



# Análise e Comparação do Desempenho de Metaheurísticas Aplicadas a Problemas de Escalonamento das Operações

TOMÁS SEQUEIRA RAMOS NUNES DE SOUSA

julho de 2025

**Análise e Comparação do Desempenho de  
Metaheurísticas Aplicadas a Problemas de  
Escalonamento das Operações**

**Tomás Sequeira Ramos Nunes de Sousa**

**Dissertação para obtenção do Grau de Mestre em  
Engenharia Mecânica, Área de Especialização em  
Gestão Industrial**

**Orientador: Dr. André Borges Guimarães Serra e Santos**

**Júri:**

Presidente: Especialista Eduardo Martins, Professor Adjunto Convidado, ISEP

Vogal: Dr. Bruno Cunha, Professor Auxiliar, Universidade Portucalense

Porto, Junho 2025



# Agradecimentos

A concretização deste trabalho representa o culminar de um ciclo académico repleto de aprendizagens, desafios e crescimento pessoal. Ao longo deste percurso, tive o privilégio de contar com o apoio, incentivo e contributo de várias pessoas, às quais dou o meu mais profundo agradecimento.

Em primeiro lugar, agradeço ao Instituto Superior de Engenharia do Porto (ISEP), pela excelente formação, pelas oportunidades de desenvolvimento académico e pela excelência do corpo docente, que ao longo dos anos contribuiu decisivamente para a minha evolução como pessoa.

Quero expressar um agradecimento muito especial ao meu orientador, Engenheiro André Serra e Santos, pela orientação constante prestada ao longo desta dissertação. O seu aconselhamento e suporte foram fundamentais para a realização deste trabalho.

À minha família, expresso uma imensa gratidão. O vosso amor, sacrifício, confiança e incentivo permanente foram o alicerce que sustentou todo o meu percurso académico. Sem o vosso apoio, este trabalho não teria sido possível.

Aos meus amigos, que estiveram sempre presentes nos bons e maus momentos, o meu sincero obrigado. Pela amizade, pelas palavras de encorajamento nos momentos de maior pressão e pelas pausas necessárias para descontrair. Vocês contribuíram para que esta etapa fosse superada com equilíbrio e motivação.

Por último, a todos os que, de alguma forma, fizeram parte deste caminho, o meu mais sincero agradecimento.



# Resumo

O Escalonamento das Operações é um elemento fundamental para o funcionamento eficiente de indústrias de serviços e manufatura, sendo o processo de decisão responsável pela alocação de recursos limitados entre atividades ao longo do tempo. Apesar de um bom Escalonamento das Operações proporcionar diversas vantagens, este acarreta alguns problemas de elevada complexidade e de difícil resolução. As Metaheurísticas surgem como técnicas de otimização que podem ser aplicadas na resolução destes problemas atingindo uma solução aproximada, pelo que a sua utilização tem vindo a crescer ao longo dos anos. Neste trabalho, foram selecionadas quatro Metaheurísticas (*Genetic Algorithm*, *Particle Swarm Optimization*, *Tabu Search* e *Simulated Annealing*) e um tipo específico de Problema de Escalonamento Industrial (Job-Shop com *makespan* como critério de otimização), com base na sua relevância e popularidade na comunidade científica, para serem o alvo de foco de testes estatísticos. Para cada Metaheurística foram reunidos dados de vinte artigos científicos, posteriormente os dados foram organizados para servirem de base à análise estatística. Inicialmente, procedeu-se à realização de uma análise descritiva com os dados dos artigos, com o objetivo de compreender a distribuição dos artigos por diferentes categorias e extrair observações preliminares. Seguidamente, com recurso ao *software* estatístico IBM SPSS *Statistics*, realizaram-se cinco análises de inferência estatística: Evolução dos Resultados ao longo dos Anos, Comparação de Desempenho entre as Metaheurísticas, Variabilidade de Resultados dentro de cada Metaheurística, Influência dos Métodos de Comparação de Resultados e Análise por tipo de Metaheurística (Híbrida e Não Híbrida). Estas análises permitiram chegar a diversas conclusões, entre as quais se destaca a ausência de correlação significativa entre o ano de publicação e a proximidade ao resultado ótimo, por de parte das Metaheurísticas, a evidência de que o *Genetic Algorithm* proporciona, em média, as maiores reduções de *makespan* entre as Metaheurísticas analisadas e o facto de este algoritmo também se destacar pela sua consistência, apresentando a menor variabilidade de resultados. Assim, este trabalho oferece uma fonte de informação valiosa para a seleção de Metaheurísticas, no contexto industrial, mais especificamente para os Problemas de Escalonamento Industrial do tipo Job-Shop que apresentam *makespan* como critério de otimização, realçando o facto que a seleção de uma Metaheurística deve ser baseada numa avaliação multicritério, e não apenas com base no desempenho médio de cada Metaheurística.

**Palavras-chave:** Escalonamento das Operações, Metaheurísticas, Hibridização, Job-Shop, *Makespan*, Resultado Ótimo, Análise Descritiva, Inferência Estatística



# Abstract

Operations Scheduling is a fundamental element in the efficient operation of service and manufacturing industries and is the decision-making process responsible for allocating limited resources between activities over time. Despite the fact that good Operations Scheduling offers a number of advantages, it also entails some highly complex problems that are difficult to solve. Metaheuristics emerge as optimization techniques that can be applied to solve these problems and achieve an approximate solution, which is why their use has been growing over the years. In this work, four Metaheuristics (Genetic Algorithm, Particle Swarm Optimisation, Tabu Search and Simulated Annealing) and a specific type of Industrial Scheduling Problem (Job-Shop with makespan as an optimization criterion) were selected, based on their relevance and popularity in the scientific community, to be the focus of statistical tests. For each Metaheuristic, data was gathered from twenty scientific articles and then organized to serve as the basis for statistical analysis. Initially, a descriptive analysis was carried out on the article data, with the aim of understanding the distribution of articles by different categories and extracting preliminary observations. Then, using IBM SPSS Statistics software, five statistical inference analyses were carried out: Evolution of Results over the Years, Comparison of Performance between Metaheuristics, Variability of Results within each Metaheuristic, Influence of Methods for Comparing Results and Analysis by Type of Metaheuristic (Hybrid and Non-Hybrid). These analyses led to a number of conclusions, including the absence of a significant correlation between the year of publication and proximity to the optimum result, on the part of the Metaheuristics, evidence that the Genetic Algorithm provides, on average, the greatest reductions in makespan among the Metaheuristics analyzed and the fact that this algorithm also stands out for its consistency, showing the least variability in results. Thus, this work offers a valuable source of information for the selection of Metaheuristics, in the industrial context, more specifically for Industrial Job-Shop Scheduling Problems that have makespan as an optimization criterion, emphasizing the fact that the selection of a Metaheuristic should be based on a multi-criteria evaluation, and not just on the average performance of each Metaheuristic.

**Keywords:** Operations Scheduling, Metaheuristics, Hybridization, Job-Shop, Makespan, Optimal Result, Descriptive Analysis, Statistical Inference



# Índice

Lista de Figuras.....	xi
Lista de Tabelas.....	xiii
Acrónimos e Siglas .....	xv
1. Introdução .....	1
1.1. Contextualização .....	1
1.2. Objetivos .....	2
1.3. Metodologia .....	3
1.4. Estrutura do Relatório .....	3
2. Revisão Bibliográfica .....	5
2.1. Escalonamento das Operações.....	5
2.1.1. Problemas de Escalonamento .....	6
2.1.1.1. Problema de Afetação.....	7
2.1.1.2. Problema de Sequenciação .....	7
2.1.1.3. Problema de Calendarização .....	7
2.1.2. Definições.....	8
2.1.3. Representação .....	9
2.1.3.1. Ambiente de Produção .....	9
2.1.3.2. Características ou Restrições .....	10
2.2. Metaheurísticas.....	11
2.2.1. Conceitos Básicos .....	15
2.2.1.1. Codificação.....	15
2.2.1.2. Avaliação.....	16
2.2.1.3. Estruturas de Vizinhança.....	17
2.2.2. Simulated Annealing .....	18
2.2.3. Genetic Algorithm .....	20
2.3. Análise Bibliométrica.....	24
2.3.1. Revisão do Estado da Arte – Casos de Estudo .....	25
3. Métodos e Aplicação.....	31
3.1. Problemas de Job-Shop .....	32
3.1.1. Genetic Algorithm .....	32
3.1.2. Particle Swarm Optimization.....	39
3.1.3. Tabu Search.....	45
3.1.4. Simulated Annealing .....	51
4. Resultados e Discussão .....	59
4.1. Análise Descritiva .....	59

4.2. Inferência Estatística e Discussão de Resultados .....	67
4.2.1. Análise de Evolução dos Resultados ao longo dos Anos.....	68
4.2.2. Análise de Comparação de Desempenho entre as Metaheurísticas .....	76
4.2.3. Análise da Variabilidade de Resultados dentro de cada Metaheurística .....	80
4.2.4. Análise da Influência dos Métodos de Comparação de Resultados .....	81
4.2.5. Análise por tipo de Metaheurística: Híbrida e Não Híbrida .....	83
4.2.6. Conclusões das Análises Estatísticas .....	87
5. Conclusão.....	91
5.1. Conclusões Finais.....	91
5.2. Limitações e Trabalhos Futuros .....	93
Referências .....	95
Declaração de Integridade .....	105
Apêndice A .....	107
Apêndice B .....	111
Apêndice C .....	121
Apêndice D .....	127
Apêndice E .....	129

# Lista de Figuras

Figura 1 - Classificação de Metaheurísticas inspiradas na Natureza (adaptado de [37]) .....	14
Figura 2 - Tipos de Codificação Linear .....	16
Figura 3 - Tipos de Estrutura de Vizinhança comuns de Permutação .....	17
Figura 4 - Representação do funcionamento do Simulated Annealing (adaptado de [32]) .....	19
Figura 5 - Métodos de Crossover .....	22
Figura 6 - Distribuição Temporal de Resultados de Artigos .....	61
Figura 7 - Distribuição de Resultados de Artigos por Metaheurística .....	61
Figura 8 - Distribuição de Resultados Ótimos de Artigos por Metaheurística .....	62
Figura 9 - Distribuição de Resultados de Artigos por Método de Comparação .....	62
Figura 10 - Distribuição de Métodos de Comparação por Metaheurística .....	63
Figura 11 - Distribuição de Metaheurísticas Híbridas e Não Híbridas por tipo de Metaheurística .....	64
Figura 12 - Variação de <i>Makespan</i> (%) por Metaheurística .....	65
Figura 13 - Proximidade de Resultado Ótimo (%) por Metaheurística .....	65
Figura 14 - Variação de <i>Makespan</i> (%) por Método de Comparação .....	66
Figura 15 - Variação de <i>Makespan</i> (%) por Metaheurística Híbrida ou Não Híbrida .....	66
Figura 16 - Proximidade de Resultado Ótimo (%) por Metaheurística Híbrida ou Não Híbrida .....	67
Figura 17 - Gráfico de Dispersão das Metaheurísticas relativamente à Variação de <i>Makespan</i> (%) .....	68
Figura 18 - Gráfico de Dispersão das Metaheurísticas relativamente à Proximidade de Resultado Ótimo (%) .....	69
Figura 19 - Gráfico de Dispersão do GA relativamente à Variação de <i>Makespan</i> (%) .....	70
Figura 20 - Gráfico de Dispersão do GA relativamente à Proximidade de Resultado Ótimo (%) .....	71
Figura 21 - Gráfico de Dispersão do PSO relativamente à Variação de <i>Makespan</i> (%) .....	71
Figura 22 - Gráfico de Dispersão do PSO relativamente à Proximidade de Resultado Ótimo (%) .....	72
Figura 23 - Gráfico de Dispersão do TS relativamente à Variação de <i>Makespan</i> (%) .....	73
Figura 24 - Gráfico de Dispersão do TS relativamente à Proximidade de Resultado Ótimo (%) .....	74
Figura 25 - Gráfico de Dispersão do SA relativamente à Variação de <i>Makespan</i> (%) .....	75
Figura 26 - Gráfico de Dispersão do SA relativamente à Proximidade de Resultado Ótimo (%) .....	75
Figura 27 - CV da Variação de <i>Makespan</i> para cada Metaheurística .....	81
Figura 28 - CV da Proximidade ao Ótimo para cada Metaheurística .....	81



# Lista de Tabelas

Tabela 1 - <i>Template</i> do algoritmo de <i>Simulated Annealing</i> (adaptado de [32]) .....	19
Tabela 2 - <i>Template</i> do algoritmo de <i>Genetic Algorithm</i> (adaptado de [59]) .....	23
Tabela 3 - Pesquisa de Palavras-Chave .....	24
Tabela 4 - Classificação de 20 Artigos relevantes .....	26
Tabela 5 - 20 Artigos de <i>Genetic Algorithm</i> para Problemas de Job-Shop .....	33
Tabela 6 - 20 Artigos de <i>Particle Swarm Optimization</i> para Problemas de Job-Shop .....	39
Tabela 7 - 20 Artigos de <i>Tabu Search</i> para Problemas de Job-Shop .....	45
Tabela 8 - 20 Artigos de <i>Simulated Annealing</i> para Problemas de Job-Shop .....	51
Tabela 9 - Estatísticas Descritivas Gerais .....	64
Tabela 10 - Regressão Linear Simples das Metaheurísticas relativamente à Variação de <i>Makespan</i> (%) .....	68
Tabela 11 - Regressão Linear Simples das Metaheurísticas relativamente à Proximidade de Resultado Ótimo (%) .....	69
Tabela 12 - Regressão Linear Simples do GA relativamente à Variação de <i>Makespan</i> (%) .....	70
Tabela 13 - Regressão Linear Simples do GA relativamente à Proximidade de Resultado Ótimo (%) .....	71
Tabela 14 - Regressão Linear Simples do PSO relativamente à Variação de <i>Makespan</i> (%) .....	72
Tabela 15 - Regressão Linear Simples do TS relativamente à Variação de <i>Makespan</i> (%) .....	73
Tabela 16 - Regressão Linear Simples do TS relativamente à Proximidade de Resultado Ótimo (%) .....	74
Tabela 17 - Regressão Linear Simples do SA relativamente à Variação de <i>Makespan</i> (%) .....	75
Tabela 18 - Regressão Linear Simples do SA relativamente à Proximidade de Resultado Ótimo (%) .....	76
Tabela 19 - Teste de Shapiro-Wilk para os dados da Variação de <i>Makespan</i> (%) por Metaheurística .....	76
Tabela 20 - Teste de Levene para os dados da Variação de <i>Makespan</i> (%) por Metaheurística .....	77
Tabela 21 - Teste de Kruskal-Wallis para os dados da Variação de <i>Makespan</i> (%) por Metaheurística .....	77
Tabela 22 - Teste de Shapiro-Wilk para os dados da Proximidade de Resultado Ótimo (%) por Metaheurística .....	78
Tabela 23 - Teste de Levene para os dados da Proximidade de Resultado Ótimo (%) por Metaheurística .....	79
Tabela 24 - Teste de Kruskal-Wallis para os dados da Proximidade de Resultado Ótimo (%) por Metaheurística .....	79
Tabela 25 - Teste de Shapiro-Wilk para os dados da Variação de <i>Makespan</i> (%) por Método de Comparação .....	82
Tabela 26 - Teste de Levene para os dados da Variação de <i>Makespan</i> (%) por Método de Comparação .....	82

Tabela 27 - Teste de Kruskal-Wallis para os dados da Variação de <i>Makespan</i> (%) por Método de Comparação .....	83
Tabela 28 - Teste de Shapiro-Wilk para os dados da Variação de <i>Makespan</i> (%) por Tipo de Metaheurística .....	84
Tabela 29 - Teste de Levene para os dados da Variação de <i>Makespan</i> (%) por Tipo de Metaheurística .....	84
Tabela 30 - Teste de Mann-Whitney para os dados da Variação de <i>Makespan</i> (%) por Tipo de Metaheurística .....	85
Tabela 31 - Teste de Shapiro-Wilk para os dados da Proximidade de Resultado Ótimo (%) por Tipo de Metaheurística .....	86
Tabela 32 - Teste de Levene para os dados da Proximidade de Resultado Ótimo (%) por Tipo de Metaheurística .....	86
Tabela 33 - Teste de Mann-Whitney para os dados da Proximidade de Resultado Ótimo (%) por Tipo de Metaheurística .....	87

# Acrónimos e Siglas

## Lista de Acrónimos e Siglas

ABC	<i>Artificial Bee Colony</i>
ACO	<i>Ant Colony Optimization</i>
AISA	<i>Artificial Immune and Simulated Annealing</i>
BObGA	<i>Multi-objective Genetic Algorithm</i>
BObTS	<i>Multi-objective Tabu Search Algorithm</i>
CEGA-DRL	<i>Co-Evolutionary Genetic Algorithm integrated with Deep Reinforcement Learning</i>
CPSO	<i>Cultural Particle Swarm Optimization</i>
CV	<i>Coeficiente de Variação</i>
DDPSO	<i>Decline Disturbance Particle Swarm Optimization</i>
DPEST	<i>Dissertação/Projeto/Estágio</i>
DTSMA	<i>Decentralized Tabu Search in Multi Agent system</i>
EDD	<i>Earliest Due Date</i>
eMPGA	<i>enhanced Multi-Population Genetic Algorithm</i>
ENSGA II	<i>Enhanced NSGA II</i>
FAM-SPT	<i>First Available Machine-Shortest Processing Time</i>
GA	<i>Genetic Algorithm</i>
GAM	<i>Genetic Algorithm-Mixed</i>
GA_OP	<i>Genetic Algorithm with Operation based representation</i>
GTS	<i>Genetic Algorithm with Tabu Search and Shifting bottlenecks</i>
HDPSO	<i>Hybrid Discrete Particle Swarm Optimization</i>
HES-SA	<i>Hybrid Evolution Strategies-Simulated Annealing</i>
HGA	<i>Hybrid Genetic Algorithm</i>
H-GA-TS	<i>Hybrid Genetic Algorithm and Tabu Search</i>
HGTS	<i>Hybrid Genetic Tabu Search</i>
HICSA	<i>Hybrid Imperialist Competitive Simulated Annealing</i>
HILS	<i>Hybrid Iterated Local Search</i>
HMQPSO	<i>Hybrid Multi-Phase Quantum Particle Swarm Optimization</i>
HPSO	<i>Hybrid Particle Swarm Optimization</i>

HPTS	<i>Hybrid of Particle swarm optimization, Tabu search and Simulated annealing</i>
HQPSO	<i>Hybrid Quantum Particle Swarm Optimization</i>
HSAQ	<i>Hybrid Simulated Annealing with Quenching</i>
HSGA	<i>Hybrid Search Genetic Algorithm</i>
HSISA	<i>Hybrid Sorting Immune Simulated Annealing</i>
HTSA	<i>Hybrid Tabu Search Algorithm</i>
ICA	<i>Imperialist Competitive Algorithm</i>
IDESAA	<i>Improved Differential Evolution Simulated Annealing Algorithm</i>
IGA	<i>Improved Genetic Algorithm</i>
IGA-FCSSVNS	<i>Improved Genetic Algorithm Hybrid with Feasibility Correction Strategy and Self-Learning Variable Neighborhood Search</i>
IGA4F	<i>Improved Genetic Algorithm for FJSP-F</i>
ISEP	Instituto Superior de Engenharia do Porto
ITS	<i>Intensification-driven Tabu Search</i>
LAHC-HF	<i>Late-Acceptance Hill Climbing-Homayouni &amp; Fontes</i>
MAGATS	<i>Multi-Agent Genetic Algorithm Tabu Search</i>
MATS	<i>Multi-Agent Tabu Search</i>
MCDE/TS	<i>Multi-operator based Differential Evolution with a Communication strategy Tabu Search</i>
MEM	Mestrado em Engenharia Mecânica
MGA	<i>Memetic Genetic Algorithm</i>
MIP	<i>Mixed Integer Programming</i>
MISSA	<i>Multi-strategy Improved Sparrow Search Algorithm</i>
MOHPIOSA	<i>Multi-Objective Hybrid Pigeon Inspired Optimization and Simulated Annealing algorithm</i>
MOPSO	<i>Multi-Objective Particle Swarm Optimization</i>
MRP	<i>Material Requirements Planning</i>
MTEA	<i>Multiobjective Tabu Evolutionary Algorithm</i>
MTSA	<i>Multi-Temperature Simulated Annealing</i>
MuSTAM	<i>Multi-Start Tabu Search-based Multi-Agent Model</i>
NN	<i>Neural Network</i>
PAP	Planeamento Agregado de Produção

PDP	Plano Diretor de Produção
PE	Planeamento Estratégico
PPSOGA	<i>Parallel Particle Swarm Optimization Genetic Algorithm</i>
PSO	<i>Particle Swarm Optimization</i>
QPSO	<i>Quantum Particle Swarm Optimization</i>
RHPSO	<i>Reactive Hybrid Particle Swarm Optimization</i>
SA	<i>Simulated Annealing</i>
SA-HH	<i>Simulated Annealing-based Hyper-Heuristic</i>
SLGA	<i>Self-Learning Genetic Algorithm</i>
SSA	<i>Sparrow Search Algorithm</i>
TDOE	<i>Taguchi's Design of Experiment</i>
TS	<i>Tabu Search</i>
TS-NSGA	<i>Two-Stage Non-dominated Sorting Genetic Algorithm</i>
TSPCB	<i>Tabu Search algorithm with a fast Public Critical Block neighborhood structure</i>



# 1. Introdução

O trabalho desenvolvido foi realizado no âmbito da unidade curricular de Dissertação/Projeto/Estágio (DPEST), inserida no 2º semestre do 2º ano do Mestrado em Engenharia Mecânica (MEM) – ramo de Gestão Industrial do Instituto Superior de Engenharia do Porto (ISEP).

Este capítulo é constituído por quatro subcapítulos, de forma a dar uma contextualização ao trabalho realizado, identificar os objetivos, expor a metodologia de investigação utilizada e terminando com uma explicação da estrutura do relatório e dos conteúdos abordados.

## 1.1. Contextualização

Atualmente e mais do que nunca, as empresas são confrontadas com inúmeros desafios e dificuldades que as pressionam para atender a procura incessante de um mercado em contínuo crescimento. Para tal, têm que continuar a se adaptar ao que o mercado lhes impõe, de forma a tentar manter os seus níveis de competitividade. Assim, olhando numa perspetiva industrial, as indústrias têm o papel de fornecer produtos e serviços de qualidade e a preços adequados. No entanto, isto por si também apresenta as suas dificuldades, sendo que é aqui que entra o Escalonamento de Operações como estratégia para ajudar a atingir estes objetivos.

O Escalonamento das Operações é um componente essencial para a eficiente operação de indústrias de serviços e manufatura. Dado o quão competitivo é o mercado atual, realça-se a importância do Escalonamento das Operações para a grande parte das indústrias, devido ao quão pequenas são as margens de lucro. O Escalonamento das Operações faz a alocação de escassos recursos entre atividades ao longo do tempo. A sua boa aplicação numa indústria tem inúmeras vantagens como a redução de custos, aumento de lucro, redução de tarefas em atraso, entre muitas outras. No entanto, acarreta alguns problemas de complexidade elevada e de difícil resolução, o que leva a muitas indústrias, que não possuem os recursos adequados e necessários para os confrontar, a afastarem-se desta estratégia apesar dos seus diversos benefícios [1].

As Metaheurísticas apresentam-se como técnicas aproximadas, muito utilizadas na resolução destes problemas de otimização de elevada complexidade. São algoritmos que podem ser aplicados na resolução destes problemas atingindo uma solução aproximada, recorrendo a estratégias para explorar de forma eficiente o espaço de soluções e evitar os Ótimos Locais. Existem diversas Metaheurísticas que são utilizadas para resolver problemas complexos de Escalonamento, pelo que o seu número continua a aumentar com o passar dos anos. Novas

Metaheurísticas proporcionam maneiras mais eficientes de abordar problemas de otimização tradicionais e novas formas de atacar problemas ainda sem resolução. É importante ter em conta que, apesar de existirem Metaheurísticas específicas para a resolução de determinados tipos de problemas de otimização, a melhor estratégia consiste em nos familiarizar-nos primeiro com as características das Metaheurísticas e o seu campo de aplicações, antes de as atribuímos a um dado problema de Escalonamento [2].

### **1.2. Objetivos**

O trabalho desenvolvido neste relatório tem como objetivo pesquisar e reunir a informação acerca do Escalonamento das Operações e de Metaheurísticas, e, de seguida, proceder à sua análise e compreensão. Com isto pretende-se não só adquirir um conhecimento mais aprofundado destes dois temas, desde que papéis assumem nas indústrias até às suas características e conceitos-chave cruciais, como também perceber a relação existente entre o Escalonamento das Operações e as Metaheurísticas, de que forma esta relação se procede e o quão importante esta é no tecido industrial atual. Assim, este trabalho tem como papel oferecer uma fonte de informação de boa qualidade e bem estruturada acerca destes temas e, com isto, proceder à análise e comparação de desempenho das Metaheurísticas selecionadas para o trabalho, quando aplicadas a Problemas de Escalonamento Industrial do tipo Job-Shop.

Para a realização dos objetivos referidos proceder-se-á , então, aos seguinte passos:

- Estudo do conceito de Escalonamento de Operações, do seu papel nas indústrias e dos vários tipos de Escalonamento existentes;
- Identificar e compreender os tipos de Problemas de Escalonamento existentes e a respetiva forma de os definir e representar;
- Estudo do conceito de Metaheurísticas e dos seus três conceitos básicos inerentes;
- Identificação e análise dos vários tipos de Metaheurísticas existentes;
- Estudo aprofundado de duas Metaheurísticas mais populares e utilizadas no contexto industrial;
- Estudo e seleção de quatro Metaheurísticas populares no contexto industrial a analisar para o trabalho;
- Estudo e seleção do tipo de Problema de Escalonamento Industrial a analisar para o trabalho;
- Estudo, seleção e organização de artigos científicos considerados relevantes e pertinentes ao tema do trabalho, de acordo com as Metaheurísticas e Problema de Escalonamento Industrial selecionados;
- Exposição e organização dos dados reunidos relativamente aos artigos científicos selecionados, com recurso à realização de uma análise descritiva;

- Análise e comparação de desempenho das Metaheurísticas selecionadas, com recurso à realização de testes de inferência estatística.

### **1.3. Metodologia**

De forma a se alcançar os objetivos estabelecidos anteriormente, adotou-se a seguinte metodologia de pesquisa:

- Pesquisa bibliográfica acerca do Escalonamento de Operações, do seu papel nas indústrias e dos vários tipos de Escalonamento existentes;
- Pesquisa bibliográfica sobre os tipos de Problemas de Escalonamento existentes e respetivas formas de definição e representação;
- Pesquisa bibliográfica acerca de Metaheurísticas e dos três conceitos básicos que lhe são inerentes;
- Pesquisa bibliográfica sobre os tipos de Metaheurísticas existentes;
- Pesquisa bibliográfica acerca das quatro Metaheurísticas e tipo de Problema de Escalonamento Industrial selecionados para o trabalho.

Assim, selecionaram-se primeiro as fontes de informação mais adequadas para o tema do trabalho. De seguida, após uma cuidada leitura e análise da diversa informação reunida, procedeu-se à seleção da mais relevante que iria integrar o relatório e concluiu-se com a devida organização e apresentação desta neste trabalho.

### **1.4. Estrutura do Relatório**

O presente relatório está organizado em cinco capítulos: Introdução, Revisão Bibliográfica, Métodos e Aplicação, Resultados e Discussão e Conclusão.

O primeiro capítulo, inicia-se com uma contextualização ao tema do trabalho, seguido da exposição dos objetivos propostos, da metodologia adotada e da estrutura do relatório.

No segundo capítulo, são contextualizados os dois temas fundamentais deste trabalho, o Escalonamento de Operações e as Metaheurísticas, dividindo-se por isso em dois subcapítulos:

- Introdução ao tema do Escalonamento de Operações, abordando o seu papel nas indústrias e os seus diferentes tipos existentes, bem como a definição dos vários Problemas de Escalonamento usualmente presentes nos processos industriais e a respetiva forma de os definir e representar;
- Introdução ao tema das Metaheurísticas, abordando o papel que assumem relativamente aos problemas complexos de Escalonamento e os vários tipos de Metaheurísticas existentes, caracterizando os três conceitos básicos que lhe são inerentes e realizando uma análise aprofundada acerca das duas Metaheurísticas específicas, consideradas relevantes para o trabalho.

## Introdução

No terceiro capítulo, são selecionadas as quatro Metaheurísticas que são o alvo de análise e comparação deste trabalho, bem como o tipo de Problema de Escalonamento Industrial no qual estas são aplicadas. Seguidamente, reuniram-se artigos científicos relevantes, mediante as Metaheurísticas e tipo de Problema de Escalonamento Industrial selecionados, organizando os dados provenientes deles em tabelas, sendo estes caracterizados através de categorias.

No quarto capítulo, é feita uma análise descritiva dos dados reunidos nos artigos selecionados, com o objetivo de se perceber como estes se distribuem em várias categorias. De seguida, realizaram-se testes de inferência estatística que permitiram retirar diversas conclusões quanto ao desempenho das quatro Metaheurísticas em foco deste trabalho.

No quinto capítulo, são apresentadas as principais conclusões atingidas ao longo da realização do trabalho, das limitações enfrentadas durante o seu desenvolvimento e das sugestões de trabalhos futuros.

## 2. Revisão Bibliográfica

O capítulo 2 vai apresentar e contextualizar os dois temas fundamentais para a realização desta dissertação: Escalonamento das Operações e Metaheurísticas. Inicialmente, será realizado um enquadramento acerca do Escalonamento das Operações, incluído os principais problemas que este apresenta e as diferentes formas de os representar.

Relativamente às Metaheurísticas, será realizado o seu enquadramento e uma explicação de três conceitos essenciais para o seu funcionamento.

### 2.1. Escalonamento das Operações

Um eficaz Escalonamento das Operações faz toda a diferença nos processos de produção modernos, assumindo um papel crucial nas indústrias de serviços e de manufatura. Graves [3], define o Escalonamento das Operações como a alocação de recursos de produção disponíveis ao longo do tempo, de forma a melhor satisfazer um conjunto de critérios bem definidos. Baker [4] define-o como a alocação de recursos ao longo do tempo para se realizar um conjunto de atividades. A partir destas definições pode-se concluir que o Escalonamento das Operações é um processo de tomada de decisão que vai alocar recursos às operações. Consoante a situação, estes recursos e atividades podem assumir diferentes formas. Recursos poderão ser, por exemplo, professores de uma escola ou mecânicos de uma oficina de reparações enquanto as atividades poderão ser os deveres dos professores de uma escola ou as reparações de carros numa oficina de reparações. Assim, podem-se ver as atividades como o que é realizado pelos recursos [5].

Os primeiros estudos relacionados ao Escalonamento das Operações remontam a 1950, quando diversos investigadores das áreas de investigação de operações, engenharia industrial e gestão se depararam com o problema de gerir as várias atividades que ocorriam numa oficina. Bons algoritmos de Escalonamento das Operações poderiam levar à redução dos custos de um processo de manufatura, permitindo à empresa manter-se competitiva. Nos finais de 1960, cientistas de computação começaram também a participar no estudo de Escalonamento das Operações. Dado que os recursos computacionais eram escassos, um uso eficiente destes poderia levar à redução dos custos envolvidos na execução dos programas de computador. Estes dois fatores aliados, ofereceram uma razão económica para o estudo mais aprofundado do Escalonamento das Operações [5].

O Escalonamento das Operações é uma das subfases do Planeamento e Controlo da Produção, sendo que este, por sua vez, se encontra dividido em cinco fases: Planeamento Estratégico (PE), Planeamento Agregado de Produção (PAP), Plano Diretor de Produção (PDP), *Material Requirements Planning* (MRP) e Programação da Produção. Assim, o Escalonamento das Operações é realizado na última fase do Planeamento e Controlo da Produção, ou seja, durante a Programação da Produção, caracterizando-se por um elevado nível de detalhe e um horizonte temporal curto [6].

As características da indústria são cruciais para determinar o tipo de Escalonamento, sendo que existem quatro tipos principais:

- Escalonamento Estático: todas as atividades são conhecidas e preparadas desde o início, não se alterando durante todo o processo de manufatura, independentemente de novas atividades aparecerem no decorrer deste. Este tipo de Escalonamento é notório pela sua falta de flexibilidade [7, 8];
- Escalonamento Dinâmico: novas atividades podem ser adicionadas durante o processo, datas de entrega podem ser alteradas, bem como o estado de algumas das máquinas. Deste modo, caracteriza-se por considerar acontecimentos em tempo real [7, 8];
- Escalonamento Determinístico: é assumido que os parâmetros das atividades, como tempos de processamento, são conhecidos à priori e com precisão, o que ocorre tipicamente em Problemas de Escalonamento académicos [9];
- Escalonamento Estocástico: contrariamente ao Escalonamento Determinístico, os parâmetros das atividades, como tempos de processamento ou datas de entrega, não são conhecidos com precisão, tornando-se estes parâmetros apenas conhecidos quando acabam por ocorrer. Este tipo de escalonamento aproxima-se mais da realidade pois apresenta a forte componente de incerteza e aleatoriedade presente nos ambientes reais de produção [9].

### **2.1.1. Problemas de Escalonamento**

Problemas de Escalonamento podem ser vistos, de forma geral, como problemas relacionados com a alocação de recursos ao longo do tempo para realizar um conjunto de atividades pertencentes a processos, de entre eles sendo os computacionais e os de manufatura os mais importantes [10].

Existem dois tipos de restrições exequíveis que normalmente se encontram nos Problemas de Escalonamento: os limites existentes nas capacidades das máquinas e as restrições tecnológicas na ordem da qual algumas atividades podem ser realizadas. Baker & Trietsche [11] descrevem o Escalonamento das Operações como um processo de decisão, que, de forma a resolver estes dois tipos de restrições e solucionar um Problema de Escalonamento, procura responder a duas questões:

- Quais recursos devem ser alocados para realizar cada operação?

- Quando é que cada operação deve ser realizada?

A partir destas duas questões pode-se concluir que o Escalonamento das Operações envolve afetação (distribuição das operações aos recursos), sequenciação (definição da ordem temporal mais correta para a realização das operações) e calendarização (distribuição das operações no tempo). Ou seja, um Problema de Escalonamento divide-se em Problemas de Afetação, Problemas de Sequenciação e Problemas de Calendarização [4].

Normalmente, um Problema de Escalonamento resolve-se de forma ordenada começando primeiro pela afetação, seguido da sequenciação e finalmente a calendarização.

#### **2.1.1.1. Problema de Afetação**

O Problema de Afetação responde à primeira questão colocada por Baker & Trietsche [11]. Um Problema de Afetação existe quando múltiplas operações podem ser realizadas por múltiplos recursos, sendo necessário distribuir estes recursos pelas operações [9].

Blazewicz et al. [10] descrevem o Problema de Afetação da seguinte maneira: “Conhecidas  $n$  operações pertencentes a  $T = \{ T_1, T_2, \dots, T_n \}$ ,  $m$  máquinas pertencentes a  $P = \{ P_1, P_2, \dots, P_m \}$  e  $s$  recursos adicionais pertencentes a  $R = \{ R_1, R_2, \dots, R_s \}$ , como deverão as máquinas de  $P$  e os recursos de  $R$  ser distribuídos pelas operações de  $T$  de forma a otimizar uma determinada medida de desempenho?”.

#### **2.1.1.2. Problema de Sequenciação**

O Problema de Sequenciação responde à segunda questão colocada por Baker & Trietsche [11] pelo que se sucede, na maioria das vezes, ao Problema de Afetação. Um Problema de Sequenciação determina a ordem em que as operações deverão ser executadas. Esta ordem de execução das operações só se poderá realizar se a distribuição das operações pelas máquinas já tiver sido concluída, pelo que, geralmente, o Problema de Sequenciação apenas se resolve após o Problema de Afetação.

O Problema de Sequenciação pode ser descrito da seguinte maneira: dado um caso onde se possui  $n$  operações pertencentes a  $T = \{ T_1, T_2, \dots, T_n \}$  e uma sequência de fabrico  $S = \{ S_1, S_2, \dots, S_n \}$ , como se deve proceder à ordenação das operações  $T$  na sequência  $S$  com o objetivo de se otimizar uma determinada medida de desempenho. Apesar de o Problema de Sequenciação, inicialmente, aparentar ser simples a sua natureza combinatória torna-o bastante complexo [6].

#### **2.1.1.3. Problema de Calendarização**

O Problema de Calendarização também responde à segunda questão colocada por Baker & Trietsche [11] sucedendo-se ao Problema de Sequenciação e correspondendo à última fase do Escalonamento das Operações. Um Problema de Calendarização determina quando é que as operações se deverão realizar nos seus respetivos recursos.

O Problema de Calendarização pode ser descrito da seguinte maneira: dado um caso onde se possui  $n$  operações pertencentes a  $T = \{ T_1, T_2, \dots, T_n \}$ ,  $m$  máquinas pertencentes a  $P = \{ P_1, P_2, \dots, P_m \}$ , uma distribuição das operações  $T$  pelas máquinas de  $P$ , uma sequência de fabrico  $S = \{ S_1, S_2, \dots, S_n \}$  para cada máquina de  $P$  e um intervalo de tempo  $[a,b]$ , como se deve proceder à distribuição das operações  $T$  no intervalo de tempo  $[a,b]$  com o objetivo de se otimizar uma determinada medida de desempenho [6]. Este problema é, então, responsável por determinar o início de todas as operações.

### 2.1.2. Definições

Os Problemas de Escalonamento são uma parte integral do Escalonamento das Operações, sendo que é importante saber e perceber os conceitos chave associados a estes. Desta forma, tem-se atividade ou tarefa, operação, máquina ou recurso e oficina.

Uma *Atividade* ou *Tarefa* é uma unidade básica de trabalho que é realizada num determinado tempo. Uma *Operação* é uma parte de uma atividade que apenas é realizada num único recurso. Uma *Máquina* ou *Recurso* é uma unidade produtiva, responsável pela execução de uma ou mais operações. A *Oficina* corresponde ao local onde os recursos utilizados na execução das diversas operações se encontram disponíveis [6].

