedings of the 4th international conference on genetic algorithms, San Diego, CA, USA, July 1991. San Diego, CA: Morgan Kaufmann; 1991. p. 416–421
20. Candan C, Dreo J, Saveant P, Vidal V. Parallel divide-and-evolve: experiments with Open-MP on a multicore machine. In: Proceedings of GECCO, Dublin, Ireland, July 2011. p. 1571–1578.
21. Cerf R. Asymptotic convergence of genetic algorithms. *Adv Appl Probab.* 1998;30(2):521–50.
22. Cheang SM, Leung KS, Lee KH. Genetic parallel programming: design and implementation. *Evol Comput.* 2006;14(2):129–56.
23. Collet P, Lutton E, Schoenauer M, Louchet J. Take it EASEA. In: Proceedings of the 6th international conference on parallel problem solving from nature (PPSN VI), Paris, France, Sept 2000, vol. 1917 of Lecture Notes in Computer Science. London: Springer; 2000. p. 891–901
24. Collins RJ, Jefferson DR. Selection in massively parallel genetic algorithms. In: Belew RK, Booker LB, editors. Proceedings of the 4th international conference on genetic algorithms, San Diego, CA, USA, July 1991. San Diego, CA: Morgan Kaufmann; 1991. p. 249–256.
25. Corno F, Reorda M, Squillero G. The selfish gene algorithm: a new evolutionary optimization strategy. In: Proceedings of the 13th annual ACM symposium on applied computing (SAC), Atlanta, Georgia, USA, 1998. p. 349–355.
26. Cramer AM, Sudhoff SD, Zivi EL. Evolutionary algorithms for minimax problems in robust design. *IEEE Trans Evol Comput.* 2009;13(2):444–53.
27. Dawkins R. *The selfish gene.* Oxford: Oxford University Press; 1989.
28. De Jong K. An analysis of the behavior of a class of genetic adaptive systems. PhD Thesis, University of Michigan, Ann Arbor, 1975.
29. de Veronese PL, Krohling RA. Differential evolution algorithm on the GPU with C-CUDA. In: Proceedings of IEEE world congress on computational intelligence, Barcelona, Spain, July 2010. p. 1878–1884.
30. Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. In: Proceedings of the 6th symposium on operating system design and implementation (OSDI), San Francisco, CA, 2004. p. 137–147.
31. Droste S, Jansen T, Wegener I. On the analysis of the (1+1) evolutionary algorithm. *Theor Comput Sci.* 2002;276:51–81.

32. Du K-L, Swamy MNS. Neural networks and statistical learning. London: Springer; 2014.148
- 8 Topics in Evolutionary Algorithms
33. Eiben AE, Aarts EHL, Van Hee KM. Global convergence of genetic algorithms: a Markov chain analysis. In: Proceedings of the 1st workshop on parallel problem solving from nature (PPSN I), Dortmund, Germany, Oct 1990. Berlin: Springer; 1991. p. 3–12.
34. Emmerich MTM, Giannakoglou KC, Naujoks B. Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Trans Evol Comput.* 2006;10(4):421–39.
35. Ewald G, Kurek W, Brdys MA. Grid implementation of a parallel multiobjective genetic algorithm for optimized allocation of chlorination stations in drinking water distribution systems: Chojnice case study. *IEEE Trans Syst Man Cybern Part C.* 2008;38(4):497–509.
36. Fok K-L, Wong T-T, Wong M-L. Evolutionary computing on consumer graphics hardware. *IEEE Intell Syst.* 2007;22:69–78.
37. Folino G, Pizzuti C, Spezzano G. A scalable cellular implementation of parallel genetic programming. *IEEE Trans Evol Comput.* 2003;7(1):37–53.
38. Ge H, Sun L, Yang X, Yoshida S, Liang Y. Cooperative differential evolution with fast variable interdependence learning and cross-cluster mutation. *Appl Soft Comput.* 2015;36:300–14.
39. Goh C-K, Tan KC. A competitive-cooperative coevolutionary paradigm for dynamic multi-objective optimization. *IEEE Trans Evol Comput.* 2009;13(1):103–27.
40. Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Reading, MA, USA: Addison-Wesley; 1989.
41. Goldberg DE, Deb K, Korb B. Messy genetic algorithms: motivation, analysis, and first results. *Complex Syst.* 1989;3:493–530.
42. Gong Y-J, Chen W-N, Zhan Z-H, Zhang J, Li Y, Zhang Q, Li J-J. Distributed evolutionary algorithms and their models: a survey of the state-of-the-art. *Appl Soft Comput.* 2015;34:286–300.
43. Grefenstette JJ. Deception considered harmful. In: Whitley LD, editor. Foundations of genetic algorithms, vol. 2. Morgan Kaufmann: San Mateo, CA; 1993. p. 75–91.
44. Hastings EJ, Guha RK, Stanley KO. Interactive evolution of particle systems for computer graphics and animation. *IEEE Trans Evol Comput.* 2009;13(2):418–32.
45. Herrmann JW. A genetic algorithm for minimax optimization problems. In: Proceedings of the congress on evolutionary computation (CEC), Washington DC, July 1999, vol. 2. p. 1099–1103.
46. He J, Yao X. Drift analysis and average time complexity of evolutionary algorithms. *Artif*

Intell. 2001;127:57–85.

47. He J, Yao X. From an individual to a population: an analysis of the first hitting time of population-based evolutionary algorithms. *IEEE Trans Evol Comput.* 2002;6(5):495–511.

48. He J, Yao X. Analysis of scalable parallel evolutionary algorithms. In: *Proceedings of the IEEE congress on evolutionary computation (CEC)*, Vancouver, BC, Canada, July 2006. p. 120–127.

49. He J, Yu X. Conditions for the convergence of evolutionary algorithms. *J Syst Arch.* 2001;47(7):601–12.

50. Holland J. *Adaptation in natural and artificial systems.* Ann Arbor, Michigan: University of Michigan Press; 1975.

51. Holland JH. Building blocks, cohort genetic algorithms and hyperplane-defined functions. *Evol Comput.* 2000;8(4):373–91.

52. Horn J. Finite Markov chain analysis of genetic algorithms with niching. In: *Proceedings of*

the 5th international conference on genetic algorithms, Urbana, IL, July 1993. San Francisco, CA: Morgan Kaufmann Publishers; 1993. p. 110–117

53. Jansen T, De Jong KA, Wegener I. On the choice of the offspring population size in evolu-

tionary algorithms. *Evol Comput.* 2005;13(4):413–40.

54. Jansen T, Wegener I. The analysis of evolutionary algorithms—a proof that crossover really

can help. *Algorithmica.* 2002;33:47–66. References  
149

55. Jin H, Frumkin M, Yan J. The OpenMP implementation of NAS parallel benchmarks and its

performance. MRJ Technology Solutions, NASA Contract NAS2-14303, Moffett Field, CA, Oct 1999.

56. Jin Y, Sendhoff B. Reducing fitness evaluations using clustering techniques and neural network

ensembles. In: *Proceedings of genetic and evolutionary computation*, Seattle, WA, USA, July 2004. p. 688–699.

57. Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. *J Global Optim.* 1998;13(4):455–92.

58. Kim H-S, Cho S-B. An efficient genetic algorithms with less fitness evaluation by clustering.

In: *Proceedings of IEEE congress on evolutionary computation (CEC)*, Seoul, Korea, May 2001. p. 887–894.

59. Koza JR. *Genetic programming: on the programming of computers by means of natural selection.* Cambridge, MA: MIT Press; 1992.

60. Krawiec K, Bhanu B. Coevolution and linear genetic programming for visual learning. In: *Proceedings of genetic and evolutionary computation conference (GECCO)*, Chicago, Illinois,

- USA, vol. 2723 of Lecture Notes of Computer Science. Berlin: Springer; 2003. p. 332–343
61. Krohling RA, Coelho LS. Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems. *IEEE Trans Syst Man Cybern Part B*. 2006;36(6):1407–16.
62. Lassig J, Sudholt D. Design and analysis of migration in parallel evolutionary algorithms. *Soft Comput*. 2013;17:1121–44.
63. Lastra M, Molina D, Benitez JM. A high performance memetic algorithm for extremely high-dimensional problems. *Inf Sci*. 2015;293:35–58.
64. Lehman J, Stanley KO. Abandoning objectives: evolution through the search for novelty alone. *Evol Comput*. 2011;19(2):189–223.
65. Lehre PK, Yao X. On the impact of mutation-selection balance on the runtime of evolutionary algorithms. *IEEE Trans Evol Comput*. 2012;16(2):225–41.
66. Leung Y, Gao Y, Xu Z-B. Degree of population diversity: a perspective on premature convergence in genetic algorithms and its Markov chain analysis. *IEEE Tran Neural Netw*. 1997;8(5):1165–76.
67. Liu J, Zhong W, Jiao L. A multiagent evolutionary algorithm for constraint satisfaction problems. *IEEE Trans Syst Man Cybern Part B*. 2006;36(1):54–73.
68. Liu J, Zhong W, Jiao L. A multiagent evolutionary algorithm for combinatorial optimization problems. *IEEE Trans Syst Man Cybern Part B*. 2010;40(1):229–40.
69. Mallipeddi R, Lee M. An evolving surrogate model-based differential evolution algorithm. *Appl Soft Comput*. 2015;34:770–87.
70. Manderick B, Spiessens P. Fine-grained parallel genetic algorithms. In: Schaffer JD, editor. *Proceedings of the 3rd international conference on genetic algorithms*, Fairfax, Virginia, USA, June 1989. San Mateo, CA: Morgan Kaufmann; 1989. p. 428–433.
71. Merelo-Guervos JJ. Fluid evolutionary algorithms. In: *Proceedings of IEEE congress on evolutionary computation*, Barcelona, Spain, July 2010. p. 1–8.
72. Meri K, Arenas MG, Mora AM, Merelo JJ, Castillo PA, Garcia-Sanchez P, Laredo JLJ. Cloud-based evolutionary algorithms: an algorithmic study. *Natural Comput*. 2013;12(2):135–47.
73. Meyer-Spradow J, Loviscach J. Evolutionary design of BRDFs. In: Chover M, Hagen H, Tost D, editors. *Eurographics 2003 short paper proceedings*. Spain: Granada; 2003. p. 301–6.
74. Muhlenbein H. Parallel genetic algorithms, population genetics and combinatorial optimization. In: Schaffer JD, editor. *Proceedings of the 3rd international conference on genetic algorithms*, Fairfax, Virginia, USA, June 1989. San Mateo, CA: Morgan Kaufman; 1989. p. 416–421.

75. Muhlenbein H, Schomisch M, Born J. The parallel genetic algorithm as a function optimizer.  
In: Proceedings of the 4th international conference on genetic algorithms, San Diego, CA, July 1991. p. 271–278.150  
8 Topics in Evolutionary Algorithms
76. Munawar A, Wahib M, Munawar A, Wahib M. Theoretical and empirical analysis of a GPU based parallel Bayesian optimization algorithm. In: Proceedings of IEEE international conference on parallel and distributed computing, applications and technologies, Higashi Hiroshima, Japan, Dec 2009. p. 457–462.
77. Nara K, Takeyama T, Kim H. A new evolutionary algorithm based on sheep flocks heredity model and its application to scheduling problem. In: Proceedings of IEEE international conference on systems, man, and cybernetics, Tokyo, Japan, Oct 1999, vol. 6. p. 503–508.
78. Niwa T, Iba H. Distributed genetic programming: empirical study and analysis. In: Proceedings of the 1st annual conference on genetic programming, Stanford University, CA, USA, July 1996. p. 339–344.
79. Nix AE, Vose MD. Modeling genetic algorithms with markov chains. *Ann Math Artif Intell.* 1992;5:79–88.
80. Omidvar MN, Li X, Mei Y, Yao X. Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Trans Evol Comput.* 2014;18(3):378–93.
81. Ong YS, Nair PB, Kean AJ. Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA J.* 2003;41(4):687–96.
82. O'Reilly UM, Oppacher F. The troubling aspects of a building-block hypothesis for genetic programming. In: Whitley LD, Vose MD, editors. *Foundations of genetic algorithm 3.* San Francisco, CA: Morgan Kaufmann; 1995. p. 73–88
83. Panait L. Theoretical convergence guarantees for cooperative coevolutionary algorithms. *Evol Comput.* 2010;18(4):581–615.
84. Poli R. Parallel distributed genetic programming. In: Come D, Dorigo M, Glover F, editors. *New ideas in optimization.* New York: McGraw-Hill; 1999.
85. Poli R. Exact schema theory for GP and variable-length GAs with one-point crossover. *Genetic Progr Evol Mach.* 2001;2:123–63.
86. Poli R, Langdon WB. Schema theory for genetic programming with one-point crossover and point mutation. *Evol Comput.* 2001;6(3):231–52.
87. Poli R, McPhee NF. General schema theory for genetic programming with subtree-swapping

- crossover: part i. *Evol Comput.* 2003;11(1):53–66.
88. Poli R, McPhee NF. General schema theory for genetic programming with subtree-swapping crossover: part ii. *Evol Comput.* 2003;11(2):169–206.
89. Potter MA, de Jong KA. A cooperative coevolutionary approach to function optimization. In: *Proceedings of the 3rd conference on parallel problem solving from nature (PPSN III)*, Jerusalem, Isreal, Oct 1994. Berlin: Springer; 1994. p. 249–257.
90. Potter MA, De Jong KA. Cooperative coevolution: an architecture for evolving coadapted subcomponenets. *Evol Comput.* 2000;8(1):1–29.
91. Qi X, Palmieri F. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space, part 1: basic properties of selection and mutation. *IEEE Trans Neural Netw.* 2004;5(1):102–19.
92. Ratle A. Accelerating the convergence of evolutionary algorithms by fitness landscape approximation. In: *Parallel problem solving from nature (PPSN V)*, 1998. p. 87–96.
93. Regis RG, Shoemaker CA. Local function approximation in evolutionary algorithms for the optimization of costly functions. *IEEE Trans Evol Comput.* 2004;8(5):490–505.
94. Reza A, Vahid Z, Koorush Z. MLGA: a multilevel cooperative genetic algorithm. In: *Proceedings of the IEEE 5th international conference on bio-inspired computing: theories and applications (BIC-TA)*, Changsha, China, Sept 2010. p. 271–277.
95. Rosin C, Belew R. New methods for competitive coevolution. *Evol Comput.* 1997;15(1):1–29.
96. Rudolph G. Convergence analysis of canonical genetic algorithm. *IEEE Trans Neural Netw.* 1994;5(1):96–101.
97. Rudolph G. Finite Markov chain results in evolutionary computation: a tour d’horizon. *Fundamenta Informaticae.* 1998;35:67–89. References 151
98. Rudolph G. Self-adaptive mutations may lead to premature convergence. *IEEE Trans Evol Comput.* 2001;5:410–4.
99. Salami M, Hendtlass T. A fast evaluation strategy for evolutionary algorithms. *Appl Soft Comput.* 2003;2(3):156–73.
100. Sastry K, Goldberg DE, Pelikan M. Don’t evaluate, inherit. In: *Proceedings of genetic evolutionary computation conference (GECCO)*, San Francisco, CA, USA, July 2001. p. 551–558.
101. Schmidt MD, Lipson H. Coevolution of fitness predictors. *IEEE Trans Evol Comput.* 2008;12(6):736–49.

102. Schutte JF, Reinbolt JA, Fregly BJ, Haftka RT, George AD. Parallel global optimization with the particle swarm algorithm. *Int J Numer Methods Eng.* 2004;61(13):2296–315.
103. Shi Y, Krohling RA. Co-evolutionary particle swarm optimization to solve min-max problems. In: *Proceedings of the congress on evolutionary computation (CEC), Honolulu, HI, May 2002, vol. 2.* p. 1682–1687.
104. Smith RE, Dike BA, Stegmann SA. Fitness inheritance in genetic algorithms. In: *Proceedings of ACM symposium on applied computing, Nashville, Tennessee, USA, 1995.* p. 345–350.
105. Smith J, Vavak F. Replacement strategies in steady state genetic algorithms: static environments. In: Banzhaf W, Reeves C, editors. *Foundations of genetic algorithms, vol. 5.* CA: Morgan Kaufmann; 1999. p. 219–233.
106. Stephens CR, Poli R. Coarse-grained dynamics for generalized recombination. *IEEE Trans Evol Comput.* 2007;11(4):541–57.
107. Stephens CR, Waelbroeck H. Schemata evolution and building blocks. *Evol Comput.* 1999;7:109–29.
108. Sudholt D. A new method for lower bounds on the running time of evolutionary algorithms. *IEEE Trans Evol Comput.* 2013;17(3):418–35.
109. Sudholt D. How crossover speeds up building-block assembly in genetic algorithms. *Evol Comput* 2016.
110. Szumlanski SR, Wu AS, Hughes CE. Conflict resolution and a framework for collaborative interactive evolution. In: *Proceedings of the 21st national conference on artificial intelligence (AAAI), Boston, Massachusetts, USA, July 2006.* p. 512–517.
111. Takagi H. Interactive evolutionary computation: fusion of the capacities of EC optimization and human evaluation. *Proc IEEE.* 2001;89(9):1275–96.
112. Tasoulis DK, Pavlidis NG, Plagianakos VP, Vrahatis MN. Parallel differential evolution. In: *Proceedings of the IEEE congress on evolutionary computation, Portland, OR, USA, June 2004.* p. 2023–2029.
113. Thomsen R, Rickers P, Krink T. A religion-based spatial model for evolutionary algorithms. In: *Proceedings of the 6th international conference on parallel problem solving from nature (PPSN VI), Paris, France, September 2000, vol. 1917 of Lecture Notes in Computer Science.* London: Springer; 2000. p. 817–826.

114. van den Bergh F, Engelbrecht A. A cooperative approach to particle swarm optimization. *IEEE Trans Evol Comput.* 2004;8(3):225–39.
115. Vose M, Liepins G. Punctuated equilibria in genetic search. *Complex Syst.* 1991;5:31–44.
116. Weber M, Neri F, Tirronen V. Distributed differential evolution with explorative-exploitative population families. *Genetic Progr Evol Mach.* 2009;10:343–471.
117. Whitley D, Starkweather T. GENITOR II: a distributed genetic algorithm. *J Exp Theor Artif Intell.* 1990;2(3):189–214.
118. Whitley D, Yoo NW. Modeling simple genetic algorithms for permutation problems. In: Whitley D, Vose M, editors. *Foundations of genetic algorithms, vol. 3.* San Mateo, CA: Morgan Kaufmann; 1995. p. 163–184.
119. Wickramasinghe W, van Steen M, Eiben A. Peer-to-peer evolutionary algorithms with adaptive autonomous selection. In: *Proceedings of the 9th annual conference on genetic and evolutionary computation (GECCO), London, U.K., July 2007.* p. 1460–1467. 152
- 8 Topics in Evolutionary Algorithms
120. Wong M-L, Cui G. Data mining using parallel multiobjective evolutionary algorithms on graphics hardware. In: Sobrevilla P, editors. *Proceedings of IEEE world congress on computational intelligence, Barcelona, Spain, July 2010.* p. 3815–3822.
121. Wong M-L, Wong T-T, Fok K-L. Parallel evolutionary algorithms on graphics processing unit. In: *Proceedings of the IEEE congress on evolutionary computation, Edinburgh, UK, Sept 2005.* p. 2286–2293.
122. Xu L, Zhang F. Parallel particle swarm optimization for attribute reduction. In: *Proceedings of the 8th ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing, Qingdao, China, July 2007, vol. 1.* p. 770–775.
123. Yang Z, Tang K, Yao X. Large scale evolutionary optimization using cooperative coevolution. *Inf Sci.* 2008;178(15):2985–99.
124. Yao X, Liu Y. A new evolutionary system for evolving artificial neural networks. *IEEE Trans Neural Netw.* 1997;8(3):694–713.
125. Yuen SY, Cheung BKS. Bounds for probability of success of classical genetic algorithm based on Hamming distance. *IEEE Trans Evol Comput.* 2006;10(1):1–18.
126. Yu Y, Zhou Z-H. A new approach to estimating the expected first hitting time of evolutionary algorithms. *Artif Intell.* 2008;172(15):1809–32.
127. Zhang C, Chen J, Xin B. Distributed memetic differential evolution with the synergy of

- Lamarckian and Baldwinian learning. *Appl Soft Comput.* 2013;13(5):2947–59.
128. Zhong W, Liu J, Xue M, Jiao L. A multiagent genetic Algorithm for global numerical optimization. *IEEE Trans Syst Man Cybern Part B.* 2004;34(2):1128–41.
129. Zhou Z, Ong YS, Nair PB, Keane AJ, Lum KY. Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Trans Syst Man Cybern Part C.* 2007;37(1):66–76.

### cap 23 Multi Objective Optimization

#### References

1. Abbass HA, Sarker R, Newton C. PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems. In: *Proceedings of IEEE congress on evolutionary computation (CEC)*, Seoul, South Korea, May 2001. p. 971–978.
2. Abbass HA. The self-adaptive pareto differential evolution algorithm. In: *Proceedings of IEEE congress on evolutionary computation (CEC)*, Honolulu, HI, USA, May 2002. p. 831–836.
3. Agrawal S, Panigrahi BK, Tiwari MK. Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch. *IEEE Trans Evol Comput.* 2008;12(5):529–41.
4. Asafuddoula M, Ray T, Sarker R. A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Trans Evol Comput.* 2015;19(3):445–60.
5. Auger A, Bader J, Brockhoff D, Zitzler E. Theory of the hypervolume indicator: optimal  $\mu$ -distributions and the choice of the reference point. In: *Proceedings of the 10th ACM SIGEVO workshop on foundations of genetic algorithms (FOGA)*, Orlando, FL, USA, Jan 2009. p. 87–102.
6. Babbar M, Lakshmikantha A, Goldberg DE. A modified NSGA-II to solve noisy multi-objective problems. In: *Proceedings of genetic and evolutionary computation conference (GECCO)*, Chicago, IL, USA, July 2003. p. 21–27.
7. Bader J, Zitzler E. HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evol Comput.* 2011;19(1):45–76.
8. Bandyopadhyay S, Mukherjee A. An algorithm for many-objective optimization with reduced objective computations: a study in differential evolution. *IEEE Trans Evol Comput.* 2015;19(3):400–13.
9. Bandyopadhyay S, Saha S, Maulik U, Deb K. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Trans Evol Comput.* 2008;12(3):269–83.
10. Bastos-Filho CJA, Guimaraes ACS. Multi-objective fish school search. *Int J Swarm Intell Res.* 2015;6(1):18p.
11. Beausoleil RP. Moss: multiobjective scatter search applied to nonlinear multiple criteria opti-

- mization. *Eur J Oper Res.* 2006;169(2):426–49.
12. Bosman PAN, Thierens D. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans Evol Comput.* 2003;7(2):174–88.
13. Bosman PAN, Thierens D. The naive MIDEA: a baseline multi-objective EA. In: *Proceedings of the 3rd international conference on evolutionary multi-criterion optimization (EMO)*, Guanajuato, Mexico, March 2005. p. 428–442.
14. Branke J, Mostaghim S. About selecting the personal best in multiobjective particle swarm optimization. In: *Proceedings of conference on parallel problem solving from nature (PPSN IX)*, Reykjavik, Iceland, Sept 2006. Berlin: Springer; 2006. p. 523–532.404
- 23 Multiobjective Optimization
15. Branke J, Greco S, Slowinski R, Zielniewicz P. Learning value functions in interactive evolutionary multiobjective optimization. *IEEE Trans Evol Comput.* 2015;19(1):88–102.
16. Brockhoff D, Zitzler E. Objective reduction in evolutionary multiobjective optimization: theory and applications. *Evol Comput.* 2009;17(2):135–66.
17. Buchta C. On the average number of maxima in a set of vectors. *Inf Process Lett.* 1989;33(2):63–5.
18. Bui LT, Liu J, Bender A, Barlow M, Wesolkowski S, Abbass HA. DMEA: a direction-based multiobjective evolutionary algorithm. *Memetic Comput.* 2011;3:271–85.
19. Cai L, Qu S, Yuan Y, Yao X. A clustering-ranking method for many-objective optimization. *Appl Soft Comput.* 2015;35:681–94.
20. Camara M, de Toro F, Ortega J. An analysis of multiobjective evolutionary algorithms for optimization problems with time constraints. *Appl Artif Intell.* 2013;27:851–79.
21. Camara M, Ortega J, de Toro F. A single front genetic algorithm for parallel multi-objective optimization in dynamic environments. *Neurocomputing.* 2009;72:3570–9.
22. Chen Q, Guan S-U. Incremental multiple objective genetic algorithms. *IEEE Trans Syst Man Cybern Part B.* 2004;34(3):1325–34.
23. Clymont KM, Keedwell E. Deductive sort and climbing sort: new methods for non-dominated sorting. *Evol Comput.* 2012;20(1):1–26.
24. Coello CAC, Becerra RL. Evolutionary multiobjective optimization using a cultural algorithm. In: *Proceedings of IEEE swarm intelligence symposium, Indianapolis, IN, USA, April 2003.* p. 6–13.
25. Coello CAC, Cortes NC. Solving multiobjective optimization problems using an artificial immune system. *Genet Program Evolvable Mach.* 2005;6:163–90.
26. Coello CAC, Lechuga MS. MOPSO: a proposal for multiple objective particle swarm optimization. In: *Proceedings of IEEE congress on evolutionary computation (CEC)*, Honolulu,

HI, USA, May 2002. p. 1051–1056.

27. Coello CAC, Pulido GT. A micro-genetic algorithm for multiobjective optimization. In: Proceedings of the 1st international conference on evolutionary multi-criterion optimization (EMO), Zurich, Switzerland, March 2001. p. 126–140.

28. Coello CAC, Pulido GT, Lechuga MS. Handling multiple objectives with particle swarm optimization. *IEEE Trans Evol Comput.* 2004;8(3):256–79.

29. Corne DW, Jerram NR, Knowles JD, Oates MJ. PESA-II: region-based selection in evolutionary multiobjective optimization. In: Proceedings of genetic and evolutionary computation conference (GECCO), San Francisco, CA, USA, July 2001. p. 283–290.

30. Corne DW, Knowles JD. Techniques for highly multiobjective optimization: some nondomi-

nated points are better than others. In: Proceedings of the 9th ACM genetic and evolutionary computation conference (GECCO), London, UK, July 2007. p. 773–780.

31. Corne DW, Knowles JD, Oates MJ. The pareto envelope-based selection algorithm for multi-

objective optimisation. In: Proceedings of the 6th international conference on parallel problem solving from nature (PPSN VI), Paris, France, Sept 2000. Berlin: Springer; 2000. p. 839–848.

32. Costa M, Minisci E. MOPED: a multi-objective Parzen-based estimation of distribution algo-

rithm for continuous problems. In: Proceedings of the 2nd international conference on evolutionary multi-criterion optimization (EMO), Faro, Portugal, April 2003. p. 282–294.

33. Costa e Silva MA, Coelho LDS, Lebensztajn L. Multiobjective biogeography-based optimiza-

tion based on predator-prey approach. *IEEE Trans Magn.* 2012;48(2):951–954.

34. Dai X, Yuan X, Zhang Z. A self-adaptive multi-objective harmony search algorithm based on

harmony memory variance. *Appl Soft Comput.* 2015;35:541–57.

35. Deb K. Multi-objective genetic algorithms: problem difficulties and construction of test prob-

lems. *Evol Comput.* 1999;7(3):205–30.

36. Deb K. Multi-objective optimization using evolutionary algorithms. Chichester: Wiley; 2001. References

405

37. Deb K, Agrawal S, Pratap A, Meyarivan T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Proceedings of the 6th international conference on parallel problem solving from nature (PPSN VI), Paris, France, Sept 2000. Berlin: Springer; 2000. p. 849–858.

38. Deb K, Jain H. An evolutionary many-objective optimization algorithm using

reference-point

based non-dominated sorting approach, part i: solving problems with box constraints. *IEEE Trans Evol Comput.* 2013;18(4):577–601.

39. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput.* 2002;6(2):182–97.
40. Deb K, Saxena DK. On finding Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. *KanGAL Report, No.2005011.* 2005.
41. Deb K, Sinha A, Kukkonen S. Multi-objective test problems, linkages, and evolutionary methodologies. In: *Proceedings of genetic and evolutionary computation conference (GECCO)*, Seattle, WA, USA, July 2006. p. 1141–1148.
42. Deb K, Sundar J. Reference point based multiobjective optimization using evolutionary algorithms. In: *Proceedings of the 8th genetic and evolutionary computation conference (GECCO)*, Seattle, WA, USA, July 2006. p. 635–642.
43. Depolli M, Trobec R, Filipic B. Asynchronous master-slave parallelization of differential evolution for multi-objective optimization. *Evol Comput.* 2013;21(2):261–91.
44. di Pierro F, Khu S-T, Savic DA. An investigation on preference order ranking scheme for multiobjective evolutionary optimization. *IEEE Trans Evol Comput.* 2007;11(1):17–45.
45. Elhossini A, Areibi S, Dony R. Strength Pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization. *Evol Comput.* 2010;18(1):127–56.
46. Erickson M, Mayer A, Horn J. The niched pareto genetic algorithm 2 applied to the design of groundwater remediation systems. In: *Proceedings of the 1st international conference on evolutionary multi-criterion optimization (EMO)*, Zurich, Switzerland, March 2001. p. 681–695.
47. Fang H, Wang Q, Tu Y-C, Horstemeyer MF. An efficient non-dominated sorting method for evolutionary algorithms. *Evol Comput.* 2008;16(3):355–84.
48. Farina M, Amato P. On the optimal solution definition for many-criteria optimization problems. In: *Proceedings of the annual meeting of the North American fuzzy information processing society (NAFIPS)*, New Orleans, LA, USA, June 2002. p. 233–238.
49. Fleming PJ, Purshouse RC, Lygoe RJ. Many-objective optimization: an engineering design perspective. In: *Proceedings of international conference on evolutionary multi-criterion optimization (EMO)*, Guanajuato, Mexico, March 2005. p. 14–32.
50. Fonseca CM, Fleming PJ. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: Forrest S, editor. *Proceedings of the 5th international conference on genetic algorithms*, July 1993. San Francisco, CA: Morgan Kaufmann; 1993. p. 416–423.

51. Fonseca CM, Fleming PJ. Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part i: a unified formulation; Part ii: application example. *IEEE Trans Syst Man Cybern Part A*. 1998;28(1):26–37, 38–47.
52. Freschi F, Repetto M. Multiobjective optimization by a modified artificial immune system algorithm. In: *Proceedings of the 4th international conference on artificial immune systems (ICARIS)*, Banff, Alberta, Canada, Aug 2005. pp. 248–261.
53. Garcia-Martinez C, Cordon O, Herrera F. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *Eur J Oper Res*. 2007;180(1):116–48.
54. Ghasemi M, Ghavidel S, Ghanbarian MM, Gitizadeh M. Multi-objective optimal electric power planning in the power system using Gaussian bare-bones imperialist competitive algorithm. *Inf Sci*. 2015;294:286–304.406
- 23 Multiobjective Optimization
55. Giagkiozis I, Purshouse RC, Fleming PJ. Generalized decomposition and cross entropy methods for many-objective optimization. *Inf Sci*. 2014;282:363–87.
56. Goh C-K, Tan KC. A competitive-cooperative coevolutionary paradigm for dynamic multi-objective optimization. *IEEE Trans Evol Comput*. 2009;13(1):103–27.
57. Goh CK, Tan KC, Liu DS, Chiam SC. A competitive and cooperative coevolutionary approach to multi-objective particle swarm optimization algorithm design. *Eur J Oper Res*. 2010;202(1):42–54.
58. Goldberg DE. *Genetic algorithms in search, optimization, and machine learning*. Reading, MA, USA: Addison-Wesley; 1989.
59. Gong M, Jiao L, Du H, Bo L. Multiobjective immune algorithm with nondominated neighbor-based selection. *Evol Comput*. 2008;16(2):225–55.
60. Guevara-Souza M, Vallejo EE. Using a simulated Wolbachia infection mechanism to improve multi-objective evolutionary algorithms. *Nat Comput*. 2015;14:157–67.
61. Guzman MA, Delgado A, De Carvalho J. A novel multi-objective optimization algorithm based on bacterial chemotaxis. *Eng Appl Artif Intell*. 2010;23:292–301.
62. Hadka D, Reed P. Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization. *Evol Comput*. 2012;20(3):423–52.
63. Hadka D, Reed P. Borg: an auto-adaptive many-objective evolutionary computing framework. *Evol Comput*. 2013;21:231–59.
64. Hansen MP, Jaszkiwicz A. Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, Institute of Mathematical Modeling, Technical University of Denmark, Denmark; 1998.

65. He X-S, Li N, Yang X-S. Non-dominated sorting cuckoo search for multiobjective optimization. In: Proceedings of IEEE symposium on swarm intelligence (SIS), Orlando, FL, USA, Dec 2014. p. 1–7.
66. He Z, Yen GG. Many-objective evolutionary algorithm: objective space reduction and diversity improvement. *IEEE Trans Evol Comput.* 2016;20(1):145–60.
67. Hu X, Eberhart RC. Multiobjective optimization using dynamic neighborhood particle swarm optimization. In: Proceedings of congress on evolutionary computation (CEC), Honolulu, HI, USA, May 2002. p. 1677–1681.
68. Hu X, Eberhart RC, Shi Y. Particle swarm with extended memory for multiobjective optimization. In: Proceedings of IEEE swarm intelligence symposium, Indianapolis, IN, USA, April 2003. p. 193–197.
69. Huang L, He XX, Wang N, Xie Y. P systems based multi-objective optimization algorithm. *Prog Nat Sci.* 2007;17:458–65.
70. Huo Y, Zhuang Y, Gu J, Ni S. Elite-guided multi-objective artificial bee colony algorithm. *Appl Soft Comput.* 2015;32:199–210.
71. Horn J, Nafpliotis N, Goldberg DE. A niched pareto genetic algorithm for multiobjective optimization. In: Proceedings of the 1st IEEE conference on evolutionary computation, Orlando, FL, USA, June 1994. p. 82–87.
72. Ikeda K, Kita H, Kobayashi S. Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal? In: Proceedings of congress on evolutionary computation (CEC), Seoul, Korea, May 2001. p. 957–962.
73. Iorio AW, Li X. A cooperative coevolutionary multiobjective algorithm using non-dominated sorting. In: Proceedings of genetic and evolutionary computation conference (GECCO), Seattle, WA, USA, June 2004. p. 537–548.
74. Ishibuchi H, Murata T. Multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Trans Syst Man Cybern Part C.* 1998;28(3):392–403.
75. Jaimes AL, Coello CAC, Barrientos JEU. Online objective reduction to deal with many-objective problems. In: Proceedings of the 5th international conference on evolutionary multi-criterion optimization (EMO), Nantes, France, April 2009. p. 423–437. References 407
76. Jain H, Deb K. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part ii: handling constraints and extending to an adaptive approach. *IEEE Trans Evol Comput.* 2013;18(4):602–22.

77. Jensen MT. Reducing the run-time complexity of multiobjective eas: the NSGA-II and other algorithms. *IEEE Trans Evol Comput.* 2003;7(5):503–15.
78. Jiao L, Gong M, Shang R, Du H, Lu B. Clonal selection with immune dominance and energy based multiobjective optimization. In: *Proceedings of the 3rd international conference on evolutionary multi-criterion optimization (EMO)*, Guanajuato, Mexico, March 2005. p. 474–489.
79. Jiang S, Zhang J, Ong Y-S. Multiobjective optimization based on reputation. *Inf Sci.* 2014;286:125–46.
80. Keeratitumrong N, Chaiyaratana N, Varavithya V. Multi-objective co-operative co-evolutionary genetic algorithm. In: *Proceedings of the 7th international conference on parallel problem solving from nature (PPSN VII)*, Granada, Spain, Sept 2002. Berlin: Springer; 2002. p. 288–297.
81. Khan N. Bayesian optimization algorithms for multi-objective and hierarchically difficult problem. IlliGAL Report No. 2003021, Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA. 2003.
82. Khare V, Yao X, Deb K. Performance scaling of multiobjective evolutionary algorithms. In: *Proceedings of the 2nd international conference on evolutionary multi-criterion optimization (EMO)*, Faro, Portugal, April 2003. p. 376–390.
83. Knowles J. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans Evol Comput.* 2006;10(1):50–66.
84. Knowles JD, Corne DW. Approximating the nondominated front using the Pareto archived evolution strategy. *Evol Comput.* 2000;8(2):149–72.
85. Knowles JD, Corne DW. M-PAES: a memetic algorithm for multiobjective optimization. In: *Proceedings of IEEE congress on evolutionary computation (CEC)*, La Jolla, CA, USA, July 2000. p. 325–332.
86. Knowles JD, Corne DW. Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In: *Proceedings of the 4th international conference on evolutionary multi-criterion optimization (EMO)*, Matsushima, Japan, March 2007. p. 757–771.
87. Koppen M, Yoshida K. Substitute distance assignments in NSGAI for handling many-objective optimization problems. In: *Proceedings of the 4th international conference on evolutionary multi-criterion optimization (EMO)*, Matsushima, Japan, March 2007. p. 727–741.
88. Kukkonen S, Lampinen J. GDE3: the third evolution step of generalized differential evolution. In: *Proceedings of IEEE congress on evolutionary computation (CEC)*, Edinburgh, UK, Sept 2005. p. 443–450.
89. Kumar V, Chhabra JK, Kumar D. Differential search algorithm for multiobjective problems. *Procedia Comput Sci.* 2015;48:22–8.

90. Kundu D, Suresh K, Ghosh S, Das S, Panigrahi BK, Das S. Multi-objective optimization with artificial weed colonies. *Inf Sci.* 2011;181(12):2441–54.
91. Lara A, Sanchez G, Coello CAC, Schutze O. HCS: a new local search strategy for memetic multiobjective evolutionary algorithms. *IEEE Trans Evol Comput.* 2010;14(1):112–32.
92. Laumanns M, Ocenasek J. Bayesian optimization algorithms for multi-objective optimization. In: *Proceedings of the 7th international conference on parallel problem solving from nature (PPSN-VII)*, Granada, Spain, Sept 2002. Berlin: Springer; 2002. p. 298–307.
93. Laumanns M, Rudolph G, Schwefel H-P. A spatial predator-prey approach to multiobjective optimization: a preliminary study. In: *Proceedings of the 5th international conference on parallel problem solving from nature (PPSN-V)*, Amsterdam, The Netherlands, Sept 1998. Berlin: Springer; 1998. p. 241–249.
94. Laumanns M, Thiele L, Deb K, Zitzler E. Combining convergence and diversity in evolutionary multi-objective optimization. *Evol Comput.* 2002;10(3):263–82.408
- 23 Multiobjective Optimization
95. Li H, Zhang Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans Evol Comput.* 2009;13(2):284–302.
96. Li JQ, Pan QK, Gao KZ. Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems. *Int J Adv Manuf Technol.* 2011;55:1159–69.
97. Li K, Zhang Q, Kwong S, Li M, Wang R. Stable matching-based selection in evolutionary multiobjective optimization. *IEEE Trans Evol Comput.* 2014;18(6):909–23.
98. Li M, Yang S, Liu X. Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Trans Evol Comput.* 2014;18(3):348–65.
99. Li M, Yang S, Liu X. Bi-goal evolution for many-objective optimization problems. *Artif Intell.* 2015;228:45–65.
100. Li X. A non-dominated sorting particle swarm optimizer for multiobjective optimization. In: *Proceedings of genetic and evolutionary computation conference (GECCO)*, Chicago, IL, USA, July 2003. p. 37–48.
101. Li X. Better spread and convergence: particle swarm multiobjective optimization using the maximin fitness function. In: *Proceedings of genetic and evolutionary computation conference (GECCO)*, Seattle, WA, USA, June 2004. p. 117–128.
102. Li Z, Nguyen TT, Chen SM, Truong TK. A hybrid algorithm based on particle swarm and

- chemical reaction optimization for multi-object problems. *Appl Soft Comput.* 2015;35:525–40.
103. Liang Z, Song R, Lin Q, Du Z, Chen J, Ming Z, Yu J. A double-module immune algorithm for multi-objective optimization problems. *Appl Soft Comput.* 2015;35:161–74.
104. Liu D, Tan KC, Goh CK, Ho WK. A multiobjective memetic algorithm based on particle swarm optimization. *IEEE Trans Syst Man Cybern Part B.* 2007;37(1):42–50.
105. Lohn JD, Kraus WF, Haith GL. Comparing a coevolutionary genetic algorithm for multiobjective optimization. In: *Proceedings of the world on congress on computational intelligence, Honolulu, HI, USA, May 2002.* p. 1157–1162.
106. Lu H, Yen G. Rank-density-based multiobjective genetic algorithm and benchmark test function study. *IEEE Trans Evol Comput.* 2003;7(4):325–43.
107. Leong W-F, Yen GG. PSO-based multiobjective optimization with dynamic population size and adaptive local archives. *IEEE Trans Syst Man Cybern Part B.* 2008;38(5):1270–93.
108. Lopez-Jaimes A, Coello Coello CA. Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems. *Inf Sci.* 2014;277:1–20.
109. Lu Z, Zhao H, Xiao H, Wang H, Wang H. An improved multi-objective bacteria colony chemotaxis algorithm and convergence analysis. *Appl Soft Comput.* 2015;31:274–92.
110. Ma X, Qi Y, Li L, Liu F, Jiao L, Wu J. MOEA/D with uniform decomposition measurement for many-objective problems. *Soft Comput.* 2014;18:2541–64.
111. Madavan NK. Multiobjective optimization using a Pareto differential evolution approach. In: *Proceedings of IEEE congress on evolutionary computation (CEC), Honolulu, HI, USA, May 2002.* p. 1145–1150.
112. Marti L, Garcia J, Berlanga A, Molina JM. Solving complex high-dimensional problems with the multi-objective neural estimation of distribution algorithm. In: *Proceedings of the 11th genetic and evolutionary computation conference (GECCO), Montreal, Canada, July 2009.* p. 619–626.
113. Menczer F, Degeratu M, Steet WN. Efficient and scalable Pareto optimization by evolutionary local selection algorithms. *Evol Comput.* 2000;8(2):223–47.
114. Miettinen K. *Nonlinear multiobjective optimization.* Boston: Kluwer; 1999.
115. Mo H, Xu Z, Xu L, Wu Z, Ma H. Constrained multiobjective biogeography optimization algorithm. *Sci World J.* 2014;2014, Article ID 232714:12p.
116. Modiri-Delshad M, Rahim NA. Multi-objective backtracking search algorithm for economic

emission dispatch problem. *Appl Soft Comput.* 2016;40:479–94. References  
409

117. Molina J, Laguna M, Marti R, Caballero R. SSPMO: a scatter tabu search procedure for non-linear multiobjective optimization. *INFORMS J Comput.* 2007;19(1):91–100.

118. Mora AM, Garcia-Sanchez P, Merelo JJ, Castillo PA. Pareto-based multi-colony multi-objective ant colony optimization algorithms: an island model proposal. *Soft Comput.* 2013;17:1175–207.

119. Murata T, Ishibuchi H, Gen M. Specification of genetic search direction in cellular multi-objective genetic algorithm. In: *Proceedings of the 1st international conference on evolutionary multicriterion optimization (EMO), Zurich, Switzerland, March 2001.* Berlin: Springer; 2001. p. 82–95.

120. Nam DK, Park CH. Multiobjective simulated annealing: a comparative study to evolutionary algorithms. *Int J Fuzzy Syst.* 2000;2(2):87–97.

121. Nebro AJ, Durillo JJ, Luna F, Dorronsoro B, Alba E. MOCcell: a cellular genetic algorithm for multiobjective optimization. *Int J Intell Syst.* 2009;24:726–46.

122. Nebro AJ, Luna F, Alba E. New ideas in applying scatter search to multiobjective optimization.

In: *Proceedings of the 3rd international conference on evolutionary multicriterion optimization (EMO), Guanajuato, Mexico, March 2005.* p. 443–458.

123. Nebro AJ, Luna F, Alba E, Dorronsoro B, Durillo JJ, Beham A. AbYSS: adapting scatter search to multiobjective optimization. *IEEE Trans Evol Comput.* 2008;12(4):439–57.

124. Nguyen L, Bui LT, Abbass HA. DMEA-II: the direction-based multi-objective evolutionary algorithm-II. *Soft Comput.* 2014;18:2119–34.

125. Okabe T, Jin Y, Sendhoff B, Olhofer M. Voronoi-based estimation of distribution algorithm for

multi-objective optimization. In: *Proceedings of IEEE congress on evolutionary computation (CEC), Portland, OR, USA, June 2004.* p. 1594–1601.

126. Parsopoulos KE, Tasoulis DK, Pavlidis NG, Plagianakos VP, Vrahatis MN. Vector evaluated

differential evolution for multiobjective optimization. In: *Proceedings of IEEE congress on evolutionary computation (CEC), Portland, Oregon, USA, June 2004.* p. 204–211.

127. Parsopoulos KE, Tasoulis DK, Vrahatis MN. Multiobjective optimization using parallel vector

evaluated particle swarm optimization. In: *Proceedings of the IASTED international conference on artificial intelligence and applications, Innsbruck, Austria, Feb 2004.* p. 823–828.

128. Pelikan M, Sastry K, Goldberg DE. Multiobjective HBOA, clustering, and scalability. In: *Proceedings of international conference on genetic and evolutionary computation; 2005.* p. 663–670.

129. Pulido GT, Coello CAC. Using clustering techniques to improve the performance of a par-

- title swarm optimizer. In: Proceedings of genetic and evolutionary computation conference (GECCO), Seattle, WA, USA, June 2004. p. 225–237.
130. Purshouse RC, Fleming PJ. On the evolutionary optimization of many conflicting objectives. *IEEE Trans Evol Comput.* 2007;11(6):770–84.
131. Rahimi-Vahed A, Mirzaei AH. A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem. *Comput Ind Eng.* 2007;53(4):642–66.
132. Rao RV, Patel V. Multi-objective optimization of two stage thermoelectric cooler using a modified teaching-learning-based optimization algorithm. *Eng Appl Artif Intell.* 2013;26:430–45.
133. Ray T, Liew KM. A swarm metaphor for multiobjective design optimization. *Eng Optim.* 2002;34(2):141–53.
134. Reddy MJ, Kumar DN. An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design. *Eng Optim.* 2007;39(1):49–68.
135. Reynoso-Meza G, Sanchis J, Blasco X, Martinez M. Design of continuous controllers using a multiobjective differential evolution algorithm with spherical pruning. In: Applications of evolutionary computation. Lecture notes in computer science, vol. 6024. Berlin: Springer; 2010. p. 532–541.
136. Robic T, Filipic B. DEMO: differential evolution for multiobjective optimization. In: Proceedings of the 3rd international conference on evolutionary multi-criterion optimization (EMO), Guanajuato, Mexico, March 2005. p. 520–533.410
- 23 Multiobjective Optimization
137. Sadollah A, Eskandar H, Kim JH. Water cycle algorithm for solving constrained multi-objective optimization problems. *Appl Soft Comput.* 2015;27:279–98.
138. Sastry K, Goldberg DE, Pelikan M. Limits of scalability of multi-objective estimation of distribution algorithms. In: Proceedings of IEEE congress on evolutionary computation (CEC), Edinburgh, UK, Sept 2005. p. 2217–2224.
139. Sato H, Aguirre H, Tanaka K. Controlling dominance area of solutions and its impact on the performance of MOEAs. In: Proceedings of the 4th international conference on evolutionary multi-criterion optimization (EMO), Matsushima, Japan, March 2007. p. 5–20.
140. Schaffer JD. Multiple objective optimization with vector evaluated genetic algorithms. In: Grefenstette JJ, editor. Proceedings of the 1st international conference on genetic algorithms, Pittsburgh, PA, USA, July 1985. Hillsdale, NJ, USA: Lawrence Erlbaum; 1985. p. 93–100.
141. Schott JR. Fault tolerant design using single and multicriteria genetic algorithm optimization.

Master's Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA; 1995.

142. Shim VA, Tan KC, Cheong CY. An energy-based sampling technique for multi-objective restricted Boltzmann machine. *IEEE Trans Evol Comput.* 2013;17(6):767–85.

143. Shim VA, Tan KC, Chia JY, Al Mamun A. Multi-objective optimization with estimation of distribution algorithm in a noisy environment. *Evol Comput.* 2013;21(1):149–77.

144. Sierra MR, Coello CAC. Improving PSO-based multiobjective optimization using crowding,

mutation and  $\epsilon$ -dominance. In: *Proceedings of the 3rd international conference on evolutionary multi-criterion optimization (EMO)*, Guanajuato, Mexico, March 2005. p. 505–519.

145. Singh HK, Isaacs A, Ray T. A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *IEEE Trans Evol Comput.* 2011;15(4):539–56.

146. Smith KI, Everson RM, Fieldsend JE, Murphy C, Misra R. Dominance-based multiobjective

simulated annealing. *IEEE Trans Evol Comput.* 2008;12(3):323–42.

147. Soh H, Kirley M. moPGA: toward a new generation of multiobjective genetic algorithms. In:

*Proceedings of IEEE congress on evolutionary computation*, Vancouver, BC, Canada, July 2006. p. 1702–1709.

148. Soylu B, Köksalan M. A favorable weight-based evolutionary algorithm for multiple criteria

problems. *IEEE Trans Evol Comput.* 2010;14(2):191–205.

149. Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algo-

rithms. *Evol Comput.* 1994;2(3):221–48.

150. Srinivas M, Patnaik LM. Adaptive probabilities of crossover and mutation in genetic algo-

rithms. *IEEE Trans Syst Man Cybern.* 1994;24(4):656–67.

151. Tan KC, Lee TH, Khor EF. Evolutionary algorithms with dynamic population size and local

exploration for multiobjective optimization. *IEEE Trans Evol Comput.* 2001;5(6):565–88.

152. Tan KC, Yang YJ, Goh CK. A distributed cooperative coevolutionary algorithm for multiob-

jective optimization. *IEEE Trans Evol Comput.* 2006;10(5):527–49.

153. Tang HJ, Shim VA, Tan KC, Chia JY. Restricted Boltzmann machine based algorithm for multi-objective optimization. In: *Proceedings of IEEE congress on evolutionary computation (CEC)*, Barcelona, Spain, July 2010. p. 3958–3965.

154. Teo J. Exploring dynamic self-adaptive populations in differential evolution. *Soft Comput.*

2006;10(8):673–86.

155. Toffolo A, Benini E. Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evol Comput.* 2003;11(2):151–67.
156. Vasconcelos JA, Maciel JHRD, Parreiras RO. Scatter search techniques applied to electro-magnetic problems. *IEEE Trans Magn.* 2005;4:1804–7.
157. Veldhuizen DAV, Lamont GB. Multiobjective evolutionary algorithm research: a history and analysis. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, USA; 1998. References 411
158. Vrugt JA, Robinson BA, Hyman JM. Self-adaptive multimethod search for global optimization in real-parameter spaces. *IEEE Trans Evol Comput.* 2009;13(2):243–59.
159. Wagner T, Beume N, Naujoks B. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: *Proceedings of the 4th international conference on evolutionary multi-criterion optimization (EMO)*, Matsushima, Japan, March 2007. p. 742–756.
160. Wang R, Purshouse RC, Fleming PJ. Preference-inspired coevolutionary algorithms for many-objective optimization. *IEEE Trans Evol Comput.* 2013;17(4):474–94.
161. Wanner EF, Guimaraes FG, Takahashi RHC, Fleming PJ. Local search with quadratic approximations into memetic algorithms for optimization with multiple criteria. *Evol Comput.* 2008;16(2):185–224.
162. Wu Y, Jin Y, Liu X. A directed search strategy for evolutionary dynamic multiobjective optimization. *Soft Comput.* 2015;19:3221–35.
163. Xiang Y, Zhou Y. A dynamic multi-colony artificial bee colony algorithm for multi-objective optimization. *Appl Soft Comput.* 2015;35:766–85.
164. Xue J, Wu Y, Shi Y, Cheng S. Brain storm optimization algorithm for multi-objective optimization problems. In: *Proceedings of the 3rd international conference on advances in swarm intelligence*, Shenzhen, China, June 2012. Berlin: Springer; 2012. p. 513–519.
165. Yang S, Li M, Liu X, Zheng J. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput.* 2013;17(5):721–36.
166. Yang X-S. Bat algorithm for multi-objective optimization. *Int J Bio-Inspired Comput.* 2011;3(5):267–74.
167. Yen GG, Leong WF. Dynamic multiple swarms in multiobjective particle swarm optimization. *IEEE Trans Syst Man Cybern Part A.* 2009;39(4):890–911.

168. Yen GG, Lu H. Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation. *IEEE Trans Evol Comput.* 2003;7(3):253–74.
169. Zhan Z-H, Li J, Cao J, Zhang J, Chung HS-H, Shi Y-H. Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems. *IEEE Trans Cybern.* 2013;43(2):445–63.
170. Zhang Q, Li H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput.* 2007;11(6):712–31.
171. Zhang Q, Liu W, Li H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: *Proceedings of the IEEE conference on evolutionary computation (CEC), Trondheim, Norway, May 2009.* p. 203–208.
172. Zhang Q, Zhou A, Jin Y. Global multiobjective optimization via estimation of distribution algorithm with biased initialization and crossover. In: *Proceedings of the genetic and evolutionary computation conference (GECCO), London, UK, July 2007.* p. 617–622.
173. Zhang Q, Zhou A, Jin Y. RM-MEDA: a regularity model-based multi-objective estimation of distribution algorithm. *IEEE Trans Evol Comput.* 2008;12(1):41–63.
174. Zhang X, Tian Y, Cheng R, Jin Y. An efficient approach to non-dominated sorting for evolutionary multi-objective optimization. *IEEE Trans Evol Comput.* 2015;19(2):201–15.
175. Zhong X, Li W. A decision-tree-based multi-objective estimation of distribution algorithm. In: *Proceedings of international conference on computational intelligence and security, Harbin, China, Dec 2007.* p. 114–118.
176. Zhou A, Zhang Q, Jin Y. Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *Trans Evol Comput.* 2009;13(5):1167–89.
177. Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results. *Evol Comput.* 2000;8(2):173–95.
178. Zitzler E, Kunzli S. Indicator-based selection in multiobjective search. In: *Proceedings of the 8th international conference on parallel problem solving from nature (PPSN VIII), Birmingham, UK, Sept 2004.* Berlin: Springer; 1998. p. 832–842.412
- 23 Multiobjective Optimization
179. Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. TIK-Report 103, Departmentt of Electrical Engineering, Swiss Federal Institute of Technology, Switzerland. 2001.

180. Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength pareto evolutionary algo-  
rithm. In: Proceedings of evolutionary methods for design, optimisation and control. CIMNE,  
Barcelona, Spain; 2002. p. 95–100.
181. Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and  
the  
strength Pareto approach. IEEE Trans Evol Comput. 1999;3(4):257–71.
182. Zitzler E, Thiele L, Laumanns M, Fonseca CM, da Fonseca VG. Performance  
assessment of  
multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput. 2003;7:117–32.

[Relação Problemas/Métodos citado no Termo de Aceite de Entrega de 9 de outubro]

As **metaheurísticas** são métodos **poderosos e flexíveis** para resolver **problemas de otimização complexos** em diversas áreas. Diferentemente de **métodos exatos**, que buscam a **solução ótima com garantias teóricas**, as metaheurísticas oferecem **soluções aproximadas de alta qualidade** em um **tempo razoável**, sendo especialmente úteis para **problemas grandes** ou com **múltiplos critérios**. Este texto apresenta a aplicação de diferentes metaheurísticas — como **algoritmos genéticos**, **algoritmos de colônia de formigas**, e **busca local** — em áreas como **telecomunicações**, **transporte**, **finanças**, e **bioinformática**, demonstrando sua eficácia na resolução de **desafios práticos e estratégicos**.

### 1. Telecomunicações

- **Otimização de Redes UMTS:** Algoritmos Genéticos para planejamento de células, alocação de recursos, e otimização da qualidade de serviço.
- **Roteamento em Redes:** Algoritmos de colônia de formigas para encontrar rotas eficientes em redes de comunicação.

### 2. Transporte e Logística

- **Gestão do Tráfego Aéreo:** Algoritmos Genéticos para resolução de conflitos em voo e gerenciamento de tráfego em aeroportos.
- **Roteamento de Veículos:** Algoritmos de colônia de formigas e programação por restrições para problemas de coleta e entrega, transporte público, e logística em geral.

### 3. Planejamento e Programação

- **Planejamento da Produção:** Algoritmos Genéticos e programação linear inteira para otimizar a produção em fábricas.

- **Programação de Tarefas:** Algoritmos de busca local e metaheurísticas para otimizar a alocação de recursos e sequenciamento de tarefas.

#### 4. Finanças e Economia

- **Otimização de Portfólios:** Algoritmos evolutivos para maximizar o retorno e minimizar o risco em investimentos.
- **Gerenciamento de Riscos:** Algoritmos de simulação para avaliar e mitigar riscos em finanças.

#### 5. Bioinformática e Ciência dos Materiais

- **Projeto de Proteínas:** Algoritmos evolutivos para prever a estrutura de proteínas.
- **Desenvolvimento de Novos Materiais:** Algoritmos de otimização para descobrir novos materiais com propriedades desejáveis.

## APÊNDICE 3

## Termo de Aceite de Entrega

### Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

**Data da Reunião (“gate”) de aprovação:** 16 de out. de 2024

**Participantes da Entrega** [matriculados em Residência em IA]:

Victor Guerreiro Pimenta

**Entrega:** [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Busca de artigos de 2023 a 2024 de acordo com a metodologia documentada em
  - ☰ Relatório de Busca de Artigos
    - Metodologia de Busca
    - String de Busca
    - Amostragem de Artigos
- Atualização da ☰ Relação Problemas/Métodos v0.1 de acordo com a busca realizada.
- Reflexão sobre etimologia, origem e significado de Meta-, Hiper-, Heurística



- ☰ Etimologia de Meta-Heurística
  - **Origem e etimologia das heurísticas:** O significado de "**heurística**" vem do grego, relacionando-a com a ideia de descoberta. Frase de Arquimedes Eureka! (Anc. Greek: εὕρηκα)
  - **Definição de heurísticas:** Estratégias simples e baseadas em experiência para resolver problemas.
  - **Introdução às metaheurísticas:** Nível superior de abstração, guiando o design de heurísticas.
  - **Hiper-heurísticas:** Nível mais alto, capazes de selecionar e combinar múltiplas metaheurísticas.
  - **Evolução das heurísticas:** A teoria da evolução é utilizada para explicar como as heurísticas se desenvolveram desde os organismos mais simples até os seres humanos.

**Planejamento:** [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Aprofundar entendimento sobre técnicas específicas e suas respectivas referências fundamentais a partir do livro **Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature**.
- Produzir resumo sobre cada algoritmo e técnica.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

---

## ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

[Relatório de Busca de Artigos citado no Termo de Aceite de Entrega de 16 de outubro]

Residência em IA  
Discente: Victor Guerreiro Pimenta

# Relatório de Busca de Artigos Científicos no Google Acadêmico: Aplicações de Metaheurísticas

## Introdução

Este relatório apresenta os resultados da busca de artigos científicos no Google Acadêmico sobre **aplicações de metaheurísticas** em diversas áreas de interesse. O objetivo foi identificar estudos recentes e relevantes, publicados entre **2023 e 2024**, para compreender as **tendências emergentes** e os **avanços mais significativos** no uso dessas técnicas. Este período foi escolhido estrategicamente para garantir que os resultados reflitam as **contribuições científicas mais atuais**, considerando a rápida evolução das pesquisas em metaheurísticas.

## Metodologia de Busca

A busca foi conduzida utilizando a plataforma Google Acadêmico, com foco em publicações entre 2023 e 2024, garantindo a inclusão de trabalhos **recentes e relevantes**. Para ampliar a abrangência e precisão, foram utilizadas **strings de busca baseadas em operadores booleanos** (AND, OR), considerando termos que cobrem tanto áreas consolidadas quanto emergentes no uso de metaheurísticas. As **strings foram elaboradas** para explorar uma ampla diversidade de aplicações, refletindo o potencial interdisciplinar da abordagem.

As consultas incluíram combinações específicas, como **"metaheuristic" OR "metaheuristics"** associado a domínios de **bioinformática, genômica, proteômica, engenharia estrutural, design de materiais, sistemas complexos, logística, redes sociais, telecomunicações e aprendizado de máquina**, entre outros. Essa estratégia garantiu a inclusão de temas como **veículos autônomos, redes inteligentes (smart grids), agricultura de precisão, gestão de resíduos, circuitos eletrônicos e processamento de**

**linguagem natural**, além de tópicos especializados, como **cibernética** e **redes de computadores**. Cada string foi projetada para capturar estudos que aplicam metaheurísticas em contextos específicos, **maximizando a relevância dos resultados retornados**.

Além disso, o uso dessas combinações possibilitou cobrir campos emergentes, como **IoT (Internet das Coisas)** e **grandes modelos de linguagem (LLMs)**, ao mesmo tempo que explorou aplicações tradicionais em **manufatura, aeronáutica, finanças, controle de sistemas e programação de tarefas**. Dessa forma, o processo de busca assegurou uma **cobertura interdisciplinar robusta**, permitindo a identificação de pesquisas que abordam metaheurísticas de forma inovadora e aplicada. **A construção rigorosa das strings**, aliada ao foco em áreas variadas, contribuiu para a identificação de publicações científicas de alta relevância em diferentes contextos de otimização.

## Strings de Busca

("metaheuristic" OR "metaheuristics") AND ("industrial")

("metaheuristic" OR "metaheuristics") AND ("electricity")

("metaheuristic" OR "metaheuristics") AND ("bioinformatics")

("metaheuristic" OR "metaheuristics") AND ("scheduling")

("metaheuristic" OR "metaheuristics") AND ("supply chain")

("metaheuristic" OR "metaheuristics") AND ("machine learning")

("metaheuristic" OR "metaheuristics") AND ("logistics")

("metaheuristic" OR "metaheuristics") AND ("telecommunication")

("metaheuristic" OR "metaheuristics") AND ("routing")

("metaheuristic" OR "metaheuristics") AND ("image")

("metaheuristic" OR "metaheuristics") AND ("robotics")

("metaheuristic" OR "metaheuristics") AND ("control systems")

("metaheuristic" OR "metaheuristics") AND ("trade")

("metaheuristic" OR "metaheuristics") AND ("finance")

("metaheuristic" OR "metaheuristics") AND ("materials design")

("metaheuristic" OR "metaheuristics") AND ("structural design")

("metaheuristic" OR "metaheuristics") AND ("manufacturing")

("metaheuristic" OR "metaheuristics") AND ("engineering")

("metaheuristic" OR "metaheuristics") AND ("automotive" OR  
"autonomous vehicles")

("metaheuristic" OR "metaheuristics") AND ("aerospace")

("metaheuristic" OR "metaheuristics") AND ("chemical")

("metaheuristic" OR "metaheuristics") AND ("environmental" OR  
"sustainability")

("metaheuristic" OR "metaheuristics") AND ("investment" OR  
"portfolio")

("metaheuristic" OR "metaheuristics") AND ("software" OR  
"programming")

("metaheuristic" OR "metaheuristics") AND ("genomics")

("metaheuristic" OR "metaheuristics") AND ("proteomics" OR  
"protein")

("metaheuristic" OR "metaheuristics") AND ("phylogenetics")

("metaheuristic" OR "metaheuristics") AND ("natural language processing" OR "large language models")

("metaheuristic" OR "metaheuristics") AND ("reinforcement learning")

("metaheuristic" OR "metaheuristics") AND ("data mining" OR "big data")

("metaheuristic" OR "metaheuristics") AND ("internet of things" OR "IoT")

("metaheuristic" OR "metaheuristics") AND ("circuits")

("metaheuristic" OR "metaheuristics") AND ("computer networks")

("metaheuristic" OR "metaheuristics") AND ("complex systems")

("metaheuristic" OR "metaheuristics") AND ("social networks")

("metaheuristic" OR "metaheuristics") AND ("waste management")

("metaheuristic" OR "metaheuristics") AND ("cybersecurity")

("metaheuristic" OR "metaheuristics") AND ("agricultural" or "precision agriculture")

("metaheuristic" OR "metaheuristics") AND ("satellite")

("metaheuristic" OR "metaheuristics") AND ("smart grids")

## Amostragem de Artigos

Para cada string de busca, foram selecionados os artigos mais relevantes, com limite de **10** por tema. Com o critério de inclusão da string diretamente no título.