Habitualmente, o número de atividades é representado por  $n$  e o número de máquinas por  $m$ , enquanto  $j$  refere-se a uma atividade e  $i$  refere-se a uma máquina. Com isto estabelecido, tem-se as seguintes definições [5, 9]:

- $P_{ij}$  – representa o tempo de processamento da atividade  $j$  na máquina  $i$ . Se uma atividade é só processada em uma só máquina, representa-se por  $p_j$ ;
- $r_j$  – representa a data de lançamento da atividade  $j$ , ou seja, o momento a partir do qual a atividade pode começar o seu processamento;
- $d_j$  – representa a data de entrega para a atividade  $j$ . Se a data de entrega não for respeitada, a encomenda ainda pode ser entregue mas irá acarretar custos adicionais para a empresa;
- $w_j$  – representa o peso da atividade  $j$  em relação às outras atividade do sistema produtivo, ou seja, reflete o seu grau de importância;
- $C_j$  – representa o tempo de conclusão da atividade  $j$ ;
- $L_j$  – representa o atraso ou antecipação de uma atividade em relação à data estabelecida. É definido através da equação (1), que resulta da diferença do tempo de conclusão e da data de entrega de uma atividade;

$$L_j = C_j - d_j \quad (1)$$

- $T_j$  – representa o atraso positivo de uma atividade em relação à data estabelecida. A diferença entre  $T_j$  e  $L_j$  é que o  $T_j$  nunca é negativo. É definido através da equação (2);

$$T_j = \max(C_j - d_j, 0) \quad (2)$$

- $E_j$  – representa o quão cedo uma atividade termina em relação à data estabelecida. É definido através da equação (3), que resulta da diferença da data de entrega e do tempo de conclusão de uma atividade.

$$E_j = \max(d_j - C_j, 0) \quad (3)$$

### 2.1.3. Representação

Os Problemas de Escalonamento, apesar das suas semelhanças, apresentam geralmente características muito diferentes. Isto advém do facto de cada empresa possuir as suas próprias particularidades, a sua própria disposição dos recursos pela oficina e a sua maneira de lidar com as atividades existentes.

Devido a isto, recorrer a nomenclaturas e formas de representação torna-se extremamente benéfico para as empresas, para facilitar a análise dos problemas. Um dos métodos de representação dos Problemas de Escalonamento mais utilizado é o proposto por Graham et al. [12]. Este método apresenta uma representação simples, apresentando os campos  $\alpha|\beta|\gamma$ .  $\alpha$  representa o ambiente de produção,  $\beta$  representa as características ou restrições do processo e  $\gamma$  representa os critérios de otimização. Este método de representação é bastante utilizado por diversos autores, sendo que Pinedo [9], Leung [5] e Blazewicz et al. [10] usam variações desta.

Este documento vai utilizar o método de representação de Graham et al, mais especificamente a variação usada por Pinedo, devido a representar com o devido detalhe as diferentes características dos Problemas de Escalonamento.

#### 2.1.3.1. Ambiente de Produção

Como foi dito anteriormente, o ambiente de produção é representado por  $\alpha$ , de acordo com a representação de Graham et al. [12]. O campo  $\alpha$  indica o número de recursos e de que forma se dispõem pela oficina. Pinedo [9] descreve os nove tipos de ambientes máquina que se podem encontrar:

- **1** – Representa os Problemas de Escalonamento de Máquina Única. O caso da máquina única é caracterizado por ser o mais simples, existindo apenas um único recurso, compreendendo apenas Problemas de Sequenciação e Calendarização. É normalmente utilizado na resolução de problemas de maior complexidade;
- $P_m$  – Representa os Problemas de Escalonamento de Máquinas Paralelas Idênticas. Apresentam velocidades iguais pelo que qualquer máquina  $m$  poderá executar qualquer dada atividade;
- $Q_m$  – Representa os Problemas de Escalonamento de Máquinas Paralelas com diferentes velocidades. Como o nome indica, apresentam diferentes velocidades pelo que o tempo de processamento irá variar de acordo com o desempenho de cada máquina;

- $R_m$  – Representa os Problemas de Escalonamento de Máquinas não relacionadas. As máquinas apresentam características diferentes pelo que o tempo de processamento irá variar entre as diferentes máquinas;
- $F_m$  – Representa os Problemas de Escalonamento de *Flow Shop*. Existem  $m$  máquinas dispostas em série, sendo que cada atividade terá que ser processada em cada uma das  $m$  máquinas e seguir a mesma sequência específica de operações;
- $FF_c$  – Representa os Problemas de Escalonamento de *Flexible Flow Shop*. Trata-se de uma generalização dos ambientes de *Flow Shop* e Máquinas Paralelas. Invés de  $m$  máquinas em série, existem  $c$  níveis em série, cada um com um número determinado de máquinas em paralelo;
- $J_m$  – Representa os Problemas de Escalonamento de *Job Shop*. Existem  $m$  máquinas, onde cada atividade possui a sua própria sequência de processamento, diferindo dos Problemas de *Flow Shop* por não possuírem um fluxo de trabalho unidirecional;
- $FJ_c$  – Representa os Problemas de Escalonamento de *Flexible Job Shop*. Trata-se de uma generalização dos ambientes de *Job Shop* e Máquinas Paralelas. Invés de  $m$  máquinas em série, existem  $c$  centros de trabalho, cada um com um número determinado de máquinas em paralelo;
- $O_m$  – Representa os Problemas de Escalonamento de *Open Shop*. Existem  $m$  máquinas, sendo que não existem restrições em relação à sequência de processamento das atividades. Assim, as operações das atividades podem ser executadas por qualquer sequência.

### 2.1.3.2. Características ou Restrições

Como já foi referido, as características ou restrições do processo de fabrico são representadas por  $\beta$ , de acordo com a representação de Graham et al. [12]. O campo  $\beta$  indica as especificações das atividades e limitações existentes no sistema produtivo. Estas características ou restrições podem incluir múltiplas entradas sendo que Pinedo [9] descreve as seguintes:

- $R_j$  – Representa as Datas de Lançamento. O processamento das atividades só se inicia após a sua data de lançamento;
- $prmp$  – Representa as Interrupções. Implica que não é necessário manter uma atividade numa máquina até esta terminar. É possível interromper o processamento de uma atividade numa dada máquina e substituí-la por outra sem se perder o trabalho já realizado;
- $prec$  – Representa as Precedências. Significa que uma dada atividade só poderá ser realizada quando forem concluídas as atividades que a precedem;
- $S_{jk}$  – Representa os Tempos de *Setup* dependentes da Sequência. Os tempos de *setup* variam de acordo com a sequência de processamento das atividades  $j$  e  $k$ ;

- **fmls** – Representa as Famílias de Atividades. Existem  $n$  atividades pertencentes a  $F$  diferentes famílias de atividades. Atividades da mesma família podem ter diferentes tempos de processamento mas não precisarem de tempo de preparação entre si para o seu processamento;
- **batch (b)** – Representa o Processamento em Lotes. Uma máquina é capaz de processar  $b$  atividades, em simultâneo. Os tempos de processamento das atividades podem não ser todos iguais pelo que a atividade com maior tempo de processamento determina o tempo de conclusão do lote;
- **brkdw** – Representa as Avarias das Máquinas. As avarias de máquinas implicam que estas podem não estar continuamente disponíveis. Às máquinas estão-lhe associadas distribuições de fiabilidade que representam a probabilidade de ocorrência de avarias;
- **$M_j$**  – Representa as Restrições de Elegibilidade das Máquinas. Quando o ambiente de máquinas é  $m$  máquinas em paralelo e  $M_j$  está presente então nem todas as máquinas  $m$  são capazes de processar a atividade  $j$ ;
- **prmu** – Representa as Permutações. Após as atividades serem processadas na primeira máquina não é possível alterar a sua sequência, sendo característico de *Flow Shops*;
- **block** – Representa o Bloqueio. É um fenómeno que poderá ocorrer em *Flow Shops*. Se uma dada atividade  $x$  tiver acabado o seu processamento numa máquina mas a atividade anterior não tiver terminado o seu processamento na máquina seguinte então esta atividade  $x$  fica bloqueada, acabando por bloquear também a atividade seguinte de iniciar o processamento na sua máquina;
- **nwt** – Representa Sem Espera. Trata-se de outro fenómeno que poderá ocorrer em *Flow Shops*. As atividades não podem esperar entre duas máquinas consecutivas pelo que só podem ser processadas se, se garantir que não haverá tempo de espera em nenhuma máquina;
- **rcrc** – Representa a Recirculação. As atividades precisam ser processadas várias vezes em uma ou mais máquinas.

## 2.2. Metaheurísticas

O cérebro humano resolve diariamente problemas de otimização recorrendo a Heurísticas, desde, por exemplo, se escolher o caminho mais rápido para casa depois do trabalho ou a escolher um dado artigo para comprar. Apesar disto, o estudo científico de Heurísticas e Metaheurísticas é bastante recente [13, 14], sendo que o próprio termo “Metaheurística” foi apenas introduzido em 1986 por Glover [15].

O termo Metaheurística resulta da combinação de duas palavras “Meta” e “Heurística”. O prefixo grego “Meta” significa “para além, em um nível superior”, enquanto a palavra “Heurística”, que deriva do verbo grego *heuriskein*, significa “encontrar” ou “atingir um objetivo”. Assim, podem-se ver Metaheurísticas como algoritmos que combinam Heurísticas numa estrutura mais generalizada [16, 17].

Ao longo da história várias técnicas de resolução de problemas apresentavam as características de Heurísticas e Metaheurísticas, no entanto a utilização de Heurísticas como técnica científica para otimização é um fenómeno relativamente moderno. Várias Metaheurísticas foram desenvolvidas em diferentes períodos históricos:

- Entre 1940 e 1960 realça-se a difusão dos métodos heurísticos para as mais diversas aplicações. Em 1945 foi publicado o famoso livro *How to Solve It*, pelo matemático húngaro Polya sendo o primeiro documento onde se pode observar o uso de Heurísticas na resolução de problemas matemáticos, através de uma perspetiva científica moderna. A forma como as Heurísticas foram apresentadas no seu livro estabeleceu os fundamentos para as diversas Heurísticas que surgiram nos anos seguintes [18].
- Em 1963 dá-se a introdução de *Evolutionary Algorithms* por Rechenberg e Schwefel [19]. Estes algoritmos são caracterizados pela sua inspiração na Teoria da Evolução de Darwin [20] tentando reproduzir a evolução das espécies [21].
- Em 1975 dá-se o desenvolvimento da primeira Metaheurística Evolutiva por de parte de Holland [22], o *Genetic Algorithm* (GA). Inspirada na seleção natural, começa-se com uma população de soluções que vai evoluindo através de sucessivos cruzamentos, privilegiando-se as soluções de melhor qualidade [23].
- Em 1983 deu-se um dos maiores passos relativamente ao desenvolvimento das Metaheurísticas. Kirkpatrick et al. [24] apresentou a primeira Metaheurística que não era inspirada na seleção natural, o *Simulated Annealing* (SA). Esta Metaheurística caracterizava-se por não utilizar uma população de soluções, como as Metaheurísticas Evolutivas, inspirando-se no processo de recozimento natural dos sólidos.
- Em 1986 Glover para além de apresentar o termo “Metaheurística”, propõe outra Metaheurística que não é inspirada na seleção natural, a *Tabu Search* (TS). Tratou-se da primeira vez em que “memória” foi utilizada em uma Metaheurística, em que os movimentos de uma pesquisa anterior eram armazenados numa *Tabu List*, proibindo a repetição de movimentos iguais dali para a frente [15].
- Entre 1990 e 2000 houve o desenvolvimento de diversas Metaheurísticas importantes, destacando-se principalmente as de *Swarm Intelligence* inspiradas no comportamento de enxames presentes na natureza, como a *Ant Colony Optimization* (ACO) e *Particle Swarm Optimization* (PSO) [14, 25].
- Entre 2000 e os dias de hoje desenvolveram-se inúmeras novas Metaheurísticas realçando alguns exemplos como a *Harmony Search*, a *Artificial Bee Colony* e a *Cuckoo Search* [25].

Este crescimento consistente no desenvolvimento de novas Metaheurísticas deveu-se à necessidade de resolução de problemas cada vez mais complexos, em que os métodos tradicionais já não eram suficientes. As Metaheurísticas apresentam-se assim como soluções para resolver estes problemas em tempos e com esforço adequados.

Após se conhecer as diversas Metaheurísticas existentes e compreender o seu papel na nossa sociedade é importante perceber o que realmente é uma Metaheurística, os seus conceitos, comportamentos e propriedades associados a ela e as diferentes classificações que esta pode vir a ter. Osman e Laporte [26] definem Metaheurística como *“um processo de geração iterativo que guia uma heurística subordinada, ao combinar de forma inteligente diferentes conceitos para explorar e aproveitar o espaço de pesquisa, estratégias de ensino são utilizadas para estruturar informação com o intuito de encontrar de forma eficiente soluções otimizadas”*.

O comportamento de uma Metaheurística é determinado através de dois conceitos: Intensificação e Diversificação. Apesar de se tratar de conceitos um tanto contrários acabam por se complementar, sendo eles responsáveis por orientar as pesquisas através do espaço de soluções. Enquanto a Intensificação consiste em focar a pesquisa nas zonas mais atrativas do espaço de soluções, com o intuito de melhorar a solução atual, a Diversificação consiste em manter uma visão global do espaço de soluções [27, 28].

Achar um equilíbrio entre a Intensificação e a Diversificação, em relação às Metaheurísticas, torna-se essencial. Um bom equilíbrio permite identificar zonas no espaço de soluções com soluções de alta qualidade e por outro lado não se desperdiça tanto tempo em zonas no espaço de soluções que já foram exploradas ou então que não oferecem soluções de alta qualidade. Caso não exista o equilíbrio entre estes dois conceitos podem-se suceder complicações. Um excesso na Intensificação leva o algoritmo a convergir em Ótimos Locais e um excesso na Diversificação leva à falta de eficiência do algoritmo, dificultando a procura por soluções de alta qualidade. Importante ainda referir que as estratégias de pesquisa de cada Metaheurística dependem do seu tipo, variando consoante cada Metaheurística [29, 30].

Existem várias formas para classificar as Metaheurísticas, consoante as características que as diferenciam umas das outras. Encontram-se de seguida os métodos de classificação utilizados com maior frequência:

- Inspirada na Natureza/Não Inspirada na Natureza [31, 32, 33, 34]
  - Inspirada na Natureza: Metaheurísticas inspiradas por processos naturais. Estes algoritmos usufruem das características inerentes da natureza para o seu desenvolvimento. Exemplos destas Metaheurísticas são o GA, ACO, *Honey-bee Mating Optimization*, PSO, *Firefly Algorithm*, *Bat Algorithm*, entre outras.
  - Não Inspirada na Natureza: como o nome indica são Metaheurísticas do qual as suas origens não são relacionadas com processos naturais mas sim com computação. Exemplos destas Metaheurísticas são a TS, *Iterated Local Search*, entre outras.
- Baseada numa Solução Única/Baseada em População [32, 33, 34, 35]

- Baseada numa Solução Única: também conhecidas por “métodos de trajetória”, estas Metaheurísticas geram uma única solução, tentando melhorar esta em cada iteração consecutiva até se atingir uma solução de alta qualidade. Possuem a característica de intensificarem a procura em regiões locais. Exemplos destas Metaheurísticas são o SA, TS, *Greedy Randomized Adaptive Search Procedure*, *Variable Neighborhood Search*, *Iterated Local Search*, *Guided Local Search*, entre outras.
- Baseada em População: estas Metaheurísticas utilizam diversas soluções invés de uma só permitindo-lhes uma melhor diversificação no espaço de soluções. Exemplos destas Metaheurísticas são o GA, *Firefly Algorithm*, *Bat Algorithm*, PSO, ACO, *Evolutionary Algorithms*, entre outras.
- Baseada em Memória/Sem Memória [29, 32, 33, 34]
  - Baseada em Memória: estas Metaheurísticas utilizam a informação armazenada durante o processo iterativo para guiar a pesquisa futura em torno de uma solução de alta qualidade. Exemplos destas Metaheurísticas são o TS, ACO, entre outras.
  - Sem Memória: estas Metaheurísticas utilizam apenas a informação disponível no momento do processo de procura, não considerando soluções passadas. Exemplos destas Metaheurísticas são o SA, *Local Search*, *Greedy Randomized Adaptive Search Procedure*, entre outras.

Com o passar do tempo não só os algoritmos já existentes foram melhorando através dos esforços dos seus investigadores, como novas Metaheurísticas foram propostas. Consequentemente, métodos de classificação mais antigos de Metaheurísticas tiveram que se adaptar a esta evolução, ou seja, tiveram que se dividir em subcategorias. Exemplo disto são as Metaheurísticas inspiradas na Natureza que se podem classificar em quatro grupos, como se pode observar na Figura 1 [36, 37].

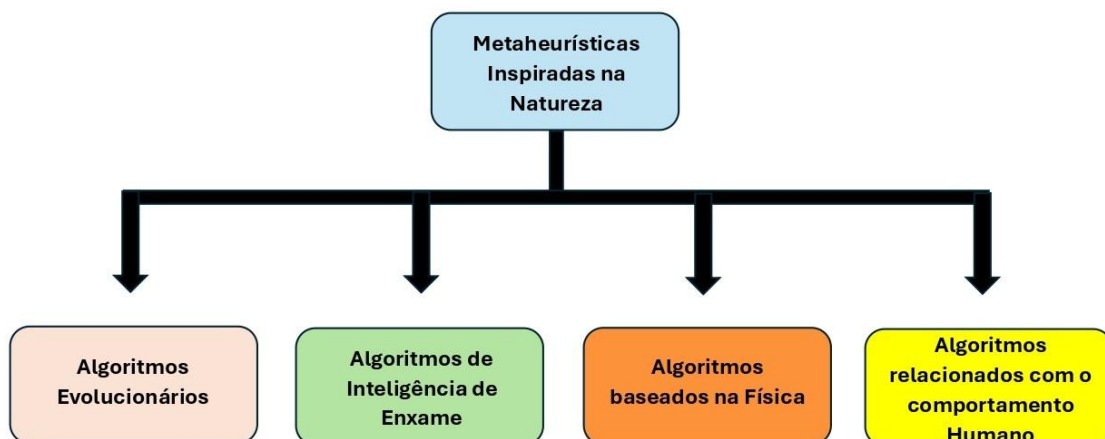


Figura 1 - Classificação de Metaheurísticas inspiradas na Natureza (adaptado de [37])

- Algoritmos Evolucionários: são inspirados pela evolução natural. Estes algoritmos começam o seu processo com uma população de soluções gerada aleatoriamente, produzindo novas soluções através de cruzamentos e mutações, restando apenas as melhores soluções;
- Algoritmos de Inteligência de Enxame: são inspirados pelos comportamentos sociais de insetos, animais, peixes, pássaros, entre outros.
- Algoritmos baseados na Física: são inspirados pelas regras da física no universo;
- Algoritmos relacionados com o comportamento social Humano: como o nome indica são inspirados puramente no comportamento social Humano, desde as suas atividades físicas até atividades não físicas como pensar.

Em suma, existem muitos mais diversos e variados métodos de classificação de Metaheurísticas para além deste, resultante da imensidão de diferentes Metaheurísticas existentes, pelo que este número continuará a crescer continuamente dado o ritmo com que novos algoritmos heurísticos são desenvolvidos atualmente. Stegherr et al. [38] apresenta uma revisão detalhada dos diferentes métodos de classificação de Metaheurísticas utilizados ao longo dos anos.

### **2.2.1. Conceitos Básicos**

Existem três conceitos importantes inseparáveis das Metaheurísticas que apesar de não se incluírem no seu procedimento são fundamentais para a sua implementação: Codificação, Avaliação e as Estruturas de Vizinhança. Estes três conceitos permitem representar, avaliar e manipular soluções [32].

#### **2.2.1.1. Codificação**

Dado que as Metaheurísticas precisam de codificar primeiramente o problema de forma a reproduzir as soluções, compreende-se facilmente a importância da codificação. A codificação permite a devida representação das soluções de um determinado problema, para além de as interpretar, ou seja, deixar as soluções compreensíveis e manipuláveis para as Metaheurísticas. Isto é essencial visto que as Metaheurísticas têm o papel de gerar, avaliar e manipular várias soluções.

Explicada a importância da codificação, a sua escolha apropriada torna-se crucial para a resolução eficiente dos diversos problemas. Talbi [32] apresenta duas classes de codificação: Codificação Linear e Codificação Não Linear. A Codificação Linear inclui as seguintes representações: Codificação Binária, Discreta, de Permutação e baseada num Vetor de Valores Reais [32, 39].

Numa Codificação Binária as variáveis de decisão são representadas por um valor binário {0,1}, sendo que a solução é codificada como um vetor de “bits”. Um exemplo de problema onde a

Codificação Binária é utilizada é o conhecido “Problema da Mochila”. Numa Codificação Discreta as variáveis de decisão são representadas por números inteiros, tratando-se de uma extensão da Codificação Binária. Numa Codificação de Permutação representa-se uma permutação das variáveis de decisão. É normalmente utilizado este tipo de codificação em Problemas de Sequenciação para representar a sequência de processamento das operações. Numa Codificação baseada num Vetor de Valores Reais a solução é codificada como um vetor de valores reais, onde os elementos são representados por números reais [32, 39].

Um exemplo de cada um dos quatro tipos de Codificação falados pode-se observar na Figura 2.

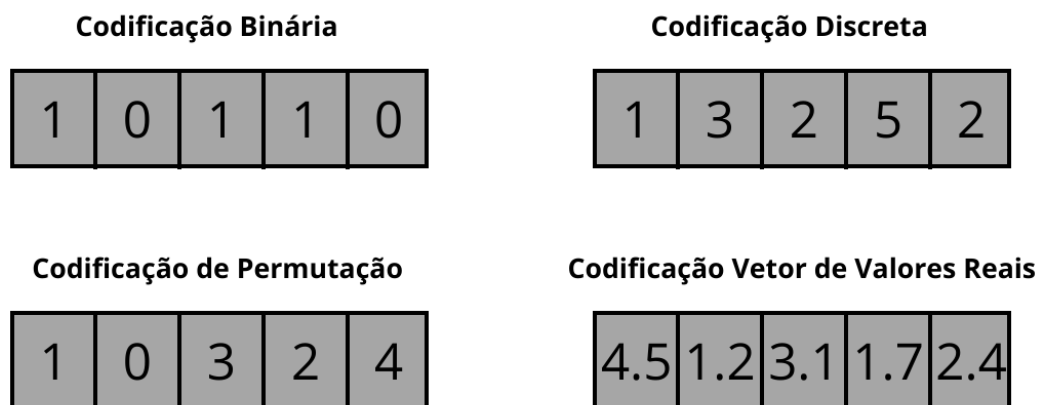


Figura 2 - Tipos de Codificação Linear

Para além dos quatros tipos de Codificação Linear já mencionados existem diversos mais, como por exemplo a Quântica [32].

Relativamente à Codificação Não Linear, esta é caracterizada por possuir maior complexidade que a Codificação Linear, representando as variáveis por árvores e com estruturas baseadas maioritariamente em grafos. Na Codificação em Árvore uma solução é representada por uma árvore de alguns objetos, sendo a Codificação mais utilizada de entre os vários tipos de Codificação Não Linear [32].

### 2.2.1.2. Avaliação

O mecanismo de Avaliação é um elemento importante ao projetar uma Metaheurística, pois ela necessita de calcular o desempenho das diversas soluções. O mecanismo de Avaliação tem como papel guiar a pesquisa em torno de soluções de alta qualidade no espaço de soluções. O mecanismo começa por calcular o desempenho de uma dada solução de Codificação, sendo que esse desempenho, depois, é utilizado para guiar a pesquisa da Metaheurística. Uma definição imprópria do mecanismo de Avaliação pode ter como consequência soluções inaceitáveis, independentemente do tipo de Metaheurística que esteja a ser utilizada [32].

### 2.2.1.3. Estruturas de Vizinhança

As Estruturas de Vizinhança apresentam um papel crucial nas Metaheurísticas, permitindo-lhes explorar o espaço de soluções na procura de soluções de alta qualidade. As Estruturas de Vizinhança determinam de que forma as Metaheurísticas se irão deslocar pelo espaço de soluções, isto é, como uma solução é transformada noutra através de um só movimento básico [40, 41].

De acordo com o problema em questão as Estruturas de Vizinhança podem variar. Estas podem apresentar uma complexidade elevada quando são desenvolvidas para resolver um problema específico ou, então, contrariamente, apresentar mecanismos simples, podendo ser usadas na resolução de muitos problemas [40]. Existem quatro Estruturas de Vizinhança comuns para problemas de permutação: Transpose, Swap, Insert e 2-Opt. Na Transpose as soluções são obtidas pela permutação de dois elementos adjacentes da solução [42]. Um exemplo disto é a transformação da solução {1,2,3,4,5,6} na solução {2,1,3,4,5,6} ao se trocarem os elementos 1 e 2. Na Swap as soluções são obtidas pela permutação de dois quaisquer elementos da solução [42, 43]. Um exemplo disto é a transformação da solução {1,2,3,4,5,6} na solução {5,2,3,4,1,6} ao se trocarem os elementos 1 e 5. Na Insert as soluções são obtidas ao remover um elemento da solução e inseri-lo noutra posição [42, 43]. Um exemplo disto é a transformação da solução {1,2,3,4,5,6} na solução {2,3,4,5,1,6}. Por último, na 2-Opt, que combina as Estruturas de Vizinhança Swap e Insert, as soluções são obtidas ao se trocarem dois elementos diferentes, invertendo a sequência de números entre eles [43, 44, 45]. Um exemplo disto é a transformação da solução {1,2,3,4,5,6} na solução {1,5,4,3,2,6} ao se selecionarem os elementos 2 e 5.

É possível observar exemplos destas Estruturas de Vizinhança na Figura 3.

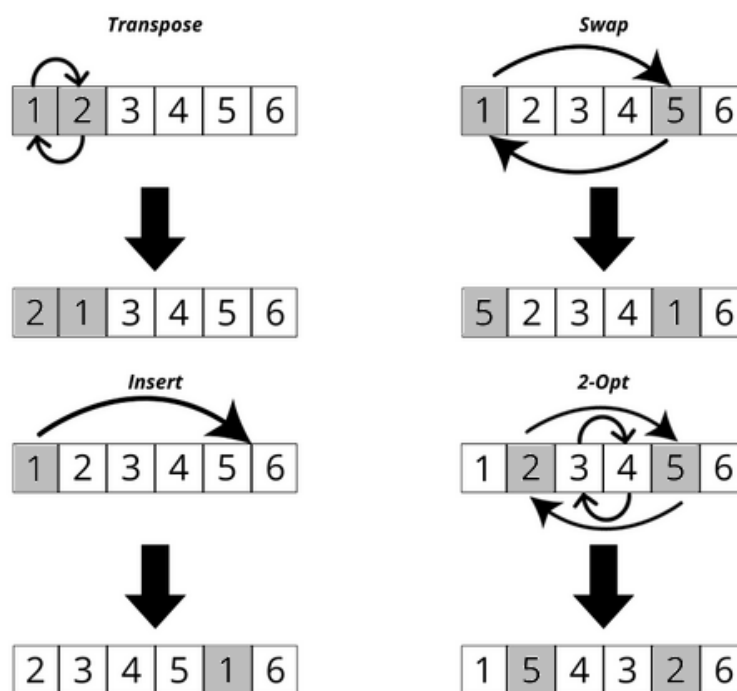


Figura 3 - Tipos de Estrutura de Vizinhança comuns de Permutação

De forma a assegurar um bom desempenho por de parte de uma Metaheurística é necessário escolher com o devido cuidado a Estrutura de Vizinhança que lhe é adequada. Isto irá permitir à Metaheurística explorar o espaço de soluções eficientemente e, conseqüentemente, encontrar mais soluções de qualidade.

### 2.2.2. Simulated Annealing

O *Simulated Annealing* (SA) é uma das Metaheurísticas mais antigas, tendo sido introduzida em 1983 por Kirkpatrick et al. [24] baseada no método de *Metropolis-Hasting*. Foi desenvolvida como técnica de resolução de problemas de otimização combinatórios.

O SA é inspirado no processo de recozimento de sólidos, no qual um sólido cristalino é aquecido, sendo de seguida arrefecido muito lentamente até atingir a sua configuração cristalina mais regular e livre de defeitos. Este algoritmo é, então, baseado na analogia entre a simulação de recozimento de sólidos e no problema de resolução de problemas de otimização, apresentando como sua vantagem principal o facto de conseguir evitar ficar preso nos Ótimos Locais [46, 47, 48].

O algoritmo começa por gerar uma solução inicial, que tanto pode ser gerada aleatoriamente ou, então, construída através de uma heurística. De seguida, o SA, em cada iteração, gera uma nova solução, designada por solução candidata, que será comparada à solução atual. Se a solução candidata apresentar um desempenho superior à solução atual, a candidata substitui a atual. Se não for o caso, a solução candidata substitui a solução atual baseado numa determinada probabilidade. Ao aceitar soluções com pior desempenho, o algoritmo oferece uma forma de evitar Ótimos Locais [35, 46].

A probabilidade de uma solução com pior desempenho ser aceite em relação à solução atual é calculada através da equação (4).

$$P(\Delta E, T) = e^{\left(\frac{-\Delta E}{T}\right)} \quad (4)$$

Onde  $\Delta E$  corresponde à diferença entre a solução candidata,  $f(s')$ , e a solução atual,  $f(s)$ , e o parâmetro  $T$  corresponde ao valor de Temperatura [32]. Para cada temperatura  $T$ , são geradas novas soluções, um determinado número de vezes. Este número é representado por  $L$  que é o *Epoch Length*, um parâmetro que determina o número de iterações para cada temperatura  $T$ , ou, o número de soluções aceites até  $T$  diminuir [49, 50].

Na Tabela 1 pode-se observar o *template* do algoritmo de SA, onde  $s$  representa a solução atual,  $s_0$  a solução inicial,  $s'$  a solução candidata,  $T$  a temperatura atual e  $T_{\max}$  a temperatura inicial.

Tabela 1 - *Template* do algoritmo de *Simulated Annealing* (adaptado de [32])

---

<b>Input:</b> Esquema de Arrefecimento
$s = s_0$
$T = T_{\max}$
<b>While</b> Critério de Interrupção não Ocorrer <b>Do</b>
<b>While</b> L <b>Do</b>
Gerar aleatoriamente $s'$
$\Delta E = f(s') - f(s)$
<b>If</b> $\Delta E \leq 0$ <b>Then</b> $s = s'$
<b>Else</b> $s = s'$ com probabilidade $P(\Delta E, T) = e^{\left(\frac{-\Delta E}{T}\right)}$
<b>EndWhile</b>
<b>Output:</b> Melhor Solução

---

Uma das características principais deste algoritmo é o facto de aceitar soluções com pior desempenho que a atual, armazenando a solução com melhor desempenho encontrada desde o início da pesquisa, de forma à solução final corresponder à melhor. Inicialmente, o valor de temperatura é elevado, o que aumenta a probabilidade de soluções com pior desempenho serem aceites e, conseqüentemente, permite explorar o espaço de soluções minuciosamente. À medida que o valor de temperatura reduz, menor será a probabilidade de se aceitarem soluções com pior desempenho, levando à concentração da pesquisa em certos locais do espaço de soluções [32, 51]. Esta informação é inferida após a devida análise da equação (4).

Na Figura 4, pode-se observar uma representação do funcionamento do SA, onde é possível perceber que quanto mais tempo passa menor fica a probabilidade de serem aceites soluções com pior desempenho. É importante realçar que soluções que apresentem um melhor desempenho são sempre aceites.

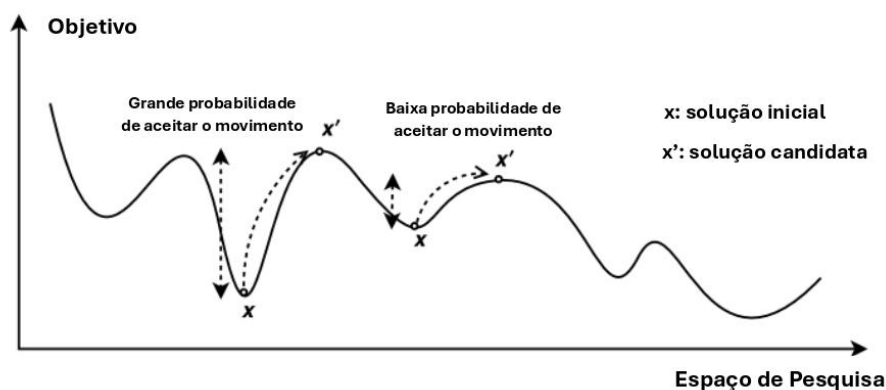


Figura 4 - Representação do funcionamento do Simulated Annealing (adaptado de [32])

Relativamente ao Esquema de Arrefecimento, se a temperatura for diminuindo lentamente obtêm-se melhores soluções, apresentando a desvantagem de ser mais dispendioso ao nível computacional [32].

Os parâmetros necessários de forma a definir o Esquema de Arrefecimento são a Temperatura Inicial, o Estado de Equilíbrio, a Função de Arrefecimento e a temperatura final que define o Critério de Fim. É a Função de Arrefecimento que é responsável por atualizar a Temperatura  $T$ , ou seja, por diminuir  $T$ , apresentando diversas formas de ser representada. De entre elas, existem três que são as mais utilizadas [32]:

- Linear, onde a Temperatura  $T$  é atualizada de acordo com a equação (5).  $T_i$  representa a Temperatura na iteração  $i$ ,  $T_0$  a Temperatura inicial e  $\beta$  o fator de diminuição situado entre  $]0,1[$ ;

$$T_i = T_0 - i \times \beta \quad (5)$$

- Geométrico, onde a Temperatura  $T$  é atualizada de acordo com a equação (6). O fator  $\alpha$  encontra-se situado entre  $]0,1[$ ;

$$T_i = \alpha \times T_0 \quad (6)$$

- Logarítmico, onde a Temperatura  $T$  é atualizada de acordo com a equação (7). Este tipo de esquema é muito lento para ser aplicado na prática, mas tem a propriedade de convergir para um Ótimo Global.

$$T_i = \frac{T_0}{\log(i)} \quad (7)$$

Estes Esquemas de Arrefecimento possuem em comum o facto de que são estáticos, ou seja, são definidos completamente *a priori*. Outros tipos de Esquemas de Arrefecimento podem ser consultados em [32, 52].

### 2.2.3. Genetic Algorithm

Outra Metaheurística antiga e igualmente popular é o *Genetic Algorithm* (GA), tendo sido introduzida em 1975 por Holland [22]. Este algoritmo pertence à classe de Algoritmos Evolutivos, sendo um dos mais conhecidos. Inicialmente, foi desenvolvido para compreender os processos dos sistemas naturais [22], pelo que mais tarde, a partir de 1980, começou a ser utilizado em problemas de otimização e *machine learning* [53, 54].

O GA é inspirado na seleção natural e na genética, utilizando técnicas igualmente inspiradas nestas como *selection*, *crossover* e *mutation*. Este algoritmo e as suas respetivas técnicas têm como ideia base o Princípio de Darwin, ou seja, a sobrevivência do mais apto de entre organismos ameaçados por predadores e catástrofes naturais. Assim, o GA parte de uma população inicial de soluções candidatas que, através das técnicas mencionadas, vai evoluindo,

selecionando as melhores soluções, sendo utilizado em diversas áreas como biologia, engenharia e ciência da computação [33, 55, 56].

O algoritmo começa por gerar uma população aleatória com diversas soluções candidatas designadas por *individuals*. Para cada um destes *individuals* é calculado o seu *fitness*, que dita a adaptabilidade solução em relação à natureza de um problema em particular, sendo que as soluções com melhores valores de *fitness* são selecionadas como *parents*. Estes *parents* selecionados dão origem a uma nova população de soluções candidatas, repetindo-se este processo por um determinado número de iterações, até o Critério de Fim ser satisfeito. Este processo tem como objetivo melhorar os valores da função objetivo de forma prática, eficiente e rápida [33, 57, 58].

Em relação ao funcionamento e implementação deste algoritmo existem 3 pontos cruciais que necessitam ser abordados, tendo sido já referidos: *selection*, *crossover* e *mutation*.

*Selection* corresponde ao processo no qual são selecionados os *individuals* responsáveis pela reprodução e criação da nova população de soluções candidatas, tornando-se estes *individuals* selecionados nos *parents* da próxima geração. O *Selection* está diretamente relacionado com o critério *fitness*, sendo este essencial na escolha dos *individuals*. Existem três métodos mais comuns para a seleção dos *parents* [33, 56, 57]:

- Seleção Proporcional, onde a probabilidade de uma solução ser selecionada é calculada pela equação (8).  $P_k$  representa a probabilidade da solução  $k$  ser selecionada e  $F(X)$  representa a função *fitness* da solução  $X$ ;

$$P_k = \frac{F(X_k)}{\sum_{j=1}^M F(X_j)} \quad (8)$$

- Seleção por Classificação, onde se faz uma classificação de todas as soluções baseadas no seu *fitness*. A melhor solução recebe a classificação de 1 enquanto a pior receber a menor classificação. A cada solução é lhe designado uma probabilidade que é proporcional à sua classificação, de acordo com as equações (9), (10), (11) e (12).  $S_k$  representa a classificação da solução  $k$  na população,  $Z$  corresponde a um valor definido pelo usuário e  $M$  representa o número de populações;

$$P_k = U - (S_k - 1) \times Z \quad (9)$$

$$S_k = \text{Classificação}(X_k) \quad (10)$$

$$\sum_{j=1}^M P_j = 1 \quad (11)$$

$$U = \frac{Z(M-1)}{2} + \frac{1}{M} \quad (12)$$

- Seleção por Torneio, onde  $Y$  ( $Y < M$ ) soluções são selecionadas aleatoriamente com uma distribuição uniforme, sendo que a melhor solução do conjunto selecionado é escolhida como *parent*. Isto repete-se até todos os *parents* serem selecionados.

O GA tem que criar soluções sucessivamente, de forma a atingir uma ótima de qualidade. Estas novas soluções (filhos) são constituídas em parte pela população *parent* e a restante pela nova população de soluções, sendo que este rácio de *parents* para filhos é um valor definido pelo utilizador. Gerar novas soluções (filhos) é o papel dos *parents* selecionados. Para tal, foram criadas várias metodologias para a criação de novas populações de soluções, com o intuito de aumentar a variedade presente no algoritmo. Aqui é que entra o *crossover*. O *crossover* corresponde ao processo no qual dois *parents* trocam material genético, sendo que através de um operador de *crossover* duas novas soluções são criadas, com os genes dos dois *parents*. Existem três métodos principais de *crossover* [33, 56, 57, 59]:

- *Crossover* de um ponto, onde um ponto de *crossover* é selecionado aleatoriamente. As duas novas soluções criadas possuem os genes de um dos *parents*, estando localizados num dos lados do ponto de *crossover* e os restantes genes pertencentes ao outro *parent* estão localizados no outro lado do ponto de *crossover*;
- *Crossover* de dois pontos, onde dois pontos de *crossover* são gerados aleatoriamente. Os genes entre os dois pontos das soluções *parent* são preservados nas mesmas posições para as novas soluções, enquanto os genes posicionados fora dos dois pontos de *crossover* são trocados para a criação das duas novas soluções;
- *Crossover* uniforme, onde os *parents* não podem ser decompostos em segmentos, sendo cada gene tratado separadamente. É decidido aleatoriamente quais genes se trocam entre os *parents* para a criação das novas soluções.

Na Figura 5, podem-se observar representações dos três métodos de *crossover* mencionados.