## SUMÁRIO

<b>Industrial.....</b>	<b>6</b>
<b>Electricity.....</b>	<b>7</b>
<b>Bioinformatics.....</b>	<b>7</b>
<b>Scheduling.....</b>	<b>8</b>
<b>Supply Chain.....</b>	<b>8</b>
<b>Machine Learning.....</b>	<b>9</b>
<b>Logistics.....</b>	<b>9</b>
<b>Telecommunication.....</b>	<b>10</b>
<b>Routing.....</b>	<b>11</b>
<b>Image.....</b>	<b>12</b>
<b>Robotics.....</b>	<b>12</b>
<b>Control Systems.....</b>	<b>13</b>
<b>Trade.....</b>	<b>14</b>
<b>Finance.....</b>	<b>14</b>
<b>Materials Design.....</b>	<b>15</b>
<b>Structural Design.....</b>	<b>15</b>
<b>Manufacturing.....</b>	<b>16</b>
<b>Engineering.....</b>	<b>17</b>
<b>Automotive or Autonomous Vehicles.....</b>	<b>18</b>
<b>Aerospace.....</b>	<b>18</b>
<b>Chemical.....</b>	<b>19</b>
<b>Environmental or Sustainability.....</b>	<b>19</b>
<b>Investment or Portfolio.....</b>	<b>20</b>
<b>Software or Programming.....</b>	<b>20</b>
<b>Genomics.....</b>	<b>21</b>
<b>Proteomics.....</b>	<b>21</b>
<b>Phylogenetics.....</b>	<b>22</b>

---

<b>Natural Language Processing or Large Language Models.....</b>	<b>22</b>
<b>Reinforcement Learning.....</b>	<b>23</b>
<b>Data Mining or Big Data.....</b>	<b>24</b>
<b>Internet of Things or IoT.....</b>	<b>25</b>
<b>Circuits.....</b>	<b>25</b>
<b>Computer Networks.....</b>	<b>26</b>
<b>Complex Systems.....</b>	<b>26</b>
<b>Social Networks.....</b>	<b>26</b>
<b>Waste Management.....</b>	<b>27</b>
<b>Cybersecurity.....</b>	<b>28</b>
<b>Agricultural or Precision Agriculture.....</b>	<b>28</b>
<b>Satellite.....</b>	<b>29</b>
<b>Smart Grids.....</b>	<b>29</b>

---

## Industrial

**An adaptive metaheuristic optimization approach for Tennessee Eastman process for an industrial fault tolerant control system**

<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0296471>

**ST-AL: a hybridized search based metaheuristic computational algorithm towards optimization of high dimensional industrial datasets**

<https://link.springer.com/article/10.1007/s00500-022-07115-7>

**A Continuous Multistage Load Shedding Algorithm for Industrial Processes Based on Metaheuristic Optimization**

<https://www.mdpi.com/2227-7390/11/12/2684>

**Advanced Statistical and Meta-Heuristic Based Optimization Fault Diagnosis Techniques in Complex Industrial Processes: A Comparative Analysis**

<https://ieeexplore.ieee.org/abstract/document/10256179>

**Machine learning tuning by diversity oriented firefly metaheuristics for Industry 4.0**

<https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.13293>

**Chapter 11 - Use of metaheuristics in industrial development and their future perspectives**

<https://www.sciencedirect.com/science/article/abs/pii/B9780323917810000119>

**Metaheuristic feature selection with deep learning enabled cascaded recurrent neural network for anomaly detection in Industrial Internet of Things environment**

<https://link.springer.com/article/10.1007/s10586-022-03719-8>

**A metaheuristic algorithm for a locomotive routing problem arising in the steel industry**

---

<https://www.sciencedirect.com/science/article/abs/pii/S0377221722008633>

**Solving large scale industrial production scheduling problems with complex constraints: an overview of the state-of-the-art**

<https://www.sciencedirect.com/science/article/pii/S1877050922023791>

**Metaheuristic-based time series clustering for anomaly detection in manufacturing industry**

<https://link.springer.com/article/10.1007/s10489-023-04594-5>

## **Electricity**

**Comparison of particle swarm optimization and other metaheuristics on electricity demand estimation: A case study of Iran**

<https://www.sciencedirect.com/science/article/abs/pii/S0360544214006379>

**Metaheuristic Optimization Methods Applied to Power Converters: A Review**

<https://ieeexplore.ieee.org/abstract/document/7024140>

**A Review of Metaheuristic Techniques for Optimal Integration of Electrical Units in Distribution Networks**

<https://ieeexplore.ieee.org/abstract/document/9311503>

**Metaheuristics Algorithms in Power Systems**

<https://link.springer.com/book/10.1007/978-3-030-11593-7>

## **Bioinformatics**

**Evolutionary Computation in bioinformatics: A survey**

<https://www.sciencedirect.com/science/article/abs/pii/S0925231224005290>

**On the role of metaheuristic optimization in bioinformatics**

<https://onlinelibrary.wiley.com/doi/full/10.1111/itor.13164>

**Applications of Nature-Inspired Metaheuristic Algorithms for Tackling Optimization Problems Across Disciplines**

<https://arxiv.org/abs/2308.10875>

**Metaheuristic Biclustering Algorithms: From State-of-the-art to Future Opportunities**

<https://dl.acm.org/doi/full/10.1145/3617590>

**Comparative Study of Population-based Metaheuristic Algorithms in Case Study of DNA Sequence Assembly**

[https://biomed.bas.bg/bioautomation/2024/vol\\_28.3/files/28.3\\_02.pdf](https://biomed.bas.bg/bioautomation/2024/vol_28.3/files/28.3_02.pdf)

**Enhancing Multiple Sequence Alignment with Genetic Algorithms: A Bioinformatics Approach in Biomedical Engineering**

<http://rmib.com.mx/index.php/rmib/article/view/1419>

**Three Metaheuristic Approaches for Tumor Phylogeny Inference: An Experimental Comparison**

<https://www.mdpi.com/1999-4893/16/7/333>

**Pelado: A Load Balancing Algorithm for Metaheuristics Optimization Applied to Biomarker Discovery**

[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4813419](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4813419)

**A proactive grey wolf optimization for improving bioinformatic systems with high dimensional data**

<https://link.springer.com/article/10.1007/s41870-024-02030-6>

**Inferring linear-B cell epitopes using 2-step metaheuristic variant-feature selection using genetic algorithm**

<https://www.nature.com/articles/s41598-023-41179-1>

## Scheduling

**Metaheuristics for Scheduling in Industrial and Manufacturing Applications**

<https://link.springer.com/book/10.1007/978-3-540-78985-7>

**Metaheuristics for scheduling of heterogeneous tasks in cloud computing environments: Analysis, performance evaluation, and future directions**

<https://www.sciencedirect.com/science/article/abs/pii/S1569190X21000678>

**A review of metaheuristic scheduling techniques in cloud computing**

<https://www.sciencedirect.com/science/article/pii/S1110866515000353>

**Metaheuristic Scheduling for Cloud: A Survey**

<https://ieeexplore.ieee.org/abstract/document/6516911>

**Metaheuristics for Scheduling in Distributed Computing Environments**

<https://link.springer.com/book/10.1007/978-3-540-69277-5>

**Metaheuristics for Production Scheduling**

<https://www.amazon.com/Metaheuristics-Production-Scheduling-Automation-Engineering/dp/1848214979>

**An Introduction to Multiobjective Metaheuristics for Scheduling and Timetabling**

[https://link.springer.com/chapter/10.1007/978-3-642-17144-4\\_4](https://link.springer.com/chapter/10.1007/978-3-642-17144-4_4)

**A survey of hybrid metaheuristics for the resource-constrained project scheduling problem**

<https://www.sciencedirect.com/science/article/abs/pii/S0377221719300980>

**Heuristics and Metaheuristics for Solving Scheduling Problems**

<https://www.igi-global.com/chapter/heuristics-metaheuristics-solving-scheduling-problems/19350>

## Supply Chain

**Revolutionizing sustainable supply chain management: A review of metaheuristics**

<https://www.sciencedirect.com/science/article/pii/S0952197623010230>

**Metaheuristics in Logistics and Supply Chain Management**

<https://onlinelibrary.wiley.com/doi/full/10.1111/j.0000-0000.2012.01042.x>

**Metaheuristic Algorithms Applied to Bioenergy Supply Chain Problems: Theory, Review, Challenges, and Future**

<https://www.mdpi.com/1996-1073/7/11/7640>

**Heuristic-based metaheuristics to address a sustainable supply chain network design problem**

<https://www.tandfonline.com/doi/abs/10.1080/21681015.2017.1422039>

**Metaheuristic Algorithms for Supply Chain Management Problems**

<https://www.igi-global.com/chapter/content/69883>

**Meta-heuristics for sustainable supply chain management: a review**

<https://www.tandfonline.com/doi/full/10.1080/00207543.2022.2045377>

## **Machine Learning**

**Tuning Metaheuristics**

<https://link.springer.com/book/10.1007/978-3-642-00483-4>

**Machine Learning into Metaheuristics: A Survey and Taxonomy**

<https://dl.acm.org/doi/abs/10.1145/3459664>

**Machine learning enhancing metaheuristics: a systematic review**

<https://link.springer.com/article/10.1007/s00500-023-08886-3>

**Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs**

[https://www.degruyter.com/document/doi/10.1515/math-2017-0029/html?trk=public\\_post\\_comment-text](https://www.degruyter.com/document/doi/10.1515/math-2017-0029/html?trk=public_post_comment-text)

**Metaheuristics in Machine Learning: Theory and Applications**

<https://link.springer.com/book/10.1007/978-3-030-70542-8>

**Combining metaheuristics with mathematical programming, constraint programming and machine learning**

<https://link.springer.com/article/10.1007/s10479-015-2034-y>

**Towards a theory of practice in metaheuristics design: A machine learning perspective**

<https://www.rairo-ita.org/articles/ita/abs/2006/02/ita06025/ita06025.html>

**Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases**

<https://www.sciencedirect.com/science/article/abs/pii/S1476927121001894>

---

## Logistics

### Metaheuristics for Logistics

<https://www.amazon.com.br/Metaheuristics-Logistics-Laurent-Deroussi/dp/1848218087>

### Logistics Management

[https://link.springer.com/chapter/10.1007/0-387-23667-8\\_15](https://link.springer.com/chapter/10.1007/0-387-23667-8_15)

### Meta-heuristics for reverse logistics: A literature review and perspectives

<https://www.sciencedirect.com/science/article/abs/pii/S0360835218305953>

### Metaheuristic Optimization for Logistics in Natural Disasters

[https://link.springer.com/chapter/10.1007/978-3-319-43709-5\\_7](https://link.springer.com/chapter/10.1007/978-3-319-43709-5_7)

### Modelling uncertainty in sustainable-green integrated reverse logistics network using metaheuristics optimization

<https://www.sciencedirect.com/science/article/abs/pii/S0360835221007324>

### Metaheuristic for solving routing problem in logistics management

<https://www.inderscienceonline.com/doi/abs/10.1504/IJOR.2009.026536>

### Optimizing Transport Logistics under Uncertainty with Simheuristics: Concepts, Review and Trends

<https://www.mdpi.com/2305-6290/6/3/42>

### Optimizing a Just-In-Time logistics network problem under fuzzy supply and demand: two parameter-tuned metaheuristics algorithms

<https://link.springer.com/article/10.1007/s00521-017-2920-0>

## Telecommunication

**Metaheuristics in Telecommunication Systems: Network Design, Routing, and Allocation Problems**

<https://ieeexplore.ieee.org/abstract/document/8255613>

**Metaheuristics and Applications to Optimization Problems in Telecommunications**

[https://link.springer.com/chapter/10.1007/978-0-387-30165-5\\_4](https://link.springer.com/chapter/10.1007/978-0-387-30165-5_4)

**Evolutionary team based on different metaheuristics for solving a real-world problem in the telecommunication domain**

<https://www.emerald.com/insight/content/doi/10.1108/ec-05-2013-0143/full/html>

**A metaheuristic and simheuristic approach for the p-Hub median problem from a telecommunication perspective**

[ictp.unb.br/jspui/handle/10482/34925](https://ictp.unb.br/jspui/handle/10482/34925)

**A Survey of Ant Colony Optimization Algorithms for Telecommunication Networks**

<https://www.igi-global.com/article/content/67331>

**Optimal Placement of Antennae in Telecommunications Using Metaheuristics**

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b845ce1be66e3d77d1bb01332f49c8bcc30c626c>

**A comprehensive review of nature inspired routing algorithms for fixed telecommunication networks**

<https://www.sciencedirect.com/science/article/abs/pii/S1383762106000178>

**A Comparative Evaluation of Nature Inspired Algorithms for Telecommunication Network Design**

<https://www.mdpi.com/2076-3417/10/19/6840>

**Multiobjective metaheuristics for frequency assignment problem in mobile networks with large-scale real-world instances**

<https://www.emerald.com/insight/content/doi/10.1108/02644401211206034/full/html>

## Routing

**Integrated clustering and routing protocol for wireless sensor networks using Cuckoo and Harmony Search based metaheuristic techniques**

<https://www.sciencedirect.com/science/article/abs/pii/S0952197617302798>

**Metaheuristics for efficient aircraft scheduling and re-routing at busy terminal control areas**

<https://www.sciencedirect.com/science/article/abs/pii/S0968090X16301504>

**Metaheuristic Algorithms for Multiagent Routing Problems**

<https://link.springer.com/article/10.1134/S0005117921100155>

**QoS Routing enhancement using metaheuristic approach in mobile ad-hoc network**

<https://www.sciencedirect.com/science/article/abs/pii/S138912861630322X>

**Improved Metaheuristics-Based Clustering with Multihop Routing Protocol for Underwater Wireless Sensor Networks**

<https://www.mdpi.com/1424-8220/22/4/1618>

**CMML: Combined metaheuristic-machine learning for adaptable routing in clustered wireless sensor networks**

<https://www.sciencedirect.com/science/article/abs/pii/S1568494622000369>

**A metaheuristic approach to fisheries survey route planning**

<https://onlinelibrary.wiley.com/doi/full/10.1111/itor.12252>

**Routing foreseeable lightpath demands using a tabu search meta-heuristic**

<https://ieeexplore.ieee.org/abstract/document/1189140>

**CFMCRS: Calibration fuzzy- metaheuristic clustering routing scheme simultaneous in on-demand WRSNs for sustainable smart city**

<https://www.sciencedirect.com/science/article/abs/pii/S0957417422016682>

**Combination between metaheuristics and simulation model for a routing problem**

<https://mediatum.ub.tum.de/doc/1254091/document.pdf#page=93>

**Improved Metaheuristic-Driven Energy-Aware Cluster-Based Routing Scheme for IoT-Assisted Wireless Sensor Networks**

<https://www.mdpi.com/2071-1050/14/13/7712>

## Image

**Metaheuristic Algorithms for Image Segmentation: Theory and Applications**

<https://link.springer.com/book/10.1007/978-3-030-12931-6>

**A survey on image segmentation using metaheuristic-based deformable models: state of the art and critical analysis**

<https://www.sciencedirect.com/science/article/abs/pii/S1568494616301053>

**Meta-Heuristic Algorithms in Medical Image Segmentation: A Review**

<https://www.igi-global.com/chapter/meta-heuristic-algorithms-in-medical-image-segmentation/192005>

**Metaheuristics Applied to Blood Image Analysis**

[https://link.springer.com/chapter/10.1007/978-3-030-56689-0\\_6](https://link.springer.com/chapter/10.1007/978-3-030-56689-0_6)

**Grey wolf optimizer and other metaheuristic optimization techniques with image processing as their applications: a review**

<https://iopscience.iop.org/article/10.1088/1757-899X/1136/1/012053/meta>

**Metaheuristics based COVID-19 detection using medical images: A review**

<https://www.sciencedirect.com/science/article/pii/S0010482522001366>

**Hybrid Metaheuristics for Image Analysis**

<https://link.springer.com/book/10.1007/978-3-319-77625-5>

**A Systematic Review of Metaheuristic-based Image Encryption Techniques**

<https://link.springer.com/article/10.1007/s11831-021-09656-w>

**Image data hiding schemes based on metaheuristic optimization: a review**

<https://link.springer.com/article/10.1007/s10462-023-10537-w>

**Applications of Metaheuristics in Hyperspectral Imaging: A Review**

[https://link.springer.com/chapter/10.1007/978-981-15-0751-9\\_93](https://link.springer.com/chapter/10.1007/978-981-15-0751-9_93)

## Robotics

**A review of metaheuristics in robotics**

<https://www.sciencedirect.com/science/article/abs/pii/S0045790615000154>

**A comparative review on mobile robot path planning: Classical or meta-heuristic methods?**

<https://www.sciencedirect.com/science/article/abs/pii/S1367578820300675>

**Recent Advances on Meta-Heuristics and Their Application to Real Scenarios**

<https://www.intechopen.com/books/3135>

**A Comprehensive Review of Coverage Path Planning in Robotics Using Classical and Heuristic Algorithms**

<https://ieeexplore.ieee.org/abstract/document/9523743>

**A Review of Fuzzy Metaheuristics for Optimal Design of Fuzzy Controllers in Mobile Robotics**

[https://link.springer.com/chapter/10.1007/978-3-031-00978-5\\_3](https://link.springer.com/chapter/10.1007/978-3-031-00978-5_3)

**A comparative analysis of metaheuristic algorithms for solving the inverse kinematics of robot manipulators**

<https://www.sciencedirect.com/science/article/pii/S2590123022002675>

**Comparative Study of Different Metaheuristics for the Trajectory Planning of a Robotic Arm**

<https://ieeexplore.ieee.org/abstract/document/6877628>

**A hybrid metaheuristic algorithm to optimise a real-world robotic cell**

<https://www.sciencedirect.com/science/article/abs/pii/S0305054816302337>

**Chaotic metaheuristic algorithms for learning and reproduction of robot motion trajectories**

<https://link.springer.com/article/10.1007/s00521-016-2717-6>

**Metaheuristics for Robotics (Computer Engineering; Optimization Heuristics)**

<https://www.amazon.com/Metaheuristics-Robotics-Engineering-Optimization-Heuristics/dp/1786303809>

**Optimization of a fuzzy controller for autonomous robot navigation using a new competitive multi-metaheuristic model**

<https://link.springer.com/article/10.1007/s00500-021-06036-1>

**Efficient metaheuristics for pick and place robotic systems optimization**

<https://link.springer.com/article/10.1007/s10845-012-0668-z>

**Adaptive Metaheuristic-Based Methods for Autonomous Robot Path Planning:  
Sustainable Agricultural Applications**

<https://www.mdpi.com/2076-3417/12/3/943>

**A comparative study of meta-heuristics for local path planning of a mobile robot**

<https://www.tandfonline.com/doi/abs/10.1080/0305215X.2020.1858074>

**Meta-heuristic approach for solving multi-objective path planning for autonomous  
guided robot using PSO–GWO optimization algorithm with evolutionary programming**

<https://link.springer.com/article/10.1007/s12652-020-02514-w>

**Research Progress of Nature-Inspired Metaheuristic Algorithms in Mobile Robot Path  
Planning**

<https://www.mdpi.com/2079-9292/12/15/3263>

## Control Systems

**Metaheuristic Optimization for the Design of Automatic Control Laws**

<https://www.wiley.com/en-us/Metaheuristic+Optimization+for+the+Design+of+Automatic+Control+Laws-p-9781118796351>

**Advanced Metaheuristics-Based Approach for Fuzzy Control Systems Tuning**

[https://link.springer.com/chapter/10.1007/978-3-319-12883-2\\_22](https://link.springer.com/chapter/10.1007/978-3-319-12883-2_22)

**Metaheuristic-Based Algorithms for Optimizing Fractional-Order Controllers—A  
Recent, Systematic, and Comprehensive Review**

<https://www.mdpi.com/2504-3110/7/7/553>

**Multi-objective meta-heuristic optimization in intelligent control: A survey on the  
controller tuning problem**

<https://www.sciencedirect.com/science/article/abs/pii/S1568494620302829>

**Controller Tuning Optimization Methods for Multi-Constraints and Nonlinear Systems**

<https://link.springer.com/book/10.1007/978-3-030-64541-0>

## Trade

**Applied Metaheuristics in International Trading: A Systematic Review**

[https://link.springer.com/chapter/10.1007/978-3-031-45438-7\\_7](https://link.springer.com/chapter/10.1007/978-3-031-45438-7_7)

**Analysis of a production system of green products considering single-level trade credit financing via a parametric approach of intervals and meta-heuristic algorithms**

<https://link.springer.com/article/10.1007/s10489-023-04493-9>

**Comparative Analysis of Metaheuristics Techniques for Trade Data Harmonization**

<https://www.scitepress.org/Papers/2023/121766/121766.pdf>

## **Finance**

**A Survey on Financial Applications of Metaheuristics**

<https://dl.acm.org/doi/abs/10.1145/3054133>

**Finance-based scheduling using meta-heuristics: discrete versus continuous optimization problems**

<https://www.emerald.com/insight/content/doi/10.1108/jfmpe-07-2014-0013/full/html>

**Meta-Heuristics Optimization Algorithms in Engineering, Business, Economics, and Finance**

<https://www.amazon.com.br/Meta-Heuristics-Optimization-Algorithms-Engineering-Economics/dp/1466620862>

**Metaheuristics application on a financial forecasting problem**

<https://ieeexplore.ieee.org/abstract/document/6557679>

**A new metaheuristic optimization model for financial crisis prediction: Towards sustainable development**

<https://www.sciencedirect.com/science/article/abs/pii/S2210537922001093>

**Adaptive Balance Optimizer: A New Adaptive Metaheuristic and its Application in Solving Optimization Problem in Finance**

<https://www.proquest.com/openview/d7fd995deffc169f7507a3c18871534b/1?pq-origsite=gscholar&cbl=5444811>

**Hybrid neural network-based metaheuristics for prediction of financial markets: a case study on global gold market**

<https://academic.oup.com/jcde/article/10/3/1110/7148134?login=false>

## **Materials Design**

**Metaheuristic-based inverse design of materials – A survey**

<https://www.sciencedirect.com/science/article/pii/S2352847819302084>

**Material Generation Algorithm: A Novel Metaheuristic Algorithm for Optimization of Engineering Problems**

<https://www.mdpi.com/2227-9717/9/5/859>

**The Use of Metaheuristics in Material Science**

<https://www.taylorfrancis.com/chapters/edit/10.1201/9781003452515-12/use-metaheuristics-material-science-mohamed-arezki-mella>

**Advances in Metaheuristic Algorithms for Optimal Design of Structures**

<https://link.springer.com/book/10.1007/978-3-319-46173-1>

**Bat algorithm as a metaheuristic optimization approach in materials and design: optimal design of a new float for different materials**

<https://link.springer.com/article/10.1007/s00521-018-3430-4>

**Mechanical Identification of Materials and Structures with Optical Methods and Metaheuristic Optimization**

<https://www.mdpi.com/1996-1944/12/13/2133>

## Structural Design

Structural design using multi-objective metaheuristics. Comparative study and application to a real-world problem

<https://link.springer.com/article/10.1007/s00158-015-1291-3>

Metaheuristics for Structural Design and Analysis

<https://www.amazon.com/Metaheuristics-Structural-Design-Analysis-Cengiz/dp/1786302349>

Metaheuristic Applications in Structures and Infrastructures

<https://www.amazon.com.br/Metaheuristic-Applications-Structures-Infrastructures-English-ebook/dp/B00BHCXNFY>

The Mosaic of Metaheuristic Algorithms in Structural Optimization

<https://link.springer.com/article/10.1007/s11831-022-09773-0>

Game theory-based metaheuristics for structural design optimization

<https://onlinelibrary.wiley.com/doi/full/10.1111/mice.12661>

Optimum structural design of the lower control arm using meta-heuristic algorithms

<https://link.springer.com/article/10.1007/s40430-024-04998-5>

Exploring Evolutionary Algorithms for Multi-Objective Optimization in Seismic Structural Design

<https://www.mdpi.com/2076-3417/14/21/9951>

## Manufacturing

Bio-inspired-based structures optimization for additive manufacturing using metaheuristic Kriging

<https://www.tandfonline.com/doi/abs/10.1080/15376494.2023.2214915>

Meta-heuristic algorithms for integrating manufacturing and supply chain functions

<https://www.sciencedirect.com/science/article/abs/pii/S0360835224003619>

Metaheuristic-based time series clustering for anomaly detection in manufacturing industry

---

<https://link.springer.com/article/10.1007/s10489-023-04594-5>

**LSTM-based framework with metaheuristic optimizer for manufacturing process monitoring**

<https://www.sciencedirect.com/science/article/pii/S1110016823008918>

**Adaptive iterated stochastic metaheuristic to optimize holes drilling path in manufacturing industry: The Adaptive-Dhouib-Matrix-3 (A-DM3)**

<https://www.sciencedirect.com/science/article/abs/pii/S0952197623000829>

**Two-stage meta-heuristic for part-packing and build-scheduling problem in parallel additive manufacturing**

<https://www.sciencedirect.com/science/article/abs/pii/S1568494623001503>

**A survey on metaheuristics for optimization in food manufacturing industry**

<https://www.sciencedirect.com/science/article/abs/pii/S156849461630182X>

**Meta-heuristics for manufacturing problems**

<https://www.proquest.com/openview/f318213920a0f3ca36cb8fe20fee6442/1?pq-origsite=gscholar&cbl=18750&diss=y>

**A Meta-Heuristic Approach for Dynamic Process Planning in Reconfigurable Manufacturing Systems**

<https://ieeexplore.ieee.org/abstract/document/8327083>

**Metaheuristic Techniques on Cell Formation in Cellular Manufacturing System**

<https://www.joace.org/uploadfile/2013/0506/20130506040509633.pdf>

**A State-of-the-Art Review on Meta-heuristics Application in Remanufacturing**

<https://link.springer.com/article/10.1007/s11831-021-09580-z>

## Engineering

**Flood algorithm (FLA): an efficient inspired meta-heuristic for engineering optimization**

<https://link.springer.com/article/10.1007/s11227-024-06291-7>

**Developments on metaheuristic-based optimization for numerical and engineering optimization problems: Analysis, design, validation, and applications**

<https://www.sciencedirect.com/science/article/pii/S1110016823006221>

**A survey on binary metaheuristic algorithms and their engineering applications**

<https://link.springer.com/article/10.1007/s10462-022-10328-9>

**Evolutionary Algorithms and Metaheuristics: Applications in Engineering Design and Optimization**

<https://jyx.jyu.fi/bitstream/handle/123456789/56831/1/27937621.pdf>

**Advances in Metaheuristics Applications in Engineering Systems**

<https://www.taylorfrancis.com/books/mono/10.1201/9781315297651/advances-metaheuristics-timothy-ganesan-pandian-vasant-irraivan-elamvazuthi>

**Metaheuristics in civil engineering: A review**

[https://d1wqtxts1xzle7.cloudfront.net/81301149/download-libre.pdf?1645635654=&response-content-disposition=inline%3B+filename%3DMetaheuristics\\_in\\_civil\\_engineering\\_A\\_re.pdf&Expires=1734050971&Signature=GzxMKFV5We8SUskGE3G7XvaFOCBcJcFyVgNvO7Lhn-gsFnIczjO5PbcEr5-Ae9M~H1SGRroCLK9RtCxuPNDb55wWVYB3APgnDpQVAcUjSp~JIZkrmTkLU1ZWU9OXmoTYLunIRykU7RidE9WW~edyIjIO0IJM69nIFsz8kVc2GE9BABkZZdfu~4tcDWbRKN6kfuik7bTy5qmjVhe3aLkrVa89eEPM6rAspKIFamp00BwEQTPUg7CpF0ipcDDOS3dcfJ8ArXKebLaEco~xvIAvrBWTxtR8dLDp6XmrUupt1kU7VpH3NL~jIOG6w8LwcZ37qxbey9gbqlfZA8NWAPEOA\\_\\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://d1wqtxts1xzle7.cloudfront.net/81301149/download-libre.pdf?1645635654=&response-content-disposition=inline%3B+filename%3DMetaheuristics_in_civil_engineering_A_re.pdf&Expires=1734050971&Signature=GzxMKFV5We8SUskGE3G7XvaFOCBcJcFyVgNvO7Lhn-gsFnIczjO5PbcEr5-Ae9M~H1SGRroCLK9RtCxuPNDb55wWVYB3APgnDpQVAcUjSp~JIZkrmTkLU1ZWU9OXmoTYLunIRykU7RidE9WW~edyIjIO0IJM69nIFsz8kVc2GE9BABkZZdfu~4tcDWbRKN6kfuik7bTy5qmjVhe3aLkrVa89eEPM6rAspKIFamp00BwEQTPUg7CpF0ipcDDOS3dcfJ8ArXKebLaEco~xvIAvrBWTxtR8dLDp6XmrUupt1kU7VpH3NL~jIOG6w8LwcZ37qxbey9gbqlfZA8NWAPEOA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA)

**Applications of Metaheuristics in Process Engineering**

<https://link.springer.com/book/10.1007/978-3-319-06508-3>

**Optimization in Engineering Sciences: Metaheuristic, Stochastic Methods and Decision Support**

<https://www.amazon.com.br/Optimization-Engineering-Sciences-Metaheuristic-Stochastic-ebook/dp/B00P3W32R8>

**A smart metaheuristic algorithm for solving engineering problems**

<https://link.springer.com/article/10.1007/s00366-020-00951-x>

**Metaheuristics and Optimization in Civil Engineering**

<https://link.springer.com/book/10.1007/978-3-319-26245-1>

**Usage of Metaheuristics in Engineering: A Literature Review**

<https://www.igi-global.com/chapter/content/69895>

**Meta-heuristic and Evolutionary Algorithms for Engineering Optimization**

<https://www.amazon.com/Meta-heuristic-Evolutionary-Algorithms-Engineering-Optimization/dp/1119386993>

**A Walk into Metaheuristics for Engineering Optimization: Principles, Methods and Recent Trends**

<https://www.tandfonline.com/doi/abs/10.1080/18756891.2015.1046324>

**Optimization in Civil Engineering and Metaheuristic Algorithms: A Review of State-of-the-Art Developments**

[https://link.springer.com/chapter/10.1007/978-3-319-96433-1\\_6](https://link.springer.com/chapter/10.1007/978-3-319-96433-1_6)

**Engineering Optimization: An Introduction with Metaheuristic Applications**

<https://www.amazon.com/Engineering-Optimization-Introduction-Metaheuristic-Applications/dp/0470582464>

**Metaheuristics in Water, Geotechnical and Transport Engineering**

<https://www.amazon.com/Metaheuristics-Geotechnical-Transport-Engineering-Elsevier-ebook/dp/B00AMZZR1K>

**Genetic Engineering Algorithm (GEA): An Efficient Metaheuristic Algorithm for Solving Combinatorial Optimization Problems**

<https://link.springer.com/article/10.1134/S000511792403007X>

**Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems**

<https://www.frontiersin.org/journals/mechanical-engineering/articles/10.3389/fmech.2022.1126450/full>

**Optical microscope algorithm: A new metaheuristic inspired by microscope magnification for solving engineering optimization problems**

<https://www.sciencedirect.com/science/article/abs/pii/S0950705123006895>

---

## Automotive or Autonomous Vehicles

Speed Control Optimization for Autonomous Vehicles with Metaheuristics

<https://www.mdpi.com/2079-9292/9/4/551>

A systematic review on metaheuristic approaches for autonomous path planning of unmanned aerial vehicles

<https://cdnsiencepub.com/doi/full/10.1139/dsa-2023-0093>

Optimizing Path Planning for Smart Vehicles: A Comprehensive Review of Metaheuristic Algorithms

[https://library.acadlore.com/JEMSE/2023/2/4/JEMSE\\_02.04\\_05.pdf](https://library.acadlore.com/JEMSE/2023/2/4/JEMSE_02.04_05.pdf)

## Aerospace

A Systematic Study on Solving Aerospace Problems Using Metaheuristics

<https://arxiv.org/abs/2411.02574>

Optimal design of aerospace structures using recent meta-heuristic algorithms

<https://www.degruyter.com/document/doi/10.1515/mt-2021-0024/html>

Meta-heuristic global optimization algorithms for aircraft engines modelling and controller design; A review, research challenges, and exploring the future

<https://www.sciencedirect.com/science/article/abs/pii/S0376042118301416>

Multiobjective Evolutionary Algorithms in Aeronautical and Aerospace Engineering

<https://ieeexplore.ieee.org/abstract/document/6151094>

Prognostics of aerospace electromechanical actuators: comparison between model-based metaheuristic methods

<https://iopscience.iop.org/article/10.1088/1742-6596/2526/1/012073/meta>

## Chemical

### **Chemical-Reaction-Inspired Metaheuristic for Optimization**

<https://ieeexplore.ieee.org/abstract/document/5353674>

### **A novel chemistry based metaheuristic optimization method for mining of classification rules**

<https://www.sciencedirect.com/science/article/abs/pii/S0957417412005891>

### **Applications of metaheuristic optimization algorithms in model predictive control for chemical engineering processes: A systematic review**

<https://www.sciencedirect.com/science/article/abs/pii/S1367578824000415>

### **Temperature Control of a Chemical Reactor Based on Neuro-Fuzzy Tuned with a Metaheuristic Technique to Improve Biodiesel Production**

<https://www.mdpi.com/1996-1073/16/17/6187>

### **Optimizing Chemical Reaction Mechanisms: Evaluating Parameter-Free Metaheuristic Algorithms and Gradient-Based Optimization**

<https://www.tandfonline.com/doi/abs/10.1080/00102202.2024.2329303>

## **Environmental or Sustainability**

### **Revolutionizing sustainable supply chain management: A review of metaheuristics**

<https://www.sciencedirect.com/science/article/pii/S0952197623010230>

### **Meta-heuristics for sustainable supply chain management: a review**

<https://www.tandfonline.com/doi/full/10.1080/00207543.2022.2045377>

### **The Explainable Potential of Coupling Metaheuristics-Optimized-XGBoost and SHAP in Revealing VOCs' Environmental Fate**

<https://www.mdpi.com/2073-4433/14/1/109>

### **A metaheuristic Multi-Objective optimization of energy and environmental performances of a Waste-to-Energy system based on waste gasification using particle swarm optimization**

<https://www.sciencedirect.com/science/article/abs/pii/S0196890424007854>

---

## Investment or Portfolio

**Metaheuristics for Financial Investment Strategies: Applications Survey**

<https://ieeexplore.ieee.org/abstract/document/10253170>

**Meta-heuristics for portfolio optimization**

<https://link.springer.com/article/10.1007/s00500-023-08177-x>

**Meta–heuristics for Portfolio Optimization: Part I — Review of Meta–heuristics**

[https://link.springer.com/chapter/10.1007/978-3-031-36625-3\\_35](https://link.springer.com/chapter/10.1007/978-3-031-36625-3_35)

**Meta-heuristics for Portfolio Optimization: Part II—Empirical Analysis**

[https://link.springer.com/chapter/10.1007/978-3-031-36625-3\\_36](https://link.springer.com/chapter/10.1007/978-3-031-36625-3_36)

**Quantum-inspired meta-heuristic approaches for a constrained portfolio optimization problem**

<https://link.springer.com/article/10.1007/s12065-024-00929-4>

**Multi-verse metaheuristic and deep learning approach for portfolio selection with higher moments**

<https://link.springer.com/article/10.1007/s13198-023-02218-2>

## Software or Programming

**Nature-inspired metaheuristic methods in software testing**

<https://link.springer.com/article/10.1007/s00500-023-08382-8>

**A Systematic Literature Review on Robust Swarm Intelligence Algorithms in Search-Based Software Engineering**

<https://onlinelibrary.wiley.com/doi/full/10.1155/2023/4577581>

**Metaheuristics and Software Engineering: Past, Present, and Future**

<https://www.worldscientific.com/doi/abs/10.1142/S0218194021500443>

**Using meta-heuristics and machine learning for software optimization of parallel computing systems: a systematic literature review**

<https://link.springer.com/article/10.1007/s00607-018-0614-9>

**An efficient approach for metaheuristic-based optimization of composite laminates using genetic programming**

<https://link.springer.com/article/10.1007/s12008-022-01175-7>

## Genomics

**A Hybrid Metaheuristics based technique for Mutation Based Disease Classification**

<https://hrcak.srce.hr/clanak/441672>

**A Review on Machine-Learning and Nature-Inspired Algorithms for Genome Assembly**

<https://www.proquest.com/openview/6215038c43e312cd2142749bb8f942a0/1?pq-origsite=gscholar&cbl=5444811>

**Optimization of MLVA loci combination using metaheuristic methods**

<https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs231367>

**Pelado: A Load Balancing Algorithm for Metaheuristics Optimization Applied to Biomarker Discovery**

[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4813419](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4813419)

**A Survey of Metaheuristics Approaches for Application in Genomic data**

[https://www.researchgate.net/profile/Manu-Phogat/publication/319959714\\_A\\_survey\\_of\\_Meta-heuristics\\_Approaches\\_for\\_application\\_in\\_Genomic\\_data/links/5ba11439a6fdc3cb61d7c0/A-survey-of-Meta-heuristics-Approaches-for-application-in-Genomic-data.pdf](https://www.researchgate.net/profile/Manu-Phogat/publication/319959714_A_survey_of_Meta-heuristics_Approaches_for_application_in_Genomic_data/links/5ba11439a6fdc3cb61d7c0/A-survey-of-Meta-heuristics-Approaches-for-application-in-Genomic-data.pdf)

**Chapter 18 - Genome sequence assembly using metaheuristics**

<https://www.sciencedirect.com/science/article/abs/pii/B9780323917810000181>

## Proteomics

**A novel multi-objective metaheuristic algorithm for protein-peptide docking and benchmarking on the LEADS-PEP dataset**

<https://www.sciencedirect.com/science/article/abs/pii/S0010482521006909>

**Self Generating Metaheuristics in Bioinformatics: The Proteins Structure Comparison Case**

<https://link.springer.com/article/10.1023/B:GENP.0000023687.41210.d7>

**Cooperative Metaheuristics for Exploring Proteomic Data**

<https://link.springer.com/article/10.1023/A:1026080413328>

**Metaheuristics for Protein Structure Prediction: Review and Empirical Analysis**

<https://ieeexplore.ieee.org/abstract/document/10512888>

**Artificial Bee Colony Algorithm in Multi-omics Analysis: A Case Study**

[https://link.springer.com/chapter/10.1007/978-981-97-7344-2\\_11](https://link.springer.com/chapter/10.1007/978-981-97-7344-2_11)

**An Evolutionary Algorithm With Heuristic Operator for Detecting Protein Complexes in Protein Interaction Networks With Negative Controls**