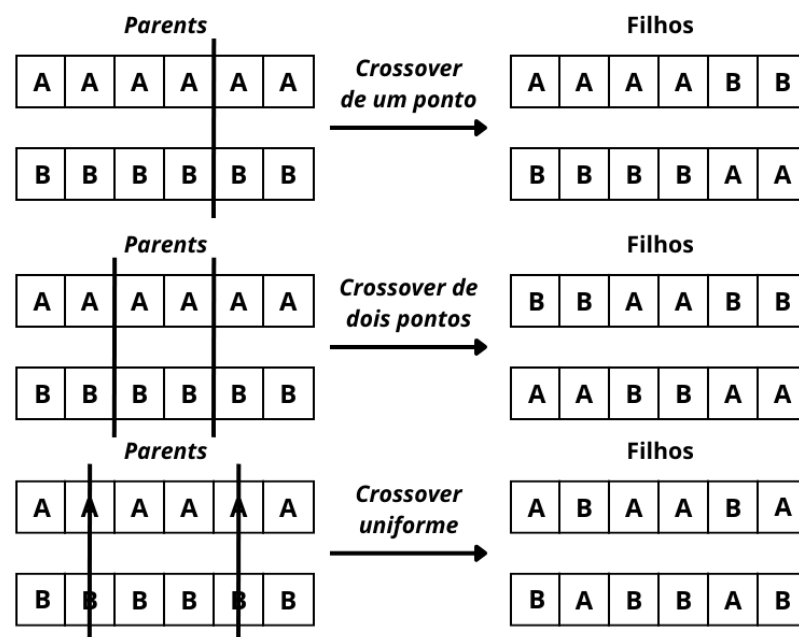


Figura 5 - Métodos de Crossover

Por último, falta falar da *mutation*. Esta é importante porque permite introduzir novo material genético nas populações e, desta forma, manter a diversidade genética. Isto é possível através

do operador de *mutation* que substitui de forma aleatória alguns genes das novas soluções. A *mutation* ao introduzir esta aleatoriedade à pesquisa previne o processo de otimização de ficar preso em Ótimos Locais. Existem dois métodos principais de *mutation* [33, 35, 56, 57, 59]:

- *Mutation* uniforme, onde um gene do *parent* é substituído por um gene gerado aleatoriamente, que se encontra no espaço de soluções;
- *Mutation* não uniforme, onde é induzida uma crescente pesquisa localizada por ótimas soluções, no qual os conjuntos de genes que são selecionados para a *mutation* são definidos por restrições, que ficam mais restritivas à medida que o algoritmo progride.

Finalmente, o Critério de Fim é responsável por determinar quando terminam as iterações do algoritmo. A importância advém do facto do Critério de Fim influenciar a convergência do algoritmo. O GA pode apresentar, então, diversos Critérios de Fim como por exemplo o número de iterações, o tempo de execução do algoritmo, quando o orçamento alocado é atingindo, entre muitos outros [33, 55].

Na Tabela 2, pode-se observar o *template* do algoritmo de GA, onde  $n$  representa o tamanho da população, MAX o número máximo de iterações,  $Y_{bt}$  a melhor solução,  $Y_i$  a solução número  $i$  e  $t$  o contador de iterações.

Tabela 2 - *Template* do algoritmo de *Genetic Algorithm* (adaptado de [59])

---

<b>Input:</b>
Tamanho da População, $n$
Número Máximo de Iterações, MAX
<b>Begin</b>
Gerar população inicial de $n$ soluções $Y_i$ ( $i = 1, 2, \dots, n$ )
Estabelecer contador de iterações $t = 0$
Calcular o valor de <i>fitness</i> para cada solução
<b>While</b> ( $t < \text{MAX}$ ) <b>Do</b>
Selecionar um par de soluções da população inicial baseado no valor de <i>fitness</i>
Aplicar <i>crossover</i> no par selecionado com probabilidade de <i>crossover</i>
Aplicar <i>mutation</i> nos filhos com probabilidade de <i>mutation</i>
Substituir antiga população com a nova população gerada
Aumentar a iteração atual $t$ em 1
<b>EndWhile</b>
<b>End</b>
<b>Output:</b> Melhor Solução, $Y_{bt}$

---

### 2.3. Análise Bibliométrica

Uma análise bibliométrica foi realizada com o intuito de compreender a relevância do tema proposto para este trabalho, bem como o seu potencial impacto, através de uma análise de como o tema tem vindo a evoluir ao longo dos anos e as principais áreas onde se destaca mais. Assim, com recurso à análise bibliométrica, foi possível reforçar as referências a utilizar e identificar os autores e palavras-chave mais importantes para o tema, consolidando a pesquisa realizada.

A análise bibliométrica foi realizada no dia 1 de dezembro de 2024, através do uso do *Web of Science Core Collection*, uma base de dados conhecida por ser um dos maiores repositórios de artigos científicos, e do *software VOSviewer*. Para as palavras-chave utilizou-se o termo “Topic”, de forma a direcionar a pesquisa para os campos mais importantes como título, resumo e palavras-chave. As palavras-chave, que estão em inglês, incluem os conceitos cruciais a este trabalho, como “Metaheurísticas”, incluindo algumas em específico que têm maior relevância para o tema como “Simulated Annealing” e “Genetic Algorithm”, “Escalonamento” com as suas diversas variantes e “Problemas de Escalonamento”. Na Tabela 3, pode-se observar a *query* realizada e os seus resultados.

Tabela 3 - Pesquisa de Palavras-Chave

Base de Dados	Query	Resultados
Web of Science	TOPIC (("Metaheuristic" OR "Metaheuristics" OR "Meta-Heuristic" OR "Meta-Heuristics" OR "Simulated Annealing" OR "Genetic Algorithm" OR "Genetic Algorithms") AND ("Job Shop Scheduling" OR "Flow Shop Scheduling" OR "Production Scheduling" OR "Resource Scheduling" OR "Operation Scheduling" OR "Operations Scheduling") AND ("Scheduling Problems" OR "Scheduling Problem"))	3573

Primeiro, realizou-se a análise de co-ocorrência, que se encontra no APÊNDICE A. Para unidade de análise escolheu-se “Author keywords” e para o número mínimo de ocorrência de uma palavra-chave definiu-se 10. De entre as palavras-chave mais significativas tem-se “genetic algorithm”, “scheduling”, “simulated annealing”, “job shop scheduling”, “makespan”, “optimization”, “production scheduling” e “multi-objective optimization”. Também se podem observar algumas palavras-chave com interesse em se destacar, como “energy consumption” e “green manufacturing”.

De seguida, realizou-se a análise de citação, que se encontra no APÊNDICE B, dividindo-se em três tipos de análise: documentos, fontes e autores. Para a análise por documentos, definiu-se para o número mínimo de citações 10, notando-se uma predominância de publicações entre 1992 e 2014. Para a análise por fontes, definiu-se para o número mínimo de documentos de uma fonte 5 e para o número mínimo de citações de uma fonte 15. Destacam-se o “International Journal of Production Research” e o “Computers & Industrial Engineering”. Para a análise por autores, definiu-se para o número mínimo de documentos de um autor 3 e para o número mínimo de citações de um autor 20, destacando-se o autor Liang Gao.

Por último, realizou-se a análise de co-citação, que se encontra no APÊNDICE C, dividindo-se em três tipos de análise: referências citadas, fontes citadas e autores citados. Para a análise por referências citadas, definiu-se para o número mínimo de citações de uma referência citada 25, notando-se 4 grandes grupos distintos. Para a análise por fontes citadas, definiu-se para o número mínimo de citações de uma fonte 30, destacando-se 3 grandes grupos distintos. Para a análise por autores citados, definiu-se para o número mínimo de citações de um autor 30. Pode-se observar que existe um autor principal, Wang.I, a partir do qual se originam 4 grandes grupos de autores.

No APÊNDICE D, pode-se observar, sob forma gráfica, quais as categorias do *Web of Science* que têm mais artigos na *query* realizada. Assim, em primeiro lugar encontra-se a “Ciência de Computação, Inteligência Artificial” e nos lugares seguintes a “Ciência de Gestão de Pesquisa de Operações”, a “Engenharia Industrial” e a “Engenharia de Manufatura”.

No APÊNDICE E, pode-se observar, graficamente, a evolução temporal do número de publicações dos artigos obtidos pela *query* realizada. Isto permite perceber de que forma foi evoluindo o interesse no tema deste trabalho. As primeiras publicações surgiram a partir do ano de 1992, em pequeno número mas com um crescimento constante. A partir do ano de 2005 houve um aumento substancial no número de publicações, acompanhado de um crescimento muito mais acentuado que se manteve até ao ano de 2019. Após este ano, o número de publicações manteve-se mais ou menos constante, notando-se um declínio notório em 2024, resultado de quando foi realizada a *query*, o ano de 2024 ainda não ter acabado. Os anos de 2019 e 2022 posicionam-se nos dois primeiros lugares, com 2022 a apresentar o maior número de publicações com 270, demonstrando-se, assim, um elevado interesse neste tema nos últimos anos.

### **2.3.1. Revisão do Estado da Arte – Casos de Estudo**

As Metaheurísticas têm assumido um papel cada vez mais importante na resolução de diversos problemas complexos de otimização. Estas técnicas de otimização têm vindo a apresentar uma utilização crescente nas indústrias, especialmente na área de Escalonamento de Operações, ao oferecerem soluções eficientes para os problemas de Escalonamento, que ameaçam as suas margens de lucro e níveis de competitividade.

Relativamente aos casos de estudo reunidos, com foco na resolução de problemas de Escalonamento através do uso de Metaheurísticas, existem um conjunto de características que os definem como o tipo de problema de Escalonamento presente, qual a Metaheurística utilizada, informações adicionais relevantes e os resultados provenientes da sua utilização. Assim, na Tabela 4 pode-se observar e analisar este conjunto de características referido para 20 artigos reunidos, de acordo com a sua relevância e qualidade para o trabalho desenvolvido.

Tabela 4 - Classificação de 20 Artigos relevantes

<b>Autor</b>	<b>Problema de Escalonamento</b>	<b>Metaheurística</b>	<b>Informações Adicionais</b>	<b>Resultados</b>
Fathollahi-Fard et al. [60]	Flow-Shop com Permutação Distribuída	<i>Simulated Annealing</i> e <i>Tabu Search</i>	Foco na Produção Sustentável e Escalonamento em Tempo Real	Redução do <i>Makespan</i>
Abreu e Prata [61]	Máquinas Paralelas Não Relacionadas	<i>Genetic Algorithm</i> Híbrido	Hibridização de GA com elementos de SA, <i>Iterated Local Search</i> , entre outros	Redução do <i>Makespan</i> e Eficácia e Eficiência na resolução de Problemas Reais
Marichelvam et al. [62]	Flow-Shop Híbrido	<i>Particle Swarm Optimization</i> Melhorado e <i>Variable Neighborhood Search</i> hibridizado com PSO	Fatores Humanos como nível de habilidade e idade são tidos em conta neste trabalho	Redução do <i>Makespan</i> e do <i>Flow Time</i> total
Stanković et al. [63]	Planeamento de Job-Shop Flexível	GA, <i>Tabu Search</i> e <i>Ant Colony Optimization</i>	Utilização de três Metaheurísticas diferentes para comparar os seus desempenhos	Minimização do processo de Produção, apresentando o GA os melhores resultados
Sacramento et al. [64]	Job-Shop Flexível	GA e <i>Tabu Search</i>	Desenvolvimento de duas Metaheurísticas Multiobjetivo e Comparação dos respetivos desempenhos	Redução do <i>Makespan</i> e <i>Tardiness</i> ; GA apresenta melhor eficiência e <i>Tabu Search</i> melhor eficácia
Abderrabi et al. [65]	Job-Shop Flexível	GA e <i>Iterated Local Search</i>	Particularidade de este caso de estudo ser proveniente da <i>supply chain</i> de um hospital	Redução do <i>Flow Time</i> total
Xue et al. [66]	Job-Shop Flexível com Lotes de Máquinas Paralelas	GA de Múltiplas Populações	É incorporado o conceito de <i>Variable Neighborhood Search</i> no GA de Múltiplas Populações	Redução do <i>Makespan</i> e aumento da eficiência de produção
Nejjarou et al. [67]	Flow-Shop Sem Espera	Híbridas e Melhoradas baseadas Nawaz Enscore Ham, GRASP e GA	Foram propostas três novas Metaheurísticas híbridas baseadas em SA	Minimização da <i>Tardiness</i> máxima
Ali et al. [68]	Job-Shop Dinâmico	GA Melhorado	Neste GA são propostos novos operadores de <i>crossover</i> virtuais	Minimização do <i>Makespan</i>
Ozsoydan e Sağır [69]	Flow-Shop Flexível Híbrido	<i>Iterated Greedy Search</i>	O desenvolvimento do algoritmo é dividido em quatro estágios, incluindo o envolvimento de uma Hiper-Heurística	Minimização do <i>Makespan</i>

Tabela 4 - Classificação de 20 Artigos relevantes (continuação)

<b>Autor</b>	<b>Problema de Escalonamento</b>	<b>Metaheurística</b>	<b>Informações Adicionais</b>	<b>Resultados</b>
Schumacher et al. [70]	Flow-Shop Híbrido de Dois Estágios	<i>Tabu Search</i>	Aliado ao <i>Tabu Search</i> são utilizadas heurísticas como a <i>Shortest Processing Time</i> e <i>Local Search</i>	Minimização do <i>Makespan</i>
Aqil e Allali [71]	Flow-Shop Híbrido com Máquinas Paralelas Não Relacionadas	<i>Iterative Local Search</i> e <i>Iterative Greedy Algorithm</i>	Aliado às Metaheurísticas são utilizados o algoritmo de Nawaz Enscore Ham e GRASP	Minimização da <i>Tardiness</i> total de todos os trabalhos
Klement et al. [72]	Sequência Dependente de Tempos de <i>Setup</i> , com Máquinas Paralelas	<i>Stochastic Descent</i> e SA	Procedeu-se à hibridização de uma <i>List Algorithm</i> com cada uma das Metaheurísticas e realizou-se uma comparação do desempenho entre elas	Minimização da <i>Tardiness</i> dos trabalhos; SA apresenta os melhores resultados
Branda et al. [73]	Flow-Shop com Manutenção Preventiva e Avaria Estocástica	GA e <i>Harmony Search</i>	Neste caso de estudo foram desenvolvidas duas novas Metaheurísticas ao modificar o GA e <i>Harmony Search</i>	Minimização do <i>Makespan</i> e redução da penalidade de <i>Earliness-Tardiness</i>
Ramos-Figueroa et al. [74]	Máquinas Paralelas Não Relacionadas	GA, <i>Grouping Genetic Algorithm</i> e PSO	É realizado um estudo comparativo entre as três Metaheurísticas para avaliar qual tem o melhor desempenho	Minimização do <i>Makespan</i> ; <i>Grouping Genetic Algorithm</i> apresenta os melhores resultados
Ebrahimi et al. [75]	Job-Shop Flexível – Escalonamento de <i>Layout</i>	Híbridas baseadas em PSO e <i>Ant Colony Optimization</i>	São desenvolvidas quatro Metaheurísticas Híbridas, realizando-se uma comparação entre os seus desempenhos	Redução de custos de energia e penalidades de <i>Tardiness</i> ; Metaheurística Híbrida entre <i>Ant Colony Optimization</i> e SA apresenta os melhores resultados
Xu et al. [76]	Flow-Shop Híbrido	GA	O GA deste caso de estudo é baseado em codificação de duas camadas	Minimização do tempo máximo de conclusão
Sahar et al. [77]	Job-Shop	GA	Este caso de estudo explica em detalhe a utilização e funcionamento do GA	Redução no tempo total de Manufatura; Encontrada a solução mais otimizada que representa o <i>Makespan</i> mínimo

Tabela 4 - Classificação de 20 Artigos relevantes (continuação)

<b>Autor</b>	<b>Problema de Escalonamento</b>	<b>Metaheurística</b>	<b>Informações Adicionais</b>	<b>Resultados</b>
Zhu et al. [78]	Flow-Shop Híbrido Distribuído	GA, <i>Artificial Bee Colony</i> , PSO e <i>Differential Evolution</i>	As 4 Metaheurísticas referidas são assistidas por <i>Q-Learning</i> , sendo o seu desempenho comparado com um <i>Iterated Greedy Algorithm</i> Melhorado	Minimização do <i>Makespan</i> ; As Metaheurísticas assistidas por <i>Q-Learning</i> apresentam os melhores resultados, nomeadamente o PSO
Daneshdoost et al. [79]	Máquina Única	SA e <i>Variable Neighborhood Search</i> Híbridos	As 2 Metaheurísticas referidas foram hibridizadas com TS e o seu desempenho foi comparado	Minimização do custo de produção total; SA Híbrido com TS apresenta os melhores resultados

A Tabela 4 é composta por artigos que compreendem 2020 e 2024, com o intuito de demonstrar como têm sido utilizadas as Metaheurísticas nos últimos anos. Após análise da Tabela 4, conclui-se que são utilizadas muitas diversas Metaheurísticas, nos mais diversos contextos, sendo algumas destas Híbridas ou Melhoradas, pelo que se pode perceber que na maioria dos casos de estudo recorrem-se a múltiplas Metaheurísticas em simultâneo, de forma a compreender qual delas irá obter o melhor desempenho para o problema. A maioria dos problemas abordados nos casos de estudo incluem Job-Shop, Flow-Shop e as suas respetivas variantes, devido a serem os tipos de problemas mais comuns no contexto industrial, bem como os mais complicados de se otimizar.

É apresentado, de seguida, uma análise mais profunda e detalhada de dois casos de estudo recentes, com o objetivo de se compreender melhor de que forma algumas das Metaheurísticas mais populares são utilizadas na resolução de Problemas de Escalonamento Industrial.

**“Multi-resource constrained flexible job shop scheduling problem with fixture-pallet combinatorial optimization” [80]**

Neste artigo, é abordado um Problema de Escalonamento sobre o qual não existe muita pesquisa realizada: Job-Shop Flexível, considerando recursos limitados de paletes de fixação em oficinas multiproduto mistas de produção. Apesar da escassa pesquisa, um estudo realizado numa indústria de produção líder de motores da China revelou que, embora recursos de paletes de fixação assumam um papel auxiliar no processo de produção, são um fator chave no que diz respeito a limitar avanços de capacidade. Assim, é proposto uma metodologia de otimização para esta variante do Problema de Job-Shop Flexível.

Primeiramente, foi construído um modelo de *Mixed Integer Programming* (MIP) com as restrições das máquinas de fixação de paletes, com o objetivo de minimizar o *Makespan*. Seguidamente, foi proposta uma nova Metaheurística Híbrida, combinando um GA Melhorado e *Variable Neighborhood Search* com estratégia corretiva de viabilidade e autoaprendizagem, para lidar com a complexidade deste problema. A estratégia corretiva de viabilidade tem o papel de resolver potenciais conflitos na seleção de máquinas e paletes de fixação, enquanto a

autoaprendizagem do *Variable Neighborhood Search* é responsável por melhorar a capacidade de otimização.

A eficácia da nova Metaheurística Híbrida proposta, foi verificada através da resolução de várias instâncias de cenários de produção real. Não só realiza a seleção de máquinas de forma otimizada e a combinação da sequência de operações, como também otimiza a combinação dos recursos de paletes de fixação. Atende cuidadosamente às necessidades de produção da oficina e tem um potencial de aplicação muito elevado atualmente.

**“A hybrid particle swarm optimization and simulated annealing algorithm for the job shop scheduling problem with transport resources” [81]**

Neste artigo, uma variante do Problema de Escalonamento de Job-Shop é abordada, no qual trabalhos têm de ser transportados para as máquinas que processam as suas operações, através de um número limitado de veículos. Uma vez que os veículos têm que entregar os trabalhos para processamento e dado que as máquinas necessitam de acabar o processamento dos seus trabalhos antes de estes serem transportados, o escalonamento das máquinas e dos veículos estão interligados. Devido a isto, este problema é mais complicado que a maioria dos Problemas de Escalonamento, sendo que estes dois tipos de escalonamento foram considerados conjuntamente de forma a assegurar a viabilidade do plano de produção e melhorar o desempenho do sistema de manufatura. Tal abordagem, tem como objetivo reduzir custos e aumentar o número de entregas dentro do prazo.

Assim, o estudo deste Problema de Escalonamento foi realizado, considerando a minimização de duas métricas de desempenho: o *Makespan* e o tempo de saída. Primeiramente, foi construído um modelo MIP, para achar soluções otimizadas, mas dada a complexidade envolvida na produção e transporte integrado, este modelo apenas conseguia lidar com pequenos problemas. Então, de forma a achar soluções de boa qualidade em tempos razoáveis, foi proposta uma Metaheurística Híbrida, combinando PSO com SA, aproveitando as capacidades de exploração do PSO e as capacidades de procura local do SA.

Para avaliar a eficácia e eficiência da Metaheurística Híbrida proposta, esta foi sujeita a experiências computacionais extensivas, de acordo com as duas métricas de desempenho. Os resultados destes experimentos demonstraram que a Metaheurística Híbrida supera os valores apresentados por outras soluções presentes na literatura, consolidando-se como uma abordagem muito robusta para este tipo de Problema de Escalonamento. Para além disto, as experiências mostraram que a produção e transporte devem ser otimizados conjuntamente, caso contrário iria resultar num aumento considerável do *Makespan*.

## Revisão Bibliográfica

### 3. Métodos e Aplicação

No capítulo 3 procedeu-se à seleção dos artigos científicos mais relevantes e adequados para o tema desta dissertação, sendo que estes serão posteriormente utilizados para analisar o desempenho das diferentes Metaheurísticas.

Assim, realizou-se uma pesquisa por artigos onde Problemas de Escalonamento Industrial do tipo Job-Shop com *makespan* como critério de otimização eram resolvidos com recurso a Metaheurísticas. Optou-se pelos problemas do tipo Job-Shop devido à elevada complexidade que apresentam, bem como pela grande frequência com que são discutidos pela comunidade científica. Para além disto, ainda foram escolhidas quatro Metaheurísticas, de forma a focalizar mais a pesquisa realizada nos problemas de Job-Shop: o *Simulated Annealing* e *Genetic Algorithm*, que foram abordados em detalhe no capítulo da Revisão Bibliográfica, e o *Particle Swarm Optimization* e *Tabu Search*, devido à sua popularidade na resolução de problemas de Job-Shop.

Este capítulo irá então apresentar os artigos de Job-Shop, divididos em quatro subcapítulos de acordo com as quatro Metaheurísticas que foram selecionadas previamente. Nestes quatro subcapítulos, cada artigo será caracterizado em três categorias:

- Qual a Metaheurística utilizada, dado que estas podem ser Híbridas, Melhoradas, incluir nomenclaturas especiais, entre outros aspetos. É importante referir estes aspetos para existir um maior detalhe relativamente a cada artigo selecionado;
- Se os Valores dos Parâmetros das respetivas Metaheurísticas são mencionados, bem como se existe uma explicação relativa à obtenção destes Valores no artigo;
- Qual o resultado que a utilização da Metaheurística proporcionou, sendo que este resultado pode ser proveniente da comparação com um valor ótimo, com outras Metaheurísticas ou até da sua utilização num caso real, e se existe menção de um resultado ótimo no respetivo artigo.

A pesquisa dos artigos científicos compreendeu o intervalo temporal de 2010 a 2025, sendo que para tal foi utilizada a base de dados do *Web of Science* e *Google Scholar*. Decidiu-se incluir artigos antigos desde 2010 com o intuito de se verificar a existência de tendências, onde certas Metaheurísticas eram utilizadas com maior frequência mas, por qualquer motivo, perderam popularidade. Outra razão foi para verificar se existiam diferenças significativas nos resultados provenientes da utilização das Metaheurísticas ao longo dos anos.

### 3.1. Problemas de Job-Shop

O Problema de Escalonamento Industrial do tipo Job-Shop encontra-se relacionado com os campos da Indústria, Economia e Gestão. Este problema faz parte de uma categoria de problemas de otimização combinatórios conhecida por Problema *NP-hard*. Os Problemas de Job-Shop lidam com um conjunto de máquinas e trabalhos com várias rotas predeterminadas através das máquinas, no qual o objetivo é organizar o escalonamento dos trabalhos de forma a minimizar certos critérios como o *makespan*, *lateness* máxima e *tardiness* total.

Estes problemas apresentam diversas características que os permite distinguir uns dos outros, desde a sua classificação, as restrições presentes e as suas funções objetivo. Relativamente à sua classificação, os Problemas de Job-Shop podem ser considerados do tipo determinístico, flexível, estático, dinâmico, entre outros. Para as restrições as três mais usuais são as de precedência, capacidade e data de lançamento e de entrega. Por último, em relação às funções objetivo as mais usuais são a minimização do *makespan*, *lateness* máxima e *tardiness* total. É importante referir que devido à atenção e consciencialização que tem sido dada para o desenvolvimento sustentável e para a redução de desperdícios nos últimos anos, a minimização dos custos energéticos é uma função objetivo que tem sido cada vez mais utilizada [82].

Ao longo das últimas décadas os Problemas de Escalonamento Industrial do tipo Job-Shop têm sido cada vez mais associados a Metaheurísticas, em detrimento das Heurísticas, devido às Metaheurísticas oferecerem melhores soluções [83]. Assim, nos subcapítulos seguintes poder-se-á observar de que forma quatro Metaheurísticas contribuem para a resolução de Problemas de Job-Shop.

#### 3.1.1. Genetic Algorithm

O *Genetic Algorithm* teve a sua origem em 1975 por Holland, tratando-se da primeira Metaheurística Evolutiva. É inspirada na seleção natural, caracterizando-se por começar com uma população de soluções que através de vários cruzamentos vai evoluindo, privilegiando-se as soluções de melhor qualidade [22, 23].

Na Tabela 5 podem-se observar os artigos selecionados, onde o GA foi aplicado na resolução de um Problema de Job-Shop, com as respetivas características relevantes associadas. Ambos os valores relativos ao Resultado da Metaheurística e Resultado Ótimo referem-se ao *makespan*.

Tabela 5 - 20 Artigos de *Genetic Algorithm* para Problemas de Job-Shop

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“Comparison of two Meta-Heuristics for the Bi-Objective Flexible Job Shop Scheduling Problem with Sequence Dependent Setup Times” [64] (2020)	GA e TS – BOBGA ( <i>Multi-objective Genetic Algorithm</i> ) e BOBTS ( <i>Multi-objective Tabu Search Algorithm</i> )	Não	Não	BOBGA = 1595.22 BOBTS = 1484.37	O BOBGA apresenta +7.47% de <i>makespan</i> médio relativamente ao BOBTS	Não
“Flexible job shop scheduling problem with sequence dependent setup time and job splitting: Hospital catering case study” [65] (2021)	GA	Sim	Sim. Foi utilizado o método <i>Taguchi</i>	788.64, 952.48, 226.12 e 705.96	O GA foi comparado a um caso industrial, levando às seguintes reduções de <i>makespan</i> : - 12.56%, - 10.36%, - 18.72% e - 11.18%	Não
“Flexible job-shop scheduling problem with parallel batch machines based on an enhanced multi-population genetic algorithm” [66] (2024)	GA de Múltiplas Populações Melhorado – eMPGA ( <i>enhanced Multi-Population Genetic Algorithm</i> )	Não	Não	2847	A redução de <i>makespan</i> do eMPGA relativamente aos outros algoritmos varia entre - 5.57% e - 16.46%	Não
“Improved Genetic Algorithm Approach Based on New Virtual Crossover Operators for Dynamic Job Shop Scheduling” [68] (2020)	GA Melhorado	Sim	Não	310.44	A redução de <i>makespan</i> do GA melhorado relativamente aos outros algoritmos e <i>P-Dispatching Rules</i> varia entre -9.23% e -24%	Não

Tabela 5 - 20 Artigos de *Genetic Algorithm* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“Multi-resource constrained flexible job shop scheduling problem with fixture-pallet combinatorial optimisation” [80] (2024)	GA Melhorado e Híbrido com VSN - IGA-FCSSVNS ( <i>Improved Genetic Algorithm Hybrid with Feasibility Correction Strategy and Self-Learning Variable Neighborhood Search</i> )	Sim	Não	599.28	O IGA-FCSSVNS apresenta entre -4.55% e -38.85% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem” [84] (2016)	GA Híbrido – GAM ( <i>Genetic Algorithm-Mixed</i> )	Sim	Sim. Foram realizados 7680 experimentos para se chegar aos melhores parâmetros do GA	Não	O GAM apresenta em média -0.02% de <i>makespan</i> relativamente a outros algoritmos da literatura	Não
“Optimisation of job shop scheduling problem using genetic algorithm and simulated annealing: a case study of manufacturing industry” [85] (2025)	GA e SA	Não	Não	GA = 1225 SA = 1247	O GA apresenta -32.25% de <i>makespan</i> relativamente ao da indústria e -1.76% de <i>makespan</i> relativamente ao SA	Não

Tabela 5 - 20 Artigos de *Genetic Algorithm* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“A Flexible Job Shop Scheduling Problem Considering On-Site Machining Fixtures: A Case Study From Customized Manufacturing Enterprise” [86] (2024)	GA Melhorado – IGA4F ( <i>Improved Genetic Algorithm for FJSP-F</i> )	Sim	Sim. Foi utilizado o método <i>Taguchi</i>	307	O IGA4F apresenta -28.94% de <i>makespan</i> em relação ao da indústria	Não
“A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem” [87] (2020)	GA – SLGA ( <i>Self-Learning Genetic Algorithm</i> )	Não	Não	181.3	O SLGA apresenta entre -0.49% e -13.95% de <i>makespan</i> médio relativamente aos outros algoritmos	163.3
“Solving multi-objective energy-saving flexible job shop scheduling problem by hybrid search genetic algorithm” [88] (2025)	GA Híbrido – HSGA ( <i>Hybrid Search Genetic Algorithm</i> )	Sim	Sim. Foi utilizada o método <i>Taguchi</i>	189.1	O HSGA apresenta entre -1.44% e -2.44% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“Optimization in job shop scheduling problem using Genetic Algorithm (study case in furniture industry)” [89] (2021)	GA	Não	Não	81780	De acordo com o artigo o GA apresenta -20.5% de <i>makespan</i> relativamente à regra EDD ( <i>Earliest Due Date</i> )	Não

Tabela 5 - 20 Artigos de *Genetic Algorithm* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“Co-Evolutionary NSGA-III with deep reinforcement learning for multi-objective distributed flexible job shop scheduling” [90] (2025)	GA Co-Evolucionário - CEGA-DRL ( <i>Co-Evolutionary Genetic Algorithm integrated with Deep Reinforcement Learning</i> )	Sim	Não	1519.2	O CEGA-DRL apresenta entre -15.66% e -31.64% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“Dynamic Events in the Flexible Job-Shop Scheduling Problem: Rescheduling with a Hybrid Metaheuristic Algorithm” [91] (2024)	Híbrida de GA, SA e VNS – GASAVNS	Não	Não	28, 31 e 217	O GASAVNS é utilizado em diversas instâncias e situações, conseguindo atingir as seguintes reduções de <i>makespan</i> : -3.45%, -16.2% e -42.7%	Não
“Dynamic Job Shop Scheduling Problem With New Job Arrivals Using Hybrid Genetic Algorithm” [92] (2024)	GA Híbrido - MGA ( <i>Memetic Genetic Algorithm</i> )	Sim	Sim. Foi utilizado o <i>Taguchi’s design of experiment</i> (TDOE)	2178.6	O MGA apresenta entre -0.046% e -0.89% de <i>makespan</i> médio relativamente aos outros algoritmos	2170.2
“A Hybrid Genetic Algorithm for Flexible Job Shop Scheduling Problem with Batch Processing Machines” [93] (2024)	GA Híbrido – HGA ( <i>Hybrid Genetic Algorithm</i> )	Sim	Sim. Foram realizados experimentos preliminares para se chegar aos parâmetros do HGA	78.25	O HGA apresenta entre -4.69% e -10.96% de <i>makespan</i> médio relativamente aos outros algoritmos	Não

Tabela 5 - 20 Artigos de *Genetic Algorithm* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“A multi-objective genetic algorithm based on two-stage reinforcement learning for green flexible shop scheduling problem considering machine speed” [94] (2024)	GA Multiobjetivo - TS-NSGA ( <i>Two-Stage Non-dominated Sorting Genetic Algorithm</i> )	Sim	Sim. Foram realizadas várias iterações até os parâmetros estabilizarem nos seus valores finais	83.47	De acordo com o artigo, o TS-NSGA apresenta -14.39% de <i>makespan</i> relativamente aos algoritmos de comparação	Não
“Energy-efficient scheduling for a flexible job shop problem considering rework processes and new job arrival” [82] (2024)	GA Melhorado - ENSGA II ( <i>Enhanced NSGA II</i> )	Sim	Não	232.7	O ENSGA II apresenta entre -2.64% e -5.18% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“A Case Study: Using Genetic Algorithm for Job Scheduling Problem” [95] (2021)	GA	Não	Não	Não	De acordo com o artigo, a utilização do GA levou a uma redução no <i>makespan</i> de 20% para o sistema em estudo	Não
“A Hybrid Genetic Algorithm for Flexible Job-shop Scheduling Problem” [96] (2014)	GA Híbrido – HGA	Sim	Sim. Foram testados vários valores para os parâmetros até se acharem os mais efetivos	173.8	O HGA apresenta entre -0.74% e -1.70% de <i>makespan</i> médio relativamente aos outros algoritmos	Não

Tabela 5 - 20 Artigos de *Genetic Algorithm* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“An improved genetic algorithm with dynamic neighborhood search for job shop scheduling problem” [97] (2023)	GA Melhorado – IGA ( <i>Improved Genetic Algorithm</i> )	Sim	Sim. Foi realizada uma exploração sistemática de 9 combinações de parâmetros. Através de vários experimentos chegaram-se aos valores de parâmetros mais favoráveis ao algoritmo	928.87	O IGA apresenta entre -21.25% e -23.11% de <i>makespan</i> médio relativamente aos outros algoritmos	879.13

A Tabela 5 é constituída por 20 artigos onde o GA foi utilizado na resolução de um Problema de Job-Shop. O GA é uma das Metaheurísticas mais utilizadas para a resolução deste tipo de problema, existindo mais artigos relativos ao GA do que aos algoritmos apresentados nos subcapítulos seguintes. Devido a isto, os artigos científicos reunidos do GA são na sua grande maioria de anos recentes, compreendendo 2020 e 2025, algo que não se irá notar para as outras Metaheurísticas. Tem-se como os artigos de exceção o [84] e [96] que são de 2016 e 2014 respetivamente.

Após uma análise da Tabela 5 percebe-se que são utilizados vários tipos de GA desde o GA simples, Melhorado e até Híbrido. Na generalidade dos artigos reunidos, o GA é comparado a outras Metaheurísticas ou algoritmos ou, então, em relação a um caso industrial, em termos de *makespan*. Assim, a utilização do GA resultou em reduções notórias de *makespan*. É de notar que as maiores reduções de *makespan* tendem a ser de artigos onde o *makespan* proveniente do GA é comparado ao de um caso industrial ou, então, ao de uma *Dispatching Rule*, enquanto as reduções de *makespan* mais modestas tendem a ser de artigos onde o *makespan* proveniente do GA é comparado ao de outras Metaheurísticas. Isto demonstra a maior capacidade que as Metaheurísticas apresentam na resolução de Problemas de Job-Shop, sendo esta a razão pela qual que quando são comparadas umas às outras as reduções de *makespan* não são tão notórias.

Apesar disto, existem algumas exceções encontradas nos artigos reunidos, desde o artigo [64] que apresenta um TS com menor *makespan* que o GA e os artigos [80], [90] e [97] que apresentam reduções bastante elevadas de *makespan* apesar do GA ser comparado com outras Metaheurísticas.

### 3.1.2. Particle Swarm Optimization

O *Particle Swarm Optimization* teve a sua origem em 1995 por Kennedy e Eberhart [98]. Faz parte da categoria de Metaheurísticas conhecida como *Swarm Intelligence*, sendo inspirada no comportamento de enxames presentes na natureza [14, 25]. Ela utiliza um conjunto de agentes que se move pelo *espaço de pesquisa* para encontrar o *global minimum* de uma função objetivo. A trajetória de cada partícula é determinada por uma regra simples que incorpora a velocidade atual da partícula e o histórico de exploração da mesma e dos seus vizinhos [99].

Na Tabela 6 podem-se observar os artigos selecionados, onde o PSO foi aplicado na resolução de um Problema de Job-Shop, com as respectivas características relevantes associadas. Ambos os valores relativos ao Resultado da Metaheurística e Resultado Ótimo referem-se ao *makespan*.

Tabela 6 - 20 Artigos de *Particle Swarm Optimization* para Problemas de Job-Shop

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“A hybrid particle swarm optimisation for flexible casting job shop scheduling problem with batch processing machine” [100] (2024)	PSO Híbrido – HPSO ( <i>Hybrid Particle Swarm Optimization</i> )	Sim	Sim. Foi realizada uma experiência ortogonal com quatro níveis de parâmetros de forma a se acharem os melhores valores para os parâmetros	114.56	O HPSO apresenta entre -0.096% e -1.12% de <i>makespan</i> médio relativamente aos outros algoritmos. Ainda apresenta -39% de <i>makespan</i> relativamente à empresa	Não
“An effective particle swarm optimization algorithm for flexible job-shop scheduling problem” [101] (2013)	PSO	Sim	Não	12	O PSO apresenta entre 0% e +9.1% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“Combining CA and PSO to Solve Flexible Job shop Scheduling Problem” [102] (2014)	PSO híbrido – CPSO ( <i>Cultural Particle Swarm Optimization</i> )	Não	Não	28.29 e 26.63	Neste artigo o CPSO foi comparado ao PSO, pelo que o CPSO apresenta entre -9.47% e -10.54% de <i>makespan</i> relativamente ao PSO	Não

Tabela 6 - 20 Artigos de *Particle Swarm Optimization* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“A NOVEL JOB-SHOP SCHEDULING STRATEGY BASED ON PARTICLE SWARM OPTIMIZATION AND NEURAL NETWORK” [103] (2019)	PSO Híbrido - PSO + NN ( <i>Neural Network</i> )	Sim	Não	1126.7	O PSO + NN apresenta entre -11.63% e -28.77% de <i>makespan</i> médio relativamente aos outros algoritmos	845
“An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem” [104] (2018)	PSO	Sim	Não	188.6	O PSO apresenta entre +5.13% e -10.74% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“Research on FJSP with transportation time constraint based on Improved Particle Swarm Optimization” [105] (2020)	PSO Híbrido – HDPSO ( <i>Hybrid Discrete Particle Swarm Optimization</i> )	Sim	Não	11.18	O HDPSO apresenta entre -1.24% e -7.45% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“An improved particle swarm optimization with decline disturbance index (DDPSO) for multi-objective job-shop scheduling problem” [106] (2014)	PSO Melhorado – DDPSO ( <i>Decline Disturbance Particle Swarm Optimization</i> )	Não	Não	1003.94	O DDPSO apresenta entre 0% e -4.55% de <i>makespan</i> médio relativamente aos outros algoritmos	Não

Tabela 6 - 20 Artigos de *Particle Swarm Optimization* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“Hybrid Particle Swarm Optimization Combined With Genetic Operators for Flexible Job-Shop Scheduling Under Uncertain Processing Time for Semiconductor Manufacturing” [107] (2018)	PSO Híbrido - HPSO + GA	Sim	Não	42.86	Neste artigo o HPSO + GA é comparado ao PSO, pelo que o HPSO + GA apresenta - 36.78% de <i>makespan</i> médio relativamente ao PSO	Não
“Reactive Hybrid Particle Swarm Optimization Based Job-shop Scheduling Problems Considering Energy Management” [108] (2018)	PSO Híbrido – RHPSO ( <i>Reactive Hybrid Particle Swarm Optimization</i> )	Não	Não	943	O RHPSO apresenta entre 0% e -0.057% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“Production Scheduling by Using ACO and PSO Techniques” [109] (2014)	PSO e ACO	Não	Não	PSO = 463.67	O PSO apresenta entre -1.15% e - 14.79% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“Performance Evaluation of Continuous and Discrete Particle Swarm Optimization in Job-Shop Scheduling Problems” [110] (2019)	PSO Contínuo e Discreto	Sim	Não	510	O PSO Discreto apresenta entre -1.64% e + 3.55% de <i>makespan</i> médio relativamente ao PSO Contínuo e à solução ótima	492.5

Tabela 6 - 20 Artigos de *Particle Swarm Optimization* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“Comparison of two variants of particle swarm optimization algorithm for solving flexible job shop scheduling problem” [111] (2016)	PSO e TRIBES	Sim	Não	10.89	O PSO e o TRIBES igualam sempre a melhor solução existente para a instância de problema, sendo que, então, não existe nem diminuição nem aumento de <i>makespan</i> pelo que a variação de <i>makespan</i> para este artigo, através do PSO, é de 0%	10.89
“Job-Shop Scheduling with Fuzzy Due Date by Multi-Objective Particle Swarm Optimization” [112] (2019)	PSO – MOPSO ( <i>Multi-Objective Particle Swarm Optimization</i> )	Não	Não	1035.4	Neste artigo o MOPSO é comparado ao NSGA-II, pelo que o MOPSO apresenta - 0.14% de <i>makespan</i> médio relativamente ao NSGA-II	Não
“A quantum behaved particle swarm optimization for flexible job shop scheduling” [113] (2016)	PSO – QPSO ( <i>Quantum Particle Swarm Optimization</i> )	Não	Não	173.8	O QPSO apresenta entre +0.70% e - 2.58% de <i>makespan</i> médio relativamente aos outros algoritmos	Não

Tabela 6 - 20 Artigos de *Particle Swarm Optimization* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“Branch-and-bound and PSO algorithms for no-wait job shop scheduling” [114] (2016)	PSO	Não	Não	1054.91	Neste artigo o PSO é comparado ao Branch-and-Bound, pelo que o PSO apresenta +10.59% de <i>makespan</i> médio relativamente ao Branch-and-Bound	Não
“Performance evaluation of different types of particle representation procedures of Particle Swarm Optimization in Job-shop Scheduling Problems” [115] (2016)	PSO	Sim	Não	542	O PSO apresenta +10.05% de <i>makespan</i> médio relativamente à solução ótima	492.5
“Improved particle swarm optimization based on blockchain mechanism for flexible job shop problem” [116] (2023)	PSO Melhorado – IPSO	Não	Não	100 e 1400	Para 100 <i>tasks</i> o IPSO apresenta entre 0% e -79.2% de <i>makespan</i> , relativamente aos outros algoritmos, e para 700 <i>tasks</i> apresenta entre -6.67% e -60.56% de <i>makespan</i>	Não
“An Improved Hybrid Quantum Particle Swarm Optimization Algorithm for FJSP” [117] (2019)	PSO – HQPSO ( <i>Hybrid Quantum Particle Swarm Optimization</i> )	Não	Não	185.1	O HQPSO apresenta entre -1.80% e -6.66% de <i>makespan</i> médio relativamente aos outros algoritmos	Não

Tabela 6 - 20 Artigos de *Particle Swarm Optimization* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
"A novel dynamic scheduling strategy for solving flexible job-shop problems" [118] (2016)	PSO – HMQPSO ( <i>Hybrid Multi-Phase Quantum Particle Swarm Optimization</i> )	Sim	Não	167.6	O HMQPSO apresenta entre -0.77% e -4.39% de <i>makespan</i> médio relativamente aos outros algoritmos	163.3
"Concurrent scheduling of jobs and AGVs in a flexible job shop system: a parallel hybrid PSO-GA meta-heuristic" [119] (2023)	PSO e GA Híbrido – PPSOGA ( <i>Parallel Particle Swarm Optimization Genetic Algorithm</i> )	Não	Não	Não	De acordo com o artigo, o PPSOGA apresenta entre -4.2% e -6.4% de <i>makespan</i> relativamente aos outros algoritmos	Não

A Tabela 6 é constituída por 20 artigos onde o PSO foi utilizado na resolução de um Problema de Job-Shop. Apesar de o PSO ser uma Metaheurística bastante conhecida não apresenta o mesmo grau de utilização que o GA na resolução deste tipo de problemas. Devido a isto, os artigos científicos reunidos do PSO abrangem uma maior janela temporal que o GA, de 2024 até 2013.

Tal como com o GA, após uma análise da Tabela 6 percebe-se que também são utilizados vários tipos de PSO desde o PSO simples, Melhorado e até Híbrido. Apesar de existirem igualmente artigos onde o PSO é comparado a outras Metaheurísticas, algoritmos ou casos industriais, em termos de *makespan*, existem muitos mais onde o PSO acaba por não ser a melhor solução, quando comparado com outra Metaheurística. Nos artigos [101], [104], [113] e [114] o PSO não apresenta o menor *makespan* de entre os algoritmos de comparação e nos artigos [106], [108], [111] e [116] o PSO iguala outra Metaheurística em termos do menor *makespan*. Isto pode ser resultado dos artigos de PSO serem mais antigos. As limitações computacionais da altura impossibilitavam a obtenção de menores valores de *makespan*, por não ser possível realizar um número elevado de iterações, dado que demoravam bastante tempo. Outro aspeto importante de se considerar é o facto de as Metaheurísticas Híbridas não serem tão utilizadas, não se aproveitando as inúmeras vantagens que estas apresentam. As ligeiras alterações que a comunidade científica foi realizando no algoritmo ao longo dos anos também não podem ser ignoradas, resultando no seu melhoramento gradual e, conseqüentemente, do *makespan* também. Por último, outra hipótese que não pode ser descartada é o facto do PSO poder não se tratar da Metaheurística mais efetiva para a resolução de Problemas de Job-Shop.

De qualquer forma, também é possível identificar para a generalidade dos artigos de PSO que quando este é utilizado e comparado relativamente a um caso industrial ou a uma *Dispatching*

*Rule*, a redução de *makespan* é bastante notória enquanto que quando é comparado a outras Metaheurísticas esta redução é mais moderada. Tem-se como exceções a isto, os artigos [107] e [116] que apresentam reduções bastante elevadas de *makespan* apesar do PSO ser comparado com outras Metaheurísticas.

### 3.1.3. Tabu Search

O *Tabu Search* teve a sua origem em 1986 por Glover, tratando-se de uma Metaheurística que não é inspirada na seleção natural. Foi a primeira vez em que “memória” foi utilizada numa Metaheurística, sendo que os movimentos de uma pesquisa anterior eram armazenados numa *Tabu List*, proibindo a repetição de movimentos iguais dali para a frente [15].

Na Tabela 7 podem-se observar os artigos selecionados, onde o TS foi aplicado na resolução de um Problema de Job-Shop, com as respetivas características relevantes associadas. Ambos os valores relativos ao Resultado da Metaheurística e Resultado Ótimo referem-se ao *makespan*.

Tabela 7 - 20 Artigos de *Tabu Search* para Problemas de Job-Shop

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“Comparison of two Meta-Heuristics for the Bi-Objective Flexible Job Shop Scheduling Problem with Sequence Dependent Setup Times” [64] (2020)	GA e TS – BObGA e BObTS	Não	Não	BObGA = 1595.22 BObTS = 1484.37	O BObTS apresenta - 6.95% de <i>makespan</i> médio relativamente ao BObGA	Não
“Metaheuristics for the job-shop scheduling problem with machine availability constraints” [120] (2018)	SA e TS	Não	Não	SA = 1660 TS = 1651.4	O TS apresenta -0.52% de <i>makespan</i> médio relativamente ao SA	1650.1

Tabela 7 - 20 Artigos de *Tabu Search* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“A hybrid algorithm for multi-objective job shop scheduling problem” [121] (2011)	TS Híbrido – TS-m	Não	Não	991.72	Neste artigo o TS-m é comparado ao HPSO, pelo que o TS-m apresenta -0.18% de <i>makespan</i> médio relativamente ao HPSO	Não
“Decentralized Tabu Searches in Multi Agent system for Distributed and Flexible Job shop Scheduling Problem” [122] (2017)	TS – DTSMA ( <i>Decentralized Tabu Search in Multi Agent system</i> )	Sim	Não	511.1, 471.3 e 467.3	O DTSMA apresenta entre +1.34% e -0.42% de <i>makespan</i> médio relativamente aos outros algoritmos para 2 fábricas, apresenta entre +0.61% e -0.074% de <i>makespan</i> médio para 3 fábricas e apresenta entre +0.38% e +0.17% de <i>makespan</i> médio para 4 fábricas	465.55
“A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem” [123] (2011)	TS Híbrido – TSPCB ( <i>Tabu Search algorithm with a fast Public Critical Block neighborhood structure</i> )	Não	Não	10.8	O TSPCB apresenta entre 0% e -28.95% de <i>makespan</i> médio relativamente aos outros algoritmos	Não

Tabela 7 - 20 Artigos de *Tabu Search* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem” [124] (2016)	TS e GA Híbrido – HA	Sim	Não	10.67	O HA apresenta entre 0% e -11.08% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“Multi-operator communication based differential evolution with sequential Tabu Search approach for job shop scheduling problems” [125] (2021)	TS Híbrido – MCDE/TS ( <i>Multi-operator based Differential Evolution with a Communication strategy Tabu Search</i> )	Sim	Sim. Foi utilizado o <i>Taguchi’s design of experiment</i>	896.14	O MCDE/TS apresenta entre -1.34% e -3.60% de <i>makespan</i> médio relativamente aos outros algoritmos	894.14
“An effective tabu search for job shop scheduling with parallel machines” [126] (2018)	TS	Sim	Sim. Foi utilizado uma <i>F-Race</i> iterativa	908.36	Neste artigo o TS é comparado ao HGA, pelo que o TS apresenta -2.89% de <i>makespan</i> médio relativamente ao HGA	904.47
“A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times” [127] (2012)	TS e GA Híbrido – GATS	Não	Não	120.83	Neste artigo o GATS é comparado ao TS, pelo que o GATS apresenta +3.87% de <i>makespan</i> médio relativamente ao TS	Não

Tabela 7 - 20 Artigos de *Tabu Search* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems” [128] (2010)	TS Híbrido – HTSA ( <i>Hybrid Tabu Search Algorithm</i> )	Sim	Não	10.8	O HTSA apresenta entre -1.82% e -29.18% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“Research on multi-agent genetic algorithm based on tabu search for the job shop scheduling problem” [129] (2019)	TS e GA Híbrido – MAGATS ( <i>Multi-Agent Genetic Algorithm Tabu Search</i> )	Sim	Não	1018.55	O MAGATS apresenta entre -0.77% e -1.86% de <i>makespan</i> médio relativamente aos outros algoritmos	1013.82
“An Effective Hybrid Imperialist Competitive Algorithm and Tabu Search for an Extended Flexible Job Shop Scheduling Problem” [130] (2019)	TS Híbrido – ICA + TS ( <i>Imperialist Competitive Algorithm + Tabu Search</i> )	Sim	Sim. Após a realização de vários experimentos chegaram-se aos melhores valores para os parâmetros do algoritmo	707.6	O ICA + TS apresenta entre -2.67% e -4.97% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“MATS-JSTL: A Multi-Agent Model Based on Tabu Search for Job Shop Problem with Time Lags” [131] (2015)	TS – MATS ( <i>Multi-Agent Tabu Search</i> )	Não	Não	814.3	O MATS apresenta entre -1.02% e -12.24% de <i>makespan</i> médio relativamente aos outros algoritmos	Não

Tabela 7 - 20 Artigos de *Tabu Search* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“Multi-agent model-based intensification-driven tabu search for solving the dual-resource constrained flexible job shop scheduling” [132] (2024)	TS – MuSTAM ( <i>Multi-Start Tabu Search-based Multi-Agent Model</i> )	Sim	Não	314.91	Neste artigo o MuSTAM é comparado ao ITS ( <i>Intensification-driven Tabu Search</i> ), pelo que o MuSTAM apresenta - 0.74% de <i>makespan</i> médio relativamente ao ITS	Não
“Robust multiobjective optimisation for fuzzy job shop problems” [133] (2017)	TS Híbrido – MTEA ( <i>Multiobjective Tabu Evolutionary Algorithm</i> )	Sim	Sim. Foi realizada uma análise paramétrica preliminar	1006.235	Neste artigo o MTEA é comparado ao MA, pelo que o MTEA apresenta - 0.41% de <i>makespan</i> médio relativamente ao MA	Não
“Flexible job-shop scheduling with transportation resources” [134] (2024)	TS	Não	Não	132.2	Neste artigo o TS é comparado ao LAHC-HF ( <i>Late-Acceptance Hill Climbing-Homayouni &amp; Fontes</i> ), pelo que o TS apresenta - 0.30% de <i>makespan</i> médio relativamente ao LAHC-HF	Não

Tabela 7 - 20 Artigos de *Tabu Search* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“A multi-objective scheduling method for distributed and flexible job shop based on hybrid genetic algorithm and tabu search considering operation outsourcing and carbon emission” [135] (2021)	TS e GA Híbrido - H-GA-TS ( <i>Hybrid Genetic Algorithm and Tabu Search</i> )	Sim	Sim. Foi realizado o <i>orthogonal design method</i>	237.6	O H-GA-TS apresenta entre -1.02% e -1.74% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“A Hybrid Metaheuristic Algorithm for Flexible Job-Shop Scheduling Problems with Transportation Constraints” [136] (2012)	TS e GA Híbrido – GTS ( <i>Genetic Algorithm with Tabu Search and Shifting bottlenecks</i> )	Não	Não	109.625	O GTS apresenta entre +0.32% e +1.06% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“Genetic tabu search for the fuzzy flexible job shop problem” [137] (2015)	TS e GA Híbrido – HGTS ( <i>Hybrid Genetic Tabu Search</i> )	Sim	Não	180.67	Neste artigo o HGTS é comparado ao PSO, pelo que o HGTS apresenta -2.17% de <i>makespan</i> médio relativamente ao PSO	176
“Evaluation of a hybrid genetic tabu search framework on job shop scheduling benchmark problems” [138] (2014)	TS e GA Híbrido	Sim	Não	825.75	O algoritmo proposto apresenta entre -0.045% e -3.67% de <i>makespan</i> médio relativamente aos outros algoritmos	825.625

A Tabela 7 é constituída por 20 artigos onde o TS foi utilizado na resolução de um Problema de Job-Shop. Semelhante ao PSO, o TS não é tão utilizado como o GA, em Problemas de Job-Shop, sendo esta a razão pela qual apresenta uma janela temporal mais alargada, de 2024 até 2010.

Após uma análise da Tabela 7 nota-se que o TS simples e Híbrido são muito utilizados. Tal como com o PSO, o TS também reúne artigos onde acaba por não oferecer a melhor solução. Nos artigos [122], [127] e [136] o TS não apresenta o menor *makespan* de entre os algoritmos de comparação e nos artigos [123] e [124] o TS iguala outra Metaheurística em termos do menor *makespan*. Isto pode-se dever às mesmas razões que foram apresentadas relativamente ao PSO: os artigos do TS serem mais antigos e as limitações computacionais associadas a isto; o facto de as Metaheurísticas Híbridas não serem tão utilizadas antigamente, não se aproveitando das suas vantagens; as ligeiras alterações que foram sendo introduzidas pela comunidade científica ao algoritmo ao longo dos anos, resultando no seu melhoramento gradual ou o facto do TS não ser a Metaheurística mais indicada para Problemas de Job-Shop. Para além disto, outro fator importante de se referir é que devido às limitações computacionais existentes, as *Tabu List*, um parâmetro importante no desempenho do TS, eram menores, o que podia resultar em maiores valores de *makespan*. Atualmente, graças à evolução tecnológica, as *Tabu List* são maiores, possibilitando a obtenção de menores valores de *makespan*.

Para a maioria dos artigos reunidos do TS também se verifica que quando este é comparado a outras Metaheurísticas, em termos de *makespan*, a redução de *makespan* é moderada, com a exceção nos artigos [123] e [128].

### 3.1.4. Simulated Annealing

O *Simulated Annealing* teve a sua origem em 1983 por Kirkpatrick et al., tratando-se da primeira Metaheurística não inspirada na seleção natural. Esta caracterizava-se por não utilizar uma população de soluções, como as Metaheurísticas Evolutivas, inspirando-se no processo de recozimento natural dos sólidos [24].

Na Tabela 8 podem-se observar os artigos seleccionados, onde o SA foi aplicado na resolução de um Problema de Job-Shop, com as respetivas características relevantes associadas. Ambos os valores relativos ao Resultado da Metaheurística e Resultado Ótimo referem-se ao *makespan*.

Tabela 8 - 20 Artigos de *Simulated Annealing* para Problemas de Job-Shop

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Varição do <i>makespan</i>	Resultado Ótimo
“Metaheuristiccs for the job-shop scheduling problem with machine availability constraints” [120] (2018)	SA e TS	Não	Não	SA = 1660 TS = 1651.4	O SA apresenta +0.52% de <i>makespan</i> médio relativamente ao TS	1650.1

Tabela 8 - 20 Artigos de *Simulated Annealing* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“Optimisation of job shop scheduling problem using genetic algorithm and simulated annealing: a case study of manufacturing industry” [85] (2025)	GA e SA	Não	Não	GA = 1225 SA = 1247	O SA apresenta - 31.03% de <i>makespan</i> relativamente ao da indústria e +1.8% de <i>makespan</i> relativamente ao GA	Não
“Dynamic Events in the Flexible Job-Shop Scheduling Problem: Rescheduling with a Hybrid Metaheuristic Algorithm” [91] (2024)	Híbrida de GA, SA e VNS – GASAVNS	Não	Não	28, 31 e 217	O GASAVNS é utilizado em diversas instâncias e situações, conseguindo atingir as seguintes reduções de <i>makespan</i> : - 3.45%, -16.2% e -42.7%	Não
“An Improved Differential Evolution Algorithm for Solving a Distributed Flexible Job Shop Scheduling Problem” [139] (2018)	SA Híbrido – IDESAA ( <i>Improved Differential Evolution Simulated Annealing Algorithm</i> )	Sim	Não	459.31 e 452.46	O IDESAA apresenta - 0.11% de <i>makespan</i> médio relativamente ao GA_OP ( <i>Genetic Algorithm with Operation based representation</i> ) para 2 fábricas e apresenta - 0.011% de <i>makespan</i> médio para 4 fábricas	418.08 e 452.17

Tabela 8 - 20 Artigos de *Simulated Annealing* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Varição do <i>makespan</i>	Resultado Ótimo
“Hyper-heuristic for flexible job shop scheduling problem with stochastic job arrivals” [140] (2023)	SA - SA-HH ( <i>Simulated Annealing-based Hyper-Heuristic</i> )	Sim	Sim. Os valores dos parâmetros foram determinados experimentalmente	243.67	Neste artigo o SA-HH é comparado ao FAM-SPT ( <i>First Available Machine-Shortest Processing Time</i> ), pelo que o SA-HH apresenta -19.56% de <i>makespan</i> médio relativamente ao FAM-SPT	Não
“The flexible job-shop scheduling problem considering deterioration effect and energy consumption simultaneously” [141] (2019)	SA Híbrido – MOHPIOSA ( <i>Multi-Objective Hybrid Pigeon Inspired Optimization and Simulated Annealing algorithm</i> )	Sim	Não	387.72	Neste artigo o MOHPIOSA é comparado ao NSGA-II, pelo que o MOHPIOSA apresenta -2.40% de <i>makespan</i> médio relativamente ao NSGA-II	Não
“Hybridization of Simulated Annealing with Quenching for Job Shop Scheduling” [142] (2015)	SA Híbrido – HSAQ ( <i>Hybrid Simulated Annealing with Quenching</i> )	Sim	Não	899.45	Neste artigo o HSAQ é comparado ao SA, pelo que o HSAQ apresenta -0.13% de <i>makespan</i> médio relativamente ao SA	Não
“Hybrid Sorting Immune Simulated Annealing Algorithm For Flexible Job Shop Scheduling” [143] (2015)	SA Híbrido – HSISA ( <i>Hybrid Sorting Immune Simulated Annealing</i> )	Sim	Não	9	O HSISA apresenta entre -10% e -52.63% de <i>makespan</i> ótimo relativamente aos outros algoritmos	Não

Tabela 8 - 20 Artigos de *Simulated Annealing* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“An Efficient Hybrid Particle Swarm Optimization for the Job Shop Scheduling Problem” [144] (2011)	PSO, TS e SA Híbrido – HPTS ( <i>Hybrid of Particle swarm optimization, Tabu search and Simulated annealing</i> )	Sim	Sim. Os valores dos parâmetros foram determinados experimentalmente numa fase de pré-processamento	1161.58	O HPTS apresenta entre -0.075% e -0.35% de <i>makespan</i> médio relativamente aos outros algoritmos	1080.02
“A Hybrid Artificial Bee Colony Algorithm with Local Search for Flexible Job-Shop Scheduling Problem” [145] (2013)	ABC ( <i>Artificial Bee Colony</i> ) e SA Híbrido	Não	Não	175.3	O algoritmo proposto apresenta entre -0.28% e -1.63% de <i>makespan</i> médio relativamente aos outros algoritmos	163.3
“A hybrid iterated local search metaheuristic for the flexible job shop scheduling problem” [146] (2018)	SA Híbrido – HILS ( <i>Hybrid Iterated Local Search</i> )	Sim	Sim. Foram realizados testes com diversas instâncias para calibrar os valores dos parâmetros do algoritmo	937.1	Neste artigo o HILS é comparado ao TS, pelo que o HILS apresenta -0.032% de <i>makespan</i> médio relativamente ao TS	Não
“A hybrid imperialist competitive algorithm for energy-efficient flexible job shop scheduling problem with variable-size sublots” [147] (2022)	SA Híbrido – HICSA ( <i>Hybrid Imperialist Competitive Simulated Annealing</i> )	Sim	Sim. Foi utilizado o método <i>Taguchi</i>	323.27	O HICSA apresenta entre -3.69% e -7.71% de <i>makespan</i> médio relativamente aos outros algoritmos	Não

Tabela 8 - 20 Artigos de *Simulated Annealing* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“Simulated-annealing-based hyper-heuristic for flexible job-shop scheduling” [148] (2023)	SA - SA-HH	Sim	Sim. Foram realizados testes com diversas instâncias para calibrar os valores dos parâmetros do algoritmo	177.7	O SA-HH apresenta entre -1.99% e -10.39% de <i>makespan</i> médio relativamente aos outros algoritmos	Não
“A Hybrid Genetic Algorithm for Solving Job Shop Scheduling Problems” [149] (2014)	SA e GA Híbrido – HGA	Sim	Não	886.08	O HGA apresenta entre -0.052% e -1.94% de <i>makespan</i> médio relativamente aos outros algoritmos	885.27
“Multi-strategy improved sparrow search algorithm for job shop scheduling problem” [150] (2024)	SA, GA e SSA ( <i>Sparrow Search Algorithm</i> ) Híbrido – MISSA ( <i>Multi-strategy Improved Sparrow Search Algorithm</i> )	Sim	Não	1087.4	O MISSA apresenta entre -0.79% e -52.30% de <i>makespan</i> médio relativamente aos outros algoritmos	1029.9
“Flexible job shop scheduling under condition-based maintenance: Improved version of imperialist competitive algorithm” [151] (2017)	SA e ICA Híbrido – ICA ( <i>Imperialist Competitive Algorithm</i> )	Sim	Sim. Foi utilizado o <i>Taguchi design method</i>	369.77	O ICA apresenta entre -1.28% e -10.89% de <i>makespan</i> médio relativamente aos outros algoritmos	Não

Tabela 8 - 20 Artigos de *Simulated Annealing* para Problemas de Job-Shop (continuação)

Artigo Científico	Meta heurística Utilizada	Valores dos Parâmetros	Explicação dos Valores	Resultado da Meta heurística	Variação do <i>makespan</i>	Resultado Ótimo
“Makespan minimization in a job shop with a BPM using simulated annealing” [152] (2013)	SA	Sim	Sim. Atingiram-se os valores ótimos dos parâmetros após uma calibragem na fase de experimentação	1560.17 e 2792.17	O SA apresenta -17.74% de <i>makespan</i> médio relativamente às <i>Dispatching Rules</i> para 300 operações e -16.18% de <i>makespan</i> médio para mais que 300 operações	Não
“A hybrid evolution strategies-simulated annealing algorithm for job shop scheduling problems” [153] (2024)	SA Híbrido - HES-SA ( <i>Hybrid Evolution Strategies-Simulated Annealing</i> )	Não	Não	1146.2	O HES-SA apresenta entre -0.26% e -0.59% de <i>makespan</i> médio relativamente aos outros algoritmos	1146.2
“Mathematical modelling and a meta-heuristic for flexible job shop scheduling” [154] (2013)	SA Híbrido – AISA ( <i>Artificial Immune and Simulated Annealing</i> )	Não	Não	881	Neste artigo o AISA é utilizado num caso de estudo de uma companhia de manufatura, sendo que a sua utilização levou a -8.3% de <i>makespan</i>	Não
“Minimising makespan in job-shops with deterministic machine availability constraints” [155] (2021)	SA – MTSA ( <i>Multi-Temperature Simulated Annealing</i> )	Sim	Sim. Foi utilizado o <i>Taguchi orthogonal experiment design</i>	1646.3	O MTSA apresenta entre -0.31% e -0.82% de <i>makespan</i> médio relativamente aos outros algoritmos	1610.75

A Tabela 8 é constituída por 20 artigos onde o SA foi utilizado na resolução de um Problema de Job-Shop. O SA apresenta a menor utilização de entre as quatro Metaheurísticas referidas, na resolução de Problemas de Job-Shop, tendo o menor número de artigos disponíveis no *Web of Science*. Assim, os artigos do SA também compreendem uma janela temporal mais alargada, de 2025 até 2011.

Após a respetiva análise da Tabela 8 é possível perceber que o SA simples e Híbrido são muito utilizados neste tipo de Problemas. O SA também apresenta artigos onde a própria Metaheurística não constitui a melhor solução, em termos de *makespan*. Nos artigos [120] e [85] um TS e GA, respetivamente, apresentam um menor *makespan* que o SA.

Relativamente aos restantes artigos, na sua grande maioria pode-se verificar que quando o SA é comparado a outras Metaheurísticas, a redução de *makespan* é modesta enquanto que quando o SA é comparado a um caso industrial ou a uma *Dispatching Rule* a redução de *makespan* é maior. Os artigos [143] e [150] constituem-se como exceções, sendo que apesar do SA estar a ser comparado a outras Metaheurísticas as respetivas reduções de *makespan* são elevadas.



## 4. Resultados e Discussão

No capítulo 4 procedeu-se à exposição e organização dos dados reunidos relativamente aos artigos selecionados, com o propósito de perceber como estes se encontram distribuídos em diversas categorias. Seguidamente, realizaram-se testes estatísticos que permitiram chegar a várias conclusões quanto ao desempenho das quatro Metaheurísticas em foco deste trabalho. Para tal, utilizou-se o *software* estatístico IBM SPSS *Statistics*, no qual foram importados todos os dados dos artigos científicos.

Assim, este capítulo encontra-se dividido em dois subcapítulos: no primeiro é explicado quais as variáveis adicionadas no SPSS, bem como a importância que estas apresentam, realizando-se de seguida uma análise descritiva dos dados e no segundo é realizada a inferência estatística, através de diversos testes estatísticos, retirando-se as respetivas conclusões mediante os resultados obtidos.

### 4.1. Análise Descritiva

Antes de se realizar a análise descritiva dos dados, foi necessário importá-los no *software* estatístico SPSS, sendo que para tal foi essencial criar todas as variáveis que permitiam caracterizar cada artigo na sua totalidade. Assim, foram criadas nove variáveis:

- Número correspondente à Referência do Artigo: uma variável escalar que tem como única função a fácil rastreabilidade dos artigos na base de dados do SPSS, com recurso ao número de referência do artigo;
- Nome da Metaheurística: uma variável nominal que tem como papel identificar qual a Metaheurística presente no dado artigo, dividindo-se em GA, PSO, TS e SA;
- Resultado da Metaheurística relativamente ao Makespan: uma variável escalar que indica o valor absoluto de *makespan* obtido em cada artigo pela respetiva Metaheurística;
- Variação do Makespan (%): uma variável escalar que indica a variação de *makespan* obtida, em percentagem. Um valor negativo significa que uma redução de *makespan* foi obtida através do uso da respetiva Metaheurística, pelo que um valor positivo significa que a utilização da dada Metaheurística não resultou numa redução do *makespan*;

## Resultados e Discussão

- Resultado Ótimo relativamente ao Makespan: uma variável escalar que indica o resultado ótimo de *makespan* num dado artigo, caso este seja mencionado;
- Ao qual a Metaheurística é comparada relativamente ao Makespan: uma variável nominal que tem como papel identificar o método de comparação utilizado para comparar o desempenho de uma Metaheurística, dividindo-se em Outras Metaheurísticas, Caso Industrial, *Dispatching Rule* e Resultado Ótimo;
- Ano de Publicação do Artigo: uma variável escalar que tem como função identificar o ano de publicação do respetivo artigo;
- Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%): uma variável escalar que indica o quão próximo o resultado em *makespan* de uma Metaheurística de um dado artigo está do seu respetivo resultado ótimo, em percentagem. Quanto mais próximo do 0% mais perto o resultado da Metaheurística se encontra do resultado ótimo;
- Metaheurística Híbrida ou Não Híbrida: uma variável nominal que indica se a Metaheurística utilizada num dado artigo é Híbrida ou Não Híbrida.

Com as nove variáveis criadas no SPSS devidamente identificadas e explicadas e com os dados dos artigos importados, pode-se proceder à análise descritiva dos dados. Como já foi referido, selecionaram-se 20 artigos científicos por cada Metaheurística em foco neste trabalho, no entanto cada artigo pode apresentar mais que um resultado, quer seja por existir mais que um resultado de *makespan* ou devido à forma como a comparação de *makespan* é realizada. Por exemplo: num dado artigo o GA é utilizado, obtendo-se então um resultado de *makespan*. No entanto, se o GA desse artigo for comparado ao *makespan* de outras duas Metaheurísticas, o mesmo artigo apresenta dois resultados diferentes para a variação de *makespan*, e se o *makespan* desse GA ainda for comparado a um Caso Industrial o mesmo artigo irá apresentar três resultados diferentes para a variação de *makespan*. Isto explica como apesar de existirem apenas 80 artigos selecionados, existem 145 resultados de artigos publicados.

Na Figura 6 pode-se observar a distribuição do número de resultados de artigos publicados de acordo com o ano de publicação do artigo.

A maioria dos resultados encontram-se concentrados nos anos mais recentes, sendo importante mencionar que na altura da realização deste trabalho ainda não se tinha atingido a metade do ano de 2025, caso contrário poderia apresentar um maior número de resultados.

## Resultados e Discussão

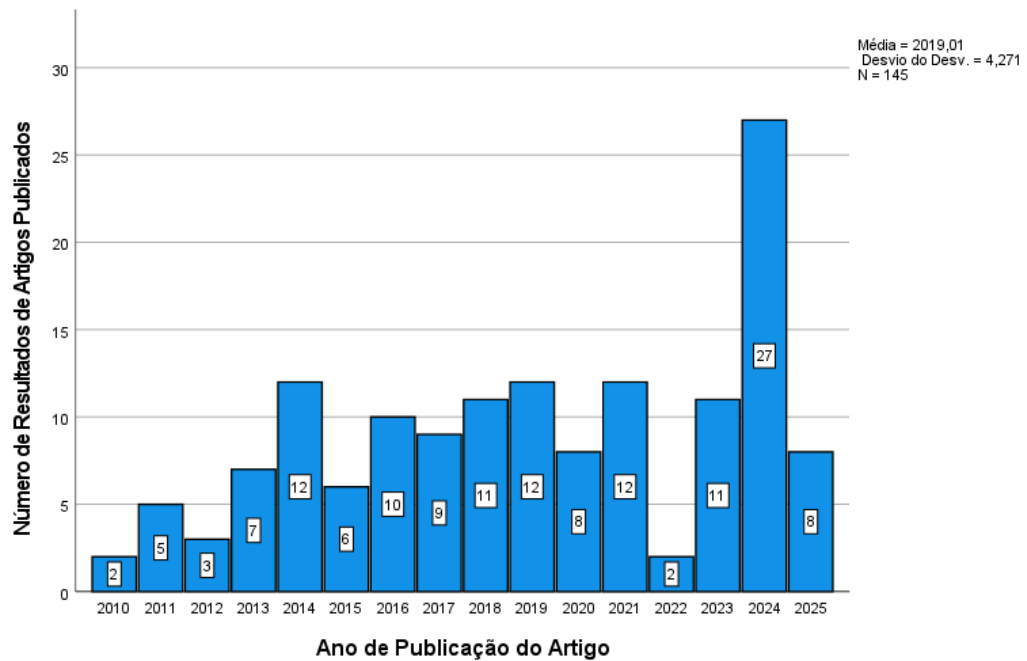


Figura 6 - Distribuição Temporal de Resultados de Artigos

Na Figura 7 pode-se observar a distribuição do número de resultados de artigos publicados de acordo com a respetiva Metaheurística.

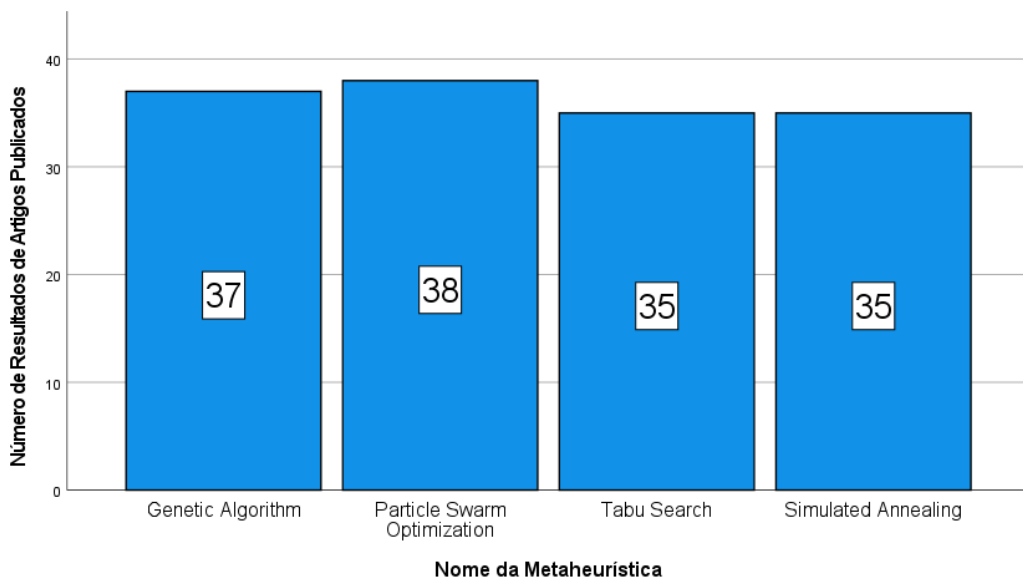


Figura 7 - Distribuição de Resultados de Artigos por Metaheurística

Pela análise da Figura 7 percebe-se que os resultados estão distribuídos de forma equilibrada, apresentando quase um número de resultados idêntico. No entanto, era o esperado, dado que foram seleccionados o mesmo número de artigos para cada Metaheurística.

Na Figura 8 pode-se observar a distribuição do número de resultados ótimos de artigos publicados de acordo com a respetiva Metaheurística.

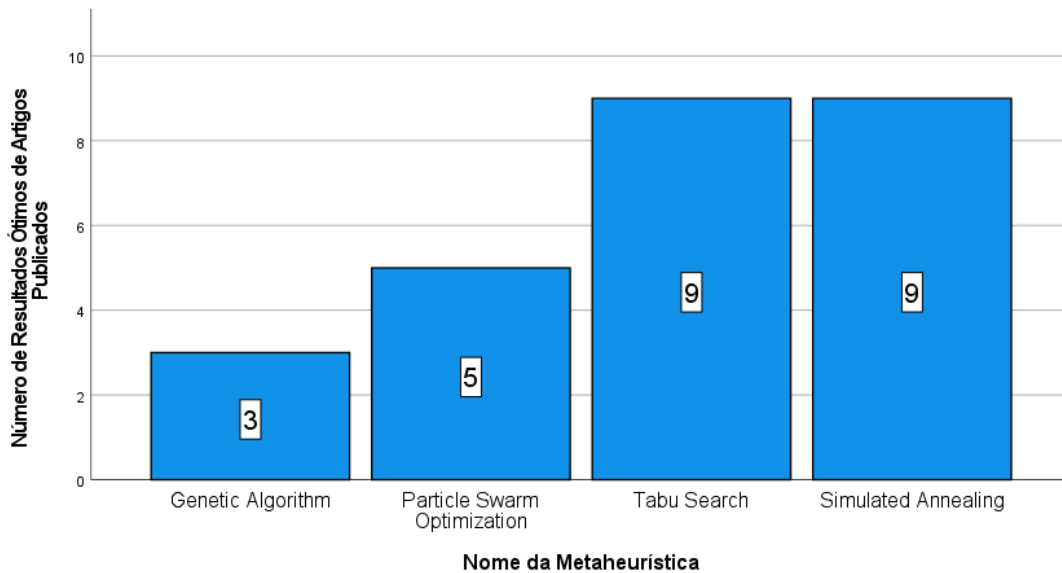


Figura 8 - Distribuição de Resultados Ótimos de Artigos por Metaheurística

Pela análise da Figura 8 percebe-se que o GA possui o menor número de resultados ótimos disponíveis, de entre as quatro Metaheurísticas, enquanto que o TS e o SA possuem o maior número de resultados ótimos disponíveis. Comparativamente ao número de resultados totais de cada Metaheurística apresentado anteriormente na Figura 7, o número de resultados ótimos disponíveis é muito baixo, o que irá acabar por influenciar na robustez de alguns dos testes estatísticos inferenciais que se irão realizar no próximo subcapítulo.

Na Figura 9 pode-se observar a distribuição do número de resultados de artigos publicados relativamente ao método de comparação.