<https://ieeexplore.ieee.org/abstract/document/10440281>

## Phylogenetics

**Three Metaheuristic Approaches for Tumor Phylogeny Inference: An Experimental Comparison**

<https://www.mdpi.com/1999-4893/16/7/333>

**An evolutionary algorithm based on parsimony for the multiobjective phylogenetic network inference problem**

<https://www.sciencedirect.com/science/article/abs/pii/S1568494623002880>

**MO-Phylogenetics: a phylogenetic inference software tool with multi-objective evolutionary metaheuristics**

<https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/2041-210X.12529>

**Exploiting multi-level parallel metaheuristics and heterogeneous computing to boost phylogenetics**

<https://www.sciencedirect.com/science/article/pii/S0167739X21003575>

**Applying a multiobjective metaheuristic inspired by honey bees to phylogenetic inference**

<https://www.sciencedirect.com/science/article/abs/pii/S0303264713001615>

**MetaPIGA v2.0: maximum likelihood large phylogeny estimation using the metapopulation genetic algorithm and other stochastic heuristics**

<https://link.springer.com/article/10.1186/1471-2105-11-379>

## **Natural Language Processing or Large Language Models**

**Exploring Metaheuristic Optimized Machine Learning for Software Defect Detection on Natural Language and Classical Datasets** <https://www.mdpi.com/2227-7390/12/18/2918>

**Leveraging Metaheuristic Optimization to Enhance Insider Threat Detection Through Email Content Natural Language Processing**

[https://link.springer.com/chapter/10.1007/978-3-031-67195-1\\_63](https://link.springer.com/chapter/10.1007/978-3-031-67195-1_63)

**A Review of Metaheuristic Optimization Techniques in Text Classification**

<https://www.ijcesen.com/index.php/ijcesen/article/view/295>

**Leveraging Large Language Models for the Generation of Novel Metaheuristic Optimization Algorithms**

<https://dl.acm.org/doi/abs/10.1145/3583133.3596401>

**Intelligent Machine Learning with Metaheuristics Based Sentiment Analysis and Classification**

[https://cdn.techscience.cn/ueditor/files/csse/TSP\\_CSSE-44-1/TSP\\_CSSE\\_24399/TSP\\_CSSE\\_24399.pdf](https://cdn.techscience.cn/ueditor/files/csse/TSP_CSSE-44-1/TSP_CSSE_24399/TSP_CSSE_24399.pdf)

**Synergies Between Natural Language Processing and Swarm Intelligence Optimization: A Comprehensive Overview**

[https://link.springer.com/chapter/10.1007/978-981-99-9718-3\\_6](https://link.springer.com/chapter/10.1007/978-981-99-9718-3_6)

---

**Metaheuristics and Large Language Models Join Forces: Towards an Integrated Optimization Approach**

<https://arxiv.org/abs/2405.18272>

**Leveraging large language model to generate a novel metaheuristic algorithm with CRISPE framework**

<https://link.springer.com/article/10.1007/s10586-024-04654-6>

## **Reinforcement Learning**

**A reinforcement learning-based metaheuristic algorithm for solving global optimization problems**

<https://www.sciencedirect.com/science/article/abs/pii/S0965997823000030>

**CRLM: A cooperative model based on reinforcement learning and metaheuristic algorithms of routing protocols in wireless sensor networks☆**

<https://www.sciencedirect.com/science/article/abs/pii/S1389128623004644>

**An inverse reinforcement learning framework with the Q-learning mechanism for the metaheuristic algorithm**

<https://www.sciencedirect.com/science/article/abs/pii/S0950705123001181>

**Performance comparison of reinforcement learning and metaheuristics for factory layout planning**

<https://www.sciencedirect.com/science/article/pii/S1755581723000718>

**Automated Design of Metaheuristics Using Reinforcement Learning Within a Novel General Search Framework**

<https://ieeexplore.ieee.org/abstract/document/9852781>

**Q-LEARNING, POLICY ITERATION AND ACTOR-CRITIC REINFORCEMENT LEARNING COMBINED WITH METAHEURISTIC ALGORITHMS IN SERVO SYSTEM CONTROL**

<https://casopisi.junis.ni.ac.rs/index.php/FUMechEng/article/view/12156>

**A Recent Publications Survey on Reinforcement Learning for Selecting Parameters of Meta-Heuristic and Machine Learning Algorithms**

<https://www.scitepress.org/Papers/2023/119543/119543.pdf>

**A review of reinforcement learning based hyper-heuristics**

<https://peerj.com/articles/cs-2141/>

**A Memetic Algorithm With Reinforcement Learning for Sociotechnical Production Scheduling**

<https://ieeexplore.ieee.org/abstract/document/10173523>

**Dynamic maintenance scheduling approach under uncertainty: Comparison between reinforcement learning, genetic algorithm simheuristic, dispatching rules Author links open overlay panel**

<https://www.sciencedirect.com/science/article/pii/S0957417424002690>

**GRAPH Reinforcement Learning for Operator Selection in the ALNS Metaheuristic**

[https://link.springer.com/chapter/10.1007/978-3-031-34020-8\\_15](https://link.springer.com/chapter/10.1007/978-3-031-34020-8_15)

**Combining hybrid metaheuristic algorithms and reinforcement learning to improve the optimal control of nonlinear continuous-time systems with input constraints**

<https://www.sciencedirect.com/science/article/abs/pii/S0045790624001071>

## **Data Mining or Big Data**

**Discovering anomalies in big data: a review focused on the application of metaheuristics and machine learning techniques**

<https://www.frontiersin.org/journals/big-data/articles/10.3389/fdata.2023.1179625/full>

**Metaheuristic Based Clustering with Deep Learning Model for Big Data Classification**

[https://www.researchgate.net/profile/Vijayalakshmi-Kumaravel-2/publication/363172030\\_Metaheuristic\\_Based\\_Clustering\\_with\\_Deep\\_Learning\\_Model\\_for\\_Big\\_Data\\_Classification/links/63102ccd1ddd44702125ea1f/Metaheuristic-Based-Clustering-with-Deep-Learning-Model-for-Big-Data-Classification.pdf](https://www.researchgate.net/profile/Vijayalakshmi-Kumaravel-2/publication/363172030_Metaheuristic_Based_Clustering_with_Deep_Learning_Model_for_Big_Data_Classification/links/63102ccd1ddd44702125ea1f/Metaheuristic-Based-Clustering-with-Deep-Learning-Model-for-Big-Data-Classification.pdf)

**Metaheuristic assisted hybrid deep classifiers for intrusion detection: a bigdata perspective**

<https://link.springer.com/article/10.1007/s11276-024-03815-0>

**A MapReduce-based big data clustering using swarm-inspired meta-heuristic algorithms**

[https://scientiairanica.sharif.edu/article\\_23492.html](https://scientiairanica.sharif.edu/article_23492.html)

**Intelligent biomedical image classification in a big data architecture using metaheuristic optimization and gradient approximation**

<https://link.springer.com/article/10.1007/s11276-023-03573-5>

**Cost-efficient resource scheduling in cloud for big data processing using metaheuristic search black widow optimization (MS-BWO) algorithm**

<https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs222048>

**Data Mining Problems Optimization by using Metaheuristic Algorithms: A Survey**

<https://sciencesforce.com/index.php/mawa/article/view/301>

**Population based metaheuristics in Spark: Towards a general framework using PSO as a case study**

<https://www.sciencedirect.com/science/article/pii/S2210650224000166>

**Preserving Privacy in Association Rule Mining Using Metaheuristic-Based Algorithms: A Systematic Literature Review**

<https://ieeexplore.ieee.org/abstract/document/10422999>

## **Internet of Things or IoT**

**Intrusion Detection in Healthcare 4.0 Internet of Things Systems via Metaheuristics Optimized Machine Learning**

<https://www.mdpi.com/2071-1050/15/16/12563>

**Hybrid Metaheuristics With Machine Learning Based Botnet Detection in Cloud Assisted Internet of Things Environment**

<https://ieeexplore.ieee.org/abstract/document/10272611>

**Internet of Things Enabled Energy Aware Metaheuristic Clustering for Real Time Disaster Management**

[https://cdn.techscience.cn/ueditor/files/csse/TSP\\_CSSE-45-2/TSP\\_CSSE\\_29463/TSP\\_CSSE\\_29463.pdf](https://cdn.techscience.cn/ueditor/files/csse/TSP_CSSE-45-2/TSP_CSSE_29463/TSP_CSSE_29463.pdf)

**Deep learning and metaheuristics application in internet of things: : A literature review**

<https://dl.acm.org/doi/abs/10.1016/j.micpro.2023.104792>

**Metaheuristics with federated learning enabled intrusion detection system in Internet of Things environment**

<https://onlinelibrary.wiley.com/doi/full/10.1111/exsy.13138>

**High-Impact Applications of IoT System-Based Metaheuristics**

[https://link.springer.com/chapter/10.1007/978-3-031-45952-8\\_6](https://link.springer.com/chapter/10.1007/978-3-031-45952-8_6)

**Design of Metaheuristic Optimization Algorithms for Deep Learning Model for Secure IoT Environment**

<https://www.mdpi.com/2071-1050/15/3/2204>

**Meta-heuristic Algorithms in IoT-Based Application: A Systematic Review**

[https://link.springer.com/chapter/10.1007/978-3-031-47448-4\\_9](https://link.springer.com/chapter/10.1007/978-3-031-47448-4_9)

## Circuits

**Mathematical Circuit Root Simplification Using an Ensemble Heuristic–Metaheuristic Algorithm**

<https://www.mdpi.com/2227-7390/11/6/1498>

**Integrated Circuit Design of Fractional-Order Chaotic Systems Optimized by Metaheuristics**

<https://www.mdpi.com/2079-9292/12/2/413>

**DSSA: Direct Simplified Symbolic analysis using metaheuristic-driven circuit modelling**

<https://www.aims sciences.org/article/doi/10.3934/jdg.2023023?viewType=HTML>

**Enhancing PID Controller Performance in RLC Circuits through Metaheuristic Optimization**

<repositorio.ugto.mx/handle/20.500.12059/10443>

**Fault Diagnosis in Analog Circuits Using Swarm Intelligence**

<https://www.mdpi.com/2313-7673/8/5/388>

## Computer Networks

**Multi-Objective Optimization in Computer Networks Using Metaheuristics**

<https://www.taylorfrancis.com/books/mono/10.1201/9781420013627/multi-objective-optimization-computer-networks-using-metaheuristics-yezid-donoso-ramon-fabregat>

**Leveraging Metaheuristics for Feature Selection With Machine Learning Classification for Malicious Packet Detection in Computer Networks**

<https://ieeexplore.ieee.org/abstract/document/10419342>

## Complex Systems

**Analysis of Reliability and Cost of Complex Systems with Metaheuristic Algorithms**

<https://ees-journal.com/index.php/journal/article/view/239>

**Meta-Heuristic and Machine Learning Modelling and Optimizing Complex Systems**

<https://www.igi-global.com/chapter/meta-heuristic-and-machine-learning-modelling-and-optimizing-complex-systems/350928>

**Metaheuristic Techniques for Fine-Tuning Parameter of Complex Systems**

<https://ieeexplore.ieee.org/abstract/document/10753120>

**Metaheuristics in Optimization of Complex Processes**

<https://repozytorium.uw.edu.pl/server/api/core/bitstreams/65d301eb-be17-4844-9d8f-07f66997a2f7/content>

**Metaheuristic and Machine Learning Optimization Strategies for Complex Systems**

<https://www.amazon.com/Metaheuristic-Machine-Learning-Optimization-Strategies/dp/B0D9VTQSYF>

**Metaheuristics: Progress in Complex Systems Optimization**

<https://www.amazon.com/Metaheuristics-Progress-Optimization-Operations-Interfaces/dp/0387719199>

## Social Networks

**A novel meta-heuristic approach for influence maximization in social networks**

<https://onlinelibrary.wiley.com/doi/full/10.1111/exsy.12676>

**Design and Implementation of Metaheuristic Algorithms for Social Network Influence Problems**

[https://grafo.etsii.urjc.es/phd/PhD\\_Thesis\\_Isaac.pdf](https://grafo.etsii.urjc.es/phd/PhD_Thesis_Isaac.pdf)

**A Dynamic Metaheuristic Algorithm for Influence Maximization in Social Networks**

[https://ecdj.ihu.ac.ir/article\\_208174\\_en.html?lang=en](https://ecdj.ihu.ac.ir/article_208174_en.html?lang=en)

**Security and privacy concerns in social networks mathematically modified metaheuristic-based approach**

<https://tarupublication.s3.ap-south-1.amazonaws.com/articles/jdm-sc-1892.pdf>

**The Use of Metaheuristics in the Evolution of Collaborative Filtering Recommender Systems: A Review**

[https://link.springer.com/chapter/10.1007/978-3-031-62922-8\\_16](https://link.springer.com/chapter/10.1007/978-3-031-62922-8_16)

**Brain Drain Optimization (BRADO) Algorithm to Solve Multi-Objective Expert Team Formation Problem in Social Networks**

<https://www.researchsquare.com/article/rs-4265402/v1>

**Maximizing the Impact on Social Networks using the Combination of PSO and GA Algorithms**

[https://www.researchgate.net/profile/Babak-Nouri-Moghaddam/publication/368438082\\_Maximizing\\_the\\_Impact\\_on\\_Social\\_Networks\\_using\\_the\\_Combination\\_of\\_PSO\\_and\\_GA\\_Algorithms/links/63e792b0dea6121757a029d5/Maximizing-the-Impact-on-Social-Networks-using-the-Combination-of-PSO-and-GA-Algorithms.pdf](https://www.researchgate.net/profile/Babak-Nouri-Moghaddam/publication/368438082_Maximizing_the_Impact_on_Social_Networks_using_the_Combination_of_PSO_and_GA_Algorithms/links/63e792b0dea6121757a029d5/Maximizing-the-Impact-on-Social-Networks-using-the-Combination-of-PSO-and-GA-Algorithms.pdf)

**A metaheuristic optimisation-based deep learning model for fake news detection in online social networks**

<https://www.inderscienceonline.com/doi/abs/10.1504/IJESDF.2024.140741?journalCode=ijesdf>

## **Waste Management**

**A significant exploration on meta-heuristic based approaches for optimization in the waste management route problems**

<https://www.nature.com/articles/s41598-024-64133-1>

**Metaheuristic optimization of waste management in circular economy**  
[thesis.essa-tlemcen.dz/handle/STDB\\_UNAM/553](https://thesis.essa-tlemcen.dz/handle/STDB_UNAM/553)

**Integrated optimization of marine oil spill response and liquid oily waste management using mathematical programming and evolutionary metaheuristic techniques**  
<https://www.sciencedirect.com/science/article/abs/pii/S0304389423021222>

**A metaheuristic Multi-Objective optimization of energy and environmental performances of a Waste-to-Energy system based on waste gasification using particle swarm optimization**  
<https://www.sciencedirect.com/science/article/abs/pii/S0196890424007854>

**Optimization Techniques in Municipal Solid Waste Management: A Systematic Review**  
<https://www.mdpi.com/2071-1050/16/15/6585>

## Cybersecurity

**Metaheuristics with Deep Learning Model for Cybersecurity and Android Malware Detection and Classification**  
<https://www.mdpi.com/2076-3417/13/4/2172>

**A systematic literature review on software security testing using metaheuristics**  
<https://link.springer.com/article/10.1007/s10515-024-00433-0>

**Bio-inspired algorithms for cybersecurity - a review of the state-of-the-art and challenges**  
<https://www.inderscienceonline.com/doi/abs/10.1504/IJBIC.2024.136199>

**A Survey on Bio-inspired Algorithms of Cybersecurity**  
[journal.alsalam.edu.iq/index.php/ajest/article/view/103](http://journal.alsalam.edu.iq/index.php/ajest/article/view/103)

**Metaheuristic Optimization for Secure Network Architecture Design**  
[ijstindex.com/index.php/ijst/article/view/48](http://ijstindex.com/index.php/ijst/article/view/48)

## Agricultural or Precision Agriculture

**Intelligent and trusted metaheuristic optimization model for reliable agricultural network**

<https://www.sciencedirect.com/science/article/pii/S0920548923000491>

**Game Theory-Based Multi-Hop Routing Protocol with Metaheuristic Optimization-Based Clustering Process in WSN for Precision Agriculture**

[https://openurl.ebsco.com/EPDB%3Agcd%3A8%3A27512168/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A178180050&crl=c&link\\_origin=scholar.google.com.br](https://openurl.ebsco.com/EPDB%3Agcd%3A8%3A27512168/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A178180050&crl=c&link_origin=scholar.google.com.br)

**Metaheuristic Algorithms in Smart Farming: An Analytical Survey**

<https://www.tandfonline.com/doi/abs/10.1080/02564602.2023.2219226>

**A Self-driven dual reinforcement model with meta heuristic framework to conquer the IoT based clustering to enhance agriculture production**

<https://scientifictemper.com/index.php/tst/article/view/1230>

**Application of Bio and Nature-Inspired Algorithms in Agricultural Engineering**

<https://link.springer.com/article/10.1007/s11831-022-09857-x>

**Review of Genetic Algorithms Applied to Agriculture**

<https://ieeexplore.ieee.org/abstract/document/10698503>

**Bio and Nature-Based Algorithmic Applications in Agricultural Engineering**

<https://nlot.iqpublication.com/documents/nlot-3.pdf>

## Satellite

**A Heterogeneous Multi-Satellite Dynamic Mission Planning Method Based on Metaheuristic Algorithms**

<https://www.mdpi.com/2079-9292/13/22/4353>

**A Review on Satellite Image Segmentation Using Metaheuristic Optimization Techniques**

[https://link.springer.com/chapter/10.1007/978-981-97-1841-2\\_4](https://link.springer.com/chapter/10.1007/978-981-97-1841-2_4)

**Multi-satellite cooperative scheduling method for large-scale tasks based on hybrid graph neural network and metaheuristic algorithm**

<https://www.sciencedirect.com/science/article/abs/pii/S1474034624000107>

**Energy-Efficient Satellite Range Scheduling Using a Reinforcement Learning-Based Memetic Algorithm**

<https://ieeexplore.ieee.org/abstract/document/10463527>

**A Multi-Objective Perspective to Satellite Design and Reliability Optimization**

<https://www.sciencedirect.com/science/article/abs/pii/S0957417424000435>

**Learning Adaptive Genetic Algorithm for Earth Electromagnetic Satellite Scheduling**

<https://ieeexplore.ieee.org/abstract/document/10241983>

**Trends, Advancements and Challenges in Intelligent Optimization in Satellite Communication**

<https://arxiv.org/abs/2410.03674>

## Smart Grids

**Review on optimization techniques used for smart grid**

<https://www.sciencedirect.com/science/article/pii/S2665917423002544>

**Role of Metaheuristics in Optimizing Microgrids Operating and Management Issues: A Comprehensive Review**

<https://www.mdpi.com/2071-1050/15/6/4982>

**AI-enabled metaheuristic optimization for predictive management of renewable energy production in smart grids**

<https://www.sciencedirect.com/science/article/pii/S2352484723011459>

**Optimization Algorithms in Smart Grids: A Systematic Literature Review**

<https://arxiv.org/abs/2301.07512>

**Meta-heuristic Techniques in Microgrid Management: A Survey**

<https://www.sciencedirect.com/science/article/pii/S2210650223000305>

**A Comparative Analysis of Meta-heuristic Algorithms for Energy Management in Smart Grids**

<https://ieeexplore.ieee.org/abstract/document/10708179>

[Relação Problemas/Métodos v0.1 citado no Termo de Aceite de Entrega de 16 de outubro]

## 1. Telecomunicações

- **Design de Redes e Qualidade de Serviço (QoS):**  
Metaheurísticas, como **Algoritmos Genéticos (GA)** e **Otimização por Enxame de Partículas (PSO)**, são aplicadas para projetar redes que equilibram **custo**,

**eficiência** e **QoS**. Por exemplo, o uso de metaheurísticas no planejamento de redes permite **reduzir a latência** enquanto **maximiza a largura de banda**.

- **Roteamento e Redes de Sensores:**

A **Otimização baseada em Colônias de Formigas (ACO)** e técnicas híbridas são frequentemente empregadas para determinar **rotas eficientes** em **redes de sensores**, garantindo **menor consumo de energia** e **confiabilidade** na transmissão.

- **Frequência e Cobertura:**

Problemas de **alocação de frequência** e **cobertura** em redes móveis utilizam metaheurísticas como **Simulated Annealing (SA)** para otimizar a **distribuição de antenas**, melhorando o **desempenho** de redes celulares.

---

## 2. Transporte e Logística

- **Tráfego Aéreo e Veículos:**

Metaheurísticas ajudam no **planejamento de rotas** de aeronaves e veículos autônomos. Algoritmos como o **PSO** permitem a **otimização de trajetórias**, **reduzindo custos operacionais** e **tempo de viagem**.

- **Logística Reversa e Sustentável:**

Otimizações robustas com metaheurísticas, como o **Algoritmo de Cuckoo Search**, são usadas para **planejar a coleta** e **reciclagem** de produtos em **cadeias logísticas reversas**.

- **Resiliência em Desastres:**

Técnicas como **Algoritmos Meméticos** auxiliam na **reorganização de redes logísticas** em **cenários emergenciais**, assegurando **respostas rápidas** e **eficientes**.

---

## 3. Planejamento e Programação

- **Produção Industrial:**

Para o **agendamento de fábricas**, metaheurísticas como o **Algoritmo de Colônia de Abelhas** oferecem **soluções eficientes**, otimizando a **sequência de tarefas** em sistemas com **múltiplas restrições**.

- **Computação em Nuvem:**

O **balanceamento de carga** em ambientes de nuvem é otimizado com técnicas

como a **Otimização por Enxame de Partículas**, garantindo uma **melhor alocação de recursos**.

- **Timetabling:**

Soluções **multiobjetivo** baseadas em **algoritmos evolutivos** são empregadas para **alocar recursos** como salas de aula e **horários de professores**, maximizando a **eficiência**.

---

## 4. Finanças e Economia

- **Portfólios e Investimentos:**

Modelos baseados em **Algoritmos Evolutivos** otimizam a **composição de portfólios**, equilibrando **retorno** e **risco** com dados de mercado **dinâmicos**.

- **Previsão Financeira:**

Técnicas híbridas que combinam **redes neurais** e metaheurísticas são aplicadas para **prever movimentos de mercado** e identificar **padrões financeiros ocultos**.

- **Crises Financeiras:**

Algoritmos **adaptativos**, como o **Algoritmo de Harmonia**, simulam cenários de crise para **prever impactos econômicos** e propor **estratégias de mitigação**.

---

## 5. Bioinformática, Genômica e Proteômica

- **Análise de Dados Biológicos:**

Metaheurísticas como o **Algoritmo de Busca de Cuckoo** auxiliam no **alinhamento de sequências genômicas**, otimizando o tempo de análise e precisão.

- **Descoberta de Biomarcadores:**

Técnicas baseadas em **Enxame de Partículas** ajudam a **identificar biomarcadores significativos** em grandes conjuntos de dados **genômicos** e **proteômicos**.

- **Proteômica e Docking:**

**Algoritmos genéticos** são usados para **prever interações moleculares**, apoiando o **design de medicamentos** mais eficazes.

---

## 6. Engenharia e Sustentabilidade

- **Otimização Estrutural:**  
Metaheurísticas como **Simulated Annealing** são empregadas no **design de estruturas resistentes a terremotos**, garantindo **segurança e eficiência de materiais**.
- **Agricultura de Precisão:**  
Modelos **bio-inspirados**, como **Algoritmos Genéticos**, são usados para **otimizar o uso de água e nutrientes** em cultivos agrícolas, promovendo **sustentabilidade**.
- **Gestão de Resíduos e Energia:**  
Algoritmos como **PSO** otimizam sistemas de **economia circular**, promovendo maior **eficiência no gerenciamento de energia e resíduos**.

---

## 7. Manufatura e Processos Industriais

- **Manufatura Aditiva:**  
A **otimização de processos aditivos** utiliza **Kriging** e **metaheurísticas híbridas** para **melhorar a qualidade de fabricação e minimizar o desperdício**.
- **Integração de Cadeias de Suprimentos:**  
Algoritmos como o **Harmony Search** otimizam o **fluxo de materiais**, **reduzindo custos logísticos e melhorando prazos**.
- **Reconfiguração Industrial:**  
**Planejamentos dinâmicos** com **Algoritmos Meméticos** adaptam fábricas para **mudanças rápidas na demanda**.

---

## 8. Inteligência Artificial e Aprendizado de Máquina

- **Tuning de Hiperparâmetros:**  
**Metaheurísticas** como **PSO** ajustam **parâmetros de redes neurais profundas**, aumentando a **precisão e eficiência** do aprendizado.
- **Redução de Dimensionalidade:**  
Técnicas de **otimização** são aplicadas para **selecionar características relevantes** em grandes volumes de dados, **reduzindo complexidade**.
- **Modelos de Linguagem Natural:**  
**Metaheurísticas** ajudam a desenvolver **algoritmos inovadores** para modelos como **BERT e GPT**, **otimizando a geração e compreensão de textos**.

---

## 9. Veículos Autônomos e Aeroespacial

- **Controle de Rotas:**  
Para **veículos autônomos** e **UAVs**, algoritmos como **ACO** otimizam trajetórias em tempo real, considerando **obstáculos** e **eficiência energética**.
- **Design Aeroespacial:**  
Técnicas **metaheurísticas** otimizam o **design de motores** e **estruturas aeronáuticas**, promovendo **redução de peso** e **aumento de eficiência**.
- **Missões Dinâmicas de Satélites:**  
**Planejamento de tarefas** em **missões espaciais** utiliza **metaheurísticas híbridas** para **maximizar cobertura** e **reduzir custos**.

---

## 10. Química e Engenharia de Processos

- **Modelagem Química:**  
Algoritmos como o **Genetic Algorithm** são aplicados no **controle de reações químicas**, ajustando **parâmetros** para **maximizar o rendimento**. Esses métodos otimizam variáveis de **temperatura**, **pressão** e **concentrações**, melhorando a **eficiência das reações** em processos industriais.
- **Engenharia Química:**  
**Otimizações adaptativas** modelam **processos químicos**, **reduzindo consumo de energia** e **aumentando a segurança**. Técnicas como **PSO** (Particle Swarm Optimization) ajudam a encontrar soluções para problemas de **controle de processos**, como a **otimização de condições operacionais** em refinarias e plantas petroquímicas.

---

## 11. Segurança Cibernética e Redes Computacionais

- **Detecção de Intrusos:**  
**Metaheurísticas combinadas com aprendizado profundo** detectam **padrões anômalos** em redes, prevenindo **ataques**. Algoritmos como **GA** (Algoritmo Genético) e **ACO** (Ant Colony Optimization) são usados para **identificar comportamentos suspeitos**, aumentando a **segurança de redes corporativas**.

- **Otimização de Arquiteturas:**  
**Redes seguras** são projetadas com técnicas como o **Algoritmo de Harmonia**, equilibrando **desempenho** e **resiliência**. Este tipo de **otimização** permite que **sistemas de segurança** em **redes de computadores** sejam mais **robustos** contra **ataques e falhas**.
- **Privacidade em Mineração de Dados:**  
**Soluções robustas** baseadas em **Enxame de Partículas** garantem **anonimato** em grandes **bases de dados**. Esses algoritmos são usados para **proteger dados sensíveis** durante a análise de grandes volumes de informações, como em **transações bancárias e dados pessoais**.

---

## 12. Internet das Coisas (IoT)

- **Gerenciamento de Energia:**  
Técnicas de **clusterização** otimizam o **consumo de energia** em dispositivos IoT, prolongando a **vida útil** das **redes**. **Algoritmos de metaheurísticas** ajudam a **distribuir recursos** de forma mais eficiente, especialmente em dispositivos de **baixo consumo energético**, como **sensores** e **dispositivos portáteis**.
- **Segurança e Privacidade:**  
**Metaheurísticas** aplicadas ao **aprendizado federado** detectam **anomalias**, protegendo **dispositivos conectados**. Isso é crucial em **redes IoT** onde a **segurança** e a **privacidade** são desafios, especialmente em **sistemas distribuídos** como **smart homes** e **dispositivos de saúde**.
- **Aplicações Inteligentes:**  
**IoT em saúde e agricultura** utiliza **PSO (Otimização por Enxame de Partículas)** para **melhorar eficiência operacional** e **reduzir custos**. Isso inclui desde o **monitoramento remoto de pacientes** até a **gestão inteligente de culturas agrícolas**, onde **sensores IoT** ajudam a otimizar o **uso de recursos**.

---

## 13. Sistemas Complexos e Big Data

- **Descoberta de Anomalias:**  
**Metaheurísticas em combinação com deep learning** identificam **padrões fora do normal** em grandes **conjuntos de dados**. Isso é particularmente útil em **Big Data** para detectar **comportamentos atípicos** que indicam **fraudes** ou **falhas** em **sistemas de grande escala**.

- **Frameworks Escaláveis:**

Plataformas como **MapReduce** e **Spark** se beneficiam de técnicas de **otimização** para **processar dados** de maneira eficiente. **Metaheurísticas** são aplicadas para melhorar o **desempenho** desses **frameworks**, otimizando como os dados são **distribuídos** e **processados** em **clusters**.

- **Privacidade em Dados:**

O uso de **algoritmos** como **Harmony Search** assegura **proteção** em sistemas de **big data**. Esses métodos são fundamentais para garantir a **privacidade** de **informações pessoais** e **sensíveis** em grandes volumes de dados, sem comprometer a **qualidade das análises**.

---

## 14. Aprendizado por Reforço e Híbridos

- **Combinação com Metaheurísticas:**

**Modelos cooperativos** ajustam **parâmetros de algoritmos**, combinando **aprendizado por reforço** com **PSO**. Isso é aplicado em problemas onde a **aprendizagem contínua** é necessária para **otimizar decisões** em **ambientes dinâmicos**, como em **jogos** ou **robôs autônomos**.

- **Controle em Tempo Real:**

Aplicações em **manutenção preditiva** utilizam **metaheurísticas** para **prever falhas** e **otimizar intervenções**. Isso permite um **gerenciamento proativo** de **sistemas**, como em **indústrias de manufatura** e **infraestruturas críticas**.

- **Planejamento em Incerteza:**

**Comparações** entre **reforço** e **algoritmos genéticos** são feitas para **otimizar decisões** sob **cenários incertos**. Esses métodos são aplicados em **ambientes** como **mercados financeiros** ou **gestão de recursos**, onde as **condições** podem mudar rapidamente.

---

## 15. Redes Sociais e Complexidade Social

- **Maximização de Influência:**

**Algoritmos** como **PSO** e **GA** modelam **propagação de influência** em **redes sociais**, otimizando **campanhas de marketing**. Eles ajudam a **identificar os influenciadores chave** em uma rede para **maximizar o impacto** de **campanhas publicitárias** ou **sociais**.

- **Recomendações Colaborativas:**  
Filtragem baseada em metaheurísticas aprimora a personalização de recomendações em plataformas digitais. Isso inclui sistemas de recomendação como os utilizados por Netflix, Amazon e Spotify, para sugerir produtos e conteúdos personalizados.
- **Deteção de Fake News:**  
Técnicas que combinam aprendizado profundo e otimização detectam notícias falsas em grandes redes sociais. Esses métodos são aplicados para identificar e mitigar a disseminação de informações falsas em plataformas como Twitter e Facebook.

---

## 16. Gestão de Resíduos e Sustentabilidade

- **Otimização em Economia Circular:**  
A economia circular busca maximizar o reaproveitamento de recursos, reduzindo o desperdício e promovendo a sustentabilidade. Modelos de planejamento de rotas para coleta e reciclagem têm sido otimizados com metaheurísticas como algoritmos genéticos e de enxame de partículas, que permitem o balanceamento entre custos logísticos e impactos ambientais. Um estudo recente apresentou uma abordagem multiobjetivo para otimizar simultaneamente o transporte e o processamento de resíduos, garantindo eficiência energética e menor emissão de carbono.
- **Energia de Resíduos:**  
A transformação de resíduos em energia é um tópico emergente na sustentabilidade, com aplicações em gasificação e gerenciamento de resíduos sólidos. Metaheurísticas como otimização por colônia de formigas têm sido usadas para modelar processos de conversão térmica e aumentar a eficiência energética. Um exemplo notável é o uso dessas técnicas para otimizar a operação de plantas de biomassa, equilibrando geração de energia e custos operacionais.
- **Integração de Recursos:**  
A integração de recursos energéticos é fundamental para melhorar a eficiência de sistemas de gestão de resíduos. Técnicas como otimização baseada em algoritmos evolutivos têm demonstrado sucesso na coordenação de múltiplas fontes energéticas, como solar, eólica e biomassa, integradas a sistemas inteligentes. Esses estudos destacam a aplicação de metaheurísticas em cenários complexos, onde variáveis ambientais e econômicas devem ser consideradas simultaneamente.

## 17. Redes Elétricas e Smart Grids

- **Gerenciamento Energético:**

A **otimização preditiva** é essencial para o **gerenciamento eficiente** de redes elétricas modernas. **Algoritmos metaheurísticos**, como **otimizadores genéticos** e **algoritmos de enxame de partículas**, têm sido amplamente empregados para **prever demandas energéticas** e distribuir **recursos** de forma eficiente. Um estudo recente utilizou **redes neurais** combinadas com **metaheurísticas** para **ajustar a alocação de energia** em **tempo real**, garantindo maior **confiabilidade** e **redução de perdas** no sistema.

- **Microgrids Sustentáveis:**

**Microgrids** são componentes chave das **redes inteligentes**, permitindo a **gestão descentralizada** de recursos. Modelos baseados em **algoritmos evolutivos** têm mostrado resultados promissores na **otimização do fluxo energético** em **microgrids**, integrando fontes **renováveis** como **solar** e **eólica**. Além disso, abordagens **multiobjetivo** foram aplicadas para **equilibrar eficiência econômica** e **sustentabilidade ambiental**, otimizando **custos** e minimizando **impactos**.

- **Sistemas Inteligentes:**

A **integração de fontes renováveis** em **redes elétricas** é um desafio crescente. **Metaheurísticas** como **otimização por colônia de formigas** e algoritmos baseados em **teoria de enxame** têm sido empregadas para **planejar a inserção de energias renováveis** em sistemas existentes. Essas técnicas auxiliam na **redução de sobrecargas** e no aumento da **estabilidade** da rede, atendendo à crescente demanda por soluções de **baixo impacto ambiental**.