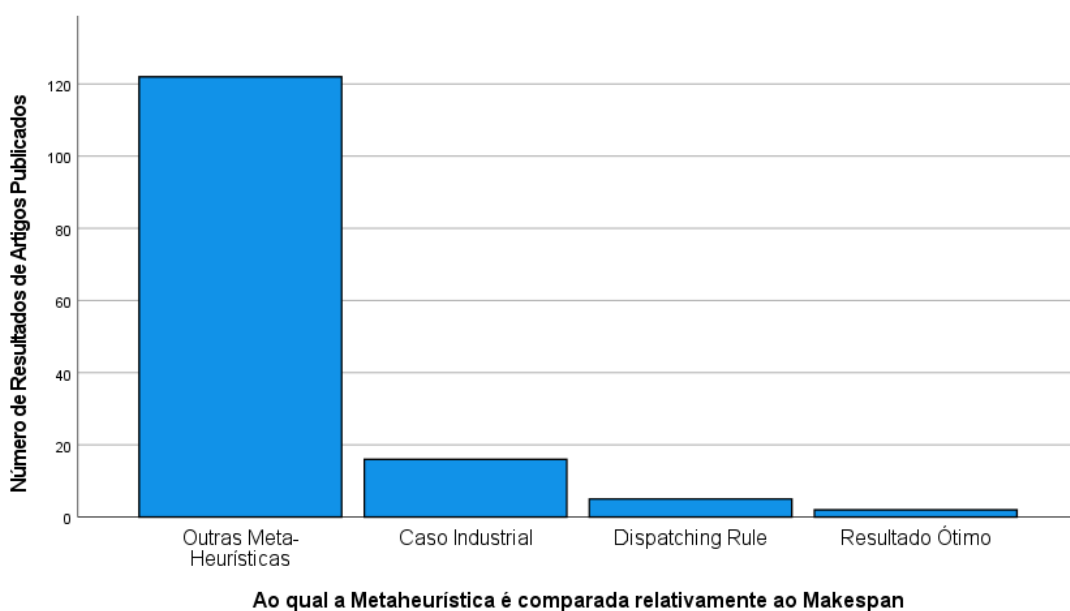


Figura 9 - Distribuição de Resultados de Artigos por Método de Comparação

Nos artigos reunidos, o *makespan* proveniente da utilização de uma dada Metaheurística tende a ser comparado ao de outras Metaheurísticas, tratando-se de uma diferença bastante notória quando se observam os restantes métodos de comparação. Apenas existem alguns casos onde o resultado de *makespan* é comparado com um Caso Industrial, *Dispatching Rule* ou com um Resultado Ótimo.

Na Figura 10 pode-se observar a distribuição dos métodos de comparação segundo o tipo de Metaheurística.

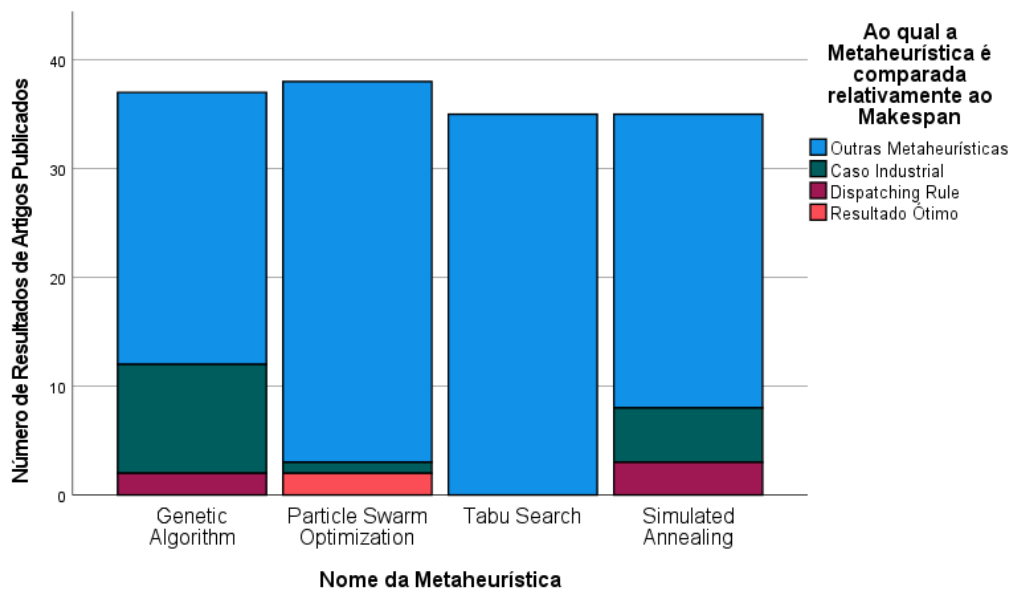


Figura 10 - Distribuição de Métodos de Comparação por Metaheurística

Para as quatro Metaheurísticas, o método de comparação Outras Metaheurísticas é o mais utilizado que é o que se estava à espera dado o que foi observado na Figura 9. O GA é a Metaheurística que apresenta o maior número de resultados comparados a um Caso Industrial e o SA é a Metaheurística que apresenta o maior número de resultados comparados a uma *Dispatching Rule*. O PSO é a única Metaheurística que apresenta resultados comparados a um Resultado Ótimo.

Na Figura 11 pode-se observar a distribuição de Metaheurísticas Híbridas e Não Híbridas de acordo com o tipo de Metaheurística.

Segundo os artigos reunidos, existem mais resultados provenientes de Metaheurísticas Não Híbridas relativamente ao GA e PSO, enquanto que o TS e SA apresentam mais resultados provenientes de Metaheurísticas Híbridas. Considerou-se a Metaheurística como Híbrida quando o próprio artigo a mencionou por esse nome, caso contrário considerou-se como Não Híbrida.

## Resultados e Discussão

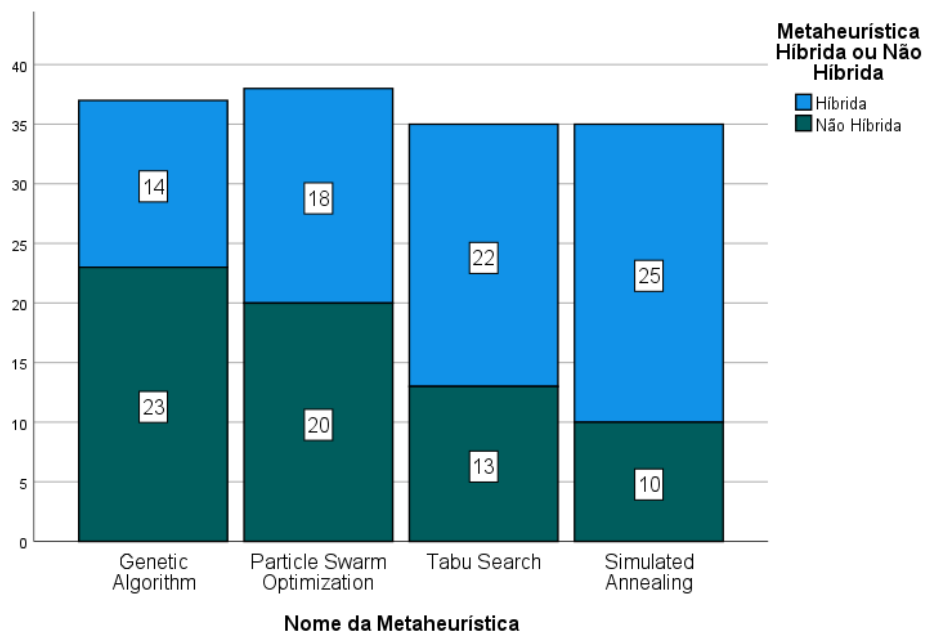


Figura 11 - Distribuição de Metaheurísticas Híbridas e Não Híbridas por tipo de Metaheurística

Na Tabela 9 podem-se observar as Estatísticas Descritivas Gerais das variáveis de maior relevância.

Tabela 9 - Estatísticas Descritivas Gerais

Estatísticas Descritivas Gerais	N	Mínimo	Máximo	Média	Desvio Padrão
<b>Resultado da Metaheurística relativamente ao <i>Makespan</i></b>	141	9	81780	1200.72	6861.43
<b>Variação do <i>Makespan</i> (%)</b>	145	-79.2	10.59	-8.28	13.95
<b>Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%)</b>	26	0	33.34	4.43	7.00

As estatísticas descritivas gerais da primeira variável da Tabela 9 não são muito importantes, dado que os valores absolutos de *makespan* variam bastante entre problemas. A média e desvio padrão elevados indicam uma grande dispersão dos valores, refletindo que os artigos analisaram problemas de diferentes dimensões. Assim, a variável do Resultado da Metaheurística relativamente ao *Makespan* serve apenas como contextualização da diversidade dos artigos analisados. Relativamente à segunda variável da Tabela 9, da Variação do *Makespan* (%), a média negativa indica que as Metaheurísticas conseguiram reduzir o *makespan* em cerca de 8%, relativamente ao método de comparação utilizado. O mínimo de -79.2% mostra que houve casos onde a redução de *makespan* foi muito significativa e o máximo de 10.59% indica que, em alguns casos, o desempenho da Metaheurística foi pior relativamente ao método de comparação. O desvio padrão de 13.95 ainda mostra uma variabilidade

## Resultados e Discussão

significativa, o que significa que o desempenho depende à partida da Metaheurística e de outros fatores. Por último, em relação à última variável da Tabela 9, da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%), a média de 4.43% mostra que, quando o resultado ótimo é conhecido, as Metaheurísticas tendem a chegar bastante perto do ótimo. O desvio padrão ainda é moderado, sugerindo uma consistência razoável. No entanto, o número de casos (N = 26) é limitado, o que fragiliza a robustez estatística da variável.

Na Figura 12 e Figura 13 pode-se observar como as variáveis Variação do Makespan (%) e Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) se distribuem de acordo com o tipo de Metaheurística.

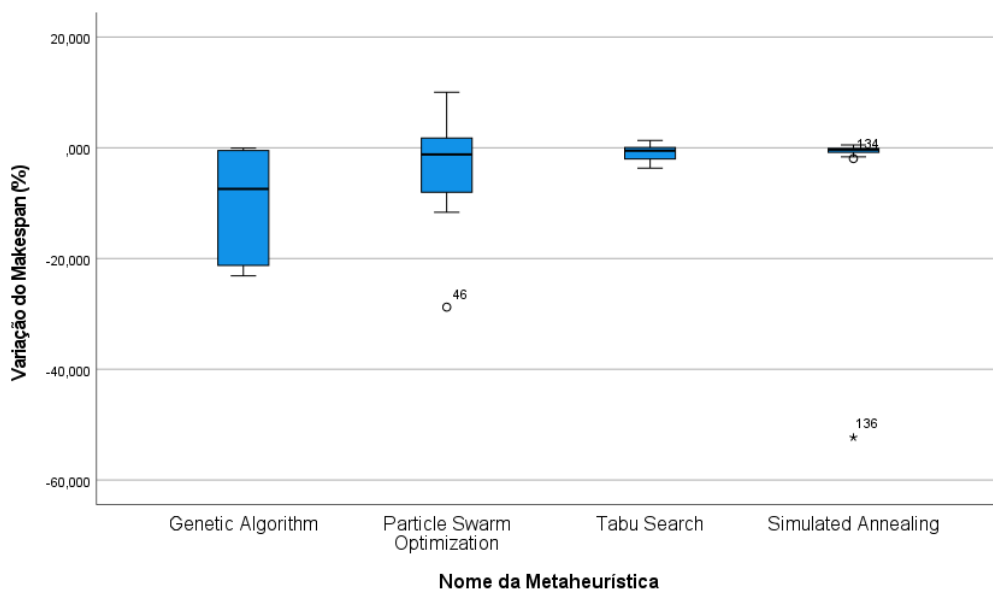


Figura 12 - Variação de *Makespan* (%) por Metaheurística

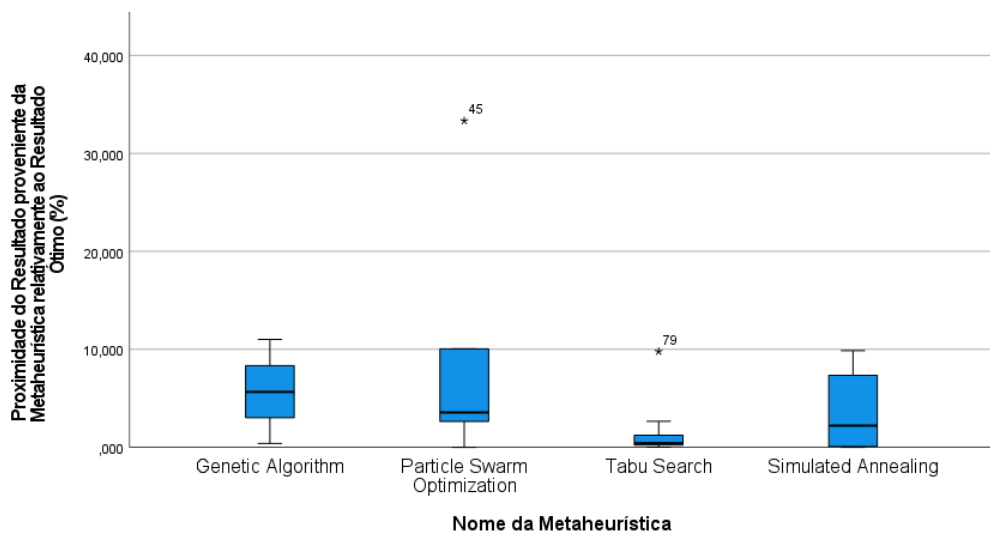


Figura 13 - Proximidade de Resultado Ótimo (%) por Metaheurística

## Resultados e Discussão

Após a análise da Figura 12 e Figura 13, o GA aparenta possuir as maiores reduções de *makespan*, enquanto o TS aparenta ter os resultados de *makespan* mais próximos do resultado ótimo.

Na Figura 14 pode-se observar como a variável Variação do Makespan (%) se distribui de acordo com o método de comparação de resultados.

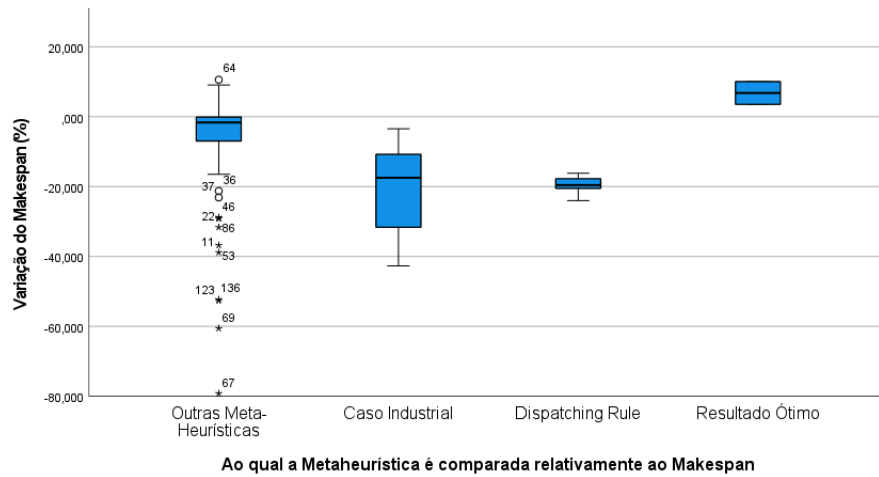


Figura 14 - Variação de *Makespan* (%) por Método de Comparação

Após a análise da Figura 14, o Caso Industrial e a *Dispatching Rule* aparentam ser os métodos de comparação que proporcionam as maiores reduções de *makespan*, apesar dos inúmeros *outliers* que as Outras Metaheurísticas apresentam. Por razões óbvias, quando o resultado de uma Metaheurística é comparado a um Resultado Ótimo não existe nenhuma redução de *makespan*.

Por último, na Figura 15 e Figura 16 pode-se observar como as variáveis Variação do Makespan (%) e Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) se distribuem consoante a Metaheurística ser Híbrida ou Não Híbrida.

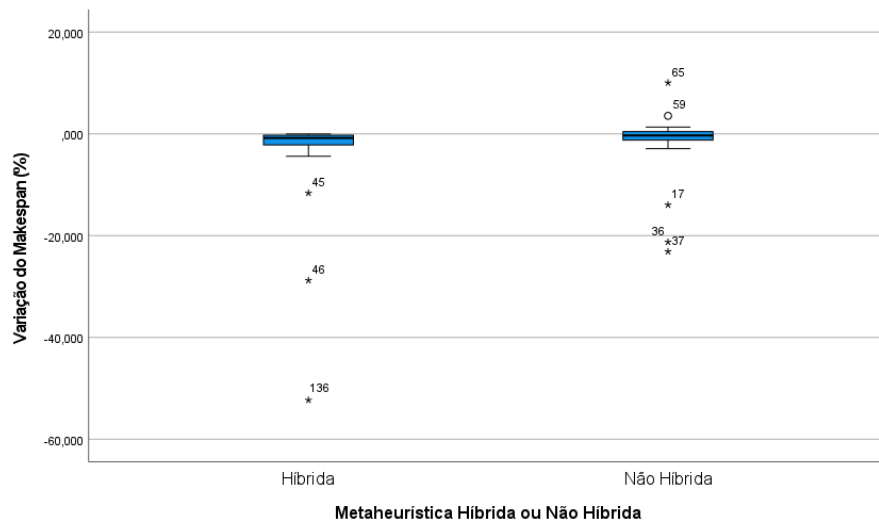


Figura 15 - Variação de *Makespan* (%) por Metaheurística Híbrida ou Não Híbrida

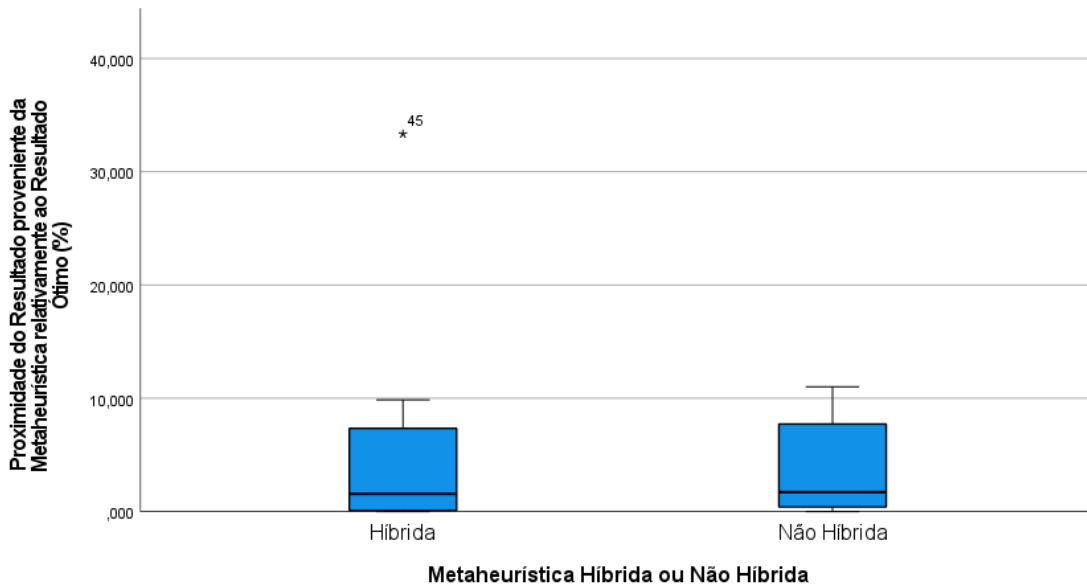


Figura 16 - Proximidade de Resultado Ótimo (%) por Metaheurística Híbrida ou Não Híbrida

Após a análise da Figura 15 e Figura 16, as Metaheurísticas Híbridas aparentam proporcionar uma maior redução de *makespan* e uma maior proximidade ao Resultado Ótimo, apesar de parecer ser por pouco.

A análise descritiva realizada aos dados reunidos permitiu oferecer um maior grau de profundidade aos artigos selecionados, percebendo-se quais os anos que apresentam o maior número de resultados de artigos publicados, de que forma estes resultados se distribuíam conforme o tipo de Metaheurística, quais os métodos de comparação mais presentes nos artigos, bem como em cada tipo de Metaheurística e de que modo as Metaheurísticas Híbridas e Não Híbridas se distribuíam segundo o tipo de Metaheurística. A análise descritiva ainda sugeriu que determinadas Metaheurísticas tendem a apresentar melhores desempenhos em termos de variação de *makespan* (como o GA e PSO) e proximidade ao resultado ótimo (como o TS e SA), que determinados métodos de comparação resultam em maiores reduções de *makespan* (como a *Dispatching Rule* e Caso Industrial) e ainda que a Metaheurística ser Híbrida ou Não Híbrida não é um fator decisivo na variação de *makespan* e proximidade ao resultado ótimo. No entanto, estas observações carecem de validação estatística formal, o que será então explorado no subcapítulo seguinte através da realização de testes estatísticos inferenciais.

## 4.2. Inferência Estatística e Discussão de Resultados

Depois da análise descritiva é necessário realizar os testes estatísticos inferenciais. Estes têm como objetivo oferecer a validação estatística que carecia das observações realizadas no subcapítulo anterior, bem como permitir chegar a outras conclusões acerca dos dados reunidos. Como tal este subcapítulo vai estar dividido em cinco subcapítulos, por cada categoria de testes estatísticos realizados, apresentando no final um último subcapítulo com as conclusões retiradas das análises estatísticas realizadas.

#### 4.2.1. Análise de Evolução dos Resultados ao longo dos Anos

Esta análise tem como papel verificar se existe uma tendência de melhoria no desempenho das Metaheurísticas com o tempo, de 2010 a 2025. Primeiro realizou-se a análise para as Metaheurísticas como um todo e de seguida para cada uma das quatro Metaheurísticas. Para esta análise as variáveis de foco foram a Variação do *Makespan* (%) e Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%), tendo sido realizado gráficos de dispersão com linhas de tendência e regressões lineares simples.

Na Figura 17 tem-se o gráfico de dispersão realizado da Variação de *Makespan* (%) por Ano de Publicação do Artigo e na Tabela 10 os resultados provenientes da regressão linear simples.

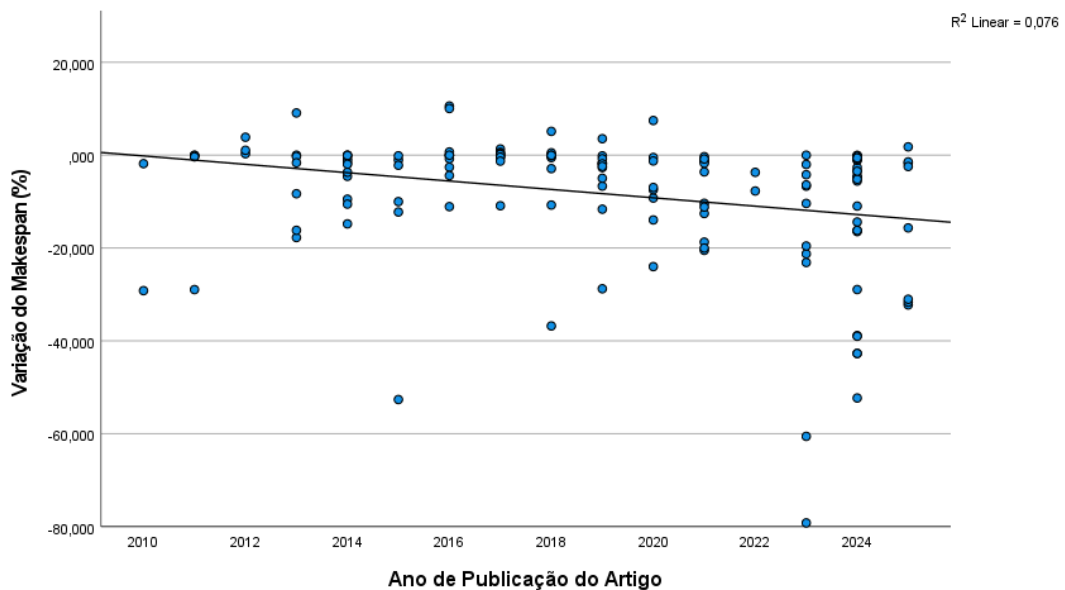


Figura 17 - Gráfico de Dispersão das Metaheurísticas relativamente à Variação de *Makespan* (%)

Tabela 10 - Regressão Linear Simples das Metaheurísticas relativamente à Variação de *Makespan* (%)

Regressão Linear Simples	R <sup>2</sup>	B	Sig
Ano de Publicação do Artigo	0.076	-0.901	< 0.001

Através da análise do gráfico de dispersão nota-se que a linha de tendência diminui ao longo dos anos, indicando inicialmente que a variação de *makespan* aumenta com o passar dos anos. Os resultados da regressão linear simples corroboram isto, demonstrando que o modelo é estatisticamente significativo (Sig < 0.001), com um coeficiente de determinação (R<sup>2</sup>) de 0.076. Isto significa que cerca de 7.6% da variação na variação de *makespan* pode ser explicada pelo ano de publicação do artigo, pelo que a maior parte da variação de *makespan* ainda depende

## Resultados e Discussão

de outros fatores não incluídos nos dados. O coeficiente de regressão ( $B = -0.901$ ) indica que, em média, a variação de *makespan* tem vindo a diminuir 0,901% por ano. Logo, pode-se concluir que, ao longo dos anos, as Metaheurísticas têm atingido maiores reduções de *makespan*.

Na Figura 18 tem-se o gráfico de dispersão realizado da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) por Ano de Publicação do Artigo e na Tabela 11 os resultados provenientes da regressão linear simples.

Este novo gráfico de dispersão apresenta uma linha de tendência com um declive negativo muito ligeiro, contrastando com a anterior que tinha um declive mais acentuado. O modelo de regressão revelou-se estatisticamente não significativo ( $Sig = 0.917$ ), com um  $R^2$  de 0. Isto significa que a variação na proximidade ao ótimo não pode ser explicada pelo ano de publicação do artigo. O B também não foi estatisticamente significativo, reforçando a ausência de uma relação relevante entre as variáveis. Logo, não se pode concluir que, ao longo dos anos, os resultados das Metaheurísticas se têm aproximado do ótimo. Importante referir que isto se pode dever ao facto de a grande maioria dos artigos reunidos não apresentar um resultado ótimo, pelo que com um maior número de resultados ótimos o teste seria mais preciso.

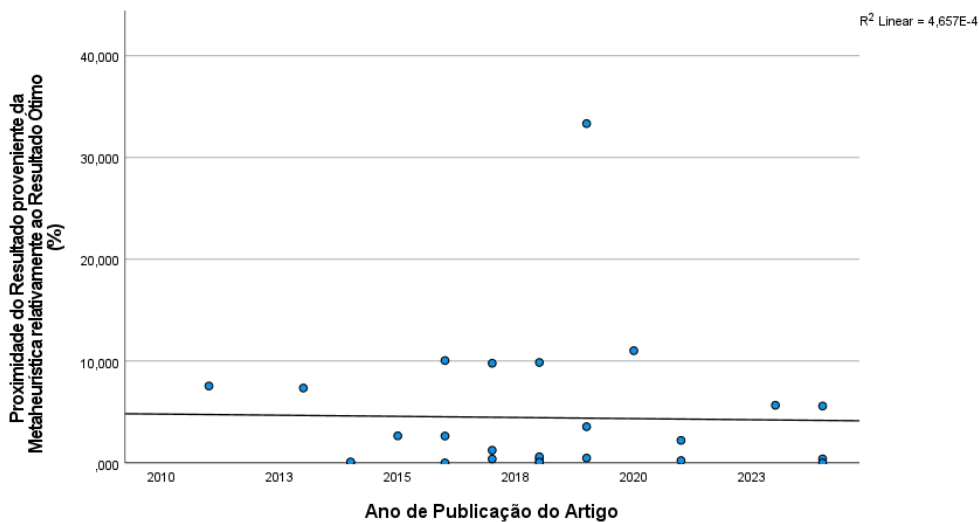


Figura 18 - Gráfico de Dispersão das Metaheurísticas relativamente à Proximidade de Resultado Ótimo (%)

Tabela 11 - Regressão Linear Simples das Metaheurísticas relativamente à Proximidade de Resultado Ótimo (%)

Regressão Linear Simples	$R^2$	B	Sig
Ano de Publicação do Artigo	0	-0.045	0.917

Com os testes realizados para as Metaheurísticas como um todo, decidiu-se também realizá-los para cada uma das quatro Metaheurísticas em foco, com o intuito de verificar se iriam existir diferenças relativas aos resultados anteriores.

## Resultados e Discussão

Começando pelo GA, na Figura 19 tem-se o gráfico de dispersão realizado da Variação de *Makespan* (%) por Ano de Publicação do Artigo e na Tabela 12 os resultados provenientes da regressão linear simples.

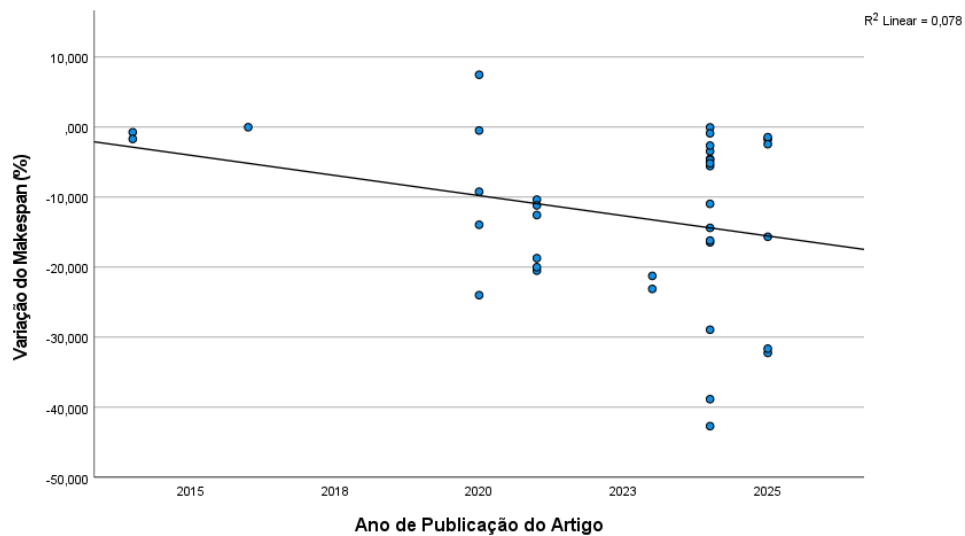


Figura 19 - Gráfico de Dispersão do GA relativamente à Variação de *Makespan* (%)

Tabela 12 - Regressão Linear Simples do GA relativamente à Variação de *Makespan* (%)

Regressão Linear Simples	R <sup>2</sup>	B	Sig
Ano de Publicação do Artigo	0.078	-1.152	0.095

O gráfico de dispersão apresenta uma linha de tendência com declive negativo acentuado, indicando inicialmente que a variação de *makespan* aumenta ao longo dos anos. No entanto, os resultados provenientes da regressão linear simples não corroboram isto totalmente. O modelo apresenta um R<sup>2</sup> de 0.078, indicando que 7.8% da variação na variação de *makespan* pode ser explicada pelo ano de publicação do artigo. Apesar do modelo não ser estatisticamente significativo ao nível de 5% (Sig = 0.095), é possível observar uma tendência de melhoria, com o B a indicar que, aproximadamente, a variação de *makespan* diminui 1.152% ao ano. Assim, para o GA há indícios de maiores reduções de *makespan* ao longo dos anos, apesar de a evidência estatística não ser significativamente forte.

Na Figura 20 tem-se o gráfico de dispersão realizado da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) por Ano de Publicação do Artigo e na Tabela 13 os resultados provenientes da regressão linear simples.

O gráfico de dispersão apresenta também uma linha de tendência com um declive negativo acentuado, indicado inicialmente que a proximidade ao ótimo aumenta ao longo dos anos. Contudo, os resultados provenientes da regressão linear simples não corroboram isto. O modelo apresenta um R<sup>2</sup> de 0.926, indicando que 92.6% da variação na proximidade ao ótimo pode ser explicada pelo ano de publicação do artigo. No entanto, o modelo revelou-se estatisticamente não significativo (Sig = 0.176), com o B a indicar que, aproximadamente, a

## Resultados e Discussão

proximidade ao ótimo diminui 2.457% ao ano. Assim, não se pode concluir que, ao longo dos anos, os resultados do GA se têm aproximado do ótimo. É importante referir novamente que é necessário ter em conta que na realização destes testes existiam muito poucos resultados ótimos face aos resultados totais.

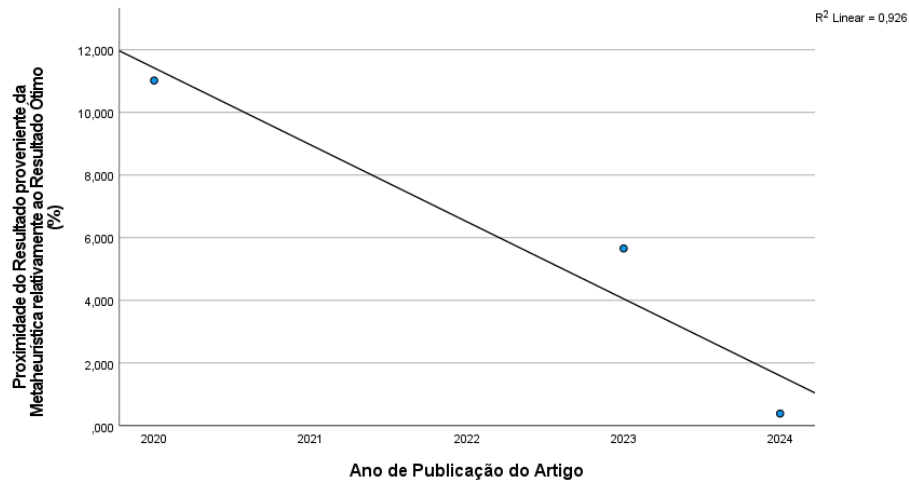


Figura 20 - Gráfico de Dispersão do GA relativamente à Proximidade de Resultado Ótimo (%)

Tabela 13 - Regressão Linear Simples do GA relativamente à Proximidade de Resultado Ótimo (%)

Regressão Linear Simples	R <sup>2</sup>	B	Sig
Ano de Publicação do Artigo	0.926	-2.457	0.176

Relativamente ao PSO, na Figura 21 tem-se o gráfico de dispersão realizado da Variação de *Makespan* (%) por Ano de Publicação do Artigo e na Tabela 14 os resultados provenientes da regressão linear simples.

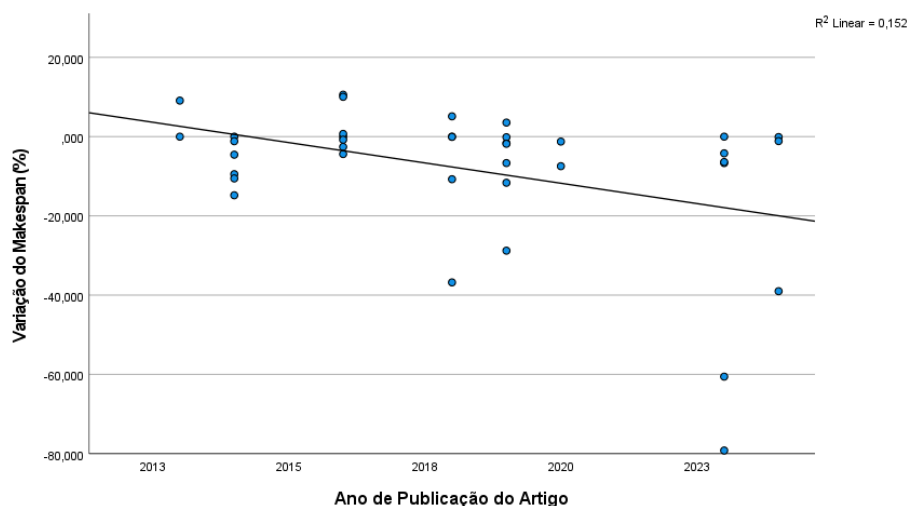


Tabela 14 - Regressão Linear Simples do PSO relativamente à Variação de *Makespan* (%)

Regressão Linear Simples	R <sup>2</sup>	B	Sig
Ano de Publicação do Artigo	0.152	-2.052	0.016

O gráfico de dispersão apresenta uma linha de tendência com declive negativo acentuado, indicando inicialmente que a variação de *makespan* aumenta ao longo dos anos. Os resultados da regressão linear simples corroboram isto, demonstrando que o modelo é estatisticamente significativo (Sig = 0.016), com um R<sup>2</sup> de 0.152. Isto significa que cerca de 15.2% da variação na variação de *makespan* pode ser explicada pelo ano de publicação do artigo. O B indica que, em média, a variação de *makespan* tem vindo a diminuir 2.052% por ano. Logo, pode-se concluir que, ao longo dos anos, o PSO tem atingido maiores reduções de *makespan*.

Na Figura 22 tem-se o gráfico de dispersão realizado da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) por Ano de Publicação do Artigo.



Figura 22 - Gráfico de Dispersão do PSO relativamente à Proximidade de Resultado Ótimo (%)

O gráfico de dispersão apresenta uma linha de tendência com declive positivo acentuado, contrariamente aos gráficos de dispersão anteriores. Logo, isto indica que para o PSO a proximidade ao ótimo tem diminuído ao longo dos anos. Isto mais certamente se deve ao facto de, como já foi referido, não existirem muitos resultados ótimos registados em artigo face aos resultados totais, não se devendo concluir à partida que o PSO tem piorado em desempenho com a passagem dos anos.

Relativamente ao TS, na Figura 23 tem-se o gráfico de dispersão realizado da Variação de *Makespan* (%) por Ano de Publicação do Artigo e na Tabela 15 os resultados provenientes da regressão linear simples.

Este gráfico de dispersão também apresenta uma linha de tendência com um declive positivo acentuado, indicando inicialmente que para o TS a variação de *makespan* diminui ao longo dos anos. No entanto, os resultados provenientes da regressão linear simples não corroboram isto totalmente. O modelo apresenta um  $R^2$  de 0.089, indicando que 8.9% da variação na variação de *makespan* pode ser explicada pelo ano de publicação do artigo. Apesar do modelo não ser estatisticamente significativo ao nível de 5% (Sig = 0.081), é possível observar uma tendência de agravamento, com o B a indicar que, aproximadamente, a variação de *makespan* aumenta 0.564% ao ano. Assim, para o TS há indícios de menores reduções de *makespan* ao longo dos anos, apesar de a evidência estatística não ser significativamente forte.