[Etimologia de Meta-Heurística citado no Termo de Aceite de Entrega de 16 de outubro]

Residência em IA

Discente: Victor Guerreiro Pimenta

### A Evolução das Heurísticas

O termo "heurística" tem suas raízes na antiga Grécia, derivando da palavra "heureka", famosa exclamação de Arquimedes ao descobrir o princípio da flutuação. A palavra "meta" significa "além" ou "acima", enquanto "hiper" indica "excessivo" ou "acima e além".

A palavra "eureka" tem origem na palavra grega heúreka, que é o pretérito perfeito do verbo heuriskéin, que significa "achar" ou "descobrir".



*16th-century illustration of Archimedes in the bath, with Hiero's crown at bottom right.*

A expressão ficou mundialmente conhecida graças ao cientista grego Arquimedes de Siracusa (287 a.C. – 212 a.C.). A história diz que Arquimedes gritou "Eureka" após descobrir a lei do peso específico dos corpos.

O rei Hierão desafiou Arquimedes a comprovar se o ourives que fabricou a coroa do rei tinha misturado prata ao ouro. Arquimedes percebeu que a quantidade de água que transbordava quando ele se banhava era igual ao volume do seu corpo. Ele concluiu que poderia medir o volume da coroa ao mergulhá-la em água e verificar a quantidade de líquido que deslocaria.



*Archimedes exclaiming Eureka. In his excitement, he forgets to dress and runs nude in the streets straight out of his bath (drawing by Pietro Scalvini, engraving by Carlo Orsolini, 1737)*

Arquimedes mergulhou a coroa, uma barra de ouro e uma barra de prata no mesmo recipiente com a mesma quantidade de água. A coroa transbordou mais água que o ouro e menos que a prata, o que confirmou que a coroa era uma mistura de ouro e prata.

Heurísticas são estratégias ou regras práticas que guiam a busca por soluções, muitas vezes fornecendo resultados satisfatórios, mas não necessariamente ótimos. São métodos baseados em experiência, intuição e palpites informados, em vez de procedimentos algorítmicos rigorosos.

Metaheurísticas são estratégias de alto nível que orientam o design ou seleção de heurísticas. Elas são frameworks que fornecem uma abordagem geral para a resolução de problemas, permitindo a adaptação de heurísticas a situações específicas. As metaheurísticas são frequentemente utilizadas para abordar problemas de otimização complexos onde métodos tradicionais podem ser computacionalmente caros ou ineficientes.

Hiper-heurísticas são metaheurísticas que operam em um nível ainda mais alto, sendo capazes de selecionar, combinar ou adaptar múltiplas metaheurísticas para resolver um problema. É uma abordagem flexível e adaptativa que pode lidar com uma ampla gama de desafios.

A teoria da evolução por seleção natural nos proporciona uma lente para compreender como essas estratégias se desenvolveram e se aperfeiçoaram ao longo do tempo. Os organismos mais simples já demonstravam comportamentos heurísticos, como a busca por alimento ou a fuga de predadores. Com o passar do tempo e a crescente complexidade dos seres vivos, as heurísticas se tornaram mais sofisticadas, permitindo a adaptação a ambientes cada vez mais desafiadores.

A evolução da mente, com suas capacidades de cognição, memória e raciocínio, foi fundamental para o desenvolvimento de heurísticas mais complexas. A capacidade de abstrair, generalizar e criar modelos mentais do mundo possibilitou a construção de regras e atalhos mentais que facilitam a tomada de decisões em situações complexas. A linguagem, por sua vez, permitiu a transmissão cultural de conhecimentos e experiências, acelerando a evolução das heurísticas.

A partir das heurísticas, surgiram as metaheurísticas, estratégias de mais alto nível que orientam a criação e seleção de heurísticas. As hiperheurísticas, por sua vez, representam a última fronteira nessa evolução, permitindo a seleção, combinação e adaptação de múltiplas metaheurísticas. Essas técnicas, cada vez mais sofisticadas, encontram aplicações em diversas áreas, desde a resolução de problemas de otimização até o desenvolvimento de sistemas de inteligência artificial.

## Termo de Aceite de Entrega

### Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

**Data da Reunião (“gate”) de aprovação:** 30 de out. de 2024

**Participantes da Entrega** [matriculados em Residência em IA]:

Victor Guerreiro Pimenta

**Entrega:** [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Resumo e estudos de referências sobre técnicas específicas em metaheurísticas.

☰ Técnicas específicas de metaheurísticas

1. Simulated Annealing.....	2
2. Genetic Algorithms.....	3
3. Genetic Programming.....	7
4. Evolutionary Strategies.....	8
5. Differential Evolution.....	9
6. Estimation of Distribution Algorithms.....	10
7. Particle Swarm Optimization.....	13
8. Artificial Immune Systems.....	16
9. Ant Colony Optimization.....	18
10. Bee Metaheuristics.....	19
11. Bacterial Foraging Algorithm.....	21
12. Harmony Search.....	22
13. Swarm Intelligence.....	23
14. Biomolecular Computing.....	27
15. Quantum Computing.....	29
16. Metaheuristics Based on Sciences.....	31
17. Memetic Algorithms.....	36
18. Tabu Search and Scatter Search.....	37
19. Search Based on Human Behaviors.....	39

**Planejamento:** [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Screening de códigos, frameworks, ou implementações das aplicações, técnicas e algoritmos de metaheurísticas levantados.
- Testar, implementar e avaliar os códigos e frameworks.

**Observação:** [caso precise fazer alguma observação, de qualquer “natureza”]

**ACEITE DA ENTREGA:**

**LEONARDO ALVES:** Em análise! ▾

[**Técnicas específicas de metaheurísticas** citado no Termo de Aceite de Entrega de 30 de outubro]

Residência em IA

Discente: Victor Guerreiro Pimenta

Objetivo: Aprofundar entendimento sobre técnicas específicas e suas respectivas referências fundamentais a partir do livro **Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature** por meio de resumo sobre as técnicas e algoritmos apresentados.

## SUMÁRIO

1. Simulated Annealing.....	2
2. Genetic Algorithms.....	3
3. Genetic Programming.....	7
4. Evolutionary Strategies.....	8
5. Differential Evolution.....	9
6. Estimation of Distribution Algorithms.....	10
7. Particle Swarm Optimization.....	13
8. Artificial Immune Systems.....	16
9. Ant Colony Optimization.....	18
10. Bee Metaheuristics.....	19
11. Bacterial Foraging Algorithm.....	21
12. Harmony Search.....	22
13. Swarm Intelligence.....	23
14. Biomolecular Computing.....	27
15. Quantum Computing.....	29
16. Metaheuristics Based on Sciences.....	31
17. Memetic Algorithms.....	36
18. Tabu Search and Scatter Search.....	37
19. Search Based on Human Behaviors.....	39

# 1. Simulated Annealing

- **2.1 Introdução**

A técnica de "annealing" refere-se ao processo de aquecimento e resfriamento lento de metais, vidros ou cristais para solidificar em uma estrutura cristalina perfeita, minimizando defeitos. Simulated Annealing (SA) simula esse processo em problemas de otimização, onde estados físicos correspondem a soluções do problema, a energia de um estado ao custo da solução e a temperatura a um parâmetro de controle. Baseado no algoritmo de Metropolis, SA utiliza um método de descida

modificado para escapar de mínimos locais, permitindo amostras em estados de energia mais baixos.

- **2.2 Simulated Annealing Básico**

Baseado na termodinâmica estatística, SA se comporta segundo a distribuição de Boltzmann, onde altas temperaturas distribuem uniformemente preferências de estado, enquanto em temperaturas baixas, apenas estados de mínima energia têm probabilidade de ocorrer. O processo de resfriamento é crucial: um resfriamento muito rápido pode levar a uma convergência prematura, e um resfriamento muito lento torna o algoritmo lento. O uso de uma programação de resfriamento mais rápida é comum para soluções subótimas.

- **Exemplo 2.1**

Para minimizar a função de Easom, uma função complexa devido a muitos mínimos locais, o `simulannealbnd` do MATLAB é utilizado, mostrando resultados próximos ao mínimo global após restringir o espaço de busca.

- **2.3 Variantes do Simulated Annealing**

Diversas variantes foram desenvolvidas para acelerar a convergência do SA:

- **Cauchy Annealing** substitui a distribuição de Boltzmann pela distribuição de Cauchy, facilitando a fuga de mínimos locais.
- **Simulated Reannealing** permite que  $T$  diminua exponencialmente, adaptando-se a insensibilidades na busca multidimensional.
- **Generalized SA** unifica métodos de annealing com uma abordagem inspirada na termodinâmica generalizada.
- **Opposition-based SA** melhora a precisão e taxa de convergência usando vizinhos opostos.
- **SA com valor global conhecido** gera pontos aleatórios com base na diferença entre o valor atual e o ótimo conhecido, assegurando a convergência com uma cota superior de iterações.
- **Fuzzy Annealing** aplica lógica fuzzy para aumentar a velocidade de cálculo e evitar mínimos locais iniciais, reduzindo a "fuzziness" gradualmente.

## 2. Genetic Algorithms

**Introdução à Computação Evolutiva:** Algoritmos evolutivos (EAs) seguem o paradigma neo-Darwiniano, que une a teoria darwiniana clássica com o selecionismo de Weismann e a genética de Mendel. A evolução pode ser acelerada por estratégias de aprendizado, como a de Lamarck ou o efeito Baldwin. EAs são ferramentas úteis para problemas de otimização complexos, não contínuos e multimodais, operando com paralelismo inerente e fácil hibridização. Cada EA contém geradores e seletores de população, estimadores de aptidão e operadores genéticos básicos: crossover, mutação e seleção.

**Diferença entre Algoritmos Evolutivos e Simulated Annealing (SA):** Ambos métodos envolvem cadeias de Markov, mas EAs utilizam populações de indivíduos em paralelo, enquanto SA opera com uma única cadeia. SA é menos eficiente para encontrar mínimos globais comparado aos EAs devido à sua natureza sequencial. A combinação de SA com EAs resulta em métodos híbridos que combinam a paralelização dos EAs com a capacidade de SA de evitar mínimos locais.

### Terminologias da Computação Evolutiva:

- **População:** Conjunto de indivíduos em uma geração.
- **Cromossomo:** Representação individual na população, que define uma solução para o problema.
- **Gene:** Elemento em um cromossomo que codifica um parâmetro do problema.
- **Alelo:** Variantes de um gene em uma posição específica de um cromossomo.
- **Genótipo e Fenótipo:** Genótipo é o código genético do organismo; fenótipo é a expressão observável dos traços.
- **Aptidão (Fitness):** Função que mede a qualidade de uma solução.
- **Seleção Natural:** Mecanismo evolutivo que propaga traços que aumentam a sobrevivência e a reprodução dos indivíduos.
- **Deriva Genética:** Processo estocástico que altera a frequência alélica ao longo do tempo devido a amostragens aleatórias.

### 3.4 Selection/Reproduction

- **Princípio da Seleção:** Baseia-se na aptidão dos indivíduos, selecionando os mais fortes para reprodução, aumentando a probabilidade de reprodução de cromossomos com maior aptidão.
- **Métodos de Seleção:**
  - **Seleção por roleta:** Probabilidade proporcional à aptidão de cada indivíduo.
  - **Seleção por ranking:** Ordena indivíduos por aptidão e atribui probabilidades lineares.

- **Seleção por torneio:** Compara grupos e seleciona o melhor de cada torneio para o pool de acasalamento. Pode incluir torneio de Boltzmann, que adiciona probabilidade na escolha.
- **Estratégia de Elitismo:** Preserva o melhor indivíduo em cada geração, melhorando a convergência ao custo de possível convergência prematura.
- **Seleção Fitness-Uniforme e Deleção:** Mantém diversidade ao distribuir uniformemente indivíduos segundo valores de aptidão e deletando aqueles com valores muito comuns.
- **Multikulti Selection:** Inspira-se na seleção natural de preferir indivíduos diferentes, promovendo diversidade.
- **Estratégias de Substituição:** Definem quantos indivíduos serão substituídos na nova geração. Inclui substituição completa, aleatória, pior, mais velho, entre outras.

### 3.5 Crossover

- **Definição:** Operador binário que troca informações entre dois cromossomos selecionados, gerando novos indivíduos.
- **Métodos de Crossover:**
  - **Crossover de um ponto:** Troca dados entre dois cromossomos a partir de um ponto.
  - **Crossover de dois pontos:** Realiza a troca entre dois pontos, minimizando a perturbação.
  - **Multiponto e Uniforme:** Trocas em múltiplos pontos ou em bits individuais, com uniformidade de probabilidade.
- **Exploração e Convergência Prematura:** Quando cromossomos tornam-se semelhantes, o crossover sozinho é insuficiente; a mutação é então necessária para reintroduzir diversidade genética.

### 3.6 Mutation

- **Definição:** Operador unário que altera aleatoriamente um gene de um cromossomo, reinserindo diversidade perdida.
- **Tipos de Mutação:**
  - **Mutação Pontual:** Substituição, deleção ou inserção de um gene ou bit.
  - **Mutação em Larga Escala:** Realiza alterações em múltiplas posições simultâneas.
  - **Operadores de Inversão e Rearranjo:** Modificam a sequência de genes ou bits em uma porção de um cromossomo.
- **Mutação Uniforme:** Para representações binárias, altera independentemente cada bit com uma probabilidade fixa, geralmente  $p = 1/L$ , onde  $L$  é o comprimento da string.

- **Taxa de Mutação:** Uma taxa alta pode tornar a busca genética mais aleatória, enquanto uma taxa baixa favorece a convergência.

### **Exploitation e Exploração**

Os algoritmos evolutivos (EAs) são guiados pelos processos de exploração e exploração. Exploração representa a busca por novas regiões no espaço de busca, enquanto a exploração aproveita as informações já obtidas. A exploração é promovida por operadores genéticos que preservam a diversidade, enquanto a exploração é impulsionada pelo procedimento de seleção. A diversidade é essencial para evitar a convergência prematura, ajustando parâmetros de controle, como a probabilidade de crossover ( $P_c$ ), a probabilidade de mutação ( $P_m$ ) e o tamanho da população ( $NP$ ). Um aumento de  $P_c$  acelera a exploração mas pode perturbar boas soluções; aumento de  $P_m$  ajuda a reintroduzir alelos perdidos; aumento de  $NP$  reduz a probabilidade de convergência prematura.

### **Adaptação dos Parâmetros de Controle**

Adaptações dos parâmetros  $P_c$  e  $P_m$  são recomendadas ao longo da busca. Inicialmente, a ênfase deve ser na exploração, mudando para a exploração em estágios avançados.  $P_m$  pode ser ajustado diminuindo linearmente com o número de gerações ou atribuído com base na aptidão relativa, protegendo as melhores soluções ao reduzir o crossover para soluções de maior aptidão.

### **Controle da Diversidade**

A diversidade genética pode ser aprimorada usando reinicialização parcial, aumento do tamanho da população ou operadores de remoção de duplicatas. A diversidade ajuda a evitar convergência prematura, e abordagens como a metodologia saw-tooth e o operador de diversidade guiado promovem alternância entre exploração e exploração com ganhos em fitness e menor número de avaliações.

### **Variação do Tamanho da População**

Estratégias de dimensionamento da população, como Messy GA e GENITOR, variam o tamanho inicial da população e adaptam o número de indivíduos conforme necessário. A abordagem piramidal de GA sem parâmetros usa múltiplas populações iterativas, ajustando-se dinamicamente à dificuldade do problema com menor necessidade de ajustes manuais.

### **Envelhecimento**

O envelhecimento é um mecanismo para manter a diversidade, associando uma idade para cada indivíduo. Indivíduos mais velhos são eventualmente substituídos, o que ajuda a evitar

a dominação de indivíduos mais experientes e permite a evolução contínua. Métodos como cohort GA gerenciam subpopulações para controlar a frequência de reprodução com base na aptidão relativa.

### **Aplicação em Otimização**

Exemplo prático demonstra o uso de GA simples na otimização da função de Rosenbrock, configurando Pc, Pm e aplicando seleção de roleta. A evolução da população ao longo das gerações mostra a eficácia dos ajustes nos parâmetros para alcançar uma solução próxima ao ótimo global.

### **Algoritmos Genéticos Bidimensionais**

Em problemas bidimensionais, como processamento de imagens, GAs convencionais perdem correlações ao codificar imagens como strings unidimensionais. GAs bidimensionais utilizam operadores de crossover que preservam melhor a estrutura, como o UNBLOX, que aplica crossover uniforme, melhorando a taxa de convergência em comparações com GAs simples.

### **Algoritmos Genéticos com Codificação Real**

Em otimizações numéricas, a codificação em ponto flutuante é preferível para representar variáveis contínuas, oferecendo vantagem na exploração de continuidades locais e proporcionando maior consistência entre execuções. Essa abordagem é mais rápida e apresenta melhores desempenhos em otimizações contínuas do que a codificação binária.

**Otimização de Sequências com Algoritmos Genéticos:** Para problemas como escalonamento e o TSP (Traveling Salesman Problem), a codificação por permutação permite a representação de sequências sem alelos ausentes ou duplicados. Operadores genéticos devem evitar soluções inviáveis ou propor métodos para reparo. Operadores unários, como inversão e troca, e operadores binários, como partial matched crossover, order crossover, cycle crossover, e recombinação de arestas são utilizados para reordenar sequências.

**Representação com Chaves Aleatórias:** Cada símbolo é codificado com um número real no intervalo (0,1). A ordenação dessas chaves resulta na solução decodificada. Essa representação evita descendentes inviáveis e facilita a aplicação de operações cruzadas. O Genetic Algorithm (GA) com chaves aleatórias garante que as soluções geradas sejam

factíveis, sendo robusto e simples.

**GA com Chaves Aleatórias Viesadas:** Uma variação do GA com chaves aleatórias, que divide a população em elite e não-elite. Na geração de descendentes, um dos pais é selecionado da elite e o outro da não-elite, favorecendo a herança de características da elite.

**Codificação de Árvores Geradoras:** Em problemas combinatórios que buscam árvores geradoras mínimas, algoritmos polinomiais conseguem soluções ótimas, enquanto problemas mais complexos são NP-difíceis. O esquema NetKeys aplica chaves aleatórias à codificação de redes, ordenando arestas para formar árvores geradoras, embora sua complexidade computacional seja alta.

**Representação Direta de Árvores:** Utiliza a codificação direta, como o predecessor code, onde cada nó de uma árvore tem um predecessor registrado. Outro exemplo é o código Dandelion, que representa árvores com complexidade  $O(n)$  e apresenta eficiência computacional elevada para redes grandes.

## 3. Genetic Programming

### 4.1 Introdução

- Genetic Programming (GP) é uma variante de algoritmos genéticos (GA) para regressão simbólica, que evolui programas de computador a partir de declarações de problemas.
- É uma técnica de busca hipereurística, adequada para descobrir estruturas subjacentes ótimas, como leis empíricas.
- Os cromossomos em GP têm comprimento e estrutura de dados variáveis, representados como árvores hierárquicas, onde nós internos são operadores matemáticos e folhas são variáveis de entrada ou constantes.
- O fenômeno de bloat afeta o GP, resultando em crescimento excessivo de ramos não codificantes, aumentando o consumo de recursos computacionais.
- GP padrão enfrenta dificuldades estruturais ao procurar soluções que exigem árvores muito cheias ou muito estreitas, devido à falta de operadores de edição de estrutura local.

## 4.2 Árvores de Sintaxe

- As variáveis e operadores algébricos são representados em genes como árvores de sintaxe, adequadas para cruzamento e mutação.
- As expressões são escritas em formato s-expressions (expressões simbólicas), que correspondem a uma estrutura de árvore.
- Exemplos de cruzamento e mutação são apresentados, ilustrando a troca e substituição de subárvores dentro das árvores de sintaxe.

**Exemplo 4.1:** Utilização de GP para gerar modelos de entrada-saída não lineares, com a complexidade do modelo controlada pelo método de mínimos quadrados ortogonais (OLS). O processo de evolução resultou na identificação correta da estrutura do modelo.

## 4.3 Causas do Bloat

- O bloat é uma consequência do uso de representações de comprimento variável, levando ao crescimento rápido de indivíduos sem melhoria proporcional na aptidão.
- A presença de código irregular e desnecessário não se traduz em melhorias na qualidade das soluções.
- Teorias clássicas explicam o bloat por meio de introns e a teoria do hitchhiking, que mostram que a seleção aleatória e o cruzamento padrão de subárvores não causam crescimento de código.
- A teoria do viés de remoção sugere que a evolução favorece a substituição de pequenos ramos por ramos de tamanho médio, levando a árvores maiores.
- Equações de evolução de tamanho foram desenvolvidas para formalizar a dinâmica do tamanho médio do programa.

# 4. Evolutionary Strategies

- Paradigma de estratégias evolutivas (ES) é uma das abordagens de algoritmos evolutivos (EAs) mais bem-sucedidas.
- Métodos de busca de gradiente evolutivo e evolução de gradiente utilizam EAs para construir informações de gradiente que direcionam a busca de forma eficiente.
- CMA-ES melhora a eficiência da busca ao supor que o espaço de solução local tem uma forma quadrática.

## 5.1 Introdução

- ES, desenvolvido inicialmente para problemas de otimização numérica, foi estendido para problemas de otimização discreta.

- Parâmetros objetivos ( $x$ ) e de estratégia ( $\sigma$ ) são codificados diretamente no cromossomo, eliminando a necessidade de codificação ou decodificação.
- Programação evolutiva é semelhante ao ES, mas não utiliza crossover, aplicando competição probabilística para seleção.

### Diferenças entre ES e GA

- **Procedimento de Seleção:** ES utiliza seleção determinística, escolhendo sempre o número especificado de melhores indivíduos; GA utiliza seleção aleatória proporcional à aptidão.
- **Ordem Relativa de Seleção e Operações Genéticas:** Em ES, a seleção ocorre após crossover e mutação; em GA, ocorre antes.
- **Adaptação de Parâmetros de Controle:** Parâmetros de estratégia em ES são evoluídos automaticamente; em GA, são específicos do problema.
- **Função de Mutação:** Em GA, a mutação visa recuperar diversidade genética; em ES, funciona como operador de busca com passo adaptativo.

### 5.2 Algoritmo Básico

- ES canônico usa apenas operações de mutação, refletindo reprodução assexuada.
- Crossover pode ser introduzido, combinando dois pais para gerar descendentes.
- Mutação gaussiana gera descendentes com genes definidos pela distribuição normal.
- Estratégias de seleção como  $(\lambda + \mu)$  e  $(\lambda, \mu)$  são utilizadas, sendo ambas métodos de amostragem determinística.

### Desempenho e Seleção

- A estratégia  $(\lambda + \mu)$  garante melhoria contínua, mas pode falhar em ambientes dinâmicos; a estratégia  $(\lambda, \mu)$  é recomendada em tais casos.
- Distribuições de probabilidade Cauchy e Levy podem ser usadas em vez de mutação gaussiana para uma variação maior em mutações.

### Análise Dinâmica

- Análise dinâmica leva a um sistema de diferenças não lineares que descreve a evolução do valor médio em ES, mostrando resultados superiores em ambientes não ruidosos.

### 5.3 Busca de Gradiente Evolutivo e Evolução de Gradiente

- O gradiente generaliza a derivada de uma função, indicando a direção de maior aumento e sua magnitude.

- Métodos clássicos de gradiente podem ser presos em ótimos locais; a busca de gradiente evolutivo utiliza EAs para construir informações de gradiente em paisagens não diferenciais.
- A busca de gradiente evolutivo implementa uma estratégia  $(1, \lambda)$ -ES, gerando novos indivíduos por mutação normal.

### Evolução de Gradiente

- Método metaheurístico baseado em população que utiliza estimativa de gradiente e atualiza vetores representando possíveis soluções em várias iterações.
- Inclui operadores de atualização, salto e renovação para evitar ótimos locais.

### Exemplos

- **Função de Rosenbrock:** Aplicação de ES com recombinação intermediária, convergindo para o mínimo global em múltiplas execuções.
- **Função Easom:** Resolução do problema com ES, alcançando o mínimo global consistentemente.

## 5. Differential Evolution

### 6.1 Introdução

- O Differential Evolution (DE) é um algoritmo evolutivo (EA) eficiente e simples, focado na otimização global de parâmetros reais.
- Utiliza uma estratégia de reprodução multiparental e um relacionamento de seleção entre indivíduos e suas progênies.
- DE se diferencia de EAs tradicionais por perturbar a população atual com diferenças escaladas entre membros selecionados aleatoriamente.
- A complexidade espacial do DE é baixa em comparação com outros otimizadores, como o CMA-ES, mas pode ser superado em funções não separáveis linearmente.

### 6.2 Algoritmo DE

- O DE realiza uma busca baseada em direção, utilizando informações direcionais da população atual.
- Para cada indivíduo, três outros indivíduos distintos são escolhidos aleatoriamente para formar um grupo parental.
- A criação de um vetor mutado e um vetor alvo ocorre através de recombinação aritmética.

- As estratégias de mutação mais utilizadas são diferenciadas por suas nomenclaturas, como DE/x/y/z, onde cada letra indica um aspecto da mutação e cruzamento.
- A mutação é um processo regular no DE, diferindo do GA, onde é ocasional.
- **Mutação Diferencial:** Usa três indivíduos distintos para evitar a convergência local prematura e garantir a convergência global. Exemplos incluem DE/rand/1 e DE/best/2.
- **Cruzamento:** O vetor mutado é combinado com o vetor atual para gerar um novo indivíduo. O cruzamento é controlado por uma probabilidade específica.
- **Seleção:** A seleção ocorre entre o indivíduo atual e seu descendente, escolhendo o que apresentar melhor valor de função objetivo.
- O DE opera com duas populações de mesmo tamanho, atualizando a população antiga com a nova com base no desempenho.
- Parâmetros de controle incluem o tamanho da população (NP), fator de escala (F) e taxa de cruzamento (Cr), que devem ser ajustados para evitar a convergência prematura.

### 6.3 Variantes de DE

- DE supera o PSO e outros EAs em qualidade de solução, mas enfrenta problemas de convergência lenta.
- **DE Adaptativa:** Ajusta F e Cr dinamicamente, resultando em uma convergência mais rápida, especialmente em problemas de alta dimensão.
- **DE com Autoajuste:** Adapta estratégias de geração de vetores e parâmetros de controle com base em experiências anteriores.
- **DE Baseado em Oposição:** Melhora a velocidade de convergência e precisão, sendo adequado para problemas de otimização ruidosos.
- **DE com Parâmetros Autoajustáveis:** Ajusta F e Cr durante a execução, melhorando o desempenho do algoritmo.

## 6. Estimation of Distribution Algorithms

### 7.1 Introdução

- EDAs são algoritmos inspirados em computação evolutiva e aprendizado de máquina, também conhecidos como probabilistic model-building GAs.
- Combinam busca baseada em população e exploração através da combinação de soluções promissoras.

- Utilizam modelos probabilísticos para guiar a exploração do espaço de busca, lidando com informações ruidosas.
- EDAs buscam regularidades ocultas em problemas e as utilizam para otimizar a busca, estimando distribuições de soluções promissoras.
- Trocam operadores genéticos clássicos pela estimativa de um modelo probabilístico, gerando novas populações a partir dele.
- Têm a vantagem de requerer menos avaliações de fitness do que os algoritmos evolutivos tradicionais, mas a complexidade computacional aumenta com a dimensionalidade.

## 7.2 Fluxograma do EDA

- O modelo probabilístico é induzido a partir de indivíduos da população atual, com a nova população gerada por amostragem deste modelo.
- O processo se divide em duas fases: seleção do modelo e ajuste do modelo.
- Representação compacta do modelo evita a convergência prematura, utilizando um vetor de probabilidades para representar a população.
- O algoritmo itera entre selecionar bons candidatos, estimar a distribuição de probabilidade e gerar novos candidatos até atender aos critérios de terminação.

## 7.3 Aprendizado Incremental Baseado em População (PBIL)

- PBIL é uma abstração do GA binário, mantendo estatísticas da população.
- A metodologia combina otimização evolutiva e escalonamento, gerando um vetor de probabilidade para criar soluções de alta qualidade.
- Assume a independência das variáveis e utiliza um modelo de variável aleatória de Bernoulli.
- O vetor de probabilidade é atualizado pela melhor solução encontrada, convergindo para valores de 0 ou 1.
- Um passo de mutação pode ser aplicado ao vetor de probabilidade aprendido para inibir a convergência prematura, perturbando-o com uma pequena probabilidade.
- Variantes de PBIL incluem mutações e aprendizado a partir de exemplos negativos para melhorar a diversidade e os resultados.
- PBIL pode ser estendido para espaços contínuos usando um modelo de distribuição Gaussiana.

## 7.6 Propriedades de Convergência

- **Estudo do PBIL:** Com seleção elitista em espaço discreto, a dinâmica do PBIL é modelada como um sistema dinâmico discreto. Um learning rate pequeno resulta em pontos fixos estáveis em relação à distância de Hamming.

- **Comparação com UMDA:** A dinâmica do UMDA é similar à de algoritmos genéticos com crossover uniforme. Ambos podem encontrar ótimos em funções lineares, mas têm limitações em problemas não lineares.
- **Modelo Markoviano:** PBIL e cGA são modelados por processos de Markov, convergindo para ótimos locais com learning rates pequenos. Condições suficientes para convergência foram apresentadas, bem como limites de probabilidade para convergência.
- **Comportamento Assintótico:** O comportamento dinâmico de modelos limites de UMDA e FDA é analisado, mostrando que ótimos locais são estáveis, mas UMDA pode ficar preso em ótimos locais dependendo da modelagem inicial.
- **Análise de Tempo de Convergência:** A complexidade temporal dos EDAs é medida pelo tempo de primeiro atingimento. A análise do cGA mostra um tempo médio exponencial de primeiro atingimento em funções pseudo-booleana.
- **Classificação da Dificuldade:** Um método para classificar a dificuldade dos problemas em EDAs é proposto, com condições probabilísticas relacionadas ao tempo de primeiro atingimento.
- **Convergência Prematura:** A dinâmica de EDAs em relação à convergência prematura é estudada, mostrando que EDAs gerais não atendem a condições necessárias para algoritmos de busca eficazes.

## 7.7 Outros EDAs

- **Desafios em Problemas de Alta Dimensão:** EDAs tradicionais enfrentam dificuldades na resolução de problemas de alta dimensão devido ao aumento dos custos computacionais. O controle de complexidade de modelo melhora o desempenho em problemas complexos.
- **Distribuições Gaussianas Multivariadas:** Vários EDAs baseados em distribuições Gaussianas foram propostos, com abordagens como EMNAGlobal e normal IDEA, que utilizam estimativas de máxima verossimilhança para modelar variáveis.
- **CMA-ES como EDA:** O CMA-ES é considerado um EDA que trata a paisagem de soluções como um espaço de funções de densidade de probabilidade.
- **GA Estocástico:** Utiliza uma estratégia de codificação estocástica para explorar o espaço de busca em regiões dinâmicas, produzindo descendentes por amostragem aleatória.
- **Algoritmos de Amostragem:** Algoritmos como o edge histogram-based e node histogram-based são projetados para problemas de permutação, com o Mallows EDA superando esses métodos em certas condições.
- **Mutação Guiada:** Um híbrido entre EA e EDA que utiliza a informação de localização das soluções locais para produzir descendentes, com desempenho superior em comparação ao PBIL, mas com maior custo computacional.

- **Código Fonte Disponível:** Fontes de código para diversos EDAs são mencionadas, incluindo extensões do cGA e BOA.

### 7.7.1 Construção de Modelos Probabilísticos em GP

- **Classificação de Algoritmos:** Dividem-se em algoritmos baseados em árvore protótipo e aqueles guiados por gramática.
- **Abordagens Baseadas em Árvore Protótipo:** Exemplos incluem PIPE, que estende o PBIL para evolução de programas, e POLE, que utiliza uma rede bayesiana para geração de novos indivíduos.
- **Desafios na Evolução de Programas:** Problemas relacionados ao número de símbolos e à correção sintática são discutidos.
- **Abordagens Baseadas em Gramática:** Exemplos como a evolução de programas com gramática estocástica são citados, abordando o aprendizado explícito e a modelagem baseada em gramática.

## 7. Particle Swarm Optimization

### 9.1 Introdução

- O PSO (Particle Swarm Optimization) utiliza múltiplas partículas autônomas que atuam em conjunto para gerar comportamentos complexos a partir de interações simples.
- Originou-se de estudos sobre comportamento social em animais, como bandos de pássaros e cardumes de peixes.
- O algoritmo evolui populações de partículas que ajustam suas posições com base nas melhores experiências individuais e coletivas.
- Comparado a algoritmos como ant colony e EAs (Evolutionary Algorithms), o PSO é mais simples, requer menos operações matemáticas e é computacionalmente menos exigente.

### 9.2 Algoritmos Básicos do PSO

- O PSO básico se baseia na experiência de cada partícula e na experiência da partícula mais bem-sucedida.
- Um enxame de partículas é definido em um espaço n-dimensional, onde cada partícula tem uma posição e uma velocidade.

- A atualização das partículas envolve seguir as melhores posições pessoais (pbest) e globais (gbest).
- Problemas de estagnação podem ocorrer, onde o enxame se prende a ótimos locais, levando à necessidade de resemear partículas.
- O PSO pode ser melhorado por técnicas como o LDWPSO (Linearly Decreasing Weight PSO), que ajusta o peso da inércia para equilibrar a busca global e local.

### 9.2.1 Bare-Bones PSO

- O Bare-bones PSO é uma versão simplificada que elimina a equação de velocidade, utilizando uma distribuição gaussiana para amostrar o espaço de busca.
- As atualizações das posições das partículas dependem das melhores posições globais e locais, mas ainda pode enfrentar convergência prematura.

### 9.2.2 Variantes do PSO Usando Distribuições Gaussiana ou Cauchy

- O PSO básico utiliza distribuições de probabilidade uniforme, mas a introdução de distribuições gaussianas ou cauchy pode melhorar a capacidade de ajuste fino e evitar mínimos locais.
- Variantes que utilizam essas distribuições são propostas para aumentar a eficácia do algoritmo em diferentes paisagens de fitness.

## 9.4 Outras Variações do PSO

- **Optimizing Liaisons Optimization:** Uma versão simplificada do PSO que não considera a melhor posição pessoal do partícula, apresentando desempenho comparável ao PSO com parâmetros mais fáceis de ajustar.
- **PSO Básico:** Um modelo síncrono onde partículas comunicam suas melhores posições e valores objetivos de forma imediata, resultando em uma troca de informações perfeita antes da atualização de posições.
- **PSO Assíncrono:** As partículas atualizam suas memórias imediatamente após mover-se, permitindo que outras partículas aproveitem essa nova informação rapidamente. Geralmente, apresenta convergência mais rápida que o modelo síncrono, mas corre o risco de ser atraído para soluções enganosas.
- **FDR-PSO (Fitness-Distance-Ratio PSO):** Utiliza informações de partículas de maior fitness, melhorando a movimentação das partículas em direção a estas, combatendo a convergência prematura.