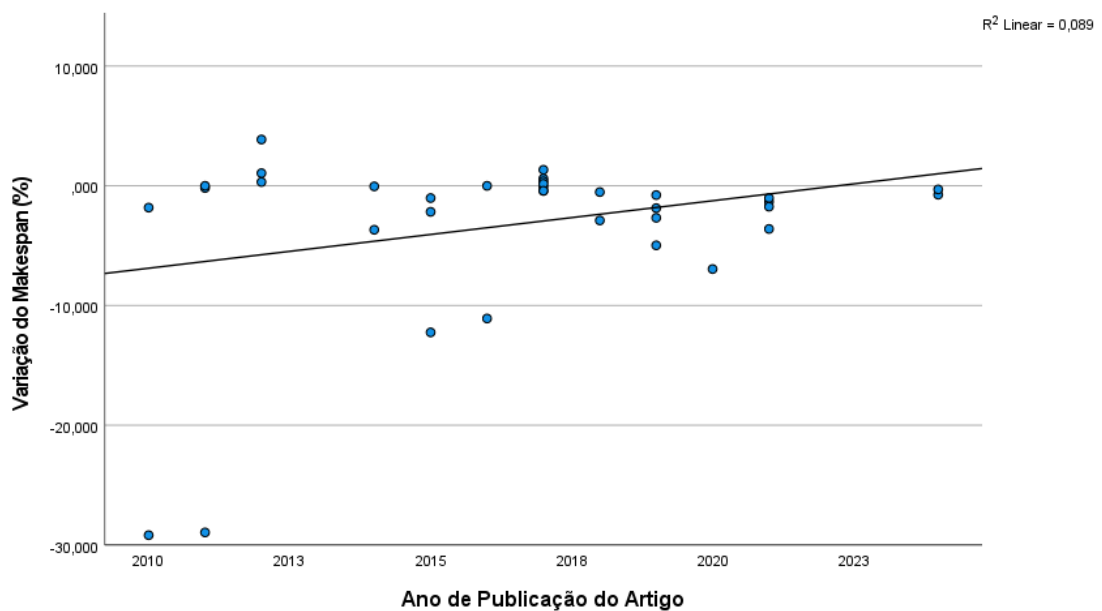


Figura 23 - Gráfico de Dispersão do TS relativamente à Variação de *Makespan* (%)

Tabela 15 - Regressão Linear Simples do TS relativamente à Variação de *Makespan* (%)

Regressão Linear Simples	$R^2$	B	Sig
Ano de Publicação do Artigo	0.089	0.564	0.081

Na Figura 24 tem-se o gráfico de dispersão realizado da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) por Ano de Publicação do Artigo e na Tabela 16 os resultados provenientes da regressão linear simples.

## Resultados e Discussão

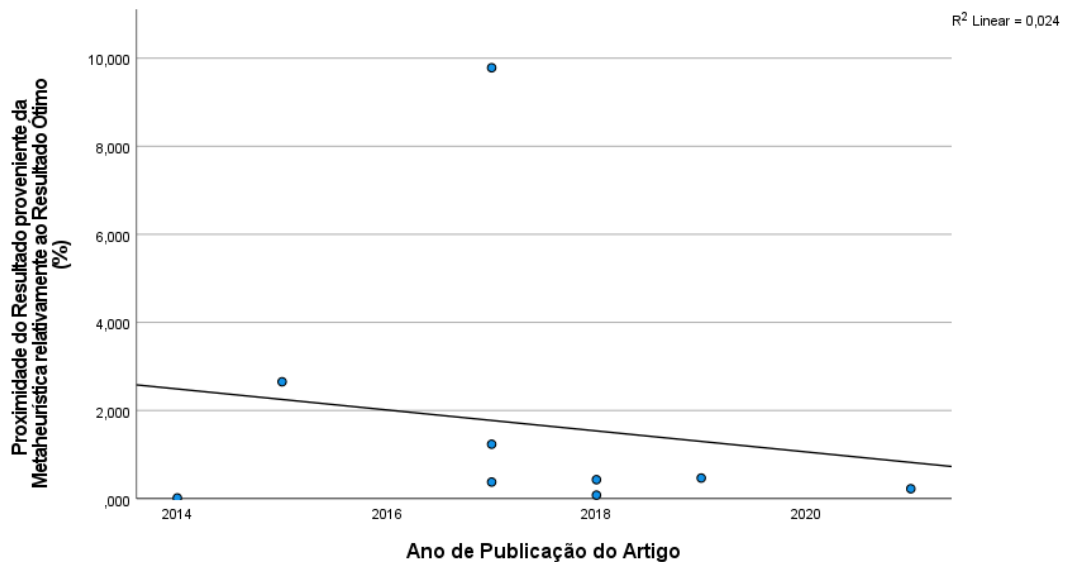


Figura 24 - Gráfico de Dispersão do TS relativamente à Proximidade de Resultado Ótimo (%)

Tabela 16 - Regressão Linear Simples do TS relativamente à Proximidade de Resultado Ótimo (%)

Regressão Linear Simples	R <sup>2</sup>	B	Sig
Ano de Publicação do Artigo	0.024	-0.238	0.688

Este gráfico de dispersão apresenta uma linha de tendência com um declive negativo ligeiro, indicando inicialmente que os resultados do TS, ao longo dos anos, se aproximam do ótimo. No entanto, o modelo de regressão revelou-se estatisticamente não significativo (Sig = 0.688), com um R<sup>2</sup> de 0.024. Isto significa que apenas 2.4% da variação na proximidade ao ótimo pode ser explicada pelo ano de publicação do artigo. O B também não foi estatisticamente significativo, reforçando a ausência de uma relação relevante entre as variáveis. Logo, não se pode concluir que, ao longo dos anos, os resultados do TS se têm aproximado do ótimo.

Por último, relativamente ao SA, na Figura 25 tem-se o gráfico de dispersão realizado da Variação de *Makespan* (%) por Ano de Publicação do Artigo e na Tabela 17 os resultados provenientes da regressão linear simples.

O gráfico de dispersão apresenta uma linha de tendência com declive negativo, indicando inicialmente que, para o SA, a variação de *makespan* aumenta ao longo dos anos. No entanto, os resultados provenientes da regressão linear simples não corroboram isto. O modelo de regressão revelou-se estatisticamente não significativo (Sig = 0.283), com um R<sup>2</sup> de 0.035. Isto significa que apenas 3.5% da variação na variação de *makespan* pode ser explicada pelo ano de publicação do artigo. O B também não foi estatisticamente significativo, reforçando a ausência de uma relação relevante entre as variáveis. Logo, não se pode concluir que o SA, ao longo dos anos, tem atingido maiores reduções de *makespan*.

## Resultados e Discussão

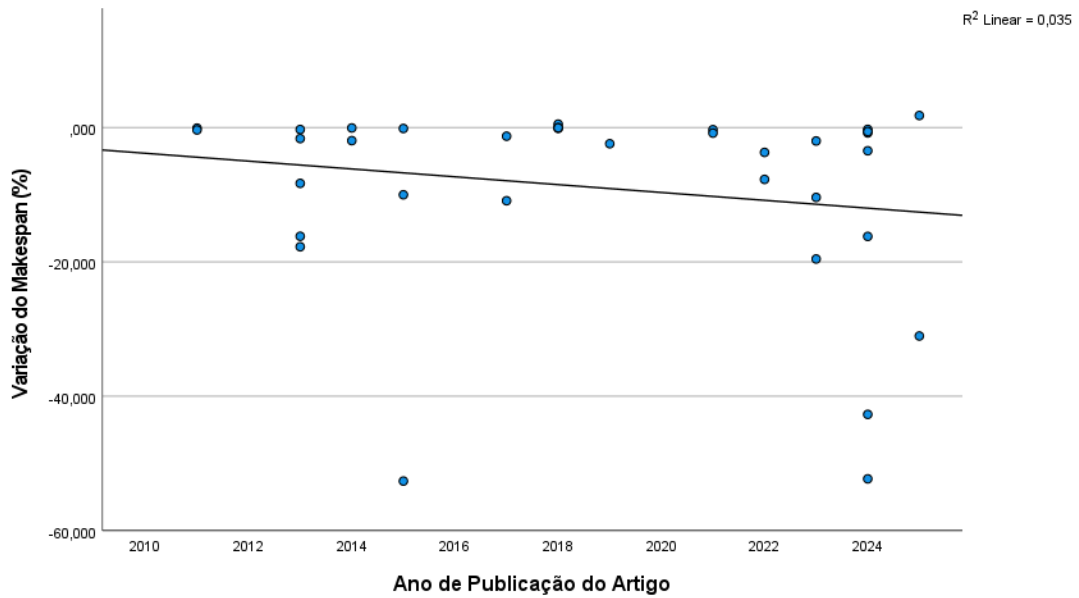


Figura 25 - Gráfico de Dispersão do SA relativamente à Variação de *Makespan* (%)

Tabela 17 - Regressão Linear Simples do SA relativamente à Variação de *Makespan* (%)

Regressão Linear Simples	R <sup>2</sup>	B	Sig
Ano de Publicação do Artigo	0.035	-0.584	0.283

Na Figura 26 tem-se o gráfico de dispersão realizado da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) por Ano de Publicação do Artigo e na Tabela 18 os resultados provenientes da regressão linear simples.

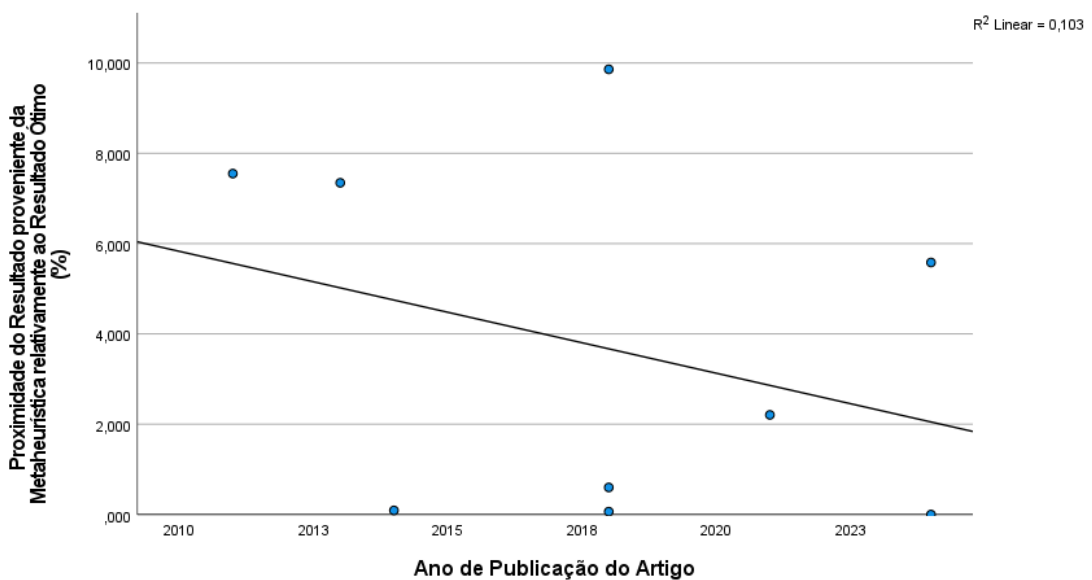


Figura 26 - Gráfico de Dispersão do SA relativamente à Proximidade de Resultado Ótimo (%)

Tabela 18 - Regressão Linear Simples do SA relativamente à Proximidade de Resultado Ótimo (%)

<b>Regressão Linear Simples</b>	<b>R<sup>2</sup></b>	<b>B</b>	<b>Sig</b>
<b>Ano de Publicação do Artigo</b>	0.103	-0.270	0.401

Este gráfico de dispersão apresenta uma linha de tendência com um declive negativo acentuado, indicando inicialmente que os resultados do SA, ao longo dos anos, se aproximam do ótimo. No entanto, o modelo de regressão revelou-se estatisticamente não significativo (Sig = 0.401), com um R<sup>2</sup> de 0.103. Isto significa que 10.3% da variação na proximidade ao ótimo pode ser explicada pelo ano de publicação do artigo. O B também não foi estatisticamente significativo, reforçando a ausência de uma relação relevante entre as variáveis. Logo, não se pode concluir que, ao longo dos anos, os resultados do SA se têm aproximado do ótimo.

#### 4.2.2. Análise de Comparação de Desempenho entre as Metaheurísticas

Esta análise tem como papel avaliar qual das quatro Metaheurísticas tende a apresentar os melhores resultados. Como tal, para esta análise as variáveis de foco foram a Variação do *Makespan* (%) e Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%), tendo sido realizado o teste estatístico não paramétrico de Kruskal-Wallis.

Antes de se chegar à decisão de utilizar este teste foi necessário primeiro verificar dois pressupostos: a existência de normalidade da variável dependente para cada grupo de Metaheurísticas e a existência de homogeneidade das variâncias entre os grupos das Metaheurísticas. Começando pela variável dependente da Variação do *Makespan* (%), realizou-se primeiro o teste de Shapiro-Wilk para verificar a normalidade por grupo. As hipóteses estabelecidas para este teste foram:

- Hipótese Nula (H<sub>0</sub>): A variável da Variação do *Makespan* (%) segue uma distribuição normal;
- Hipótese Alternativa (H<sub>1</sub>): A variável da Variação do *Makespan* (%) não segue uma distribuição normal.

Na Tabela 19 pode-se observar que todos os valores de Sig < 0.05 para Shapiro-Wilk, logo rejeita-se a H<sub>0</sub>, pelo que não existe normalidade nos grupos de dados.

Tabela 19 - Teste de Shapiro-Wilk para os dados da Variação de *Makespan* (%) por Metaheurística

<b>Teste de Shapiro-Wilk</b>		<b>Sig</b>
<b>Variação do <i>Makespan</i> (%)</b>	<b><i>Genetic Algorithm</i></b>	0.028
	<b><i>Particle Swarm Optimization</i></b>	< 0.001
	<b><i>Tabu Search</i></b>	< 0.001
	<b><i>Simulated Annealing</i></b>	< 0.001

Seguidamente foi necessário verificar a homogeneidade de variâncias entre os grupos, através do teste de Levene. As hipóteses estabelecidas para este teste foram:

- $H_0$ : As variâncias da variável da Variação do *Makespan* (%) são iguais entre os grupos;
- $H_1$ : Pelo menos um dos grupos apresenta uma variância diferente.

Na Tabela 20 pode-se observar que dois dos quatro valores de Sig < 0.05, logo rejeita-se a  $H_0$ , pelo que não existe homogeneidade de variâncias entre os grupos de dados.

Tabela 20 - Teste de Levene para os dados da Variação de *Makespan* (%) por Metaheurística

Teste de Levene		Sig
Variação do <i>Makespan</i> (%)	Com base em média	0.011
	Com base em mediana	0.081
	Com base em mediana e com gl ajustado	0.083
	Com base em média aparada	0.024

Como o grupo de dados da Variação de *Makespan* (%) não apresentava normalidade nem homogeneidade realizou-se o teste não paramétrico de Kruskal-Wallis. As hipóteses estabelecidas para este teste foram:

- $H_0$ : As distribuições da variável da Variação de *Makespan* (%) são iguais entre os grupos;
- $H_1$ : Pelo menos uma das distribuições difere entre os grupos.

Na Tabela 21 pode-se observar os valores obtidos do teste. Dado que Sig < 0.05, para o teste de Kruskal-Wallis, rejeita-se a  $H_0$ , indicando que existem diferenças estatisticamente significativas na distribuição da Variação de *Makespan* (%) entre as quatro Metaheurísticas analisadas.

Tabela 21 - Teste de Kruskal-Wallis para os dados da Variação de *Makespan* (%) por Metaheurística

Teste de Kruskal-Wallis		Posto Médio
Variação do <i>Makespan</i> (%)	<i>Genetic Algorithm</i>	50.07
	<i>Particle Swarm Optimization</i>	79.37
	<i>Tabu Search</i>	90.77
	<i>Simulated Annealing</i>	72.56
Estatística de Teste	Sig	< 0.001

Dado que para a variável Variação do *Makespan* (%), valores mais negativos indicam melhor desempenho (ou seja, maiores reduções de *makespan*), menores valores de posto médio no teste de Kruskal-Wallis correspondem a um melhor desempenho médio por parte da Metaheurística. Assim, o GA apresenta o melhor desempenho médio, seguido pelo SA, PSO e, por último, o TS que apresenta o pior desempenho médio. Estes resultados indicam que, em média, o GA proporcionou as maiores reduções no *makespan*, ao passo que o TS proporcionou as menores reduções no *makespan*.

Seguidamente foi necessário realizar o mesmo procedimento mas para a variável Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%). Realizou-se primeiro o teste de Shapiro-Wilk para verificar a normalidade por grupo. As hipóteses estabelecidas para este teste foram:

- $H_0$ : A variável da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) segue uma distribuição normal;
- $H_1$ : A variável da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) não segue uma distribuição normal.

Na Tabela 22 pode-se observar que dois dos quatro valores de Sig < 0.05 para Shapiro-Wilk, logo rejeita-se a  $H_0$ , pelo que não existe normalidade nos grupos de dados.

Tabela 22 - Teste de Shapiro-Wilk para os dados da Proximidade de Resultado Ótimo (%) por Metaheurística

Teste de Shapiro-Wilk		Sig
Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%)	<i>Genetic Algorithm</i>	0.991
	<i>Particle Swarm Optimization</i>	0.049
	<i>Tabu Search</i>	< 0.001
	<i>Simulated Annealing</i>	0.070

De seguida foi necessário verificar a homogeneidade de variâncias entre os grupos, através do teste de Levene. As hipóteses estabelecidas para este teste foram:

- $H_0$ : As variâncias da variável da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) são iguais entre os grupos;
- $H_1$ : Pelo menos um dos grupos apresenta uma variância diferente.

Na Tabela 23 pode-se observar que dois dos quatro valores de Sig < 0.05, logo rejeita-se a  $H_0$ , pelo que não existe homogeneidade de variâncias entre os grupos de dados.

Tabela 23 - Teste de Levene para os dados da Proximidade de Resultado Ótimo (%) por Metaheurística

Teste de Levene		Sig
Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%)	Com base em média	0.028
	Com base em mediana	0.265
	Com base em mediana e com gl ajustado	0.328
	Com base em média aparada	0.035

Como o grupo de dados da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) não apresentava normalidade nem homogeneidade realizou-se o teste não paramétrico de Kruskal-Wallis. As hipóteses estabelecidas para este teste foram:

- $H_0$ : As distribuições da variável da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) são iguais entre os grupos;
- $H_1$ : Pelo menos uma das distribuições difere entre os grupos.

Na Tabela 24 pode-se observar os valores obtidos do teste. Dado que  $Sig > 0.05$ , para o teste de Kruskal-Wallis, não há evidência estatística suficiente para rejeitar a  $H_0$ , indicando que não existem diferenças estatisticamente significativas na distribuição da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) entre as quatro Metaheurísticas analisadas.

Tabela 24 - Teste de Kruskal-Wallis para os dados da Proximidade de Resultado Ótimo (%) por Metaheurística

Teste de Kruskal-Wallis		Posto Médio
Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%)	<i>Genetic Algorithm</i>	17.67
	<i>Particle Swarm Optimization</i>	16.70
	<i>Tabu Search</i>	10.56
	<i>Simulated Annealing</i>	13.28
Estatística de Teste	Sig	0.375

Dado que para a variável Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%), valores mais baixos indicam melhor desempenho (ou seja, resultados mais próximos do ótimo), menores valores de posto médio no teste de Kruskal-Wallis correspondem a um melhor desempenho médio por parte da Metaheurística. Embora o TS apresente o menor posto médio e, conseqüentemente, o melhor desempenho médio, seguido pelo SA, PSO e, por último, o GA, não é possível concluir, com base na evidência

estatística, que alguma das quatro Metaheurísticas obtenha resultados mais próximos do ótimo em relação às restantes.

#### 4.2.3. Análise da Variabilidade de Resultados dentro de cada Metaheurística

Esta análise tem como papel avaliar a consistência dos resultados obtidos por cada Metaheurística e concluir qual das quatro é a mais consistente. Como tal, para esta análise as variáveis de foco foram novamente a Variação do *Makespan* (%) e Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%). Para cada grupo de Metaheurísticas quantificou-se a dispersão dos valores das duas variáveis mencionadas, comparando-se a dispersão entre os grupos. As Metaheurísticas que obtiverem menor variabilidade são, portanto, as mais consistentes mesmo que não apresentem os melhores resultados em média.

Foi então necessário calcular o Coeficiente de Variação (CV) através da equação (13).

$$CV = \frac{\text{Desvio Padrão}}{|\text{Média}|} \times 100 \quad (13)$$

Primeiramente, com recurso ao SPSS, calcularam-se a média e desvio padrão das variáveis da Variação do *Makespan* (%) e Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) para cada uma das quatro Metaheurísticas, dado que eram valores necessários para o cálculo do CV. Seguidamente aplicou-se a equação (13), tendo sido calculado o CV das duas variáveis referidas para as quatro Metaheurísticas. Na Figura 27 e Figura 28 é possível observar os valores obtidos do CV da Variação do *Makespan* (%) e do CV da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) de cada uma das Metaheurísticas, respetivamente.

Para avaliar a consistência dos resultados obtidos por cada Metaheurística, calculou-se o CV que permitiu uma análise da dispersão relativa dos resultados, sendo que valores mais baixos de CV indicavam maior consistência. Os resultados atingidos, presentes na Figura 27 e Figura 28, demonstram que o GA apresenta a maior consistência de entre as quatro Metaheurísticas analisadas, devido ao facto de apresentar os menores valores de CV em ambas as variáveis. Contrariamente, o TS revelou-se como a Metaheurística menos consistente de entre as quatro, à conta de apresentar os maiores valores de CV em ambas as variáveis, indicando maior variabilidade nos resultados obtidos.

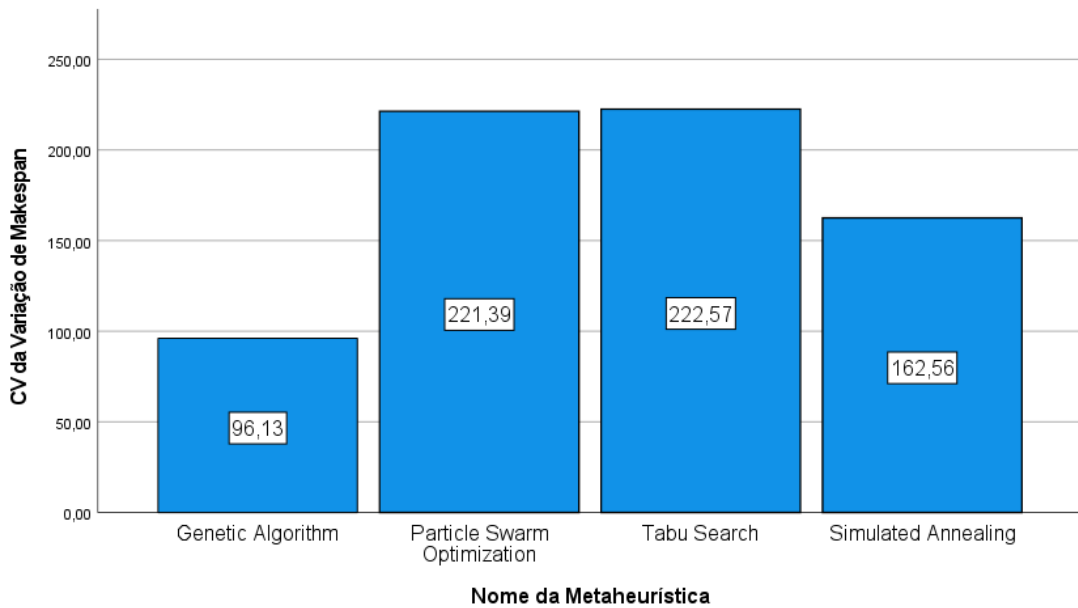


Figura 27 - CV da Variação de *Makespan* para cada Metaheurística

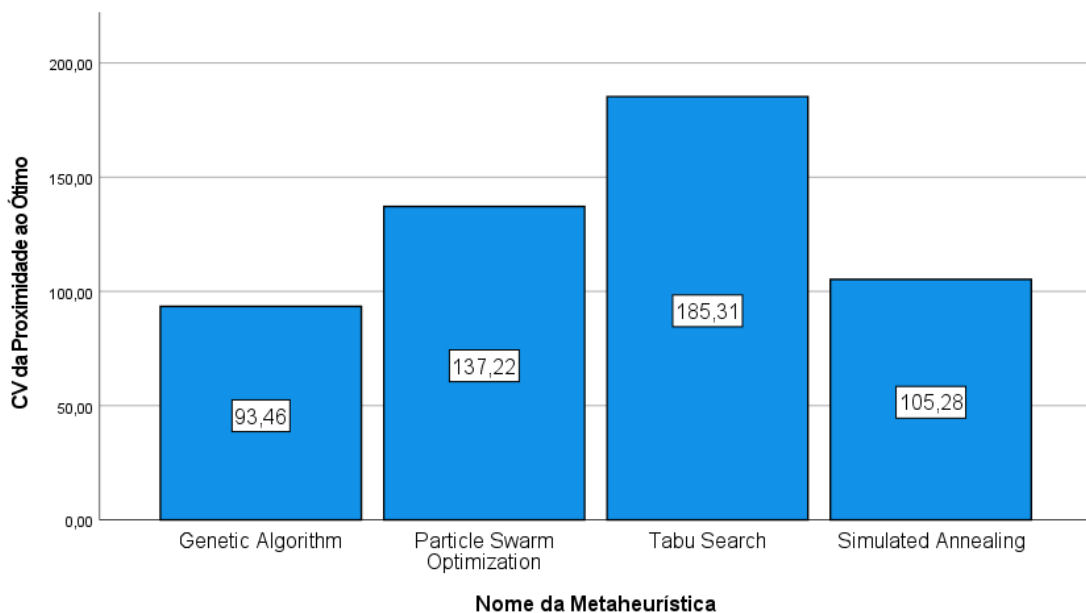


Figura 28 - CV da Proximidade ao Ótimo para cada Metaheurística

#### 4.2.4. Análise da Influência dos Métodos de Comparação de Resultados

Esta análise tem como papel avaliar se o desempenho de uma dada Metaheurística é melhor quando comparada a um determinado método de comparação. Os métodos de comparação tidos em conta foram Outras Metaheurísticas, Caso Industrial e *Dispatching Rule*. Resultado Ótimo não foi tido em consideração dado que não faria sentido pois não existe redução de *makespan* face a um resultado ótimo. Para esta análise a variável de foco foi a Variação de

*Makespan* (%). Como os métodos de comparação só têm influência na variável de Variação de *Makespan* (%), a variável Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) não consta desta análise.

Tal como na segunda análise estatística realizada, de forma a se realizar o teste estatístico indicado foi necessário verificar dois pressupostos: a existência de normalidade da variável dependente para cada método de comparação e a existência de homogeneidade das variâncias entre os métodos de comparação. Realizou-se primeiro o teste de Shapiro-Wilk para verificar a normalidade por método de comparação. As hipóteses estabelecidas para este teste foram:

- $H_0$ : A variável da Variação do *Makespan* (%) segue uma distribuição normal;
- $H_1$ : A variável da Variação do *Makespan* (%) não segue uma distribuição normal.

Na Tabela 25 pode-se observar que um dos três valores de Sig < 0.05 para Shapiro-Wilk, logo rejeita-se a  $H_0$ , pelo que não existe normalidade nos grupos de dados.

Tabela 25 - Teste de Shapiro-Wilk para os dados da Variação de *Makespan* (%) por Método de Comparação

Teste de Shapiro-Wilk		Sig
Variação do <i>Makespan</i> (%)	Outras Metaheurísticas	< 0.001
	Caso Industrial	0.171
	<i>Dispatching Rule</i>	0.896

De seguida foi necessário verificar a homogeneidade de variâncias entre os métodos de comparação, através do teste de Levene. As hipóteses estabelecidas para este teste foram:

- $H_0$ : As variâncias da variável da Variação do *Makespan* (%) são iguais entre os métodos de comparação;
- $H_1$ : Pelo menos um dos métodos de comparação apresenta uma variância diferente.

Na Tabela 26 pode-se observar que os quatros valores de Sig > 0.05, logo não há evidência estatística suficiente para rejeitar a  $H_0$ , indicando a existência de homogeneidade de variâncias entre os métodos de comparação.

Tabela 26 - Teste de Levene para os dados da Variação de *Makespan* (%) por Método de Comparação

Teste de Levene		Sig
Variação do <i>Makespan</i> (%)	Com base em média	0.266
	Com base em mediana	0.416
	Com base em mediana e com gl ajustado	0.416
	Com base em média aparada	0.306

Apesar do grupo de dados da Variação de *Makespan* (%), relativo aos métodos de comparação, apresentar homogeneidade, não apresentava normalidade. Devido a isto, realizou-se o teste não paramétrico de Kruskal-Wallis. As hipóteses estabelecidas para este teste foram:

- $H_0$ : As distribuições da variável da Variação de *Makespan* (%) são iguais entre os métodos de comparação;
- $H_1$ : Pelo menos uma das distribuições difere entre os métodos de comparação.

Na Tabela 27 pode-se observar os valores obtidos do teste. Dado que  $Sig < 0.05$ , para o teste de Kruskal-Wallis, rejeita-se a  $H_0$ , indicando que existem diferenças estatisticamente significativas na distribuição da Variação de *Makespan* (%) entre os métodos de comparação analisados.

Tabela 27 - Teste de Kruskal-Wallis para os dados da Variação de *Makespan* (%) por Método de Comparação

Teste de Kruskal-Wallis		Posto Médio
Variação do <i>Makespan</i> (%)	Outras Metaheurísticas	79.86
	Caso Industrial	27.63
	<i>Dispatching Rule</i>	22.20
Estatística de Teste	Sig	< 0.001

Dado que para a variável Variação do *Makespan* (%), valores mais negativos indicam melhor desempenho (ou seja, maiores reduções de *makespan*), menores valores de posto médio no teste de Kruskal-Wallis correspondem a um melhor desempenho médio por parte do método de comparação. Assim, a comparação à *Dispatching Rule* apresenta o melhor desempenho médio, seguido pelo Caso Industrial e, por último, as Outras Metaheurísticas que apresenta o pior desempenho médio. Estes resultados indicam que, em média, a comparação à *Dispatching Rule* proporcionou as maiores reduções no *makespan*, ao passo que a comparação a Outras Metaheurísticas proporcionou as menores reduções no *makespan*. Contudo, é importante ter em consideração que tanto o grupo de *Dispatching Rule* como o de Caso Industrial têm uma amostra pequena (5 e 16 respetivamente), quando comparado à das Outras Metaheurísticas (122), o que limita a robustez estatística destas conclusões.

#### 4.2.5. Análise por tipo de Metaheurística: Híbrida e Não Híbrida

Esta última análise tem como papel avaliar o desempenho de uma Metaheurística de acordo com o seu tipo, ou seja, avaliar de entre as Metaheurísticas Híbridas e Não Híbridas qual das duas tende a apresentar os melhores resultados. Como tal, para esta análise as variáveis de foco foram a Variação do *Makespan* (%) e Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%), tendo sido realizado o teste estatístico não paramétrico de Mann-Whitney.

Antes de se chegar à decisão de utilizar este teste foi necessário primeiro verificar os mesmos dois pressupostos: a existência de normalidade da variável dependente para cada tipo de

Metaheurística e a existência de homogeneidade das variâncias entre os dois tipos de Metaheurísticas.

Começando pela variável dependente da Variação do *Makespan* (%), realizou-se primeiro o teste de Shapiro-Wilk para verificar a normalidade por tipo de Metaheurística. As hipóteses estabelecidas para este teste foram:

- $H_0$ : A variável da Variação do *Makespan* (%) segue uma distribuição normal;
- $H_1$ : A variável da Variação do *Makespan* (%) não segue uma distribuição normal.

Na Tabela 28 pode-se observar que os dois valores de Sig < 0.05 para Shapiro-Wilk, logo rejeita-se a  $H_0$ , pelo que não existe normalidade nos grupos de dados.

Tabela 28 - Teste de Shapiro-Wilk para os dados da Variação de *Makespan* (%) por Tipo de Metaheurística

Teste de Shapiro-Wilk		Sig
Variação do <i>Makespan</i> (%)	Híbrida	< 0.001
	Não Híbrida	< 0.001

Seguidamente foi necessário verificar a homogeneidade de variâncias entre os tipos de Metaheurísticas, através do teste de Levene. As hipóteses estabelecidas para este teste foram:

- $H_0$ : As variâncias da variável da Variação do *Makespan* (%) são iguais entre os tipos de Metaheurísticas;
- $H_1$ : Pelo menos um dos tipos de Metaheurísticas apresenta uma variância diferente.

Na Tabela 29 pode-se observar que os quatros valores de Sig > 0.05, logo não há evidência estatística suficiente para rejeitar a  $H_0$ , indicando a existência de homogeneidade de variâncias entre os tipos de Metaheurísticas.

Tabela 29 - Teste de Levene para os dados da Variação de *Makespan* (%) por Tipo de Metaheurística

Teste de Levene		Sig
Variação do <i>Makespan</i> (%)	Com base em média	0.285
	Com base em mediana	0.177
	Com base em mediana e com gl ajustado	0.177
	Com base em média aparada	0.204

Apesar do grupo de dados da Variação do *Makespan* (%), relativo aos tipos de Metaheurísticas, apresentar homogeneidade, não apresentava normalidade. Devido a isto realizou-se o teste não paramétrico de Mann-Whitney. As hipóteses estabelecidas para este teste foram:

- $H_0$ : A distribuição da variável da Variação do *Makespan* (%) é igual entre os dois tipos de Metaheurísticas;
- $H_1$ : A distribuição da variável da Variação do *Makespan* (%) é diferente entre os dois tipos de Metaheurísticas.

Na Tabela 30 pode-se observar os valores obtidos do teste. Dado que  $Sig > 0.05$ , para o teste de Mann-Whitney, não há evidência estatística suficiente para rejeitar a  $H_0$ , indicando que não existem diferenças estatisticamente significativas na distribuição da Variação do *Makespan* (%) entre os dois tipos de Metaheurísticas analisadas.

Tabela 30 - Teste de Mann-Whitney para os dados da Variação de *Makespan* (%) por Tipo de Metaheurística

Teste de Mann-Whitney		Posto Médio
Variação do <i>Makespan</i> (%)	Híbrida	73.50
	Não Híbrida	72.40
Estatística de Teste	Sig	0.875

O teste de Mann-Whitney revelou que não existem diferenças estatisticamente significativas entre o desempenho das Metaheurísticas Híbridas e Não Híbridas, relativamente à variável Variação do *Makespan* (%). Apesar de as Metaheurísticas Não Híbridas apresentarem um posto médio ligeiramente inferior, o que indica maiores reduções de *makespan*, essa diferença não é significativa. Conclui-se então, com base na amostra analisada, que o tipo de Metaheurística não teve um impacto estatisticamente significativo na Variação do *Makespan* (%).

Seguidamente foi necessário realizar o mesmo procedimento mas para a variável Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%). Realizou-se primeiro o teste de Shapiro-Wilk para verificar a normalidade por tipo de Metaheurística. As hipóteses estabelecidas para este teste foram:

- $H_0$ : A variável da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) segue uma distribuição normal;
- $H_1$ : A variável da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) não segue uma distribuição normal.

Na Tabela 31 pode-se observar que os dois valores de  $Sig < 0.05$  para Shapiro-Wilk, logo rejeita-se a  $H_0$ , pelo que não existe normalidade nos grupos de dados.

Tabela 31 - Teste de Shapiro-Wilk para os dados da Proximidade de Resultado Ótimo (%) por Tipo de Metaheurística

Teste de Shapiro-Wilk		Sig
Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%)	Híbrida	< 0.001
	Não Híbrida	0.009

De seguida foi necessário verificar a homogeneidade de variâncias entre os tipos de Metaheurísticas, através do teste de Levene. As hipóteses estabelecidas para este teste foram:

- $H_0$ : As variâncias da variável da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) são iguais entre os tipos de Metaheurísticas;
- $H_1$ : Pelo menos um dos tipos de Metaheurísticas apresenta uma variância diferente.

Na Tabela 32 pode-se observar que os quatros valores de Sig > 0.05, logo não há evidência estatística suficiente para rejeitar a  $H_0$ , indicando a existência de homogeneidade de variâncias entre os tipos de Metaheurísticas.

Tabela 32 - Teste de Levene para os dados da Proximidade de Resultado Ótimo (%) por Tipo de Metaheurística

Teste de Levene		Sig
Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%)	Com base em média	0.350
	Com base em mediana	0.544
	Com base em mediana e com gl ajustado	0.547
	Com base em média aparada	0.458

Apesar do grupo de dados da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%), relativo aos tipos de Metaheurísticas, apresentar homogeneidade, não apresentava normalidade. Devido a isto realizou-se o teste não paramétrico de Mann-Whitney. As hipóteses estabelecidas para este teste foram:

- $H_0$ : A distribuição da variável da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) é igual entre os dois tipos de Metaheurísticas;
- $H_1$ : A distribuição da variável da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) é diferente entre os dois tipos de Metaheurísticas.

Na Tabela 33 pode-se observar os valores obtidos do teste. Dado que  $Sig > 0.05$ , para o teste de Mann-Whitney, não há evidência estatística suficiente para rejeitar a  $H_0$ , indicando que não existem diferenças estatisticamente significativas na distribuição da Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%) entre os dois tipos de Metaheurísticas analisadas.

Tabela 33 - Teste de Mann-Whitney para os dados da Proximidade de Resultado Ótimo (%) por Tipo de Metaheurística

Teste de Mann-Whitney		Posto Médio
Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%)	Híbrida	12.89
	Não Híbrida	14.21
Estatística de Teste	Sig	0.662

O teste de Mann-Whitney revelou igualmente que não existem diferenças estatisticamente significativas entre o desempenho das Metaheurísticas Híbridas e Não Híbridas, relativamente à variável Proximidade do Resultado proveniente da Metaheurística relativamente ao Resultado Ótimo (%). Apesar de as Metaheurísticas Híbridas apresentarem um posto médio ligeiramente inferior, o que indica mais resultados próximos do ótimo, essa diferença não é significativa. Conclui-se então, com base na amostra analisada, que o tipo de Metaheurística não teve um impacto estatisticamente significativo na Proximidade ao Resultado Ótimo.

#### 4.2.6. Conclusões das Análises Estatísticas

As análises estatísticas realizadas ao longo deste trabalho permitiram identificar e compreender diversos fatores que influenciam significativamente o desempenho das Metaheurísticas aplicadas a Problemas de Escalonamento Industrial do tipo Job-Shop. Estas análises não só evidenciaram a existência de diferenças significativas entre os desempenhos obtidos por diferentes Metaheurísticas, como também possibilitaram uma reflexão mais abrangente sobre o impacto de variáveis contextuais, como o ano de publicação dos artigos selecionados, o método de comparação utilizado relativamente à Metaheurística do dado artigo e até o tipo da Metaheurística. Para além disto, ainda foi possível aferir a consistência dos resultados produzidos por cada Metaheurística.