- **PSO Concorrente:** Simula simultaneamente PSO modificado e FDR-PSO, evitando efeitos de crosstalk entre pbest e gbest.
- **Garantia de Convergência PSO:** Utiliza uma equação de atualização de velocidade diferente para manter o partícula gbest em movimento até alcançar um mínimo local.
- **PSO Ortogonal:** Aplica um design experimental ortogonal para ajustar a velocidade de cada partícula usando uma abordagem de divisão e conquista.
- **Michigan PSO:** Cada membro da população codifica parte da solução, reduzindo a dimensão do espaço de busca. Introduce uma função de fitness local e vizinhanças dinâmicas.
- **Aprendizado Incremental de PSO:** Aumenta a população se a busca local não for satisfatória, melhorando a solução.
- **Estratégia de Utilização Eficiente da População (EPUS-PSO):** Um gerente de população elimina partículas redundantes e ajusta o número de partículas com base no status da busca de solução.
- **PSO com Líder Envelhecendo e Desafiante (ALC-PSO):** Melhora o PSO superando a convergência prematura, permitindo que novos indivíduos desafiem a liderança conforme o líder envelhece.
- **Aprendizado Baseado em Oposição:** Substitui a partícula menos apta por sua antipartícula, melhorando o desempenho do PSO.
- **PSO Adaptativo:** Avalia a distribuição da população e fitness das partículas em tempo real, ajustando parâmetros do algoritmo automaticamente.
- **PSO Caótico:** Utiliza mapas caóticos para adaptação de parâmetros, melhorando a capacidade de busca.
- **PSO Frankenstein:** Combina componentes algorítmicos para otimizar a velocidade e confiabilidade.
- **PSO Baseado em Clã:** Divide a população em clãs, onde os líderes ajustam suas posições após a seleção do melhor fitness.

- **PSO Estático:** Utiliza a analogia de energia eletrostática, onde partículas se repelem entre si, melhorando a exploração.
- **PSO de Buraco Negro Aleatório:** Gera buracos negros próximos à melhor partícula, puxando partículas aleatoriamente, ajudando a escapar de mínimos locais.
- **PSO de Aprendizado Social:** Introduce mecanismos de aprendizado social, onde partículas aprendem com outras melhores.
- **Starling PSO:** Adiciona um mecanismo de diversidade inspirado no comportamento coletivo de estorninhos.

## 9.5 PSO e Híbridos de Algoritmos Evolutivos (EAs)

- **Diferenças Estruturais:** No PSO, partículas movem-se no espaço de solução influenciadas por outras, enquanto EAs envolvem a reprodução de indivíduos para criar novas soluções.
- **Facilidade de Implementação:** O PSO é mais fácil de implementar, com menos parâmetros a ajustar em comparação aos EAs.
- **Capacidade de Memória:** O PSO possui uma memória mais eficaz, recordando pbest e gbest, o que melhora a diversidade do enxame.
- **Hibridização:** A combinação de EAs e PSO visa unir as vantagens de ambos, potencializando a eficiência na busca por soluções otimizadas.

## 9.6 Discrete PSO

- **Versão Discreta do PSO:** Adaptada para problemas com variáveis binárias. A movimentação das partículas é realizada através da probabilidade de cada bit estar em um estado específico.
- **Partículas e Velocidade:** Cada partícula é composta por variáveis binárias, e a velocidade é transformada em probabilidade usando uma função logística. O valor de velocidade é limitado para evitar que a probabilidade se aproxime de 0 ou 1.
- **Atualização de Velocidade:** Utiliza a equação básica do PSO, com  $c1r1 + c2r2$  tipicamente limitado a 4.

- **Aplicações:** Inclui técnicas como multiphase discrete PSO, onde os grupos de partículas adotam diferentes estratégias de busca, e jumping PSO, onde as partículas mudam de solução sem considerar a velocidade.

## 9.7 Multi-swarm PSOs

- **Definição de Multi-swarm PSO:** Utiliza múltiplas enxame para explorar o espaço de busca e encontrar soluções ótimas, semelhante ao comportamento de espécies de pássaros em busca de alimento.
- **Solução de Problemas Multimodais:** Adotado para resolver problemas multimodais e combater a convergência prematura.
- **Niching PSO:** Cria subswarms a partir de partículas próximas, permitindo a identificação de nichos e a fusão de subswarms durante a otimização.
- **Turbulent PSO:** Divide a população em duas subswarms que se movem em direções opostas, ajudando a evitar a convergência prematura.
- **Fuzzy Adaptive Turbulent PSO:** Combina o PSO turbulento com um controlador lógico fuzzy para ajustar dinamicamente os parâmetros de velocidade.
- **Speciation-based PSO:** Utiliza especiação espacial para localizar múltiplos ótimos locais em paralelo, mantendo a diversidade.
- **Adaptive Sequential Niche PSO:** Modifica os valores de fitness das partículas com uma função de penalidade, permitindo a descoberta sequencial de soluções ótimas.
- **Trocas de Informação entre Swarms:** Após várias gerações, as enxames trocam informações para melhorar a diversidade e a exploração.
- **Cooperative PSO:** Emprega comportamento cooperativo entre múltiplas enxames para otimizar problemas multimodais, dividindo as variáveis de decisão entre as enxames.
- **PSOPB e PS20:** Dividem a população em dois grupos, promovendo interações que mimetizam comportamentos parasitários e simbioticidade entre espécies.
- **TRIBES:** Sistema PSO sem parâmetros que se adapta ao feedback de desempenho, onde a população é dividida em tribos que trocam as melhores partículas entre si.

# 8. Artificial Immune Systems

## 10.1 Introdução

- O Sistema Imunológico Artificial (AIS) é inspirado no sistema imunológico biológico, que protege o corpo contra doenças detectando e eliminando patógenos e células tumorais.

- O sistema imunológico é robusto e se baseia em mecanismos de memória para reconhecer patógenos previamente encontrados, o que é útil em ciência da computação e detecção de intrusões.
- Composto por células como linfócitos B e T, o sistema é capaz de discriminar entre células próprias e patogênicas, mantendo diversidade em seu repertório.
- O AIS apresenta características de reconhecimento de padrões, auto-identidade, otimização e aprendizado de máquina, aplicando-se a problemas de otimização de funções multimodais.

## 10.2 Teorias Imunológicas

- Quatro teorias imunológicas são discutidas: clonal selection, immune networks, negative selection e danger theory.
- O aprendizado e a memória do sistema imunológico baseiam-se principalmente nas teorias de clonal selection e immune networks, enquanto a seleção de detectores para identificar entidades anômalas é fundamentada na negative selection.

### 10.3.1 Algoritmo de Seleção Clonal

- O algoritmo de seleção clonal se baseia na teoria de seleção clonal, onde células imunes proliferam em resposta a antígenos, gerando clones que se diferenciam em células efetoras e de memória.
- Anticorpos com boa afinidade são selecionados como "pais", que se reproduzem por mitose, e a seleção clonal é composta por etapas de clonagem, cruzamento, mutação e seleção.
- Este processo simula a evolução das células imunes, permitindo que elas aprendam e memorizem os modos dos antígenos, resultando em uma resposta imune mais eficaz.

### 10.3.2 Artificial Immune Network

- **aiNet**: Combina CLONALG com a teoria de rede imunológica para resolver problemas de otimização. Funciona como uma rede construtiva e competitiva, onde anticorpos correspondem aos nós da rede, com a concentração e afinidade dos anticorpos representando seus estados. O aprendizado é responsável pelas mudanças na concentração e afinidade dos anticorpos, visando construir um conjunto de memória que reconhece a distribuição espacial dos antígenos.
- **opt-aiNet**: Adapta aiNet para problemas de otimização multimodal, mantendo múltiplas soluções ótimas. Ajusta dinamicamente o tamanho da população e preserva soluções locais estáveis. O processo inclui avaliação de aptidão, clonagem

e mutação dos anticorpos, e eliminação de células redundantes com base na afinidade.

- **omni-aiNet**: Foca na manutenção da diversidade populacional e busca simultânea de múltiplas soluções de alta qualidade.
- **dopt-aiNet**: Melhora a diversidade e refina soluções para se adequar a otimizações dinâmicas, utilizando procedimentos de busca de linha de seção áurea.
- **dt-aiNet**: Introduce a teoria do perigo para aumentar a qualidade das soluções e a diversidade da população.

### 10.3.3 Negative Selection Algorithm

- **Inspiração**: Baseado no mecanismo de seleção negativa que detecta antígenos desconhecidos. A implementação para padrões binários opera em tempo linear.
- **Funcionamento**: Inicialmente, padrões normais são tratados como autômatos, e detectores aleatórios são gerados e comparados com os padrões autônomos. Detectores que não correspondem a padrões autônomos são retidos.
- **Aplicações**: Usado em detecção de anomalias, como em segurança de redes. A AIS também é aplicada em sistemas de detecção de intrusões.
- **Receptor Density Algorithm**: Derivado de modelos de células T imunológicas, classifica dados em anômalos ou normais com base em entradas discretas e contribui para a geração de assinaturas limpas.

### 10.3.4 Dendritic Cell Algorithm

- **Teoria do Perigo**: Serve de inspiração para um mecanismo robusto de detecção de anomalias, permitindo uma abordagem distribuída e adaptativa sem fase de treinamento.
- **Características**: Análise estatística é utilizada para identificar anomalias, apresentando alta taxa de detecção e baixa de falsos positivos. Cada célula dendrítica coleta sinais de múltiplas fontes e gera um output que é usado para correlacionar dados suspeitos.
- **Tipos de Sinais**: PAMP (indicador confiante de anomalia), Danger (indica potencial anomalia) e Safe (indica comportamento normal). A quantidade de cada sinal é ponderada para avaliar o estado do sistema monitorado.
- **Funcionamento**: A célula dendrítica amostra repetidamente sinais e antígenos, acumulando experiências até atingir um limite de migração, momento em que apresenta o antígeno. O processo é iterativo, com células substituídas após a apresentação.

- **Aplicações:** Um protótipo foi aplicado em problemas de classificação binária, realizando discriminação entre duas classes usando um timestamp como antígeno e combinações de características para as categorias de sinal.

## 9. Ant Colony Optimization

### 11.1 Introdução

- A eusocialidade evoluiu entre insetos, como as formigas e abelhas.
- As formigas possuem comportamento de forrageamento para encontrar caminhos curtos entre a comida e a colônia.
- A comunicação entre formigas ocorre através de sinais químicos e táteis, fundamentais para a organização das colônias.
- As formigas utilizam trilhas de feromônio para transmitir informações sobre caminhos, possibilitando a otimização na busca de soluções.

### 11.2 Ant-Colony Optimization

- ACO é uma abordagem metaheurística inspirada no comportamento das formigas para resolver problemas de otimização.
- As formigas artificiais constroem soluções de forma aleatória, guiadas por informações de feromônio de formigas anteriores.
- O algoritmo ACO inclui etapas de inicialização, construção de soluções, pesquisa local opcional, atualização de feromônios e critério de terminação.

#### 11.2.1 Algoritmo ACO Básico

- ACO é aplicado a problemas de otimização combinatória discretos (COPs).
- Formigas artificiais andam aleatoriamente em um grafo, coletando informações sobre soluções.
- O algoritmo ACO envolve a construção de soluções, avaliação, busca local e atualização de feromônios até a condição de término ser satisfeita.

- Um exemplo prático é dado com o Problema do Caixeiro Viajante (TSP) utilizando o sistema max-min de formigas.

### 11.2.2 ACO para Otimização Contínua

- ACO foi estendido para domínios contínuos, mantendo a estrutura conceitual do ACO.
- Aplicações em problemas contínuos e problemas mistos discretos-contínuos são mencionadas, demonstrando a versatilidade do ACO.

## 10. Bee Metaheuristics

### 12.1 Introdução

- O capítulo aborda algoritmos inspirados no comportamento das abelhas, destacando a importância da comunicação entre as abelhas para a execução de tarefas complexas, como a construção de colmeias e a colheita de pólen.
- Vários algoritmos de otimização são inspirados no comportamento das abelhas, incluindo o algoritmo Artificial Bee Colony (ABC).

### 12.2 Algoritmo da Colônia de Abelhas Artificial (ABC)

- O algoritmo ABC é utilizado principalmente em problemas de otimização contínua, simulando o comportamento da dança de uma colônia de abelhas durante a busca por alimentos.
- Apresenta um desempenho superior em problemas de otimização de funções quando comparado a outros métodos como GA, DE, ES e PSO, devido à sua capacidade de realizar buscas locais em cada iteração.

#### 12.2.1 Fluxograma do Algoritmo

- O algoritmo associa abelhas empregadas a fontes de alimentos, com cada fonte representando uma solução potencial.
- Na fase de inicialização, as fontes de alimentos são geradas aleatoriamente e associadas a abelhas empregadas.
- O processo é dividido em três fases: fase das abelhas empregadas, fase das abelhas observadoras e fase das abelhas exploradoras.
  - Na fase das abelhas empregadas, elas buscam novas fontes de alimento nas proximidades e compartilham informações através da dança.

- As abelhas observadoras escolhem uma fonte de alimento com base na qualidade, usando uma seleção por roleta.
- As abelhas exploradoras substituem as fontes que não melhoraram após um número determinado de tentativas.

### **12.2.2 Modificações no Algoritmo ABC**

- O algoritmo pode ser modificado para melhorar sua eficiência e eficácia em diferentes contextos de otimização.

### **12.2.3 Algoritmos ABC Discretos**

- Além da versão contínua, o algoritmo também pode ser adaptado para resolver problemas discretos, expandindo seu campo de aplicação.

## **12.3 Marriage in Honeybees Optimization**

- O algoritmo é inspirado no comportamento de acasalamento das abelhas e na fertilização.
- Simula a evolução de colônias de abelhas a partir de uma colônia solitária até uma colônia eusocial.
- A rainha realiza um voo de acasalamento, onde os zangões a seguem e a fertilizam no ar.
- O algoritmo utiliza três tipos de abelhas: rainha, zangões e operárias.
- A probabilidade de acasalamento é influenciada pela velocidade da rainha e pela diferença de fitness entre a rainha e o zangão.
- A energia e a velocidade da rainha são atualizadas após cada interação.
- Um operador de crossover é usado para criar as crias (broods).
- As operárias cuidam das crias e são responsáveis por uma fase de busca local no algoritmo.
- A substituição da rainha ocorre quando uma cria mais apta é identificada.

## **12.4 Bee Colony Optimization**

- Método de otimização estocástico inspirado no comportamento de forrageamento das abelhas.
- Uma população de abelhas artificiais busca a solução ótima para problemas combinatórios.
- Cada abelha gera soluções por meio de passos construtivos em várias iterações.
- Durante a fase de retorno, as abelhas compartilham suas soluções através da dança de waggle.

- A seleção das soluções a serem mantidas é probabilística, favorecendo soluções melhores.
- A metodologia é eficaz na exploração, mas menos eficiente na exploração aprofundada.
- O Weighted Bee Colony Optimization melhora a exploração ao permitir que abelhas busquem informações sobre fontes de alimento.

## 12.5 Other Bee Algorithms

- Diversas abordagens que simulam o comportamento das abelhas são discutidas, incluindo:
  - **Virtual Bee Algorithm:** Associa a população de abelhas a uma fonte de alimento e permite comunicação por dança de waggle.
  - **Bees Algorithm:** Realiza busca de vizinhança combinada com busca aleatória, aplicável a otimização contínua e combinatória.
  - **Beehive Algorithm:** Inspirado pela comunicação dentro da colmeia, aplicado em roteamento de redes.
  - **Bee Swarm Optimization:** Introduce padrões de voo modificados para equilibrar busca global e local, usando estratégias de comunicação.
  - **Bee Collecting Pollen Algorithm:** Algoritmo de otimização inspirado no comportamento de coleta de pólen.

### 12.5.1 Wasp Swarm Optimization

- Método estocástico inspirado no comportamento de colônias de vespas, focando na alocação de recursos com base no status social.
- Em situações de conflito, vespas soldados recebem mais alimento.
- A alocação de recursos é feita de forma estocástica, dependendo da força de cada opção, que é específica para cada aplicação.

# 11. Bacterial Foraging Algorithm

## 13.1 Introdução

- O comportamento de forrageamento social de **Escherichia coli** no intestino humano é explorado, destacando como as bactérias escalam gradientes de nutrientes.
- O forrageamento é modelado como um processo de otimização, onde as bactérias buscam maximizar a ingestão de energia.

- O algoritmo de forrageamento bacteriano é uma técnica estocástica baseada em populações, inspirada no comportamento de forrageamento de **E. coli**.
- O conceito de **quimiotaxia** é introduzido, descrito como a movimentação celular em resposta a gradientes químicos.
- Assumem-se vários aspectos do comportamento quimiotático das bactérias, como trajetórias em linha reta e mudanças de direção governadas por distribuições de probabilidade.

### 13.2 Algoritmo de Forrageamento Bacteriano

- O algoritmo é eficaz para problemas de otimização global, apesar de ser relativamente complexo e exigir mais tempo computacional.
- O processo é dividido em quatro estágios: **quimiotaxia, aglomeração, reprodução e eliminação e dispersão**.
  - **Quimiotaxia**: Cada bactéria produz uma solução iterativamente, com o tamanho do passo de movimento afetando a convergência e precisão do algoritmo.
  - **Aglomerção**: Bactérias se comunicam através de sinais químicos, atraindo outras para a localização do alimento.
  - **Reprodução**: Bactérias são classificadas e a metade mais apta é clonada, enquanto a outra metade é eliminada.
  - **Eliminação e dispersão**: Bactérias podem ser eliminadas ou dispersas aleatoriamente no espaço de busca, evitando a otimização local.
- O movimento quimiotático é modelado como uma combinação de **tumbles** (movimentos aleatórios) e **swims** (movimentos direcionados).
- A análise matemática do algoritmo é baseada em abordagens de descida de gradiente e a estabilidade é analisada usando teoremas de estabilidade de Lyapunov.

### 13.3 Algoritmos Inspirados em Fungos, Algas e Células Tumorais

- Algoritmos adicionais inspirados no comportamento de outros organismos, como fungos e algas, são discutidos, enfatizando a diversidade de estratégias de forrageamento e otimização.

## 12. Harmony Search

### 14.1 Introdução

Harmony Search é uma técnica de otimização metaheurística baseada em população, inspirada no processo de improvisação musical. O algoritmo utiliza conceitos como músicos, notas e harmonias, onde os músicos representam variáveis de decisão e as notas são os valores dessas variáveis. A memória de harmonia armazena vetores de solução, permitindo a geração de novas soluções através da improvisação. O algoritmo é iterativo, buscando melhorar continuamente as soluções com base em parâmetros como a taxa de consideração da memória (PHMCR) e a taxa de ajuste de pitch (PAR). A abordagem é eficaz em problemas de otimização contínuos e discretos.

## 14.2 Algoritmo de Harmony Search

O algoritmo básico de Harmony Search consiste em quatro etapas principais:

1. **Inicialização da Memória de Harmonia:** Parâmetros de controle como tamanho da memória (HMS), taxa de consideração (PHMCR), taxa de ajuste (PAR) e número máximo de iterações (NI) são definidos. Soluções viáveis são geradas aleatoriamente e armazenadas.
2. **Improvisação de Nova Harmonia:** Uma nova solução é criada escolhendo valores da memória de harmonia ou gerando aleatoriamente, com ajustes de pitch aplicados conforme necessário.
3. **Atualização da Memória de Harmonia:** A nova solução é incluída na memória, substituindo a pior solução se for melhor.
4. **Repetição até Critério de Término:** O processo se repete até que uma solução aceitável seja encontrada.

## 14.3 Variantes do Harmony Search

Várias variantes e melhorias do Harmony Search foram desenvolvidas para otimizar o desempenho:

- **Harmony Search Melhorada:** Ajusta dinamicamente os parâmetros PAR e BW ao longo das iterações.
- **Global-best Harmony Search:** Hibridiza conceitos do PSO, modificando o operador de ajuste de pitch para melhorar a taxa de convergência.
- **Self-adaptive Global-best Harmony Search:** Introduz um novo esquema de improvisação e ajuste adaptativo de parâmetros.
- **Métodos de Otimização Local:** Usam técnicas como programação quadrática sequencial e divisões em submemórias.
- **Dynamic Harmony Search:** Ajusta todos os parâmetros dinamicamente, visando uma convergência mais rápida.

Essas variantes buscam aprimorar a exploração e a convergência do algoritmo, evitando soluções locais ótimas e aumentando a eficácia geral do Harmony Search.

## 13. Swarm Intelligence

Métodos de otimização inspirados na natureza podem ser agrupados em abordagens evolutivas e métodos de inteligência de enxame. Os métodos de inteligência de enxame são populares devido à sua simplicidade, fácil adaptação a problemas e eficácia na resolução de problemas complexos de otimização.

### 15.1 Glowworm-Based Optimization

Os glowworms emitem luz bioluminescente para atrair parceiros ou presas, com a intensidade da luz proporcional à quantidade de luciferina que possuem.

#### 15.1.1 Glowworm Swarm Optimization

Este algoritmo é inspirado no comportamento natural dos glowworms e é projetado para calcular múltiplos ótimos de funções multimodais. Os glowworms se movem em direção aos vizinhos com maior emissão de luciferina, e o processo é dividido em três fases: atualização de luciferina, fase de movimento e atualização da faixa de vizinhança.

#### 15.1.2 Firefly Algorithm

Baseado na bioluminescência das fireflies, esse algoritmo também busca a otimização contínua multimodal. Fireflies são atraídas por outras com base em sua intensidade de luz, que está ligada à função objetivo. O movimento é influenciado pela intensidade da luz e pela distância entre fireflies.

### 15.2 Group Search Optimization

Inspirado no comportamento de busca de animais como leões e lobos, este algoritmo é baseado no modelo de produtor-escarador. O grupo consiste em três tipos de membros: produtores, escaradores e membros dispersos. O produtor busca novas posições e os escaradores seguem a posição do produtor, enquanto os membros dispersos evitam mínimos locais.

### 15.3 Shuffled Frog Leaping

Este método combina a estratégia de busca local de sapos com um componente de mistura, permitindo a exploração do espaço de busca de forma eficiente. A abordagem envolve saltos para as melhores posições encontradas, com a aleatoriedade introduzida para evitar armadilhas em mínimos locais.

#### 15.4 Collective Animal Search

Este método considera o comportamento coletivo de animais em busca de recursos. Utiliza mecanismos de interação social e comunicação para melhorar a eficiência da busca. O algoritmo aproveita a diversidade de comportamentos de busca entre os membros do grupo, otimizando a exploração do espaço de soluções.

#### 15.5 Cuckoo Search

- **Definição:** Algoritmo metaheurístico para otimização global, inspirado no comportamento de parasitismo das aves cucos.
- **Comportamento:** Os cucos colocam seus ovos em ninhos de aves hospedeiras. Se o ovo for descoberto, o ninho pode ser abandonado.
- **Regras do Algoritmo:**
  1. Cada cuco coloca um ovo por vez em um ninho aleatório.
  2. Ninhos com ovos de alta qualidade são preservados.
  3. Um percentual de ninhos ruins é abandonado e novos são criados.
- **Levy Flights:** Método de geração de novas soluções, caracterizado por movimentos aleatórios que evitam mínimos locais.
- **Parâmetros:** Probabilidade de abandono de ninhos ( $P_a$ ), tamanho do passo ( $\alpha$ ) e comprimento do passo aleatório ( $\lambda$ ).
- **Vantagens:** Oferece soluções melhores que outros métodos como PSO e GA.

#### 15.6 Bat Algorithm

- **Definição:** Algoritmo de otimização inspirado na ecolocalização dos morcegos, que usam sons para detectar e localizar presas.
- **Comportamento:** Os morcegos emitem pulsos sonoros e ajustam a frequência e a intensidade com base na proximidade do alvo.
- **Parâmetros:** Taxa de pulso e intensidade, que controlam a exploração do espaço de busca e a aceitação de soluções.
- **Etapas do Algoritmo:**
  1. Inicialização de parâmetros e geração de uma população de soluções.
  2. Operação de variação para mover os morcegos no espaço de busca.
  3. Busca local para melhorar soluções atuais.
  4. Substituição de soluções com base em probabilidades.

- **Resultados:** A combinação de busca local/global torna o Bat Algorithm eficiente para encontrar soluções.

## 15.7 Swarm Intelligence Inspired by Animal Behaviors

- **Conceito Geral:** Algoritmos que simulam o comportamento coletivo de animais, utilizando estratégias de otimização baseadas na natureza.

### 15.7.1 Social Spider Optimization

- **Descrição:** Algoritmo que modela o comportamento social de aranhas em busca de soluções otimizadas, enfatizando a colaboração.

### 15.7.2 Fish Swarm Optimization

- **Descrição:** Baseado em como cardumes de peixes se movem e se organizam, este algoritmo explora o espaço de solução de forma cooperativa e adaptativa.

### 15.7.3 Krill Herd Algorithm

- **Descrição:** Modela o comportamento dos krills em cardumes, utilizando interações sociais e alimentação para guiar a busca por soluções.

### 15.7.4 Cockroach-Based Optimization

- **Descrição:** Inspirado no comportamento de baratas, enfatiza a resiliência e adaptabilidade na busca de soluções otimizadas em ambientes complexos.

### 15.7.5 Seven-Spot Ladybird Optimization

- Baseia-se no comportamento de forrageamento da joaninha sete-pontos.
- Semelhança com o PSO, utilizando lbest e gbest para busca.
- O espaço é dividido em áreas (patches) para pesquisa intensiva (lenta) e extensiva (rápida).
- A joaninha utiliza feromônios em diferentes estágios de vida para localizar presas.
- A busca intensa ocorre após encontrar presas via busca extensiva, utilizando antenas e palpos para detectar movimento.

### 15.7.6 Monkey-Inspired Optimization

- Inspirado no comportamento de macacos ao escalar árvores para buscar alimento.
- O algoritmo representa ramos da árvore como perturbações entre soluções viáveis.

- Inclui métodos como Spider Monkey Optimization e Monkey King EA, que se destacam em robustez e precisão de otimização.
- O Monkey Algorithm simula processos de escalada, observação e saltos para evitar máximos locais.

### 15.7.7 Migrating-Based Algorithms

- **Self-Organizing Migrating Algorithm:** Baseado no comportamento social de indivíduos cooperativos, converge para o ótimo global em laços de migração.
- **Differential Search Algorithm:** Imita a migração de seres vivos em busca de habitats com mais recursos.
- Ambas as abordagens utilizam operadores de mutação para garantir diversidade na população.

### 15.7.8 Other Methods

- **Society and Civilization Algorithm:** Focado em problemas de otimização com restrições, usando interações sociais.
- **Gray Wolf Optimizer:** Simula a hierarquia e estratégias de caça de lobos cinzentos.
- **Diversos algoritmos:** Dog Group Wild Chase, Bird Mating Optimizer, Raven Roosting Optimization, Dolphin Partner Optimization, entre outros, inspirados em comportamentos de diferentes espécies animais para otimização.

### 15.8 Plant-Based Metaheuristics

- **Invasive Weed Optimization:** Inspirado em invasões de plantas daninhas, envolve inicialização, reprodução, dispersão espacial e exclusão competitiva.
- **Flower Pollination Algorithm:** Simula o comportamento de polinização de flores, combinando buscas locais e globais com voos de Levy.

### 15.9 Other Swarm Intelligence-Based Metaheuristics

- **Moth Flame Optimization:** Baseado na estratégia de navegação de mariposas, imita o movimento em busca de alimento em relação à luz.
- Vários métodos utilizam comportamentos coletivos e sociais de organismos para otimização, adaptando-se a diferentes problemas de pesquisa.

## 14. Biomolecular Computing

### 16.1 Introdução

- Estudo do potencial das moléculas biológicas para computação.
- Foco em DNA e computação por membrana como técnicas de computação natural.
- Organismos multicelulares são compostos por células que funcionam em paralelo.
- Células procarióticas (bactérias e arqueias) e eucarióticas (animais, plantas e fungos) têm estruturas funcionais distintas.
- P systems (sistemas de membrana) como modelos de computação e a importância das interações entre módulos moleculares.

#### 16.1.1 Redes Bioquímicas

- Redes bioquímicas surgem de interações moleculares mediadas por proteínas dentro das células.
- Três tipos de redes bioquímicas:
  - **Rede Metabólica:** Interações auto-organizadoras de reações mediadas por enzimas, formando caminhos metabólicos.
  - **Rede Genética:** Interações regulatórias entre genes, capturando como eles regulam a expressão proteica.
  - **Rede de Sinalização:** Reações mediadas por proteínas que transportam sinais químicos no ambiente celular.
- As redes são acopladas, com a rede genética modulando o comportamento das redes metabólica e de sinalização.
- Redes metabólicas e genéticas artificiais modelam comportamentos auto-organizadores e interações regulatórias.

#### 16.2 Computação em DNA

- Trabalho pioneiro de Adleman usando moléculas de DNA para resolver o Problema do Caixeiro Viajante (TSP) em sete cidades.
- Computação em DNA como uma metaheurística de otimização utilizando características biomoleculares do DNA, como hibridização e paralelismo massivo.
- Uso de tubos de solução e eletroforese em gel para produção e análise de sequências de DNA.
- Estrutura de DNA de cadeia dupla, aproveitada para correção de erros e capacidade de realizar cálculos paralelos.

- Exemplos de circuitos e portas lógicas moleculares, como elementos genéticos bacterianos para criar portas lógicas aproximando funções booleanas.
- Demonstração de portões lógicos de deoxiriboenzimas e amplificação de sinais em circuitos DNA-based.
- Demonstração de interruptores lógicos de ácidos nucleicos projetados, ampliando para portas lógicas complexas.

### 16.3 Membrane Computing

**Introdução:** Membrane computing é um modelo de computação paralelo e distribuído que abstrai modelos formais de computação a partir da estrutura e funcionamento das células vivas. Esses sistemas são inspirados nas interações bioquímicas que ocorrem dentro das células.

#### 16.3.1 Cell-Like P System:

- Estrutura básica composta por múltiplas membranas organizadas hierarquicamente.
- Cada compartimento contém um conjunto finito de objetos e regras que realizam operações de transformação e comunicação.
- As regras são aplicadas de maneira não determinística e paralela, permitindo a evolução do sistema a partir de configurações iniciais.
- O resultado da computação é representado pela quantidade de objetos em uma membrana específica.

#### 16.3.2 Computing by P System:

- Um P system é definido como uma tupla que inclui um alfabeto finito, a estrutura das membranas e os conjuntos de regras de processamento.
- A estrutura é representada por uma sequência de parênteses que descreve a hierarquia das membranas.
- A computação ocorre através de transições entre configurações, utilizando um modo de paralelismo máximo. A computação é bem-sucedida quando atinge uma configuração terminal.

#### 16.3.3 Other P Systems:

- Além dos sistemas de P semelhantes a células, existem sistemas de P semelhantes a tecidos, neurais, metabólicos e populacionais.

- Esses sistemas variam na organização hierárquica e nos componentes básicos, como neurônios ou células.
- As interações entre células em sistemas de P semelhantes a tecidos simulam a comunicação intercelular, enquanto os sistemas neurais podem ser estruturados como grafos direcionados.

#### 16.3.4 Membrane-Based Optimization:

- Sistemas de P podem ser utilizados para resolver problemas NP-difíceis em tempo linear ou polinomial, explorando a paralelização massiva e o espaço computacional exponencial.
- A computação em sistemas de P pode ser ampliada através de processos biológicos como mitose e autopoiese, permitindo a criação exponencial de novas membranas.

## 15. Quantum Computing

### 17.1 Introdução

Computação quântica é baseada na teoria da mecânica quântica, que descreve o comportamento de partículas em nível atômico. Os algoritmos quânticos têm um desempenho exponencialmente mais rápido que os algoritmos tradicionais. A proposta de computadores quânticos surgiu na década de 1980, e este tópico introduz algoritmos quânticos básicos e algoritmos híbridos metaheurísticos baseados em quântica.

### 17.2 Fundamentos

A computação quântica se baseia no princípio da superposição de estados, que permite que um sistema esteja em uma combinação de todos os seus estados possíveis simultaneamente. Isso possibilita o processamento paralelo para otimizar funções. Os operadores unitários atuam em vetores de estado discretos, permitindo a solução de problemas de otimização através de uma busca exaustiva por todas as soluções possíveis. O entrelaçamento e o paralelismo proporcionam acelerações nas computações e comunicações quânticas. O entrelaçamento permite que partículas correlacionadas exibam propriedades que não podem ser explicadas classicamente.

#### 17.2.1 Algoritmo de Busca de Grover

O algoritmo de Grover é usado para buscar em um banco de dados não estruturado, amplificando os coeficientes da superposição dos elementos que correspondem às soluções, enquanto reduz os outros. Essa técnica oferece um ganho de velocidade,

permitindo a busca em um banco de dados com  $N$  entradas em tempo  $O(\sqrt{N})$ , comparado ao tempo  $O(N)$  de uma busca clássica. Grover's pode encontrar um elemento marcado que satisfaça uma condição específica com um número mínimo de avaliações.

### **17.3 Métodos Híbridos**

Os métodos híbridos combinam técnicas clássicas com algoritmos quânticos, buscando aproveitar os benefícios de ambos. Essa seção explora duas categorias principais de algoritmos híbridos inspirados em quântica.

#### **17.3.1 Algoritmos Evolutivos Inspirados em Quântica**

Esses algoritmos utilizam conceitos quânticos para melhorar algoritmos evolutivos clássicos, como a capacidade de explorar o espaço de soluções mais rapidamente através da superposição.

#### **17.3.2 Outros Algoritmos Híbridos Inspirados em Quântica**

##### **Otimização por Enxame de Partículas Inspirada em Quântica (Quantum PSO):**

- O Quantum PSO emprega uma técnica de busca probabilística, transferindo o espaço de busca clássico para o espaço quântico.
- Durante o processo de iteração, a posição da população é inicialmente configurada. As partículas então buscam soluções de acordo com o princípio de onda das partículas, onde a posição de cada partícula é determinada probabilisticamente.
- Para avaliar cada partícula, é necessário aprender sua posição exata, o que permite calcular a aptidão (fitness) das partículas. A medição da posição é realizada utilizando métodos de Monte Carlo.