A análise de evolução dos resultados ao longo dos anos revelou a existência de correlação significativa entre o ano de publicação do artigo e a variação de *makespan*, indicando uma tendência de melhoria nos desempenhos reportados ao longo do tempo. No entanto, essa mesma tendência não se verificou relativamente à proximidade ao resultado ótimo, para a qual não se encontrou qualquer correlação estatisticamente significativa. Esta aparente contradição pode dever-se a diversos fatores. Em primeiro lugar, é importante destacar que muitos dos artigos analisados não apresentam resultados ótimos, o que limita a precisão e robustez das análises relacionadas com esta variável. Contudo, esta limitação pode não ser a única

responsável pela ausência de correlação. Outra explicação plausível reside na forma como os investigadores priorizam os objetivos das suas publicações. É comum que os autores concentrem os seus esforços na otimização e parametrização das suas próprias Metaheurísticas, de forma a demonstrar melhorias de desempenho face ao estado da arte, deixando em segundo plano uma comparação rigorosa com os resultados ótimos. Esta abordagem pode levar a melhorias aparentes no desempenho geral das Metaheurísticas (como evidenciado pela redução de *makespan*), mas sem necessariamente traduzir-se numa aproximação consistente aos resultados ótimos. Para além disto, ainda é possível que parte da evolução observada na variação de *makespan* esteja relacionada com alterações metodológicas dos artigos – por exemplo, a escolha de instâncias menos exigentes ou a comparação com algoritmos mais simples – o que pode artificialmente levar a maiores reduções de *makespan* ao longo dos anos. Por último, a ausência de padronização nas instâncias e *benchmarks* utilizados também pode ser um fator responsável. Apesar de se utilizar o problema de Job-Shop como referência, os artigos nem sempre utilizam as mesmas instâncias ou *benchmarks*. Isto significa que, ao longo do tempo, os artigos podem estar a resolver problemas diferentes com diferentes níveis de dificuldade, tornando a evolução da proximidade ao ótimo inconsistente. Por outro lado, como a variação de *makespan* é muitas vezes feita relativamente a um algoritmo base do próprio artigo, ela acaba por refletir melhorias internas, mais do que um avanço objetivo em relação ao ótimo. A análise individual por Metaheurística reforçou esta complexidade, uma vez que se observaram comportamentos distintos entre os algoritmos, demonstrando que a evolução temporal do desempenho não é linear nem uniforme, mas que depende fortemente do contexto específico e das abordagens metodológicas adotadas. Assim, ainda que possa parecer que as Metaheurísticas, com o passar do tempo, não se tenham aproximado do resultado ótimo, é importante ter em consideração estes fatores que foram mencionados antes de se tirar uma conclusão definitiva.

A análise de comparação de desempenho entre as Metaheurísticas revelou diferenças estatisticamente significativas na variação de *makespan*, mas não na proximidade ao resultado ótimo. Este resultado sugere que, embora algumas Metaheurísticas consigam reduzir substancialmente os valores de *makespan* em relação a um algoritmo de referência, essas melhorias nem sempre implicam uma aproximação mais consistente ao resultado ótimo. Esta divergência pode estar relacionada com a natureza relativa da variável da variação de *makespan*, que depende fortemente do algoritmo de comparação utilizado em cada artigo. Além disso, é ainda possível que, em alguns artigos, os valores de referência utilizados sejam modestos, o que pode inflacionar artificialmente as reduções observadas. Já a variável da proximidade ao resultado ótimo, por depender diretamente da apresentação de resultados ótimos, tende a ser mais exigente e menos sujeita a flutuações metodológicas, o que pode explicar a ausência de diferenças estatisticamente significativas entre as Metaheurísticas nesse aspeto. Assim, esta análise demonstrou que o GA apresenta, em média, as maiores reduções de *makespan*, o que poderá estar parcialmente relacionado com o facto de concentrar um maior número de artigos mais recentes, enquanto o TS apresenta as menores reduções. Para a proximidade ao ótimo não é possível retirar conclusões quanto ao desempenho de cada Metaheurística devido à falta de evidência estatística. No que diz respeito à análise da influência dos métodos de comparação de resultados, verificaram-se diferenças estatisticamente

significativas na variação de *makespan*. Os artigos que comparavam os seus resultados com *Dispatching Rules* ou com casos industriais apresentavam maiores reduções de *makespan*, enquanto que os artigos que realizavam comparações com outras Metaheurísticas apresentavam, em média, reduções mais modestas. Esta tendência é compreensível, já que *Dispatching Rules* ou métodos industriais tradicionais costumam apresentar desempenhos inferiores aos das Metaheurísticas mais recentes, o que amplia as margens de melhoria observadas. Por outro lado, quando a comparação é feita com outras Metaheurísticas, o nível de desempenho de referência é geralmente mais elevado, tornando-se mais difícil atingir reduções de *makespan* maiores. Este fator evidencia a importância crítica da escolha do método de comparação, uma vez que influencia diretamente a percepção de eficácia da Metaheurística proposta. Em relação à análise por tipo de Metaheurística (Híbrida e Não Híbrida) não se observaram diferenças estatisticamente significativas nem na variação de *makespan* nem na proximidade ao resultado ótimo. Uma das explicações possíveis para este resultado é que as Metaheurísticas Não Híbridas, apesar de conceptualmente mais simples, continuam a ser bastante eficazes na resolução de Problemas de Job-Shop, beneficiando de décadas de refinamento e aplicação prática. Por outro lado, embora as Metaheurísticas Híbridas combinem técnicas distintas e tenham, em teoria, maior potencial de exploração e intensificação, é possível que, na prática, não estejam a ser exploradas na sua plenitude. A complexidade da sua configuração, os desafios na integração eficiente dos seus componentes e a ausência de padrões consolidados de implementação podem limitar a obtenção de ganhos significativos. Adicionalmente, muitos artigos que propõem Metaheurísticas Híbridas não realizam comparações diretas com versões Não Híbridas equivalentes, o que dificulta a avaliação do verdadeiro impacto da hibridização. Também é possível que as instâncias utilizadas nos artigos não sejam suficientemente complexas para evidenciar vantagens significativas das Metaheurísticas Híbridas, o que contribui para resultados estatisticamente equivalentes. Em suma, estes achados sugerem que a eficácia das Metaheurísticas (Híbridas ou Não Híbridas) depende tanto do seu desenho técnico como do contexto em que são aplicadas e da forma como são avaliadas nos artigos.

Por último, a análise da variabilidade de resultados dentro de cada Metaheurística revelou variações consideráveis entre elas, sugerindo que, para além do desempenho médio, a consistência que cada uma das Metaheurísticas oferece também deve ser considerada como um critério relevante na avaliação de Metaheurísticas. A Metaheurística que apresentou os resultados mais consistentes foi o GA, tanto em termos de variação de *makespan* como de proximidade ao ótimo. Uma possível explicação para esta estabilidade está no facto de o GA ser a Metaheurística com o maior número de artigos recentes na amostra analisada. Esta concentração temporal pode ter levado a uma maior uniformidade metodológica, em termos de instâncias, parametrizações e critérios de avaliação, o que, por sua vez, reduziu a dispersão dos resultados. Além disso, o GA é uma das Metaheurísticas mais utilizadas em problemas de Job-Shop, o que pode ter levado ao desenvolvimento de boas práticas consolidadas na sua implementação e afinação, contribuindo desta forma para um desempenho mais previsível e estável nos artigos. A familiaridade da comunidade científica com esta Metaheurística também pode ter facilitado a sua aplicação mais cuidadosa, minimizando variações indesejadas nos resultados. Contrariamente, o TS foi a Metaheurística que apresentou a maior variabilidade de

resultados, tanto na variação de *makespan* como na proximidade ao ótimo. Apesar de não ter sido possível identificar uma causa específica para esta inconsistência, é relevante notar que o TS, tal como o PSO, está associado a um maior número de artigos mais antigos na amostra analisada. Este fator pode indicar uma maior diversidade de abordagens experimentais, diferentes estágios de maturidade na sua aplicação e menos uniformidade metodológica ao longo do tempo, o que contribui para um aumento da dispersão dos resultados. É também possível que o TS, sendo mais sensível à escolha de parâmetros, esteja mais sujeito a variações de resultado caso não seja cuidadosamente ajustado. Os resultados desta análise reforçam a importância de avaliar não só o desempenho médio, mas também a estabilidade, consistência e previsibilidade das Metaheurísticas, sobretudo quando se pretende aplicar estes algoritmos em contextos industriais, onde a robustez dos resultados assume um papel fundamental.

Em suma, os resultados quantitativos provenientes das análises estatísticas realizadas demonstram claramente que a avaliação do desempenho de Metaheurísticas no contexto de Problemas de Escalonamento Industrial do tipo Job-Shop exige uma abordagem multifatorial, abrangente e criteriosa. Fatores como a evolução temporal das Metaheurísticas, o seu tipo, o método de comparação adotado e a consistência dos resultados revelaram-se determinantes para uma interpretação rigorosa dos avanços nesta área. Neste sentido, este trabalho reforça a importância de se adotarem metodologias de análise robustas e sistemáticas na seleção e comparação de Metaheurísticas, especialmente em contextos industriais, onde a eficácia, previsibilidade e estabilidade das Metaheurísticas são essenciais para uma tomada de decisão informada. Assim, conclui-se que uma compreensão aprofundada e integrada dos diversos fatores que influenciam o desempenho das Metaheurísticas é indispensável para maximizar o seu potencial de aplicação em problemas reais.

## 5. Conclusão

Este capítulo 5 tem como papel apresentar as conclusões alcançadas ao longo da realização da dissertação. Encontra-se dividido em dois subcapítulos: Conclusões Finais e Limitações e Trabalhos Futuros. No primeiro subcapítulo, destacam-se as principais conclusões que se retiram do trabalho realizado, bem como os passos que se tomaram para a sua realização. No segundo subcapítulo, falam-se das limitações existentes durante o desenvolvimento do trabalho e, ainda, brevemente de trabalhos futuros.

### 5.1. Conclusões Finais

O trabalho desenvolvido neste relatório apresenta como foco a área industrial, mais especificamente nos complexos problemas existentes nesta área e, de que forma, se podem resolver estes de forma eficaz e eficiente, através das Metaheurísticas. Este tema é deveras importante porque, atualmente, as indústrias são cada vez mais competitivas, sendo necessário estas adotarem Metaheurísticas de forma a manterem os seus níveis de competitividade.

Assim, partiu-se inicialmente do Escalonamento das Operações, o que é, típicos problemas existentes e as suas formas de os representar, e de seguida para as Metaheurísticas, o que são, tipos diferentes de Metaheurísticas existentes e de que forma estas funcionam e contribuem para o benefício das indústrias. A revisão bibliográfica finaliza com a realização de uma análise bibliométrica para avaliar a relevância do tema do trabalho e com a análise de alguns casos de estudo recentes considerados pertinentes, de forma a se perceber o que tem sido realizado nos últimos anos relativamente ao tema do projeto.

As conclusões principais retiradas desta revisão bibliográfica são que as Metaheurísticas são cada vez mais cruciais no Escalonamento das Operações, ajudando na resolução de diversos problemas de otimização complexos, que de outra forma seriam impossíveis de se resolver de forma eficiente. Também se concluiu que é difícil categorizar Metaheurísticas específicas para dados problemas de escalonamento, pois apesar de algumas apresentarem resultados mais favoráveis em torno de problemas em particular, nem sempre acabam por ser a melhor opção, pelo que esta decisão deve advir da prática, conhecimento e experiência de pessoas dedicadas e não só do que é apresentado na literatura. Por último, ainda se pôde concluir que as Metaheurísticas não tendem a ser usadas sozinhas na resolução dos problemas de escalonamento, sendo muitas vezes utilizadas múltiplas simultaneamente ou até Metaheurísticas hibridizadas.

## Conclusão

Seguidamente, procedeu-se à seleção dos artigos científicos mais relevantes para este trabalho, tendo sido realizada uma pesquisa por artigos onde Problemas de Escalonamento Industrial do tipo Job-Shop com *makespan* como critério de otimização eram resolvidos com recurso a Metaheurísticas. Os problemas do tipo Job-Shop foram selecionados devido à sua complexidade e popularidade na comunidade científica. Como tal, o GA, PSO, TS e SA foram escolhidos como as quatro Metaheurísticas em foco. Os dados provenientes de cada artigo selecionado foram devidamente organizados em tabelas, de acordo com a Metaheurística, divididas por categorias como Metaheurística utilizada, parametrização e resultado da Metaheurística.

As principais conclusões retiradas foram que o GA demonstrava a maior utilização na resolução de problemas de Job-Shop, enquanto que o SA demonstrava a menor utilização, com base no número de artigos existentes para cada uma das Metaheurísticas na base de dados do *Web of Science*. Também se concluiu que as Metaheurísticas demonstravam a maior capacidade na resolução deste tipo de problemas, pelo que quando estas eram comparadas umas às outras as reduções de *makespan* eram modestas. Contrariamente quando estas eram comparadas a um caso industrial ou *Dispatching Rule* as reduções de *makespan* já eram notórias, apesar da existência de algumas exceções. Para além disto, ainda se evidenciou que as Metaheurísticas com maior número de artigos antigos, como o PSO e TS, apresentavam os piores resultados, indicando à partida a evolução de desempenho das Metaheurísticas ao longo do tempo.

Por último, procedeu-se à exposição e organização dos dados relativos aos artigos científicos selecionados, com o objetivo de compreender a sua distribuição por diversas categorias. Posteriormente, realizaram-se testes estatísticos que permitiram atingir várias conclusões relativamente ao desempenho das quatro Metaheurísticas em foco, recorrendo-se, para tal, ao *software* estatístico IBM SPSS *Statistics*.

As análises estatísticas realizadas permitiram obter uma perspetiva aprofundada sobre a aplicação de Metaheurísticas na resolução de problemas de Job-Shop. Através da análise de evolução de resultados ao longo dos anos, verificou-se que os progressos tecnológicos e metodológicos não se refletem de forma uniforme nas métricas de desempenho, levantando questões quanto à consistência e qualidade das abordagens adotadas nos artigos científicos.

A análise de comparação de desempenho entre as Metaheurísticas evidenciou diferenças significativas entre os diferentes algoritmos, confirmando que a seleção da Metaheurística pode influenciar os resultados obtidos. Esta conclusão reforça então a necessidade de avaliar cuidadosamente qual a Metaheurística mais apropriada para o dado problema.

Através da análise de influência dos métodos de comparação de resultados, também se verificou que o método de comparação influencia os resultados registados, sugerindo que o contexto experimental escolhido nos artigos pode afetar substancialmente a perceção da eficácia das Metaheurísticas.

No que diz respeito à análise por tipo de Metaheurística (Híbrida e Não Híbrida), não foram observadas diferenças estatisticamente significativas nos resultados. Esta ausência de diferença levantou duas hipóteses relevantes: ou as Metaheurísticas Não Híbridas continuam a ser

suficientemente competitivas, ou as Híbridas ainda não estão a ser exploradas no seu potencial máximo.

Por fim, a análise da variabilidade de resultados dentro de cada Metaheurística veio salienta que, para além da média de desempenho, é igualmente essencial considerar a consistência de cada Metaheurística. Algumas Metaheurísticas demonstraram ser mais robustas e previsíveis, o que representa uma vantagem em contextos industriais, onde a variabilidade de resultados é um fator crítico.

Em suma, estas análises estatísticas revelaram que o desempenho das Metaheurísticas é dependente de múltiplos fatores e que a sua avaliação não deve ser feita de forma isolada, mas sim com base numa abordagem abrangente, que considere as múltiplas características essenciais à definição e ao destaque de cada Metaheurística.

## **5.2. Limitações e Trabalhos Futuros**

Relativamente às limitações sentidas durante a realização deste trabalho, estas advêm principalmente do tempo limitado disponível para o seu desenvolvimento. Devido a isto, o trabalho não pode englobar um maior número de Metaheurísticas, bem como alargar o estudo a outros tipos de Problemas de Escalonamento Industrial, o que teria permitido uma análise mais abrangente. Para além disto, o facto de muitos dos artigos científicos selecionados não apresentarem resultado ótimo limitou a robustez da análise estatística realizada.

Finalmente, em relação aos trabalhos futuros, estes estão diretamente relacionados com as limitações anteriormente referidas. Seria relevante expandir o estudo a outras Metaheurísticas e a diferentes tipos de Problemas de Escalonamento Industrial que não foram incluídos nesta dissertação, especialmente aqueles considerados mais relevantes e importantes. Adicionalmente, também se poderia estudar e selecionar mais artigos relativos às Metaheurísticas e Problema de Escalonamento escolhidos para a dissertação, de forma a enriquecer as análises estatísticas realizadas e, porventura, atingir novas conclusões.

## Conclusão

## Referências

- [1] G. P. Georgiadis, A. P. Elekidis, e M. C. Georgiadis, «Optimization-Based Scheduling for the Process Industries: From Theory to Real-Life Industrial Applications», *Processes*, vol. 7, n. 7, Jul. 2019, doi: 10.3390/pr7070438.
- [2] J. Torres-Jiménez e J. Pavón, «Applications of metaheuristics in real-life problems», *Progress in Artificial Intelligence*, vol. 2, n. 4, pp. 175–176, Abr. 2014, doi: 10.1007/s13748-014-0051-8.
- [3] S. C. Graves, «A Review of Production Scheduling», <https://doi.org/10.1287/opre.29.4.646>, vol. 29, n. 4, pp. 646–675, Ago. 1981, doi: 10.1287/OPRE.29.4.646.
- [4] K. R. Baker, *Introduction to sequencing and scheduling*. New York: John Wiley & Sons, 1974.
- [5] J. Y. T. Leung, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. New York: Chapman & Hall/CRC, 2004.
- [6] A. Santos, «Auto-parametrização de meta-heurísticas para problemas de escalonamento em ambiente industrial», Universidade do Minho, Guimarães, 2020.
- [7] J. Koskinen, C. Raduly-Baka, M. Johnsson, e O. S. Nevalainen, «Rolling horizon production scheduling of multi-model PCBs for several assembly lines», *Int J Prod Res*, vol. 58, n. 4, pp. 1052–1073, Fev. 2020, doi: 10.1080/00207543.2019.1609708.
- [8] D. Ouelhadj e S. Petrovic, «A survey of dynamic scheduling in manufacturing systems», Agosto de 2009. doi: 10.1007/s10951-008-0090-8.
- [9] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 3rd ed. New York: Springer, 2008.
- [10] J. Błażewicz, K. Ecker, E. Pesch, G. Schmidt, e J. Węglarz, *Handbook on Scheduling: From Theory to Applications*. Springer, 2007.
- [11] K. R. Baker e Dan. Trietsch, *Principles of sequencing and scheduling*. New Jersey: John Wiley & Sons, 2009.
- [12] R. L. Graham, E. L. Lawler, J. K. Lenstra, e A. H. G. R. Kan, «Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey», *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979, doi: 10.1016/S0167-5060(08)70356-X.
- [13] R. Martí, M. Sevaux, e K. Sörensen, «Fifty years of metaheuristics», *Eur J Oper Res*, Abr. 2024, doi: 10.1016/J.EJOR.2024.04.004.
- [14] K. Sörensen, M. Sevaux, e F. Glover, «A History of Metaheuristics», em *Handbook of Heuristics*, vol. 2–2, Springer International Publishing, 2018, pp. 791–808. doi: 10.1007/978-3-319-07124-4\_4.
- [15] F. Glover, «Future Paths for Integer Programming and Links to Artificial Intelligence», *Comput Oper Res*, vol. 13, n. 5, pp. 533–549, 1986, doi: 10.1016/0305-0548(86)90048-1.

## Referências

- [16] A. Kaveh, *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, 3rd ed. Springer, 2021.
- [17] L. Bianchi, M. Dorigo, L. M. Gambardella, e W. J. Gutjahr, «A survey on metaheuristics for stochastic combinatorial optimization», *Nat Comput*, vol. 8, n. 2, pp. 239–287, 2009, doi: 10.1007/s11047-008-9098-4.
- [18] G. Polya, *How to Solve It*. Princeton University Press, 1945.
- [19] A. Auger, «Convergence results for the  $(1,\lambda)$ -SA-ES using the theory of  $\phi$ -irreducible Markov chains», *Theor Comput Sci*, vol. 334, n. 1–3, pp. 35–69, Abr. 2005, doi: 10.1016/J.TCS.2004.11.017.
- [20] C. Darwin, *The Origin of The Species*. 1859.
- [21] A. S. Wu e W. Banzhaf, «Variable-Length Representation and Noncoding Segments for Evolutionary Algorithms», 1998.
- [22] J. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.
- [23] S. Katoch, S. S. Chauhan, e V. Kumar, «A review on genetic algorithm: past, present, and future», *Multimed Tools Appl*, vol. 80, n. 5, pp. 8091–8126, 2021, doi: 10.1007/S11042-020-10139-6.
- [24] S. Kirkpatrick, C. D. Gelatt, e M. P. Vecchi, «Optimization by simulated annealing», *Science (1979)*, vol. 220, n. 4598, pp. 671–680, 1983, doi: 10.1126/SCIENCE.220.4598.671.
- [25] S. M. Almufti, «Historical survey on metaheuristics algorithms», 2019.
- [26] I. H. Osman e G. Laporte, *Metaheuristics: A bibliography*, vol. 63. Annals of Operations Research, 1996.
- [27] X. S. Yang, S. Deb, e S. Fong, «Metaheuristic algorithms: Optimal balance of intensification and diversification», *Applied Mathematics and Information Sciences*, vol. 8, n. 3, pp. 977–983, Mai. 2014, doi: 10.12785/amis/080306.
- [28] F. Glover e M. Samorani, «Intensification, Diversification and Learning in metaheuristic optimization», 1 de Outubro de 2019, *Springer New York LLC*. doi: 10.1007/s10732-019-09409-w.
- [29] C. Blum e A. Roli, «Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison Metaheuristics in Combinatorial Optimization», 2003.
- [30] B. Morales-Castañeda, D. Zaldívar, E. Cuevas, F. Fausto, e A. Rodríguez, «A better balance in metaheuristic algorithms: Does it exist?», *Swarm Evol Comput*, vol. 54, Mai. 2020, doi: 10.1016/j.swevo.2020.100671.
- [31] X.-She. Yang, *Nature-inspired optimization algorithms*. Elsevier, 2014.
- [32] E.-Ghazali. Talbi, *Metaheuristics : from design to implementation*. John Wiley & Sons, 2009.
- [33] O. Bozorg-Haddad, M. Solgi, e H. A. Loáiciga, *Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization*. John Wiley & Sons, 2017.
- [34] A. Gogna e A. Tayal, «Metaheuristics: Review and application», *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 25, n. 4, pp. 503–526, 2013, doi: 10.1080/0952813X.2013.782347.
- [35] I. Boussaïd, J. Lepagnot, e P. Siarry, «A survey on optimization metaheuristics», em *Information Sciences*, 2013, pp. 82–117. doi: 10.1016/j.ins.2013.02.041.
- [36] A. W. Mohamed, A. A. Hadi, e A. K. Mohamed, «Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm», *International Journal of Machine Learning and Cybernetics*, vol. 11, n. 7, pp. 1501–1529, 2020, doi: 10.1007/s13042-019-01053-x.

## Referências

- [37] P. Agrawal, H. F. Abutarboush, T. Ganesh, e A. W. Mohamed, «Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019)», *IEEE Access*, vol. 9, pp. 26766–26791, 2021, doi: 10.1109/ACCESS.2021.3056407.
- [38] H. Stegherr, M. Heider, e J. Hähner, «Classifying Metaheuristics: Towards a unified multi-level classification system», *Nat Comput*, vol. 21, n. 2, pp. 155–171, Jun. 2022, doi: 10.1007/s11047-020-09824-0.
- [39] T. Van Luong, N. Melab, e E. G. Talbi, «GPU computing for parallel local search metaheuristic algorithms», *IEEE Transactions on Computers*, vol. 62, n. 1, pp. 173–185, Jan. 2013, doi: 10.1109/TC.2011.206.
- [40] A. S. Santos, A. M. Madureira, e M. L. R. Varela, «The Influence of Problem Specific Neighborhood Structures in Metaheuristics Performance», *Journal of Mathematics*, vol. 2018, 2018, doi: 10.1155/2018/8072621.
- [41] K. Tamssaouet e S. Dauzère-Pérès, «A general efficient neighborhood structure framework for the job-shop and flexible job-shop scheduling problems», *Eur J Oper Res*, vol. 311, n. 2, pp. 455–471, Dez. 2023, doi: 10.1016/j.ejor.2023.05.018.
- [42] M. Den Besten e T. Stützle, «Neighborhoods Revisited: An Experimental Investigation into the Effectiveness of Variable Neighborhood Descent for Scheduling», *4th Metaheuristics International Conference*, pp. 545–549, 2001.
- [43] M. Sevkli e M. E. Aydin, «Variable Neighbourhood Search for Job Shop Scheduling Problems», *Journal of Software*, vol. 1, n. 2, pp. 34–39, 2006, doi: 10.4304/jsw.1.2.34-39.
- [44] R. K. Ahuja, O. Ergun, J. B. Orlin, e A. P. Punnen, «A survey of very large-scale neighborhood search techniques», *Discrete Appl Math (1979)*, vol. 123, n. 1–3, pp. 75–102, 2002.
- [45] H. Liu e L. Zhang, «Optimizing a Disassembly Sequence Planning with Success Rates of Disassembly Operations via a Variable Neighborhood Search Algorithm», *IEEE Access*, vol. 9, pp. 157540–157549, 2021, doi: 10.1109/ACCESS.2021.3101221.
- [46] D. Henderson, S. H. Jacobson, e A. W. Johnson, «The Theory and Practice of Simulated Annealing», em *Handbook of Metaheuristics*, Kluwer Academic Publishers, 2006, pp. 287–319. doi: 10.1007/0-306-48056-5\_10.
- [47] P. J. M. van Laarhoven e E. H. L. Aarts, *Simulated Annealing: Theory and Applications*. 1987.
- [48] X.-S. Yang, «Engineering Optimization: An Introduction with Metaheuristic Applications», John Wiley & Sons, 2010, 12, pp. 181–187.
- [49] M.-W. Park e Y.-D. Kimto, «A Systematic Procedure for Setting Parameters in Simulated Annealing Algorithms», *Computers Ops Res*, vol. 25, n. 3, pp. 207–217, 1998.
- [50] M. Andresen, H. Bräsel, M. Mörig, J. Tusch, F. Werner, e P. Willenius, «Simulated annealing and genetic algorithms for minimizing mean flow time in an open shop», *Math Comput Model*, n. 48, pp. 1279–1293, 2008, doi: 10.1016/j.ejor.2007.02.057.
- [51] D. Delahaye, S. Chaimatanan, e M. Mongeau, «Simulated annealing: From basics to applications», em *Handbook of Metaheuristics*, vol. 272, Springer, 2019, pp. 1–35. doi: 10.1007/978-3-319-91086-4\_1i.
- [52] N. Siddique e H. Adeli, «Simulated Annealing, Its Variants and Engineering Applications», *International Journal on Artificial Intelligence Tools*, vol. 25, n. 6, p. 24, 2016, doi: 10.1142/S0218213016300015.
- [53] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [54] K. De Jong, «Genetic Algorithms: A 10 Year Perspective», *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, pp. 169–177, 1985, doi: 10.4324/9781315799674.

## Referências

- [55] M. Kumar, M. Husian, N. Upreti, e D. Gupta, «Genetic Algorithm: Review and Application», *International Journal of Information Technology and Knowledge Management*, vol. 2, n. 2, pp. 451–454, 2010.
- [56] P. K. Yadav e Dr. N. L. Prajapati, «An Overview of Genetic Algorithm and Modeling», *International Journal of Scientific and Research Publications*, vol. 2, n. 9, pp. 1–4, Set. 2012.
- [57] M. Bogdanović, «On Some Basic Concepts of Genetic Algorithms as a Meta-Heuristic Method for Solving of Optimization Problems», *Journal of Software Engineering and Applications*, vol. 04, n. 08, pp. 482–486, 2011, doi: 10.4236/jsea.2011.48055.
- [58] D. I. Merino, E. N. Reyes, e C. Steidley, «Genetic Algorithms: Theory and Application», em *1998 Annual Conference*, 1998, pp. 32981–32986. doi: 10.18260/1-2--7144.
- [59] S. Katoch, S. S. Chauhan, e V. Kumar, «A review on genetic algorithm: past, present, and future», *Multimed Tools Appl*, vol. 80, n. 5, pp. 8091–8126, Fev. 2021, doi: 10.1007/s11042-020-10139-6.
- [60] A. M. Fathollahi-Fard, L. Woodward, e O. Akhrif, «A distributed permutation flow-shop considering sustainability criteria and real-time scheduling», *J Ind Inf Integr*, vol. 39, pp. 1–27, Mai. 2024, doi: 10.1016/j.jii.2024.100598.
- [61] L. R. de Abreu e B. de A. Prata, «A genetic algorithm with neighborhood search procedures for unrelated parallel machine scheduling problem with sequence-dependent setup times», *Journal of Modelling in Management*, vol. 15, n. 3, pp. 809–828, Ago. 2020, doi: 10.1108/JM2-12-2018-0209.
- [62] M. K. Marichelvam, M. Geetha, e Ö. Tosun, «An improved particle swarm optimization algorithm to solve hybrid flowshop scheduling problems with the effect of human factors – A case study», *Comput Oper Res*, vol. 114, pp. 1–9, Fev. 2020, doi: 10.1016/j.cor.2019.104812.
- [63] A. Stanković, G. Petrović, Ž. Čojbašić, e D. Marković, «An application of metaheuristic optimization algorithms for solving the flexible job-shop scheduling problem», *Operational Research in Engineering Sciences: Theory and Applications*, vol. 3, n. 3, pp. 13–28, Dez. 2020, doi: 10.31181/oresta20303013s.
- [64] J. Sacramento, J. Pedrosa, N. Chibeles-Martins, e T. Pinto-Varela, «Comparison of two Meta-Heuristics for the Bi-Objective Flexible Job Shop Scheduling Problem with Sequence Dependent Setup Times», *30th European Symposium on Computer Aided Process Engineering*, vol. 48, pp. 181–186, 2020.
- [65] F. Abderrabi *et al.*, «Flexible job shop scheduling problem with sequence dependent setup time and job splitting: Hospital catering case study», *Applied Sciences (Switzerland)*, vol. 11, n. 4, pp. 1–27, Fev. 2021, doi: 10.3390/app11041504.
- [66] L. Xue, S. Zhao, A. Mahmoudi, e M. R. Feylizadeh, «Flexible job-shop scheduling problem with parallel batch machines based on an enhanced multi-population genetic algorithm», *Complex and Intelligent Systems*, vol. 10, n. 3, pp. 4083–4101, Jun. 2024, doi: 10.1007/s40747-024-01374-7.
- [67] O. Nejjarou, S. Aqil, e M. Lahby, «Hybrid meta-heuristic solving no-wait flow shop scheduling minimizing maximum tardiness», *Evol Intell*, vol. 17, pp. 3935–3959, Jul. 2024, doi: 10.1007/s12065-024-00965-0.
- [68] K. Ben Ali, A. J. Telmoudi, e S. Gattoufi, «Improved Genetic Algorithm Approach Based on New Virtual Crossover Operators for Dynamic Job Shop Scheduling», *IEEE Access*, vol. 8, pp. 213318–213329, Dez. 2020, doi: 10.1109/ACCESS.2020.3040345.
- [69] F. Burcin Ozsoydan e M. Sağır, «Iterated greedy algorithms enhanced by hyper-heuristic based learning for hybrid flexible flowshop scheduling problem with sequence dependent setup times: A case study at a manufacturing plant», *Comput Oper Res*, vol. 125, pp. 1–15, Jan. 2021, doi: 10.1016/j.cor.2020.105044.

## Referências

- [70] C. Schumacher, P. Buchholz, K. Fiedler, e N. Gorecki, «Local Search and Tabu Search Algorithms for Machine Scheduling of a Hybrid Flow Shop Under Uncertainty», em *Proceedings of the 2020 Winter Simulation Conference*, Dortmund: IEEE Press, 2020, pp. 1456–1467.
- [71] S. Aqil e K. Allali, «Local search metaheuristic for solving hybrid flow shop problem in slabs and beams manufacturing», *Expert Syst Appl*, vol. 162, pp. 1–17, Dez. 2020, doi: 10.1016/j.eswa.2020.113716.
- [72] N. Klement, M. A. Abdeljaouad, L. Porto, e C. Silva, «Lot-sizing and scheduling for the plastic injection molding industry—A hybrid optimization approach», *Applied Sciences (Switzerland)*, vol. 11, n. 3, pp. 1–13, Fev. 2021, doi: 10.3390/app11031202.
- [73] A. Branda, D. Castellano, G. Guizzi, e V. Popolo, «Metaheuristics for the flow shop scheduling problem with maintenance activities integrated», *Comput Ind Eng*, vol. 151, Jan. 2021, doi: 10.1016/j.cie.2020.106989.
- [74] O. Ramos-Figueroa, M. Quiroz-Castellanos, E. Mezura-Montes, e O. Schütze, «Metaheuristics to solve grouping problems: A review and a case study», *Swarm Evol Comput*, vol. 53, Mar. 2020, doi: 10.1016/j.swevo.2019.100643.
- [75] A. Ebrahimi, H. W. Jeon, S. Lee, e C. Wang, «Minimizing total energy cost and tardiness penalty for a scheduling-layout problem in a flexible job shop system: A comparison of four metaheuristic algorithms», *Comput Ind Eng*, vol. 141, pp. 1–21, Mar. 2020, doi: 10.1016/j.cie.2020.106295.
- [76] W. Xu, H. Y. Sun, A. L. Awaga, Y. Yan, e Y. J. Cui, «Optimization approaches for solving production scheduling problem: A brief overview and a case study for hybrid flow shop using genetic algorithms», *Advances in Production Engineering And Management*, vol. 17, n. 1, pp. 45–56, Mar. 2022, doi: 10.14743/apem2022.1.420.
- [77] H. Sahar, B. Herrou, e S. Sekkat, «Optimization of Job Shop Scheduling Problem by Genetic Algorithms: Case Study», *Management and Production Engineering Review*, vol. 14, n. 3, pp. 44–56, Set. 2023, doi: 10.24425/mper.2023.147189.
- [78] Q. Zhu, K. Gao, W. Huang, Z. Ma, e A. Slowik, «Q-Learning-Assisted Meta-Heuristics for Scheduling Distributed Hybrid Flow Shop Problems», *Computers, Materials and Continua*, vol. 80, n. 3, pp. 3573–3589, Set. 2024, doi: 10.32604/cmc.2024.055244.
- [79] F. Daneshdoost, M. Hajiaghahi-Keshteli, R. Sahin, e S. Niroomand, «Tabu Search Based Hybrid Meta-Heuristic Approaches for Schedule-Based Production Cost Minimization Problem for the Case of Cable Manufacturing Systems», *Informatica (Netherlands)*, vol. 33, n. 3, pp. 499–522, 2022, doi: 10.15388/21-INFOR471.
- [80] M. Liu, J. Lv, S. Du, Y. Deng, X. Shen, e Y. Zhou, «Multi-resource constrained flexible job shop scheduling problem with fixture-pallet combinatorial optimisation», *Comput Ind Eng*, vol. 188, Fev. 2024, doi: 10.1016/j.cie.2024.109903.
- [81] D. B. M. M. Fontes, S. M. Homayouni, e J. F. Gonçalves, «A hybrid particle swarm optimization and simulated annealing algorithm for the job shop scheduling problem with transport resources», *Eur J Oper Res*, vol. 306, n. 3, pp. 1140–1157, Mai. 2023, doi: 10.1016/j.ejor.2022.09.006.
- [82] E. Albayrak e S. Önüt, «Energy-efficient scheduling for a flexible job shop problem considering rework processes and new job arrival», *International Journal of Industrial Engineering Computations*, vol. 15, n. 4, pp. 871–886, 2024, doi: 10.5267/j.ijiec.2024.7.004.
- [83] M. Abdolrazzagh-Nezhad e S. Abdullah, «Job Shop Scheduling: Classification, Constraints and Objective Functions», *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 11, n. 4, pp. 386–391, 2017.

## Referências

- [84] N. Kundakci e O. Kulak, «Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem», *Comput Ind Eng*, vol. 96, pp. 31–51, Jun. 2016, doi: 10.1016/j.cie.2016.03.011.
- [85] R. K. Phanden, S. Gupta, B. Wolde, R. Kumar, e A. Chakraborty, «Optimisation of job shop scheduling problem using genetic algorithm and simulated annealing: a case study of manufacturing industry», *International Journal of System Assurance Engineering and Management*, Jan. 2025, doi: 10.1007/s13198-025-02714-7.
- [86] J. Li, X. Li, L. Gao, e Q. Liu, «A Flexible Job Shop Scheduling Problem Considering On-Site Machining Fixtures: A Case Study From Customized Manufacturing Enterprise», *IEEE Transactions on Automation Science and Engineering*, 2024, doi: 10.1109/TASE.2024.3485810.
- [87] R. Chen, B. Yang, S. Li, e S. Wang, «A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem», *Comput Ind Eng*, vol. 149, Nov. 2020, doi: 10.1016/j.cie.2020.106778.
- [88] L. Hao, Z. Zou, e X. Liang, «Solving multi-objective energy-saving flexible job shop scheduling problem by hybrid search genetic algorithm», *Comput Ind Eng*, vol. 200, Fev. 2025, doi: 10.1016/j.cie.2024.110829.
- [89] S. L. Aquinaldo, N. R. Cucuk, e Yuniaristanto, «Optimization in job shop scheduling problem using Genetic Algorithm (study case in furniture industry)», *IOP Conf Ser Mater Sci Eng*, vol. 1072, n. 1, Fev. 2021, doi: 10.1088/1757-899x/1072/1/012019.
- [90] Y. Hou, X. Liao, G. Chen, e Y. Chen, «Co-Evolutionary NSGA-III with deep reinforcement learning for multi-objective distributed flexible job shop scheduling», *Comput Ind Eng*, vol. 203, Mai. 2025, doi: 10.1016/j.cie.2025.110990.
- [91] S. K. Fuladi e C. S. Kim, «Dynamic Events in the Flexible Job-Shop Scheduling Problem: Rescheduling with a Hybrid Metaheuristic Algorithm», *Algorithms*, vol. 17, n. 4, Abr. 2024, doi: 10.3390/a17040142.
- [92] K. Ben Ali, S. Bechikh, A. Louati, H. Louati, e E. Kariri, «Dynamic Job Shop Scheduling Problem With New Job Arrivals Using Hybrid Genetic Algorithm», *IEEE Access*, vol. 12, pp. 85338–85354, Jun. 2024, doi: 10.1109/ACCESS.2024.3401080.
- [93] T. Wang, Y. Teng, C. Zhang, Y. Gao, e X. Li, «A Hybrid Genetic Algorithm for Flexible Job Shop Scheduling Problem with Batch Processing Machines», em *Proceedings of the 2024 27th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 1893–1898. doi: 10.1109/CSCWD61410.2024.10580140.
- [94] M. Zhuang, W. Zhang, H. Tang, X. Li, e K. Wang, «A multi-objective genetic algorithm based on two-stage reinforcement learning for green flexible shop scheduling problem considering machine speed», *Expert Syst Appl*, vol. 258, Dez. 2024, doi: 10.1016/j.eswa.2024.125189.
- [95] B. Tağtekin, M. U. Öztürk, e M. K. Sezer, «A Case Study: Using Genetic Algorithm for Job Scheduling Problem», Jun. 2021, [Em linha]. Disponível em: <http://arxiv.org/abs/2106.04854>
- [96] S. X. Wang, C. Y. Zhang, e L. L. Jin, «A Hybrid Genetic Algorithm for Flexible Job-shop Scheduling Problem», em *Advanced Materials Research*, 2014, pp. 1179–1184. doi: 10.4028/www.scientific.net/AMR.889-890.1179.
- [97] K. Hu, L. Wang, J. Cai, e L. Cheng, «An improved genetic algorithm with dynamic neighborhood search for job shop scheduling problem», *Mathematical Biosciences and Engineering*, vol. 20, n. 9, pp. 17407–17427, 2023, doi: 10.3934/mbe.2023774.
- [98] J. Kennedy e R. Eberhart, «Particle Swarm Optimization», em *International Conference on Neural Networks*, IEEE, 1995, pp. 1942–1948.