##### **Quantum PSO para Otimização Binária:**

- Este método utiliza qubits para representar probabilisticamente uma partícula, eliminando a necessidade de atualizar a velocidade, característica tradicional do PSO.
- No lugar dos fatores de peso de inércia e coeficientes de aceleração, apenas o ângulo de rotação é necessário para modificar a posição das partículas.
- Um portão de rotação proposto, que inclui um portão de rotação de coordenadas, é utilizado para atualizar os qubits, juntamente com uma abordagem dinâmica para determinar a magnitude do ângulo de rotação.

##### **Busca Tabu Inspirada em Quântica:**

- Este algoritmo combina o Algoritmo Evolutivo Quântico (Quantum EA) com a busca tabu.

- Integra estratégias de diversificação e intensificação do Quantum EA, modificado por um novo estado quântico de atualização.
- Essa combinação não apenas previne a convergência prematura, mas também facilita a obtenção mais rápida da solução ótima.
- Uma população de estados qubit é mantida, representando cadeias binárias. O processo de medição dos qubits é uma operação probabilística que aumenta a diversificação, enquanto um portão de rotação quântica é usado para buscar regiões atrativas, aumentando a intensificação.

#### **Algoritmo de Busca Gravitacional Binária Inspirado em Quântica:**

- Combina o algoritmo de busca gravitacional com princípios de computação quântica, proporcionando uma ferramenta de otimização robusta para problemas binários.
- Neste algoritmo, o ângulo de rotação determina a nova posição do agente, substituindo a atualização de aceleração típica do algoritmo de busca gravitacional.
- A magnitude do ângulo de rotação é usada para substituir a massa gravitacional, facilitando uma abordagem mais eficaz para a otimização.

#### **Algoritmo de Busca Gravitacional Quântico para Otimização Contínua:**

- Além das versões binárias, também existem implementações que aplicam princípios quânticos à busca gravitacional para otimização contínua, ampliando ainda mais a gama de problemas que podem ser resolvidos.

## **16. Metaheuristics Based on Sciences**

### **18.1 Busca Baseada nas Leis de Newton**

- **Gravitational Search Algorithm:** Técnica de otimização estocástica que utiliza a metáfora da interação gravitacional. Os agentes são representações de soluções que possuem massa, posição, massa gravitacional ativa e passiva. Soluções melhores (maiores massas) se movem mais lentamente, promovendo uma exploração eficaz do espaço de busca. O algoritmo é comparável ao PSO em suas equações de atualização de posição e velocidade, mas utiliza a lei da gravidade para essa atualização. Um algoritmo binário de busca gravitacional também altera

probabilisticamente as coordenadas.

- **Artificial Physics Optimization:** Algoritmo global baseado em populações, onde cada entidade é tratada como um indivíduo físico com atributos de massa, posição e velocidade. A atração é determinada pela relação entre massa e valor da função objetivo, fazendo com que indivíduos se movam em direção a regiões com melhores valores de fitness.
- **Central Force Optimization:** Metaheurística determinística que simula a movimentação de massas sob a influência da gravidade. Cada execução com os mesmos parâmetros resulta em resultados idênticos, e a atualização da aceleração depende da posição e fitness dos agentes.
- **Vibration Damping Optimization:** Algoritmo que utiliza o conceito de amortecimento de vibrações, aplicado na otimização de problemas como o agendamento de máquinas em paralelo.

## 18.2 Busca Baseada nas Leis Eletromagnéticas

- **Charged System Search:** Algoritmo de otimização multiagente que usa a lei de Coulomb para modelar a interação entre partículas carregadas. A movimentação dos agentes é baseada em seus valores de fitness e distâncias.
- **Electromagnetism-Like Algorithm:** Algoritmo inspirado na teoria do eletromagnetismo, onde partículas carregadas exercem forças atrativas ou repulsivas. Cada solução é representada por uma partícula que tem um vetor de posição e uma carga relacionada ao seu valor da função objetivo.
- **Ions Motion Optimization:** Baseado no comportamento de íons, onde íons de cargas semelhantes se repelem e de cargas opostas se atraem. As soluções são representadas por íons que se movem em direção aos melhores íons opostos.
- **Magnetic Optimization Algorithm:** Inspirado pela teoria do campo magnético, onde partículas magnéticas se movem em direção a outras partículas mais adequadas. A estrutura em rede ajuda na exploração local antes de buscar a melhor solução global.
- **Optimization by Optics:** Ray optimization modela agentes como raios de luz que se movem com base nas leis de refração. Cada pico na função a ser otimizada é tratado

como um espelho convexo e cada vale como um espelho côncavo.

### 18.3 Busca Baseada em Princípios de Energia Térmica

- **States of Matter Search:** Inspirado pelos estados da matéria, este algoritmo emula moléculas em diferentes estados (gás, líquido e sólido) para simular exploração e exploração. Começa com uma fase gasosa (exploração) e ajusta as intensidades até chegar à fase sólida (exploração pura).
- **Heat Transfer Search:** Algoritmo de otimização baseado nas leis da termodinâmica e transferência de calor, onde agentes interagem para alcançar um estado de equilíbrio térmico através de condução, convecção e radiação.
- **Kinetic Gas Molecule Optimization:** Baseado na teoria cinética dos gases, onde moléculas se atraem através de forças elétricas fracas. Este algoritmo é eficaz em convergir rapidamente para mínimos globais, melhorando a precisão em comparação com outras abordagens como PSO e Gravitational Search Algorithm. As moléculas exploram o espaço de busca até encontrarem a região com menor temperatura e energia cinética.

### 18.4 Pesquisa Baseada em Fenômenos Naturais

#### 18.4.1 Pesquisa Baseada em Fluxos de Água

- **Algoritmo de Gotas de Água Inteligentes:** Inspirado no comportamento das gotas de água em rios, este algoritmo otimiza o caminho que uma gota toma até o destino, removendo solo ao longo do caminho. O caminho ideal é aquele com menor resistência do solo.
- **Algoritmo do Ciclo da Água:** Simula o movimento da água das fontes para os oceanos. O oceano representa a melhor solução, enquanto os rios representam soluções boas, mas não as melhores. O processo de evaporação e precipitação corresponde à fase de exploração do algoritmo, evitando que ele fique preso em ótimos locais.
- **Algoritmo do Grande Dilúvio:** Baseado em uma solução única, este algoritmo permite a aceitação de soluções inferiores durante a busca. A aceitação depende de um critério chamado nível, permitindo explorar soluções menos ótimas para escapar de ótimos locais.

- **Algoritmo de Crescimento de Plantas:** Inspirado nos mecanismos de crescimento das plantas, esse algoritmo envolve etapas como a germinação da semente e o crescimento em direção à fonte de luz, incorporando conceitos de fototropismo e dominância apical.

#### 18.4.2 Pesquisa Baseada em Cosmologia

- **Algoritmo Big Bang Big Crunch:** Este método otimiza globalmente, inspirado pela teoria da evolução do universo. A fase de Big Bang dispersa soluções aleatoriamente, enquanto a fase de Big Crunch ordena essas soluções, convergindo para um ponto ideal.
- **Busca Espiral ou Vórtice:** Baseado em fenômenos naturais como galáxias e furacões, o algoritmo cria trajetórias em espiral para os agentes de busca, permitindo uma exploração diversificada e uma exploração intensa em fases específicas.
- **Otimização Baseada em Nuvens:** Simula o comportamento das nuvens na atmosfera. As regiões do espaço de busca têm valores de umidade e pressão do ar, e as nuvens se movem de áreas de alta pressão para baixa pressão, representando a busca por soluções.
- **Busca Relâmpago:** Inspirado pelo fenômeno do relâmpago, utiliza projetéis para modelar comportamentos de busca. O algoritmo explora soluções em um espaço restrito, utilizando um comportamento aleatório para aumentar a diversidade.
- **Otimização Dirigida pelo Vento:** Um algoritmo baseado em população que simula o movimento de pequenas parcelas de ar. O movimento é influenciado por forças gravitacionais e outras, permitindo uma busca eficiente no espaço de soluções.

#### 18.4.3 Otimização Baseada em Buracos Negros

- **Otimização Baseada em Buracos Negros:** Um algoritmo que simula o comportamento de buracos negros, onde soluções candidatas (estrelas) são atraídas para um buraco negro, que representa a melhor solução. Estrelas que se aproximam demais do buraco negro são “absorvidas” e substituídas por novas soluções geradas aleatoriamente.
- **Otimização de Buracos Negros de Massa Estelar:** Uma técnica que aproveita a gravidade dos buracos negros, mostrando melhor desempenho em relação a outras

técnicas como PSO e busca de cucos.

- **Otimização do Multiverso:** Inspirada em buracos brancos, buracos negros e wormholes, esta abordagem utiliza modelos matemáticos para realizar exploração, exploração local e busca local.

## 18.5 Ordenação

A ordenação e a pesquisa são fundamentais para o processamento de dados em computação. Existem diversos algoritmos de ordenação:

- **Beadsort:** Baseado na movimentação de contas em varas paralelas, semelhante a um ábaco.
- **Rainbow Sort:** Baseia-se nos conceitos de refração e dispersão da luz.
- **Spaghetti Sort:** Representa os números a serem ordenados através de espaguetes não cozidos.
- **Centrifugal Sort:** Utiliza a densidade dos líquidos para representar os números, onde a aceleração gravitacional ajuda na ordenação.
- **Friction-based Sorting:** Este método associa a ordenação ao atrito, onde cada elemento tem uma “resistência” que influencia sua posição final na sequência.

## 18.6 Algorithmic Chemistries

Os algoritmos de química artificial simulam processos químicos reais, atribuindo coeficientes cinéticos, definindo representações moleculares e focando em estados eficientes de conservação de energia. Essas químicas são vistas como modelos computacionais. Um modelo de energia simples pode guiar o uso de recursos de forma eficiente, mantendo as moléculas dentro de limites razoáveis de comprimento e permitindo que o algoritmo se comporte termodinamicamente e cineticamente similar à química real.

### 18.6.1 Chemical Reaction Optimization

A otimização de reações químicas (CRO) é uma metaheurística baseada em populações para problemas de otimização combinatória, inspirada nas interações entre moléculas durante reações químicas. O objetivo é encontrar produtos com a menor energia potencial. O modelo CRO se baseia na conservação de energia, onde as moléculas possuem

estruturas que tendem a ter energia potencial cada vez menor. Os agentes manipulados no CRO são moléculas, que possuem estrutura molecular, energia potencial e cinética, número de colisões e valores mínimos.

As reações químicas ocorrem na forma de colisões e são classificadas em quatro tipos: colisão ineficaz de parede, decomposição, colisão intermolecular ineficaz e síntese. A inicialização do algoritmo envolve a geração aleatória de moléculas, seguida pela determinação de suas energias potenciais e atribuição de energias cinéticas iniciais. Durante as iterações, um evento de reação química é escolhido aleatoriamente, e a solução é mantida se for melhor que as anteriores. A CRO é modelada como uma cadeia de Markov, mostrando que pode convergir para uma solução ótima global com alta probabilidade.

### **Chemical Reaction Networks**

Redes de reações químicas modelam formalmente a química em soluções bem misturadas, sendo utilizadas para descrever o processamento de informações em redes celulares. Elas demonstram a capacidade de computação Turing universal com uma probabilidade de erro pequena. Redes que garantem convergência em respostas corretas decidem apenas predicados semilineares. A complexidade de tempo do algoritmo é inferior à quadrática em relação ao número total de moléculas de entrada.

---

### **18.7 Biogeography-Based Optimization**

A biogeografia estuda a distribuição geográfica de organismos ao longo do tempo. Modelos matemáticos descrevem migração, especiação e extinção de espécies. No contexto da otimização, a Biogeography-Based Optimization (BBO) é um algoritmo estocástico baseado em populações, onde um conjunto de soluções é chamado de arquipélago e cada solução é um habitat.

Na BBO, índices de adequação do habitat (HSI) determinam a adequação de uma solução. O algoritmo utiliza operadores de migração e mutação para compartilhar informações entre soluções. A taxa de imigração e emigração de cada habitat é calculada com base em sua adequação. Um habitat com alta adequação tem uma taxa de emigração maior e uma taxa de imigração menor. O fluxo de operações na BBO envolve a geração inicial de soluções, avaliação de adequação e iterações onde as taxas de imigração e emigração são calculadas continuamente.

---

## **18.8 Methods Based on Mathematical Concepts**

Métodos que utilizam conceitos matemáticos em otimização buscam melhorar a eficiência dos algoritmos baseados em natureza, integrando princípios matemáticos fundamentais para otimizar a busca por soluções. Esses métodos visam não apenas aumentar a eficácia dos algoritmos, mas também garantir uma convergência mais robusta e rápida em comparação com abordagens puramente heurísticas. A utilização de conceitos como álgebra linear, análise de variáveis e teoria da probabilidade permite a construção de algoritmos mais sofisticados, que podem lidar com problemas complexos e dinâmicos, oferecendo soluções mais precisas e confiáveis.

### **18.8.1 Opposition-Based Learning**

O aprendizado baseado em oposição é uma abordagem inovadora que utiliza soluções opostas para melhorar a eficiência do processo de otimização. Essa técnica, que se destaca por sua originalidade, é aplicada em diversos algoritmos de otimização, incluindo aqueles inspirados na natureza, para explorar o espaço de soluções de forma mais abrangente. A ideia central é gerar uma solução oposta a cada solução encontrada, o que permite uma avaliação mais ampla das possibilidades disponíveis.

Essa técnica busca reduzir a convergência para mínimos locais, que é um desafio comum em otimização, e aumenta a probabilidade de encontrar soluções globais mais eficientes. O aprendizado baseado em oposição promove uma exploração mais dinâmica do espaço de busca, ajudando a evitar a estagnação em regiões pouco promissoras. Em muitas aplicações práticas, essa abordagem tem mostrado melhorias significativas em desempenho e eficiência, tornando-se uma ferramenta valiosa em algoritmos de otimização modernos.

## **17. Memetic Algorithms**

### **19.1 Introdução**

O conceito de meme, introduzido por Dawkins em 1976, refere-se a unidades de informação cultural que se propagam entre indivíduos por meio da imitação. Memes são comparáveis a genes na evolução cultural, atuando como blocos de construção de informações que são adaptadas e refinadas ao longo do tempo. Memetic computation, portanto, combina busca global baseada em populações com melhorias locais, equilibrando exploração e exploração do espaço de busca. Um aspecto crucial é a identificação de representações meméticas

adequadas, onde a seleção memética determina a imitação e a variação memética se refere a como os memes são assimilados e gerados.

### **19.2 Algoritmos Culturais**

Os algoritmos culturais são uma abordagem computacional que utiliza o conhecimento cultural para melhorar o desempenho evolutivo, dividindo o processo em dois espaços: espaço populacional e espaço de crenças. O espaço populacional contém soluções possíveis, enquanto o espaço de crenças armazena conhecimentos que influenciam a evolução da população, permitindo que os indivíduos aprendam com os sucessos e fracassos dos outros. Este sistema dual ajuda a evitar a convergência prematura, permitindo que o conhecimento armazenado, como conhecimento situacional e histórico, guie a evolução. Operações como aceitação e influência são realizadas para comunicar experiências entre os dois espaços.

### **19.3 Algoritmos Meméticos**

Os algoritmos meméticos, também conhecidos como busca local genética, são uma forma de algoritmo cultural que incorpora refinamento local durante a evolução. Inspirados na evolução cultural, eles combinam a adaptação evolutiva de uma população com o aprendizado individual, sendo mais rápidos que algoritmos genéticos simples. O algoritmo é considerado uma estratégia Lamarckiana, onde a evolução e o aprendizado se entrelaçam. Embora apresentem características da evolução cultural, faltam a transmissão e variação de memes típicas de um sistema evolutivo verdadeiro.

#### **19.3.1 Algoritmos Meméticos Baseados em Simplicidade**

Uma variante dos algoritmos meméticos é o uso de um quadro probabilístico que analisa quando favorecer a evolução em comparação ao aprendizado individual. Essa estrutura busca otimizar a probabilidade de alcançar um ótimo global, ajustando a intensidade de aprendizado de cada indivíduo. Algumas classes de algoritmos meméticos incorporam transmissão e seleção de memes em seu design, onde o material memético é parte do genótipo e refinamentos locais são realizados a partir de memes decodificados. Mecanismos de recompensa são utilizados para decidir quais memes devem ser selecionados para melhorias futuras.

### **19.4 Aplicação: Buscando Sequências de Baixa Autocorrelação**

Um exemplo prático da aplicação de algoritmos meméticos é a busca por sequências de baixa autocorrelação. Nesse contexto, a eficiência dos algoritmos meméticos pode ser explorada para identificar padrões de dados com baixa dependência entre os elementos, demonstrando a eficácia da abordagem memética em resolver problemas complexos. Essa aplicação ilustra como a combinação de busca global e refinamento local pode ser utilizada para otimizar processos e alcançar soluções inovadoras.

## 18. Tabu Search and Scatter Search

### 20.1 Tabu Search

- **Definição:** Tabu search é um método estocástico de otimização global baseado em solução única, originalmente desenvolvido para problemas de otimização combinatória (COP) e, posteriormente, estendido para otimização contínua. É uma técnica de busca local que imita a estrutura da memória humana para melhorar a tomada de decisões.
- **Conceito de Tabu:** O termo "tabu" refere-se a movimentos ou soluções proibidas. O método utiliza estratégias e informações aprendidas para evitar revisitar soluções anteriores e prevenir a convergência prematura para ótimos locais.
- **Funcionamento:** O algoritmo começa com uma solução inicial, gerando soluções vizinhas a partir dela, utilizando uma lista de tabu que registra estados previamente visitados. As melhores soluções vizinhas são selecionadas, mesmo que sejam piores que a atual, a fim de evitar ficarem presas em ótimos locais.
- **CrITÉrios de AspiraÇão:** Permitem que soluções tabus sejam escolhidas se forem melhores que a melhor solução encontrada até o momento, permitindo a exploração de soluções que de outra forma seriam descartadas.
- **Exemplo Prático:** Em um exemplo aplicado ao Problema do Caixeiro Viajante (TSP) com 52 cidades, a busca tabu produziu uma solução com comprimento de rota de aproximadamente 7783 após 100 iterações.
- **Paralelismo e Diversificação:** Modelos de busca tabu paralela foram desenvolvidos para acelerar a exploração do espaço de soluções. Além disso, técnicas de diversificação são incorporadas para alternar entre a busca tabu e fases de perturbação, explorando novas regiões do espaço de busca.

#### 20.1.1 Iterative Tabu Search

- **Definição:** A Iterative Tabu Search combina busca tabu com operadores de perturbação, evitando que o algoritmo fique preso em ótimos locais. O algoritmo modifica a solução atual por meio de perturbações e, em seguida, aplica a busca tabu para aprimorar a nova solução.

- **Funcionamento:** Após cada perturbação, se a nova solução obtida é melhor, ela substitui a anterior; caso contrário, a busca continua com a solução perturbada.
- **Exemplo Prático:** No caso do TSP de Berlin52, a busca tabu iterativa convergiu para uma solução com comprimento de rota de 7782.9844 em 100 iterações, destacando a necessidade de estratégias mais sofisticadas para escapar de mínimos locais.

## 20.2 Scatter Search

- **Definição:** Scatter search é um algoritmo metaheurístico baseado em população, que combina soluções para gerar novas. Inicia-se a partir de soluções obtidas por técnicas heurísticas, selecionando um subconjunto das melhores soluções para formar novas soluções.
- **Mecanismo de Elitismo:** Inclui um mecanismo de elitismo, onde as melhores soluções são mantidas e utilizadas para a geração de novas soluções. Este processo é repetido até que uma solução satisfatória seja alcançada.
- **Aplicações:** Scatter search é utilizado para resolver problemas de otimização combinatória e não lineares, beneficiando-se do uso de memória adaptativa e mecanismos que exploram informações anteriores.

## 20.3 Path Relinking

- **Definição:** Path relinking é uma forma generalizada da scatter search, que utiliza a exploração da memória para melhorar a busca por soluções. Esta técnica combina soluções que já foram encontradas para criar novas, aproveitando a relação entre elas.
- **Funcionamento:** As soluções são unidas em um caminho que religa as melhores soluções, permitindo que o algoritmo explore soluções que podem não ter sido visitadas diretamente.
- **Benefícios:** O uso de path relinking permite uma exploração mais ampla do espaço de soluções, aumentando as chances de encontrar ótimos globais em problemas complexos.

## 19. Search Based on Human Behaviors

Este capítulo aborda metaheurísticas de busca inspiradas em comportamentos humanos durante a resolução de problemas. Abaixo estão os subcapítulos e seus conteúdos específicos.

### 21.1 Seeker Optimization Algorithm

O **Seeker Optimization Algorithm** é uma metaheurística populacional voltada para problemas de otimização com parâmetros reais, simulando o ato de busca dos humanos. Funciona com um conjunto de soluções denominado **equipe de busca**, onde os indivíduos são chamados de **seekers**. A direção da busca é escolhida com base no gradiente empírico, enquanto o comprimento do passo é decidido por uma regra fuzzy.

- **Direção de busca:** Determinada por uma combinação aleatória do comportamento egoísta, altruísta e proativo do seeker.
- **Egoistic direction:** Define-se pela diferença entre a melhor posição pessoal e a posição atual do seeker.
- **Altruistic direction:** Determinada pela diferença entre a melhor posição na vizinhança e a posição atual.
- **Proactive direction:** Avaliada com base nas últimas três posições do seeker.

A atualização da posição é dada pela soma da posição atual e um fator de passo, que é gerado por uma função de pertinência gaussiana. Este algoritmo apresenta uma velocidade de convergência mais rápida e melhor capacidade de busca global em comparação com algoritmos como PSO e DE.

### 21.2 Teaching–Learning-Based Optimization

O **Teaching–Learning-Based Optimization (TLBO)** é uma abordagem populacional que se inspira na filosofia do ensino e aprendizado, onde um grupo de **aprendizes** busca melhorar seu desempenho acadêmico.

- **Fases do algoritmo:** Consiste em duas fases principais: a fase do professor, onde o aprendiz aprende com o melhor indivíduo (professor), e a fase do aprendiz, onde os aprendizes interagem entre si para melhorar seus resultados.
- **Atualização de posição:** Durante a fase do professor, a posição de cada aprendiz é atualizada em direção à posição do professor, considerando a média dos aprendizes.

- **Fase de aprendizagem:** Cada aprendiz escolhe aleatoriamente um colega e se movimenta em direção a ele, dependendo da qualidade do ajuste (fitness) de cada um.

O TLBO é um algoritmo livre de parâmetros que tem se destacado por sua rápida convergência e resultados superiores quando comparado a GA e PSO.

### 21.3 Imperialist Competitive Algorithm

O **Imperialist Competitive Algorithm** é inspirado na evolução sociopolítica humana e busca resolver problemas de otimização contínua.

- **Estrutura do algoritmo:** Começa com um conjunto de soluções aleatórias (países iniciais) que são avaliadas pela função de custo. Os melhores se tornam **imperialistas**, controlando outros países, chamados **colônias**.
- **Operadores principais:**
  - **Assimilação:** Faz com que as colônias se aproximem do imperialista.
  - **Revolução:** Causa mudanças súbitas nas características de alguns países.

O algoritmo continua até que um critério de parada seja atingido. O método é adaptado para resolver problemas como o TSP, onde as colônias aprendem com as políticas de países mais fortes.

### 21.4 Várias Metaheurísticas Inspiradas em Comportamentos Humanos

Nesta seção, exploramos algoritmos **metaheurísticos** inspirados em diversos comportamentos sociais e cognitivos humanos. Cada um desses algoritmos imita aspectos específicos das atividades humanas e dos processos de tomada de decisão, aproveitando esses comportamentos para resolver problemas complexos de otimização.

- **League Championship Algorithm (LCA)**

O **League Championship Algorithm (LCA)** é uma metaheurística estocástica baseada em população, projetada para **otimização global contínua**, inspirada nos campeonatos esportivos. No LCA, equipes competem em uma liga ao longo de várias iterações (chamadas de semanas). Cada equipe representa uma solução candidata, e os jogos entre pares de equipes são pontuados com base no "valor de fitness" ou força de cada equipe. Com o tempo, as equipes ajustam suas estratégias com base nos resultados anteriores, evoluindo para melhores soluções. Esse mecanismo se assemelha ao **Particle Swarm Optimization (PSO)**, mas as soluções evoluem com base nos ajustes estratégicos derivados das análises das partidas, assim como os treinadores ajustariam as formações das equipes com base no desempenho passado e nas estratégias dos adversários.

- **Golden Ball Metaheuristic**

A **Golden Ball** é uma metaheurística modelada no futebol, utilizando uma estrutura de **multi-população** para abordar problemas de otimização combinatória. Aqui, a população é dividida em equipes, cada uma com metodologias de treinamento específicas lideradas por um "treinador". As temporadas, que contêm várias semanas de treinamento e competição, permitem que as equipes refinam suas estratégias e melhorem seu desempenho. Ao final de cada temporada, ocorre um processo de transferência, permitindo que jogadores (soluções) e treinadores (métodos de treinamento) se movam entre as equipes, criando novas dinâmicas e incentivando a diversidade. Esse processo cíclico continua até que uma condição de término seja atendida, refinando gradualmente as soluções.

- **Squeaky Wheel Optimization (SWO)**

**Squeaky Wheel Optimization (SWO)** é particularmente eficaz para problemas combinatórios com componentes interdependentes. Inicialmente, um algoritmo ganancioso gera uma solução, e elementos problemáticos (aqueles que degradam a qualidade da solução) são identificados e priorizados. Essa ordem de prioridade é usada para reconstruir novas soluções em iterações sucessivas. O SWO busca de maneira eficaz tanto o espaço de soluções quanto o espaço de prioridades, ajustando o foco para componentes difíceis de resolver. Para aprimorar o processo, o **Evolutionary Squeaky Wheel Optimization** incorpora operadores de seleção e mutação para reter componentes bem-sucedidos, enquanto reconstrói partes subótimas, permitindo uma otimização mais focada e eficiente.

- **Exchange Market Algorithm (EMA)**

O **Exchange Market Algorithm (EMA)** emula a negociação no mercado de ações para resolver problemas de otimização contínua. Os indivíduos na população representam acionistas, e seus rankings de fitness ditam suas ações de negociação. O mercado categoriza os indivíduos em grupos com base no ranking; os indivíduos de maior ranking permanecem estáveis, enquanto os outros assumem maiores riscos, explorando soluções menos favoráveis. Em mercados não voláteis, indivíduos de baixo ranking imitam as negociações dos elitistas, mas em condições voláteis, eles assumem maiores riscos, explorando áreas desconhecidas do espaço de busca. Esse equilíbrio entre estabilidade e exploração ajuda o EMA a encontrar **ótimos globais**.

- **Group Counseling Optimization (GCO)**

**Group Counseling Optimization (GCO)** simula a dinâmica de resolução de problemas em grupo, onde indivíduos (soluções) melhoram compartilhando experiências e insights dentro de um grupo. Essa metaheurística imita o processo social de aconselhamento, onde os indivíduos se ajustam com base no feedback do grupo e adotam estratégias de aprendizagem compartilhada. O GCO é particularmente eficaz para **otimização multi-objetivo**, fornecendo soluções diversas que refletem experiências coletivas. A estrutura social desse algoritmo

permite aprendizagem adaptativa e suporta uma resolução robusta de problemas em espaços complexos e multidimensionais.

- **Human Learning Optimization (HLO)**

**Human Learning Optimization (HLO)** modela diversos processos de aprendizagem na cognição humana, incluindo aprendizagem individual e social. Combina exploração aleatória com reaprendizado, gerando novas soluções ao emular o refinamento individual, a influência do grupo e a curiosidade exploratória. O HLO tem demonstrado eficácia em tarefas de otimização multidimensionais, como o **problema da mochila**, onde modos de aprendizagem diversos ajudam o algoritmo a navegar por um complexo campo de soluções, equilibrando profundidade e amplitude em sua busca.

- **Creative Thinking-Based Optimization (CTO)**

**Creative Thinking-Based Optimization (CTO)** aplica princípios do **pensamento criativo** à otimização contínua, emulando sessões de brainstorming onde novas soluções surgem da ideação em grupo e refinamento iterativo. CTO inclui **Brainstorm Optimization**, que utiliza métodos de brainstorming para explorar caminhos inovadores de resolução de problemas. Essa abordagem é paralelizada e computacionalmente eficiente, adequada para resolver problemas complexos com espaços de solução de alta dimensão, simulando a ideação coletiva e a resolução espontânea de problemas.

- **Immigrant Population Search (IPS)**

**Immigrant Population Search (IPS)** tira inspiração dos comportamentos humanos de migração para explorar melhores habitats dentro do espaço de soluções. Nesse contexto, soluções viáveis representam "habitats" favoráveis com atributos de alta qualidade, e a população do algoritmo migra em direção a melhores opções. Cada grupo populacional avalia e se move em direção a soluções mais ótimas, com a função de fitness representando a qualidade de vida do ambiente. Esse movimento semelhante à migração fomenta a diversidade e ajuda a convergir para soluções de alta qualidade.

- **Democracy-Inspired PSO with Peer Groups**

Essa variação do **Particle Swarm Optimization (PSO)** integra princípios de governança democrática. No **Democracy-Inspired PSO**, partículas formam grupos de pares e selecionam líderes com base em um processo de votação, escolhendo entre dois líderes potenciais. Essa estrutura democrática incentiva influências diversas, já que as partículas podem escolher seguir um líder ou uma figura de oposição, adicionando flexibilidade e resiliência ao processo de busca. Esse método é bem adequado para resolver problemas de otimização multimodais, onde estratégias diversas de solução podem melhorar o desempenho da busca.

Esses algoritmos ilustram como as metaheurísticas podem simular efetivamente diversos comportamentos humanos e estruturas organizacionais, oferecendo soluções robustas para

desafios complexos de otimização. Através de mecanismos que lembram aprendizagem, competição, tomada de risco e resolução cooperativa de problemas, esses algoritmos navegam por vastos espaços de solução, equilibrando exploração e exploração para encontrar soluções ótimas de forma eficiente.

## APÊNDICE 4

## Termo de Aceite de Entrega

### Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

**Data da Reunião (“gate”) de aprovação:** 6 de nov. de 2024

**Participantes da Entrega** [matriculados em Residência em IA]:

Victor Guerreiro Pimenta

**Entrega:** [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Screening de códigos, frameworks, ou implementações das aplicações, técnicas e algoritmos de metaheurísticas levantados, **presentes no github**. Segundo os critérios:
  - **Recência:** Intervalo de tempo desde a última atualização do projeto.
  - **Número de Contribuições:** Quantidade de commits e pull requests feitas no repositório.
  - **Estrelas:** Quantidade de estrelas atribuídas pelos usuários no repositório GitHub.
  - **Documentação:** Conjunto de informações e instruções disponíveis para orientar o uso e desenvolvimento do projeto.
  - **Licença:** Tipo de licença que rege o uso, modificação e distribuição do código.
  - **Testes Unitários:** Presença de testes automatizados que verificam a funcionalidade das partes individuais do código.
- Links do github compilados em [Busca de códigos heurísticas](#)

**Planejamento:** [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Testar, implementar e avaliar os códigos e frameworks.

**Observação:** [caso precise fazer alguma observação, de qualquer “natureza”]

## ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go!](#)

[Busca de códigos heurísticas citado no Termo de Aceite de Entrega de 6 de novembro]

## Metodologia de Busca

A busca foi realizada exclusivamente na plataforma GitHub, devido à facilidade de comparação e análise quantitativa e qualitativa de projetos hospedados na plataforma. Foram utilizados termos de busca específicos para cada algoritmo, seguidos da palavra-chave “Python” para garantir que somente implementações nesta linguagem fossem listadas. A análise se limitou às implementações mais populares e bem avaliadas em cada categoria de algoritmo, utilizando os critérios previamente estabelecidos para compilar e comparar as opções.

Critérios:

- **Recência:** Intervalo de tempo desde a última atualização do projeto.
- **Número de Contribuições:** Quantidade de commits e pull requests feitas no repositório.
- **Estrelas:** Quantidade de estrelas atribuídas pelos usuários no repositório GitHub.
- **Documentação:** Conjunto de informações e instruções disponíveis para orientar o uso e desenvolvimento do projeto.
- **Licença:** Tipo de licença que rege o uso, modificação e distribuição do código.
- **Testes Unitários:** Presença de testes automatizados que verificam a funcionalidade das partes individuais do código.

### 1. Meta-heuristic level

[thieu1995/mealpy](https://github.com/thieu1995/mealpy)

**A Collection Of The State-of-the-art Metaheuristic Algorithms In Python  
(Metaheuristic/Optimizer/Nature-inspired/Biology)**

<https://github.com/SISDevelop/SwarmPackagePy>

**Library of swarm optimization algorithms.**

### 2. Simulated Annealing

[perrygeo/simanneal](https://github.com/perrygeo/simanneal)

**Python module for *Simulated Annealing* optimization**

[shinmorino/sqaod](#)

**Solvers/*annealers* for *simulated quantum annealing* on CPU and CUDA(NVIDIA GPU).**

### 3. Genetic Algorithms

[ahmedfgad/GeneticAlgorithmPython](#)

**Source code of PyGAD, a *Python 3* library for building the *genetic algorithm* and training machine learning *algorithms* (Keras & PyTorch).**

### 4. Genetic Programming

[cavalab/ellyn](#)

***python*-wrapped version of *ellen*, a linear *genetic programming* system for *symbolic regression* and classification.**

[hengzhe-zhang/PS-Tree](#)

**An open source *python* library for non-linear piecewise *symbolic regression* based on *Genetic Programming***

[trevorstephens/gplearn](#)

**Genetic Programming in Python, with a scikit-learn inspired API**

### 5. Evolutionary Strategies

[alirezamika/evostra](#)

**A fast Evolution Strategy implementation in Python**

---

[openai/evolution-strategies-starter](#)

Code for the paper "*Evolution Strategies as a Scalable Alternative to Reinforcement Learning*"

6. Differential Evolution

[nathanrooy/differential-evolution-optimization](#)

A simple, bare bones, implementation of *differential evolution* optimization.

[malmazuke/differential-evolution-for-portfolio-optimisation](#)

A *Python* implementation of *Differential Evolution*, used in the context of Portfolio Optimization.

[tiagoCuervo/EvoFuzzy](#)

A *Python* implementation of the *Differential Evolution* algorithm for the optimization of Fuzzy Inference Systems.

7. Estimation of Distribution Algorithms

[VicentePerezSoloviev/EDAspy](#)

*Estimation of Distribution algorithms Python* package

[unclearness/EBNA](#)

Implementation of *Estimation of Bayesian Network Algorithm* (EBNA), which is an *Estimation of Distribution Algorithm* (EDA) in *Python*

8. Particle Swarm Optimization

[ljvmiranda921/pyswarms](#)

---

A research toolkit for *particle swarm optimization* in *Python*

[nathanrooy/particle-swarm-optimization](#)

Learn about *particle swarm optimization* (PSO) through *Python*!

[ujjwalkhandelwal/pso\\_particle\\_swarm\\_optimization](#)

Implemented fully documented *Particle Swarm Optimization* algorithm (basic model with few advanced features) using *Python* programming language

[jerrytheo/psopy](#)

A SciPy compatible super fast *Python* implementation for *Particle Swarm Optimization*.

## 9. Artificial Immune Systems

[AIS-Package/aisp](#)

This is an open source *Python* package that provides an implementation of techniques inspired by the artificial immune system, enabling the easy and intuitive use of algorithms based on immunology.

## 10. Ant Colony Optimization

[pjmattingly/ant-colony-optimization](#)

## Public archive

Implementation of the *Ant Colony Optimization* algorithm (*python*)

[ShengzhiWu/anco](#)

A *GPU-accelerated ant colony optimization library for Python 3 based on taichi*.

[rhgrant10/acopy](#)

---

## ***A Python implementation of the Ant Colony Optimization Meta-Heuristic***

### **11. Bee Metaheuristics**

[rwuilbercq/Hive](#)

**Artificial Bee Colony Algorithm in *Python*.**

[amineremache/qbso-fs](#)

***Python* implementation of QBSO-FS : a Reinforcement Learning based Bee Swarm Optimization *metaheuristic* for Feature Selection problem.**

### **12. Bacterial Foraging Algorithm**

<https://github.com/SISDevelop/SwarmPackagePy>

**Library of swarm optimization algorithms.**

### **13. Harmony Search**

[lanl/pyHarmonySearch](#)

**pyHarmonySearch is a pure *Python* implementation of the *harmony search* (HS) global optimization algorithm.**

### **14. Swarm Intelligence**

<https://github.com/SISDevelop/SwarmPackagePy>

**Library of swarm optimization algorithms.**

[Pattio/DeepSwarm](#)

**Neural Architecture Search Powered by *Swarm Intelligence*** 🐜

## 15. Biomolecular Computing

### [biopython/biopython](#)

Official git repository for Biopython (originally converted from CVS)

### [scikit-bio/scikit-bio](#)

*scikit-bio*: a community-driven *Python* library for bioinformatics, providing versatile data structures, algorithms and educational resources.

### [michellab/BioSimSpace](#)

Code and resources for the EPSRC BioSimSpace project.

## 16. Quantum Computing

### [CQCL/pytket-docs](#)

User manual and example notebooks for the *pytket quantum computing* toolkit

## 17. Metaheuristics Based on Sciences

### [himanshuRepo/Gravitational-Search-Algorithm](#)

Python Code for *Gravitational Search Algorithm (GSA)*

### [SajadAHMAD1/Chaotic-GSA-for-Engineering-Design-Problems](#)

All nature-inspired algorithms involve two processes namely exploration and exploitation. For getting optimal performance, there should be a proper balance between these processes. Further, the majority of the optimization algorithms suffer from local minima entrapment problem and slow convergence speed. To alleviate these problems, researchers ...

## 18. Memetic Algorithms

[YuJerryShi/memetic\\_thin\\_film\\_py](#)

Now in Python! This software designs thin-film optical structures to match target spectral properties using the memetic algorithm

[hurlenko/memetic-algorithm](#)

Simple implementation of *Memetic Algorithm* written in python3

## 19. Tabu Search and Scatter Search

[mcfadd/Job\\_Shop\\_Schedule\\_Problem](#)

Parallel *Tabu Search* and Genetic Algorithm for the Job Shop Schedule Problem with Sequence Dependent Set Up Times

[stromatolith/peabox](#)

an evolutionary algorithm toolbox written in python

## 20. Search Based on Human Behaviors

[andaviaco/tblo](#)

*Teaching-learning-based optimization* algorithm implementation.

[jorisdhondt/imperialist-competitive-algorithm](#)

Implementation of the *imperialist competitive algorithm* (ICA)

<https://github.com/sorend/fylearn/tree/main>

## Termo de Aceite de Entrega

### Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

**Data da Reunião (“gate”) de aprovação:** 14 de nov. de 2024

**Participantes da Entrega** [matriculados em Residência em IA]:

Victor Guerreiro Pimenta

**Entrega:** [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Análise de frameworks:

- [MealPy](#) Possui base para definir **custom Problem class** com **Tipos de Variáveis** da solução, **Função Objetivo** e **Parâmetros de Otimização**.  
**Não possui Simulated Annealing. Somente Population Based.**  
**Contém grande parte dos algoritmos dos outros Frameworks levantados**
- [Simanneal](#) Annealer exige definição de **Movimento** e **Energia**.
- [PyGAD](#) Suporte para Deep Learning (**Keras** e **PyTorch**).
- [gplearn](#) e [ellyn](#) Suporte para Regressão Simbólica com sklearn API.
- [openai/evolution-strategies-starter](#) **Code for the paper "Evolution Strategies as a Scalable Alternative to Reinforcement Learning"**
- Outros Frameworks em [Mapeamento de códigos, frameworks e implementações](#).

**Planejamento:** [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Modelar problemas de otimização levantados com os Framework (preferencialmente MealPy) e procurar bases de dados ou simuladores relacionados para testes.

**Observação:** [caso precise fazer alguma observação, de qualquer “natureza”]

## ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go! ▾](#)

[Mapeamento de códigos, frameworks e implementações citado no Termo de Aceite de Entrega de 14 de novembro]

Residência em IA  
Discente: Victor Guerreiro Pimenta

Análise e mapeamento de códigos, frameworks e implementações das aplicações, técnicas e algoritmos de metaheurísticas.

Testes realizados em `Testes frameworks heurísticas.ipynb`.

## 1. Meta-heuristic level

[thieu1995/mealpy](#)

**A Collection Of The State-of-the-art Metaheuristic Algorithms In Python  
(Metaheuristic/Optimizer/Nature-inspired/Biology)**

- [MealPy](#) Possui base para definir **custom Problem class** com **Tipos de Variáveis** da solução, **Função Objetivo** e **Parâmetros de Otimização**.  
**Não possui Simulated Annealing**. Somente **Population Based**.  
**Contém grande parte dos algoritmos em outros Frameworks levantados**

<https://github.com/SISDevelop/SwarmPackagePy>

Library of swarm optimization algorithms.

- [SwarmPackagePy](#) Todos os alg. implementados no Mealpy.

## 2. Simulated Annealing

[perrygeo/simanneal](#)

**Python module for Simulated Annealing optimization**

- [simanneal](#) Annealer exige definição de **Movimento** e **Energia**.

---

### [shinmorino/sqaod](#)

Solvers/*annealers* for *simulated quantum annealing* on CPU and CUDA(NVIDIA GPU).

- Requerimento de GPU

### 3. Genetic Algorithms

### [ahmedfgad/Genetic Algorithm Python](#)

Source code of PyGAD, a *Python 3* library for building the *genetic algorithm* and training machine learning *algorithms* (Keras & PyTorch).

- [PyGAD](#) Suporte para Deep Learning (**Keras** e **PyTorch**).

### 4. Genetic Programming

### [cavalab/ellyn](#)

*python-wrapped* version of *ellen*, a linear *genetic programming* system for *symbolic regression* and classification.

- Suporte para Regressão Simbólica

### [hengzhe-zhang/PS-Tree](#)

An open source *python* library for non-linear piecewise *symbolic regression* based on *Genetic Programming*

- Suporte para Regressão Simbólica. Obtive problemas de instalação.

### [trevorstevens/gplearn](#)

Genetic Programming in Python, with a scikit-learn inspired API

- [gplearn](#) Suporte para Regressão Simbólica

## 5. Evolutionary Strategies

[alirezamika/evostra](#)

A fast Evolution Strategy implementation in Python

[openai/evolution-strategies-starter](#)

Code for the paper "*Evolution Strategies as a Scalable Alternative to Reinforcement Learning*"

<https://openai.com/index/evolution-strategies/>

## 6. Differential Evolution

[nathanrooy/differential-evolution-optimization](#)

A simple, bare bones, implementation of *differential evolution* optimization.

- Aplicável em problemas com valores reais.

[malmazuke/differential-evolution-for-portfolio-optimisation](#)

A *Python* implementation of *Differential Evolution*, used in the context of Portfolio Optimization.

[tiagoCuervo/EvoFuzzy](#)

A *Python* implementation of the *Differential Evolution* algorithm for the optimization of Fuzzy Inference Systems.

## 7. Estimation of Distribution Algorithms

[VicentePerezSoloviev/EDAspy](#)

*Estimation of Distribution algorithms Python package*

---

## [unclearness/EBNA](#)

Implementation of *Estimation of Bayesian Network Algorithm* (EBNA), which is an *Estimation of Distribution Algorithm* (EDA) in *Python*

### 8. Particle Swarm Optimization

#### [lvmiranda921/pyswarms](#)

A research toolkit for *particle swarm optimization* in *Python*

#### [nathanrooy/particle-swarm-optimization](#)

Learn about *particle swarm optimization* (PSO) through *Python*!

#### [ujjwalkhandelwal/psoparticle\\_swarm\\_optimization](#)

Implemented fully documented *Particle Swarm Optimization* algorithm (basic model with few advanced features) using *Python* programming language

#### [jerrytheo/psopy](#)

A SciPy compatible super fast *Python* implementation for *Particle Swarm Optimization*.

### 9. Artificial Immune Systems

#### [AIS-Package/aisp](#)

This is an open source *Python* package that provides an implementation of techniques inspired by the artificial immune system, enabling the easy and intuitive use of algorithms based on immunology.

### 10. Ant Colony Optimization

#### [pjmattingly/ant-colony-optimization](#)

## Public archive

Implementation of the *Ant Colony Optimization* algorithm (*python*)

[ShengzhiWu/anco](#)

A GPU-accelerated ant colony optimization library for Python 3 based on taichi.

[rhgrant10/acopy](#)

A Python implementation of the *Ant Colony Optimization Meta-Heuristic*

### 11. Bee Metaheuristics

[rwuilbercq/Hive](#)

Artificial *Bee Colony* Algorithm in *Python*.

[amineremache/qbso-fs](#)

*Python* implementation of QBSO-FS : a Reinforcement Learning based *Bee Swarm Optimization metaheuristic* for Feature Selection problem.

### 12. Bacterial Foraging Algorithm

<https://github.com/SISDevelop/SwarmPackagePy>

Library of swarm optimization algorithms.

### 13. Harmony Search

[lanl/pyHarmonySearch](#)

pyHarmonySearch is a pure *Python* implementation of the *harmony search* (HS) global optimization algorithm.

### 14. Swarm Intelligence

---

<https://github.com/SISDevelop/SwarmPackagePy>

Library of swarm optimization algorithms.

[Pattio/DeepSwarm](#)

Neural Architecture Search Powered by *Swarm Intelligence* 

## 15. Biomolecular Computing

[biopython/biopython](#)

Official git repository for Biopython (originally converted from CVS)

[scikit-bio/scikit-bio](#)

scikit-*bio*: a community-driven *Python* library for bioinformatics, providing versatile data structures, algorithms and educational resources.

[michellab/BioSimSpace](#)

Code and resources for the EPSRC BioSimSpace project.

## 16. Quantum Computing

[CQCL/pytket-docs](#)

User manual and example notebooks for the pytket *quantum computing* toolkit

## 17. Metaheuristics Based on Sciences

[himanshuRepo/Gravitational-Search-Algorithm](#)

Python Code for *Gravitational Search Algorithm* (GSA)

---

## [SajadAHMAD1/Chaotic-GSA-for-Engineering-Design-Problems](#)

All nature-inspired algorithms involve two processes namely exploration and exploitation. For getting optimal performance, there should be a proper balance between these processes. Further, the majority of the optimization algorithms suffer from local minima entrapment problem and slow convergence speed. To alleviate these problems, researchers ...

### 18. Memetic Algorithms

#### [YuJerryShi/memetic\\_thin\\_film\\_py](#)

Now in Python! This software designs thin-film optical structures to match target spectral properties using the memetic algorithm

#### [hurlenko/memetic-algorithm](#)

Simple implementation of *Memetic Algorithm* written in python3

### 19. Tabu Search and Scatter Search

#### [mcfadd/Job\\_Shop\\_Schedule\\_Problem](#)

Parallel *Tabu Search* and Genetic Algorithm for the Job Shop Schedule Problem with Sequence Dependent Set Up Times

#### [stromatolith/peabox](#)

an evolutionary algorithm toolbox written in python

### 20. Search Based on Human Behaviors

#### [andaviaco/tblo](#)

*Teaching–learning-based optimization* algorithm implementation.

[\*\*jorisdhondt/imperialist-competitive-algorithm\*\*](#)

Implementation of the *imperialist competitive algorithm* (ICA)

<https://github.com/sorend/fylearn/tree/main>

## APÊNDICE 5

## Termo de Aceite de Entrega

### Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

**Data da Reunião (“gate”) de aprovação:** 27 de nov. de 2024

**Participantes da Entrega** [matriculados em Residência em IA]:

Victor Guerreiro Pimenta

**Entrega:** [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Modelagem de problemas de otimização levantados com o MealPy
  - Modelagem de Problemas de Otimização com MealPy.ipynb
- Bases de Dados e Simuladores
- **Localização de torres de celular e áreas de cobertura:** OpenCellID
- **Topologias de rede:** TopGen, SNDLib, NS-3
- **Cenários de tráfego aéreo simulados:** BlueSky, OpenSky Network
- **Dados geográficos:** OpenStreetMap, VRP Instances, Geonames, OSMnX
- **Dados financeiros:** Yahoo Finance, Quandl, Kaggle Datasets - Finance
- **Dados de biologia estrutural:** Protein Data Bank, UniProt, Rosetta Commons
- **Dados de materiais:** Materials Project, AFLOW, Open Quantum Materials Database, Matminer
- **Problemas combinatórios e de roteamento:** TSPLIB, DIMACS Challenge Instances
- **Planejamento e programação:** Google OR-Tools, PSPLIB, MISTA Benchmark
- **Simulação de tráfego urbano:** SUMO
- **Simulação de cenários climáticos:** Weather Research and Forecasting Model (WRF)

**Planejamento:** [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Testar diferentes algoritmos nos problemas modelados, comparar e analisar os resultados.

**Observação:** [caso precise fazer alguma observação, de qualquer “natureza”]

## ACEITE DA ENTREGA:

LEONARDO ANTÔNIO ALVES: Em análise! ▾

[**Bases de Dados e Simuladores** citado no Termo de Aceite de Entrega de 27 de novembro]

## Telecomunicações

- **OpenCellID**: Base de dados com localizações e coberturas de torres de celular, útil para otimização de redes UMTS.
  - **TopGen**: Gerador de grafos topológicos para redes simuladas.
  - **NS-3 (Network Simulator 3)**: Simulador de redes avançado que permite testar diferentes configurações de redes móveis e roteamento.
  - **SNDLib (Survivable Network Design Library)**: Dados de topologias de redes reais e simuladas com atributos como largura de banda e latência.
- 

## Transporte e Logística

- **BlueSky**: Simulador de tráfego aéreo que fornece cenários realistas com base em dados públicos.
  - **OpenSky Network**: Dados de tráfego aéreo em tempo real, incluindo posições de aeronaves e rotas.
  - **TSPLIB**: Instâncias clássicas de problemas de roteamento de veículos e TSP (Traveling Salesman Problem).
  - **VRP Instances (Vehicle Routing Problem)**: Base de dados padrão para problemas de roteamento de veículos.
  - **SUMO (Simulation of Urban Mobility)**: Simulador de tráfego urbano para modelagem de rotas, redes de transporte público e veículos.
- 

## Planejamento e Programação

- **Google OR-Tools**: Ferramenta que suporta problemas de planejamento e fornece instâncias de dados para sequenciamento de tarefas e planejamento de produção.
  - **PSPLIB (Project Scheduling Problem Library)**: Base de dados para problemas de programação de tarefas com restrições de recursos.
-

- **MISTA Benchmark:** Dados para problemas de programação em indústrias e produção.
- 

## Finanças e Economia

- **Yahoo Finance:** Fonte de dados históricos de ações, índices e criptomoedas para problemas de otimização de portfólios.
  - **Quandl:** Plataforma com diversos datasets financeiros, como taxas de câmbio, commodities e derivativos.
  - **Kaggle Datasets - Finance:** Coleção de dados financeiros, incluindo risco de crédito e preços de ativos.
- 

## Bioinformática

- **Protein Data Bank (PDB):** Banco de dados com estruturas de proteínas e interações moleculares.
  - **Rosetta Commons:** Ferramenta para simulação e otimização de estruturas proteicas.
  - **UniProt:** Base de dados de sequências e anotações proteicas para modelagem.
- 

## Ciência dos Materiais

- **Materials Project:** Base de dados com propriedades de materiais, como resistência mecânica e térmica.
  - **AFLOW (Automatic Flow Framework):** Dados de materiais projetados para modelagem de propriedades eletrônicas e térmicas.
  - **Open Quantum Materials Database (OQMD):** Dados para otimização e previsão de propriedades de materiais.
  - **Matminer:** Ferramenta para explorar e coletar datasets de materiais.
- 

## Outros

- **TSP e Problemas Combinatórios: DIMACS Challenge Instances,** uma coleção para otimização combinatória.
-

- **Simulação de Cenários Climáticos: Weather Research and Forecasting Model (WRF)**, útil para modelar restrições ambientais em problemas de logística.
- **Dados Geográficos e Urbanos: Geonames e OSMnx** para análise e otimização de redes urbanas.

## APÊNDICE 6

## Termo de Aceite de Entrega

### Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

**Data da Reunião (“gate”) de aprovação:** 5 de dez. de 2024

**Participantes da Entrega** [matriculados em Residência em IA]:

Victor Guerreiro Pimenta

**Entrega:** [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Estudo comparativo de algoritmos nos problema mais clássicos, considerando principalmente a variável tempo de convergência:
  - Estudo Comparativo
- Gráfico do panorama de fundamentos e ferramentas levantados:
  - Panorama da Otimização Meta-Heurística

**Planejamento:** [descrever o que pretende fazer para realizar a próxima ENTREGA]

**Observação:** [caso precise fazer alguma observação, de qualquer “natureza”]


Realizar uma compilação melhor estruturada dos conteúdos levantados.

## ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

[**Estudo Comparativo** citado no Termo de Aceite de Entrega de 5 de dezembro]

## Problemas e Algoritmos Comparados

 Estudo Comparativo.ipynb

Neste estudo, foram comparados quatro algoritmos metaheurísticos populares: **Otimização por Enxame de Partículas (PSO)**, **Algoritmo Genético (GA)**, **Otimização por Colônia de Formigas (ACO)** e **Otimização por Forrageamento Bacteriano (BA)**. Esses algoritmos foram aplicados a quatro problemas clássicos de otimização: **Problema do Caixeiro Viajante (TSP)**, **Problema de Roteamento de Veículos (VRP)**, **Problema da Árvore de Steiner** e **Particionamento de Grafos**, e comparados em termos de **tempo de convergência** e **fitness** obtido.

Além disso, um **problema de otimização de hiperparâmetros de SVM** foi incluído para demonstrar como os algoritmos podem ser aplicados a problemas de otimização contínuos, como a seleção dos melhores parâmetros para um modelo de **Máquinas de Vetores de Suporte (SVM)**.

1. **Problema do Caixeiro Viajante (TSP):** O **TSP** é um problema clássico onde o objetivo é encontrar o caminho mais curto que visita um conjunto de cidades uma única vez e retorna à cidade de origem. Esse problema é NP-difícil e é amplamente estudado em algoritmos de otimização.
2. **Problema de Roteamento de Veículos (VRP):** O **VRP** busca encontrar as rotas ótimas para uma frota de veículos que devem entregar mercadorias a diversos locais, minimizando a distância total percorrida, respeitando as capacidades dos veículos e as janelas de tempo de entrega. Esse problema é essencial para logística e transporte.
3. **Problema da Árvore de Steiner:** O **Problema da Árvore de Steiner** busca encontrar a árvore de peso mínimo em um grafo que conecta um subconjunto de vértices, conhecidos como terminais. É um problema importante em redes de comunicação e transporte.
4. **Particionamento de Grafos:** O **Problema de Particionamento de Grafos** visa dividir um grafo em dois ou mais subgrafos, minimizando o número de arestas entre os subgrafos e mantendo as partições balanceadas. Este problema tem diversas aplicações, como em computação paralela e design de circuitos.

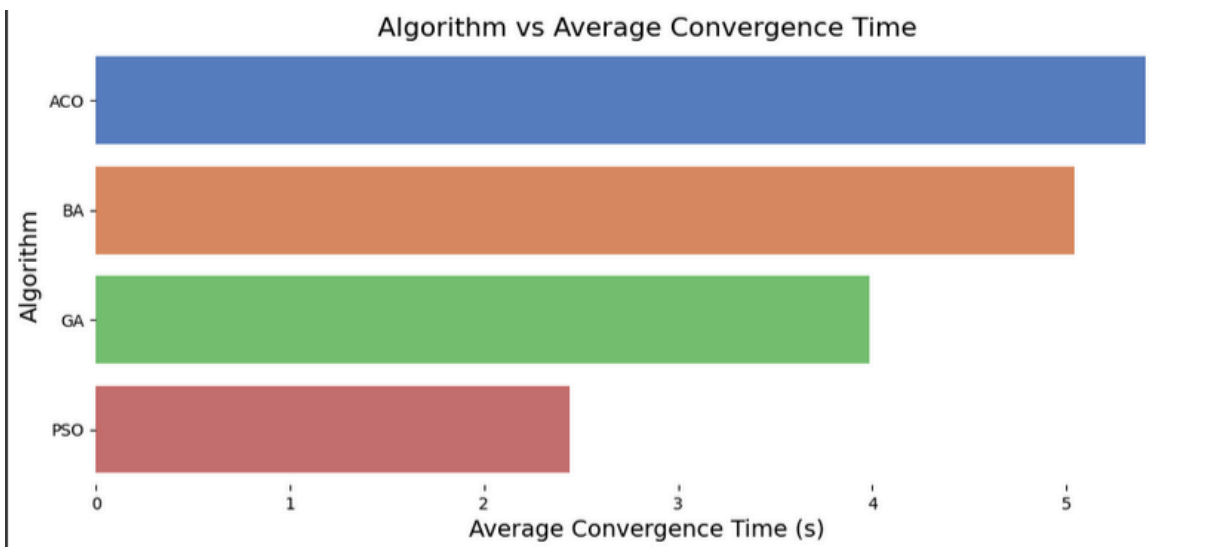
5. **Otimização de Hiperparâmetros do SVM:** A **otimização de hiperparâmetros de SVM** busca ajustar os parâmetros de um modelo de SVM, como o valor de C, o tipo de kernel, o grau do polinômio e o parâmetro gamma, para melhorar a performance do modelo em termos de acurácia.

## Algoritmos Metaheurísticos

1. **Otimização por Enxame de Partículas (PSO):** O PSO é um algoritmo baseado em população inspirado no comportamento social de pássaros ou cardumes. Cada partícula se move pelo espaço de soluções, ajustando sua posição com base em sua própria experiência e na experiência do grupo, buscando encontrar soluções ótimas de forma rápida.
2. **Algoritmo Genético (GA):** O GA é um algoritmo evolutivo inspirado na seleção natural. Utiliza operações como crossover, mutação e seleção para evoluir uma população de soluções e encontrar a solução ótima. É eficaz tanto em problemas combinatórios quanto contínuos.
3. **Otimização por Colônia de Formigas (ACO):** O ACO simula o comportamento de formigas, onde elas depositam feromônio ao longo dos caminhos percorridos, direcionando outras formigas para soluções promissoras. Ao longo do tempo, caminhos mais curtos e eficientes são reforçados pela presença do feromônio.
4. **Otimização por Forrageamento Bacteriano (BA):** O BA é inspirado no comportamento de bactérias que se movem em busca de fontes de alimento, realizando adaptações contínuas ao seu ambiente. A otimização é realizada por uma população de "bactérias", que exploram o espaço de soluções de maneira eficiente.

## Resultados

	Problem	Algorithm	Best Fitness	Convergence Time (s)
0	TSP	PSO	0.000000	0.452745
1	TSP	GA	0.000000	1.317694
2	TSP	ACO	0.000000	2.851813
3	TSP	BA	0.000000	0.620995
4	VRP	PSO	37.798256	0.689916
5	VRP	GA	30.867281	1.360806
6	VRP	ACO	0.000000	2.470138
7	VRP	BA	0.000000	0.780354
8	Steiner Tree	PSO	0.547753	0.825998
9	Steiner Tree	GA	0.974656	1.826646
10	Steiner Tree	ACO	0.000000	3.196491
11	Steiner Tree	BA	0.095410	0.437777
12	Graph Partitioning	PSO	1.482271	0.403330
13	Graph Partitioning	GA	0.000000	0.853870
14	Graph Partitioning	ACO	0.000000	1.444447
15	Graph Partitioning	BA	0.000000	0.351018
16	SVM optimizer	PSO	0.982456	9.830268
17	SVM optimizer	GA	0.976608	14.560991
18	SVM optimizer	ACO	0.982456	17.071673
19	SVM optimizer	BA	0.976608	23.017656



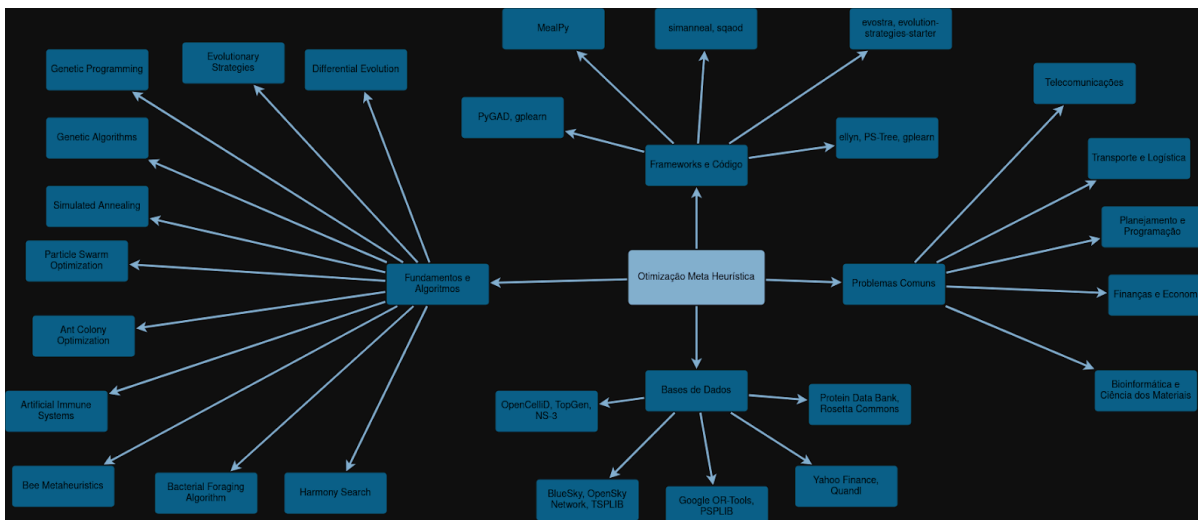
Os experimentos mostraram resultados interessantes em relação ao **tempo de convergência** e à **qualidade das soluções**:

- O **PSO** obteve o **menor tempo de convergência** entre todos os algoritmos testados. Isso ocorre porque o PSO realiza uma busca eficiente no espaço de soluções, ajustando rapidamente as partículas em direção às melhores soluções. Esse desempenho rápido é especialmente útil em problemas contínuos, como o TSP, o VRP e o Problema da Árvore de Steiner.
- O **Algoritmo Genético (GA)**, embora mais lento do que o PSO, produziu resultados competitivos em termos de **fitness**, principalmente em problemas combinatórios, como o TSP e o VRP, onde as operações de crossover e mutação são eficazes para explorar o espaço de soluções.
- O **ACO** e o **BA** também apresentaram bons resultados, mas com tempos de convergência mais elevados em comparação ao PSO. Ambos os algoritmos são eficientes na exploração de espaços de soluções complexos, mas exigem mais iterações para atingir soluções ótimas.
- No caso da **otimização de hiperparâmetros do SVM**, os algoritmos metaheurísticos ajudaram a ajustar os parâmetros como C, kernel, grau e gamma, proporcionando uma melhoria significativa na acurácia do modelo SVM em comparação com uma busca por grid tradicional.

## Conclusão

Em resumo, o **PSO** se destacou pelo seu **tempo de convergência rápido**, tornando-se a escolha ideal para problemas como o TSP e VRP, onde a obtenção de soluções rápidas é crucial. No entanto, para problemas como o **Problema da Árvore de Steiner** e o **Particionamento de Grafos**, algoritmos como **GA**, **ACO** e **BA** podem produzir soluções de alta qualidade, mas a custos de um tempo de convergência maior. O estudo também demonstrou a eficácia das metaheurísticas na **otimização de hiperparâmetros do SVM**, melhorando o desempenho do modelo em termos de acurácia.

[Panorama da Otimização Metaheurística citado no Termo de Aceite de Entrega de 5 de dezembro]



Este diagrama oferece uma visão abrangente sobre a **Otimização Metaheurística**, abordando diferentes aspectos que envolvem técnicas, frameworks, problemas e bases de dados utilizados na aplicação dessas técnicas. O diagrama está organizado em quatro seções principais: **Fundamentos e Algoritmos**, **Frameworks e Código**, **Problemas Comuns** e **Bases de Dados**.

## 1. Fundamentos e Algoritmos

A seção de **Fundamentos e Algoritmos** aborda diversas técnicas de otimização metaheurística, que são utilizadas para resolver problemas complexos e de difícil solução de maneira eficiente. Cada algoritmo tem suas peculiaridades, mas todos buscam encontrar soluções ótimas ou quase ótimas por meio de aproximações. Entre os algoritmos apresentados, destacam-se:

- **Simulated Annealing (SA):** Uma técnica baseada em simulação de resfriamento de materiais, usada para escapar de mínimos locais e encontrar uma solução global aproximada.
- **Algoritmos Genéticos (GA):** Inspirados na seleção natural, estes algoritmos buscam evoluir soluções por meio de operadores de seleção, cruzamento e mutação.
- **Algoritmos de Programação Genética (GP):** Uma forma de evolução de programas, usando uma representação simbólica para gerar soluções de problemas complexos.
- **Algoritmos de Colônia de Formigas (ACO):** Simulam o comportamento das formigas na busca por alimentos, ideal para problemas de roteamento, como o problema do vendedor viajante (TSP).

- **Otimização por Enxame de Partículas (PSO):** Simula o movimento de partículas em um espaço de busca, ajustando sua posição de acordo com a melhor solução encontrada.
- **Evolução Diferencial (DE):** Um algoritmo de otimização que gera novas soluções baseadas na diferença entre vetores de soluções.
- **Sistemas Imunes Artificiais (AIS):** Inspirados no sistema imunológico, são usados para aprender e adaptar-se a problemas dinâmicos.

## 2. Frameworks e Código

Na seção **Frameworks e Código**, são apresentados os frameworks mais utilizados para implementar as técnicas de otimização metaheurística. Estes frameworks oferecem implementações de código que facilitam a aplicação dos algoritmos em diferentes problemas. Por exemplo, para **Algoritmos Genéticos**, os frameworks como **PyGAD** e **gplearn** são amplamente utilizados, enquanto o **PSO** pode ser implementado usando o **pyswarms** ou **particle-swarm-optimization**.

**Mealpy** é uma biblioteca Python que se destaca por oferecer uma vasta gama de algoritmos metaheurísticos prontos para uso. Ela permite a fácil implementação de técnicas como PSO, Algoritmos Genéticos, Colônia de Formigas, Busca Harmônica e muitos outros, facilitando a experimentação com diferentes algoritmos de otimização. Mealpy oferece uma interface simples para implementar e customizar os algoritmos, além de permitir a integração com outras bibliotecas de aprendizado de máquina, o que a torna uma ferramenta poderosa para problemas de otimização. A flexibilidade do Mealpy é uma de suas grandes vantagens, permitindo ajustar parâmetros e implementar novas variantes de algoritmos com facilidade. Além disso, Mealpy possui uma documentação detalhada e exemplos práticos que ajudam a acelerar o processo de aprendizado e aplicação de metaheurísticas.

## 3. Problemas Comuns

A seção de **Problemas Comuns** descreve as áreas de aplicação das técnicas de otimização metaheurística. A otimização em **Telecomunicações** pode envolver o planejamento de redes UMTS ou o roteamento eficiente de redes usando algoritmos de colônia de formigas. **Transporte e Logística** incluem problemas como a gestão de tráfego aéreo, onde algoritmos genéticos são aplicados para resolver conflitos em voo, ou o roteamento de veículos usando colônia de formigas. Em **Planejamento e Programação**, técnicas de otimização podem ser usadas para o planejamento da produção e programação de tarefas. Na área de **Finanças e Economia**, as técnicas são aplicadas na otimização de portfólios e gerenciamento de riscos. Já em **Bioinformática e Ciência dos Materiais**, a

otimização pode ser utilizada para o projeto de proteínas e desenvolvimento de novos materiais.

#### 4. Bases de Dados

Finalmente, a seção de **Bases de Dados** apresenta fontes de dados frequentemente usadas para testar e treinar os algoritmos de otimização. Por exemplo, na área de **Telecomunicações**, a base de dados **OpenCellID** contém informações sobre localizações de torres de celular, enquanto na **Logística**, a **TSPLIB** fornece instâncias clássicas de problemas de roteamento de veículos e TSP. Em **Bioinformática**, o **Protein Data Bank (PDB)** é uma base importante para a pesquisa em estruturas de proteínas.

#### Considerações Finais

Este diagrama ilustra de maneira clara a inter-relação entre as técnicas de otimização metaheurística, os frameworks utilizados para implementá-las, os problemas práticos onde elas são aplicadas e as bases de dados que alimentam os algoritmos. A otimização metaheurística é uma ferramenta poderosa para resolver problemas complexos e encontrar soluções de alta qualidade de maneira eficiente.