## Referências

- [99] K. Kameyama, «Particle Swarm Optimization - A Survey», Julho de 2009, *Institute of Electronics, Information and Communication, Engineers, IEICE*. doi: 10.1587/transinf.E92.D.1354.
- [100] W. Zhang, M. Zhuang, H. Tang, X. Li, e S. Guo, «A hybrid particle swarm optimisation for flexible casting job shop scheduling problem with batch processing machine», *IET Collaborative Intelligent Manufacturing*, vol. 6, n. 4, Dez. 2024, doi: 10.1049/cim2.12117.
- [101] M. Nouiri, A. Jemai, A. Chiheb Ammari, A. Bekrar, e S. Niar, «An effective particle swarm optimization algorithm for flexible job-shop scheduling problem», Rabat, Out. 2013.
- [102] W. Zhou, Y. Bu, e Y. Zhou, «Combining CA and PSO to Solve Flexible Job shop Scheduling Problem», em *26th Chinese Control and Decision Conference (CCDC)*, IEEE Industrial Electronics (IE) Chapter, Singapore, 2014, pp. 1031–1036.
- [103] Z. Zhang, Z. L. Guan, J. Zhang, e X. Xie, «A NOVEL JOB-SHOP SCHEDULING STRATEGY BASED ON PARTICLE SWARM OPTIMIZATION AND NEURAL NETWORK», *International Journal of Simulation Modelling*, vol. 18, n. 4, pp. 699–707, Dez. 2019, doi: 10.2507/IJSIMM18(4)CO18.
- [104] M. Nouiri, A. Bekrar, A. Jemai, S. Niar, e A. C. Ammari, «An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem», *J Intell Manuf*, vol. 29, n. 3, pp. 603–615, Mar. 2018, doi: 10.1007/s10845-015-1039-3.
- [105] L. Bi e K. Chen, «Research on FJSP with transportation time constraint based on Improved Particle Swarm Optimization», Yinchuan: IEEE Computer Society, Conference Publishing Services, Ago. 2020, pp. 130–137.
- [106] F. Zhao, J. Tang, J. Wang, e Jonrinaldi, «An improved particle swarm optimization with decline disturbance index (DDPSO) for multi-objective job-shop scheduling problem», *Comput Oper Res*, vol. 45, pp. 38–50, Mai. 2014, doi: 10.1016/j.cor.2013.11.019.
- [107] T. Jamrus, C. F. Chien, M. Gen, e K. Sethanan, «Hybrid Particle Swarm Optimization Combined With Genetic Operators for Flexible Job-Shop Scheduling Under Uncertain Processing Time for Semiconductor Manufacturing», *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, n. 1, pp. 32–41, Fev. 2018, doi: 10.1109/TSM.2017.2758380.
- [108] S. Kawaguchi e Y. Fukuyama, «Reactive Hybrid Particle Swarm Optimization Based Job-shop Scheduling Problems Considering Energy Management», em *Proceedings of TENCON 2018*, Tokyo: IEEE, Out. 2018, pp. 2181–2186.
- [109] F. A. Toader, «Production Scheduling by Using ACO and PSO Techniques», em *12th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS*, Suceava: IEEE, Mai. 2014, pp. 170–175.
- [110] N. I. Anuar, M. H. F. M. Fauadi, e A. Saptari, «Performance Evaluation of Continuous and Discrete Particle Swarm Optimization in Job-Shop Scheduling Problems», em *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, 2019. doi: 10.1088/1757-899X/530/1/012044.
- [111] S. Kamel e S. Boubaker, «Comparison of two variants of particle swarm optimization algorithm for solving flexible job shop scheduling problem», em *International Conference on Recent Advances in Computer Systems (RACS 2015)*, Atlantis Press, 2016, pp. 40–45.
- [112] M. Abdolrazzagh-Nezhad e S. Sarbishegi, «Job-Shop Scheduling with Fuzzy Due Date by Multi-Objective Particle Swarm Optimization», em *5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, IEEE, Dez. 2019.

## Referências

- [113] M. R. Singh e S. S. Mahapatra, «A quantum behaved particle swarm optimization for flexible job shop scheduling», *Comput Ind Eng*, vol. 93, pp. 36–44, Mar. 2016, doi: 10.1016/j.cie.2015.12.004.
- [114] A. AitZai, B. Benmedjdoub, e M. Boudhar, «Branch-and-bound and PSO algorithms for no-wait job shop scheduling», *J Intell Manuf*, vol. 27, n. 3, pp. 679–688, Jun. 2016, doi: 10.1007/s10845-014-0906-7.
- [115] N. I. Anuar e A. Saptari, «Performance evaluation of different types of particle representation procedures of Particle Swarm Optimization in Job-shop Scheduling Problems», em *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Mar. 2016. doi: 10.1088/1757-899X/114/1/012055.
- [116] M. U. Sana, Z. Li, F. Javaid, M. W. Hanif, e I. Ashraf, «Improved particle swarm optimization based on blockchain mechanism for flexible job shop problem», *Cluster Comput*, vol. 26, n. 5, pp. 2519–2537, Out. 2023, doi: 10.1007/s10586-021-03349-6.
- [117] Q. Zhang e S. Hu, «An Improved Hybrid Quantum Particle Swarm Optimization Algorithm for FJSP», em *ACM International Conference Proceeding Series*, Association for Computing Machinery, 2019, pp. 246–252. doi: 10.1145/3318299.3318359.
- [118] T. Ning, M. Huang, X. Liang, e H. Jin, «A novel dynamic scheduling strategy for solving flexible job-shop problems», *J Ambient Intell Humaniz Comput*, vol. 7, n. 5, pp. 721–729, Out. 2016, doi: 10.1007/s12652-016-0370-7.
- [119] A. Amirteimoori e R. Kia, «Concurrent scheduling of jobs and AGVs in a flexible job shop system: a parallel hybrid PSO-GA meta-heuristic», *Flex Serv Manuf J*, vol. 35, n. 3, pp. 727–753, Set. 2023, doi: 10.1007/s10696-022-09453-y.
- [120] K. Tamssaouet, S. Dautère-Pérès, e C. Yugma, «Metaheuristics for the job-shop scheduling problem with machine availability constraints», *Comput Ind Eng*, vol. 125, pp. 1–8, Nov. 2018, doi: 10.1016/j.cie.2018.08.008.
- [121] J. Li, Q. Pan, S. Xie, Gao. Kaizhou, e Y. Wang, «A hybrid algorithm for multi-objective job shop scheduling problem», em *2011 Chinese Control and Decision Conference*, 2011, pp. 3630–3634.
- [122] B. Marzouki, O. B. Driss, e K. Ghedira, «Decentralized Tabu Searches in Multi Agent system for Distributed and Flexible Job shop Scheduling Problem», em *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, IEEE Computer Society, Jul. 2017, pp. 1019–1026. doi: 10.1109/AICCSA.2017.133.
- [123] J. Q. Li, Q. K. Pan, P. N. Suganthan, e T. J. Chua, «A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem», *International Journal of Advanced Manufacturing Technology*, vol. 52, n. 5–8, pp. 683–697, Fev. 2011, doi: 10.1007/s00170-010-2743-y.
- [124] X. Li e L. Gao, «An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem», *Int J Prod Econ*, vol. 174, pp. 93–110, Abr. 2016, doi: 10.1016/j.ijpe.2016.01.016.
- [125] S. Mahmud, A. Abbasi, R. K. Chakraborty, e M. J. Ryan, «Multi-operator communication based differential evolution with sequential Tabu Search approach for job shop scheduling problems», *Appl Soft Comput*, vol. 108, Set. 2021, doi: 10.1016/j.asoc.2021.107470.
- [126] T. K. Zubaran e M. R. P. Ritt, «An effective tabu search for job shop scheduling with parallel machines», em *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, Munich: Institute of Electrical and Electronics Engineers, Ago. 2018, pp. 913–919.
- [127] Q. Zhang, H. Manier, e M. A. Manier, «A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded

## Referências

- processing times», *Comput Oper Res*, vol. 39, n. 7, pp. 1713–1723, Jul. 2012, doi: 10.1016/j.cor.2011.10.007.
- [128] J. Q. Li, Q. K. Pan, e Y. C. Liang, «An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems», *Comput Ind Eng*, vol. 59, n. 4, pp. 647–662, Nov. 2010, doi: 10.1016/j.cie.2010.07.014.
- [129] C. Peng, G. Wu, T. Warren Liao, e H. Wang, «Research on multi-agent genetic algorithm based on tabu search for the job shop scheduling problem», *PLoS One*, vol. 14, n. 9, Set. 2019, doi: 10.1371/journal.pone.0223182.
- [130] W. Tessaro Lunardi, H. Voos, e L. Henrique Cherri, «An Effective Hybrid Imperialist Competitive Algorithm and Tabu Search for an Extended Flexible Job Shop Scheduling Problem», *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19)*, pp. 204–211, Abr. 2019, doi: 10.1145/3297280.
- [131] M. Harrabi e O. B. Driss, «MATS–JSTL: A multi-agent model based on tabu search for job shop problem with time lags», em *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2015, pp. 39–46. doi: 10.1007/978-3-319-24069-5\_4.
- [132] F. Farjallah, H. E. Nouri, e O. B. Driss, «Multi-agent model-based intensification-driven tabu search for solving the dual-resource constrained flexible job shop scheduling», *Journal of Information and Telecommunication*, 2024, doi: 10.1080/24751839.2024.2425478.
- [133] J. José Palacios, I. González-Rodríguez, C. R. Vela, e J. Puente, «Robust multiobjective optimisation for fuzzy job shop problems», *Applied Soft Computing Journal*, vol. 56, pp. 604–616, Jul. 2017, doi: 10.1016/j.asoc.2016.07.004.
- [134] L. Berterottière, S. Dauzère-Pérès, e C. Yugma, «Flexible job-shop scheduling with transportation resources», *Eur J Oper Res*, vol. 312, n. 3, pp. 890–909, Fev. 2024, doi: 10.1016/j.ejor.2023.07.036.
- [135] W. Xu, Y. Hu, W. Luo, L. Wang, e R. Wu, «A multi-objective scheduling method for distributed and flexible job shop based on hybrid genetic algorithm and tabu search considering operation outsourcing and carbon emission», *Comput Ind Eng*, vol. 157, Jul. 2021, doi: 10.1016/j.cie.2021.107318.
- [136] Q. Zhang, H. Manier, e M.-A. Manier, «A Hybrid Metaheuristic Algorithm for Flexible Job-Shop Scheduling Problems with Transportation Constraints», em *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, ACM Digital Library, Jul. 2012, pp. 441–448.
- [137] J. J. Palacios, M. A. González, C. R. Vela, I. González-Rodríguez, e J. Puente, «Genetic tabu search for the fuzzy flexible job shop problem», *Comput Oper Res*, vol. 54, pp. 74–89, 2015, doi: 10.1016/j.cor.2014.08.023.
- [138] S. Meeran e M. S. Morshed, «Evaluation of a hybrid genetic tabu search framework on job shop scheduling benchmark problems», *Int J Prod Res*, vol. 52, n. 19, pp. 5780–5798, 2014, doi: 10.1080/00207543.2014.911417.
- [139] W. Xiuli e L. Xiajing, «An Improved Differential Evolution Algorithm for Solving a Distributed Flexible Job Shop Scheduling Problem», em *2018 IEEE 14th International Conference on Automation Science and Engineering*, Munich: Institute of Electrical and Electronics Engineers, Ago. 2018, pp. 968–973.
- [140] K. C. W. Lim, L. P. Wong, e J. F. Chin, «Hyper-heuristic for flexible job shop scheduling problem with stochastic job arrivals», *Manuf Lett*, vol. 36, pp. 5–8, Jul. 2023, doi: 10.1016/j.mfglet.2022.12.009.
- [141] X. Wu, X. Shen, e C. Li, «The flexible job-shop scheduling problem considering deterioration effect and energy consumption simultaneously», *Comput Ind Eng*, vol. 135, pp. 1004–1024, Set. 2019, doi: 10.1016/j.cie.2019.06.048.

## Referências

- [142] K. Akram e K. Kamal, «Hybridization of Simulated Annealing with Quenching for Job Shop Scheduling», em *2015 International Conference on Fluid Power and Mechatronics*, Harbin: IEEE, Ago. 2015, pp. 825–829.
- [143] N. Shivasankaran, P. S. Kumar, e K. Venkatesh Raja, «Hybrid Sorting Immune Simulated Annealing Algorithm For Flexible Job Shop Scheduling», *International Journal of Computational Intelligence Systems*, vol. 8, n. 3, pp. 455–466, 2015.
- [144] X.-F. Zhang, M. Koshimura, H. Fujita, e R. Hasegawa, «An Efficient Hybrid Particle Swarm Optimization for the Job Shop Scheduling Problem», em *2011 IEEE International Conference on Fuzzy Systems*, Taipei: IEEE, Jun. 2011, pp. 622–626.
- [145] A. Thammano e A. Phu-Ang, «A Hybrid Artificial Bee Colony Algorithm with Local Search for Flexible Job-Shop Scheduling Problem», em *Procedia Computer Science*, Elsevier B.V., 2013, pp. 96–101. doi: 10.1016/j.procs.2013.09.245.
- [146] D. De C Bissoli e A. R. S. Amaral, «A hybrid iterated local search metaheuristic for the flexible job shop scheduling problem», em *Proceedings - 2018 44th Latin American Computing Conference, CLEI 2018*, Institute of Electrical and Electronics Engineers Inc., Out. 2018, pp. 149–157. doi: 10.1109/CLEI.2018.00026.
- [147] Y. Li, Z. Yang, L. Wang, H. Tang, L. Sun, e S. Guo, «A hybrid imperialist competitive algorithm for energy-efficient flexible job shop scheduling problem with variable-size sublots», *Comput Ind Eng*, vol. 172, Out. 2022, doi: 10.1016/j.cie.2022.108641.
- [148] K. C. W. Lim, L. P. Wong, e J. F. Chin, «Simulated-annealing-based hyper-heuristic for flexible job-shop scheduling», *Engineering Optimization*, vol. 55, n. 10, pp. 1635–1651, 2023, doi: 10.1080/0305215X.2022.2106477.
- [149] H. Piroozfard, A. Hassan, A. M. Moghadam, e A. Derakhshan Asl, «A Hybrid Genetic Algorithm for Solving Job Shop Scheduling Problems», em *Advanced Materials Research*, Trans Tech Publications, 2014, pp. 559–563. doi: 10.4028/www.scientific.net/AMR.845.559.
- [150] Z. Li, C. Zhao, G. Zhang, D. Zhu, e L. Cui, «Multi-strategy improved sparrow search algorithm for job shop scheduling problem», *Cluster Comput*, vol. 27, n. 4, pp. 4605–4619, Jul. 2024, doi: 10.1007/s10586-023-04200-w.
- [151] M. Zandieh, A. R. Khatami, e S. H. A. Rahmati, «Flexible job shop scheduling under condition-based maintenance: Improved version of imperialist competitive algorithm», *Applied Soft Computing Journal*, vol. 58, pp. 449–464, Set. 2017, doi: 10.1016/j.asoc.2017.04.060.
- [152] M. Rojas-Santiago, P. Damodaran, S. Muthuswamy, e M. C. Vélez-Gallego, «Makespan minimization in a job shop with a BPM using simulated annealing», *International Journal of Advanced Manufacturing Technology*, vol. 68, n. 9–12, pp. 2383–2391, Out. 2013, doi: 10.1007/s00170-013-4858-4.
- [153] B. Khurshid e S. Maqsood, «A hybrid evolution strategies-simulated annealing algorithm for job shop scheduling problems», *Eng Appl Artif Intell*, vol. 133, Jul. 2024, doi: 10.1016/j.engappai.2024.108016.
- [154] V. Roshanaei, A. Azab, e H. Elmaraghy, «Mathematical modelling and a meta-heuristic for flexible job shop scheduling», *Int J Prod Res*, vol. 51, n. 20, pp. 6247–6274, Out. 2013, doi: 10.1080/00207543.2013.827806.
- [155] S. W. Lin e K. C. Ying, «Minimising makespan in job-shops with deterministic machine availability constraints», *Int J Prod Res*, vol. 59, n. 14, pp. 4403–4415, 2021, doi: 10.1080/00207543.2020.1764125.

# Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade. Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Declaro que o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.

NOME: Tomás Sequeira Ramos Nunes de Sousa

ISEP, Porto, 14 de junho de 2025







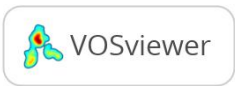
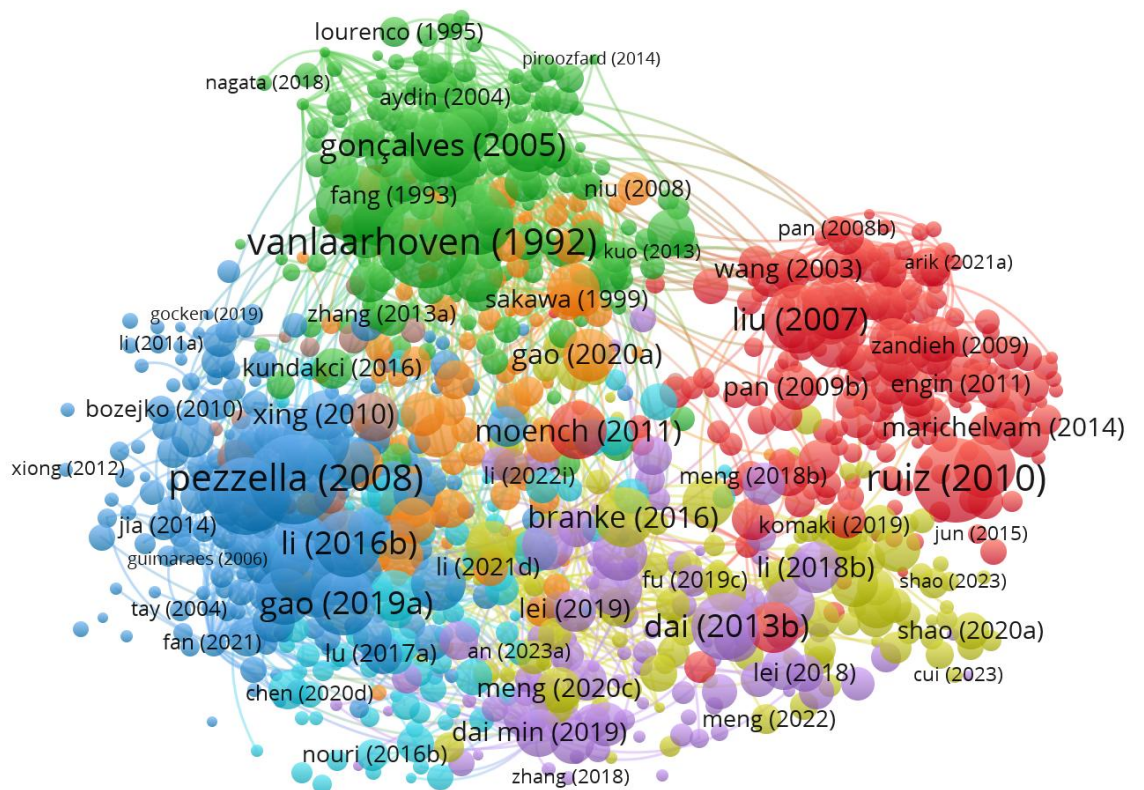


## Apêndice A

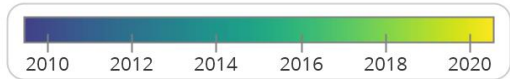
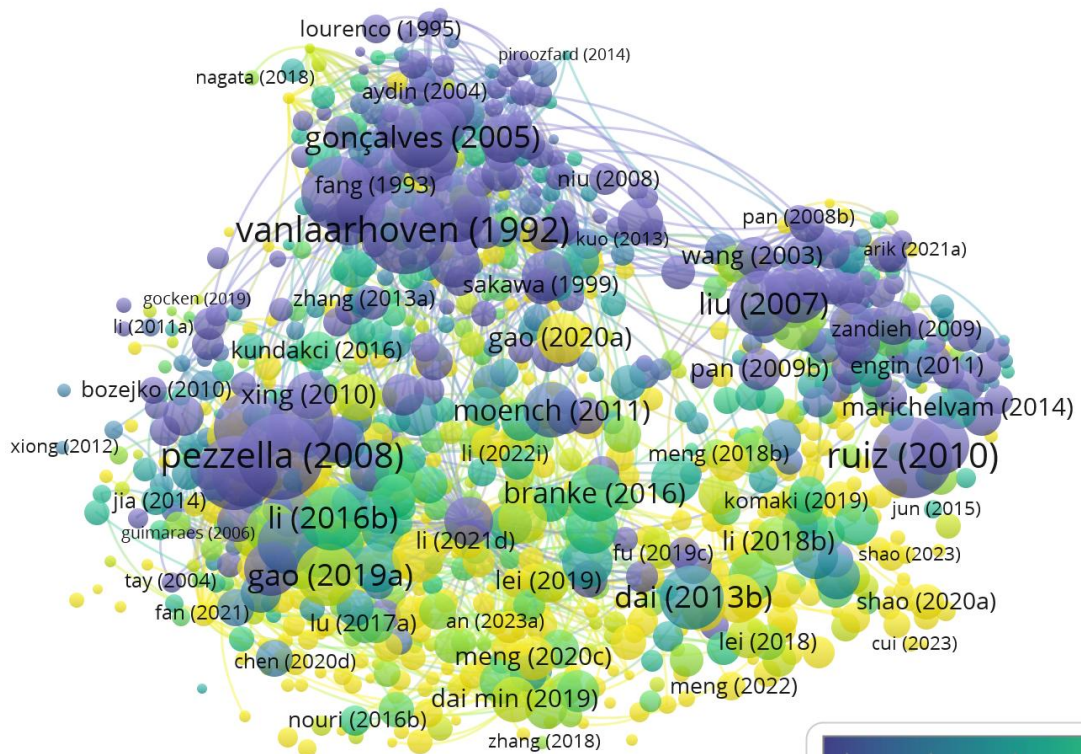
# Apêndice B

## Análise de Citação por Documentos

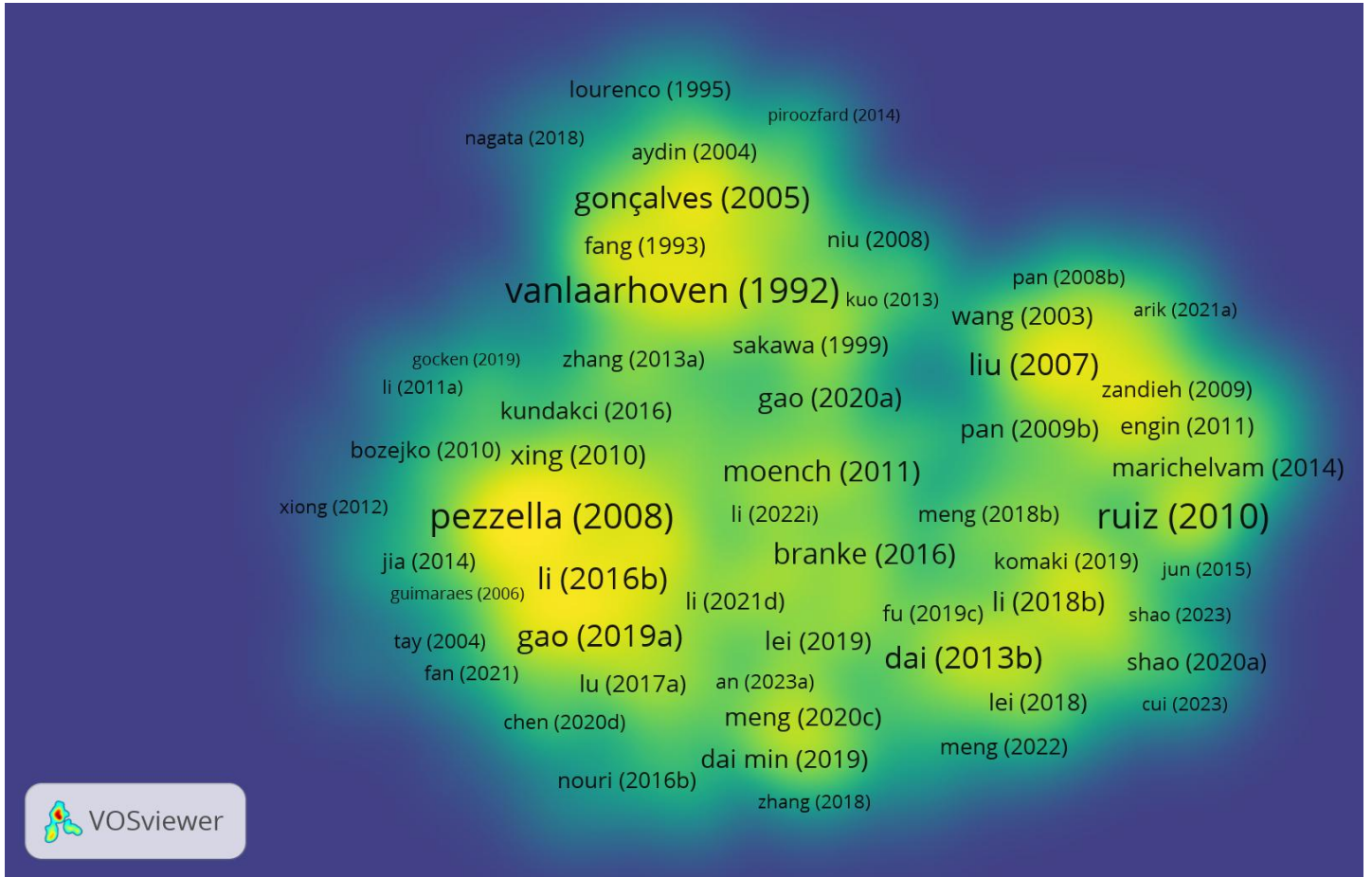
- Vista de Redes



- **Temporal**

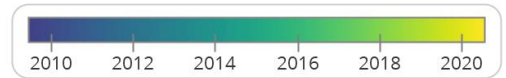
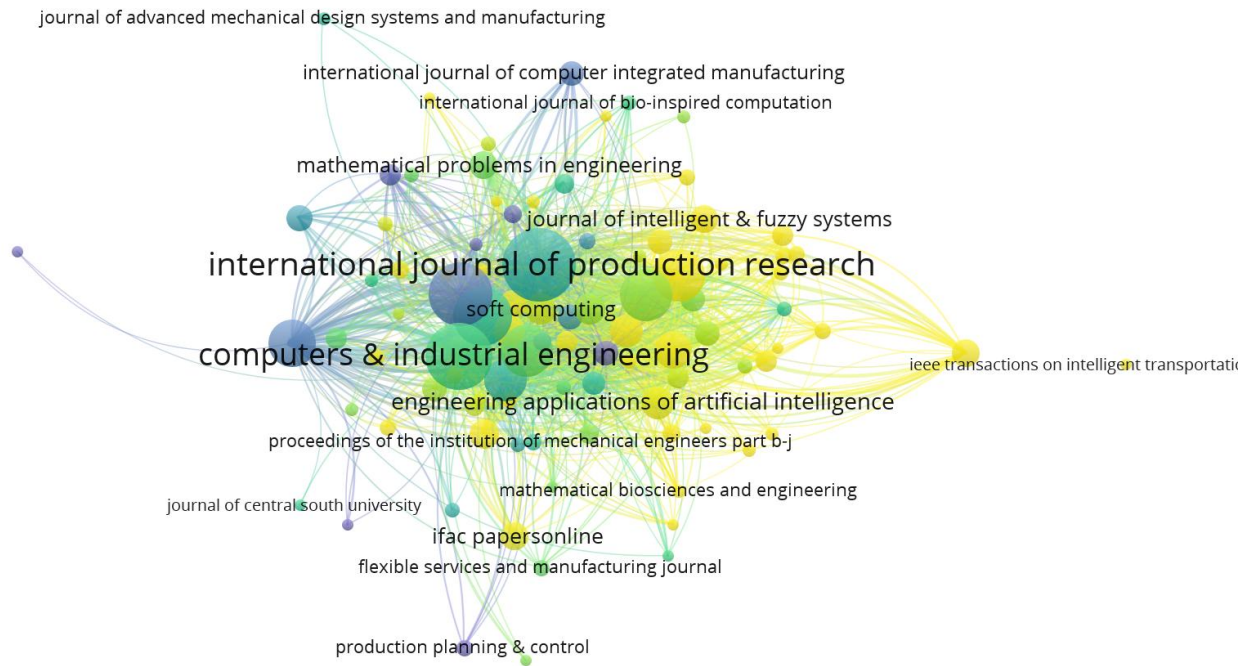


- **Densiométrica**





- **Temporal**



## Apêndice B

- **Densiométrica**



### Análise de Citação por Autores

- Vista de Redes

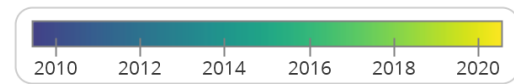
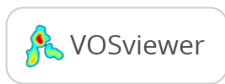


## Apêndice B

- **Temporal**



dimitrakopoulos, roussos



- **Densiométrica**

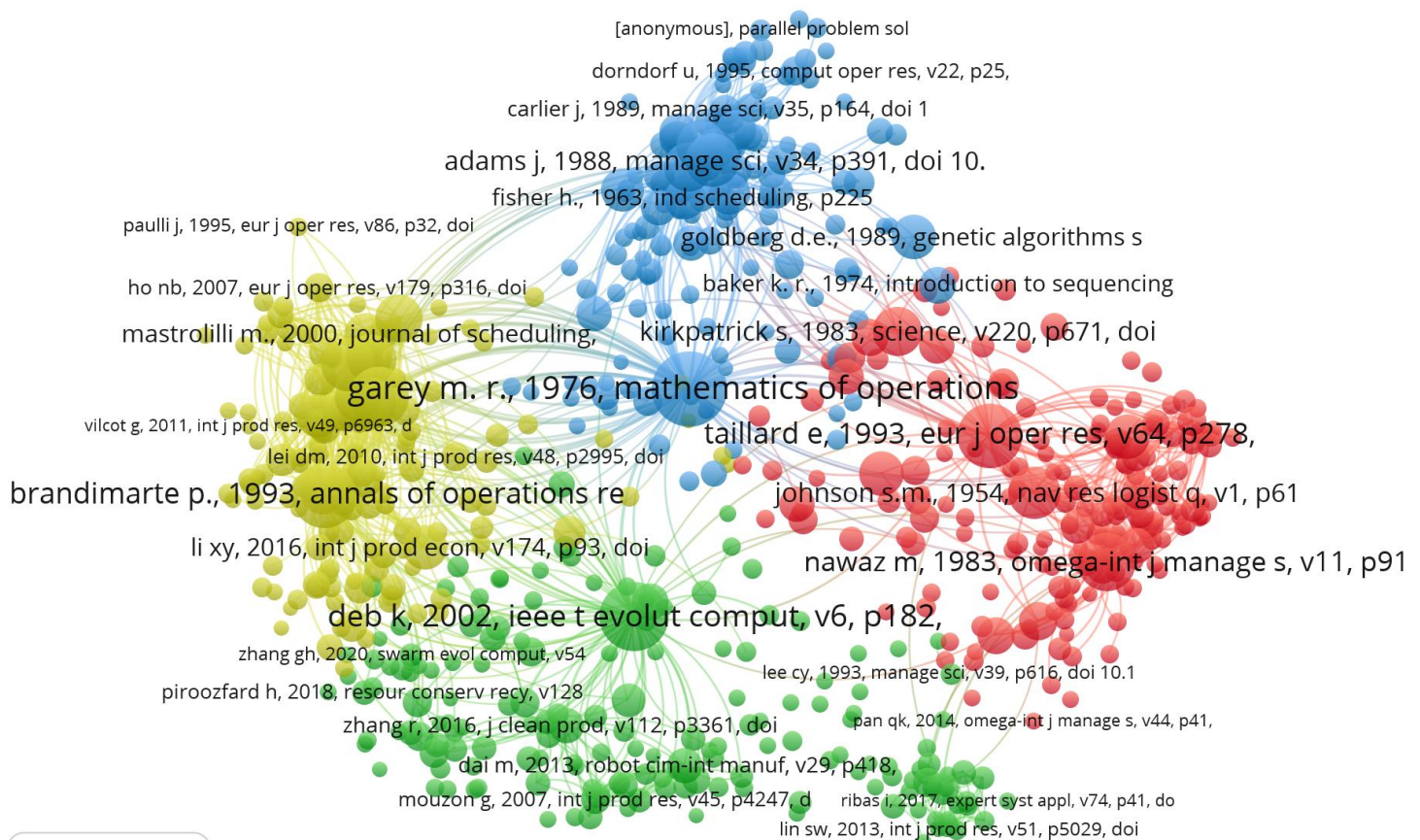


## Apêndice B

# Apêndice C

## Análise de Co-Citação por Referências Citadas

- Vista de Redes



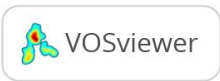
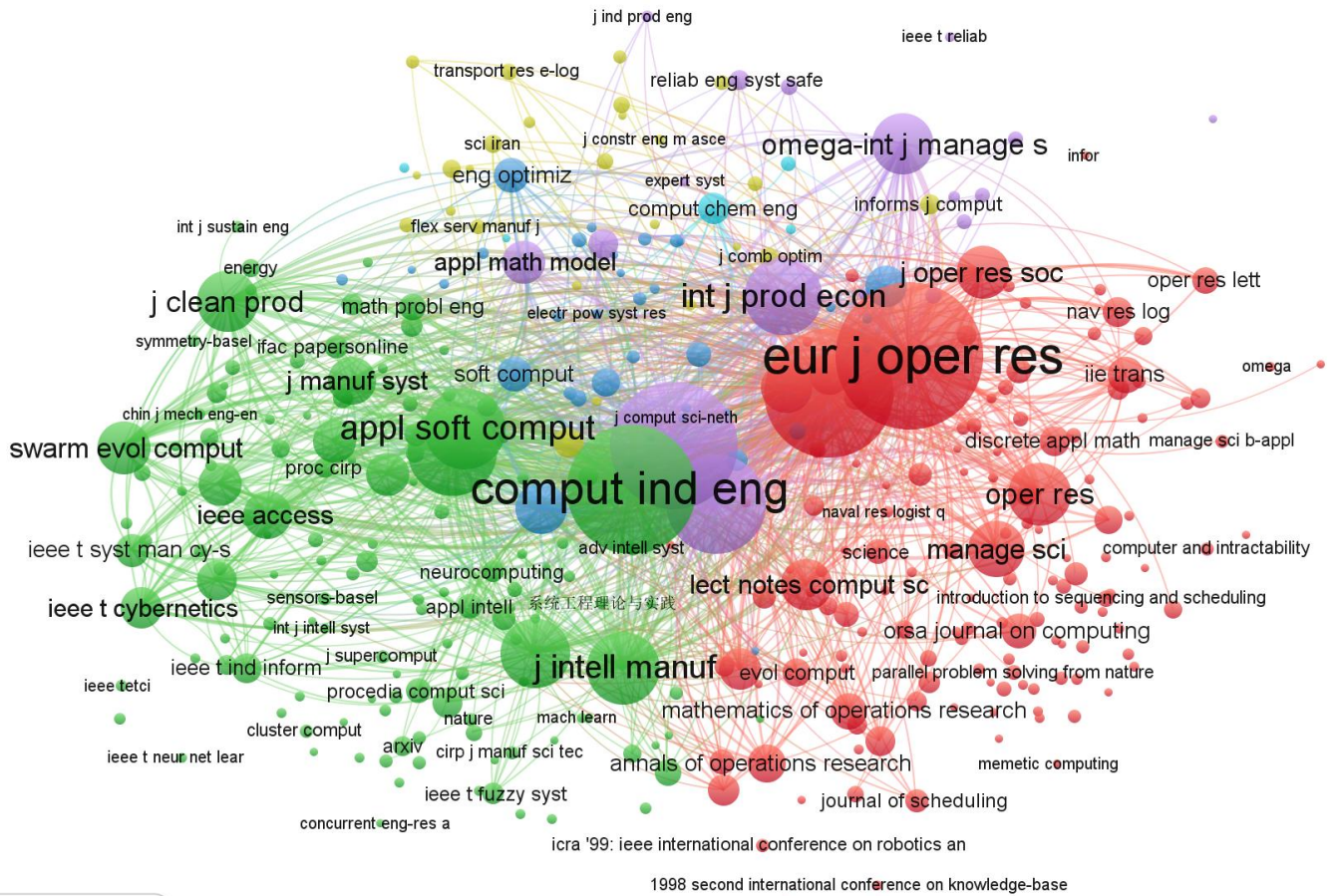
- **Densiométrica**

[anonymous], parallel problem sol  
dorndorf u, 1995, comput oper res, v22, p25,  
carlier j, 1989, manage sci, v35, p164, doi 1  
adams j, 1988, manage sci, v34, p391, doi 10.  
fisher h., 1963, ind scheduling, p225  
paulij j, 1995, eur j oper res, v86, p32, doi  
goldberg d.e., 1989, genetic algorithms s  
ho nb, 2007, eur j oper res, v179, p316, doi  
baker k. r., 1974, introduction to sequencing  
mastrolilli m., 2000, journal of scheduling, kirkpatrick s, 1983, science, v220, p671, doi  
garey m. r., 1976, mathematics of operations  
vilcot g, 2011, int j prod res, v49, p6963, d  
taillard e, 1993, eur j oper res, v64, p278,  
lei dm, 2010, int j prod res, v48, p2995, doi  
brandimarte p., 1993, annals of operations re johnson s.m., 1954, nav res logist q, v1, p61  
li xy, 2016, int j prod econ, v174, p93, doi  
nawaz m, 1983, omega-int j manage s, v11, p91  
deb k, 2002, ieee t evolut comput, v6, p182,  
zhang gh, 2020, swarm evol comput, v54  
lee cy, 1993, manage sci, v39, p616, doi 10.1  
piroozfard h, 2018, resour conserv recy, v128  
zhang r, 2016, j clean prod, v112, p3361, doi pan qk, 2014, omega-int j manage s, v44, p41,  
dai m, 2013, robot cim-int manuf, v29, p418,  
mouzon g, 2007, int j prod res, v45, p4247, d ribas i, 2017, expert syst appl, v74, p41, do  
lin sw, 2013, int j prod res, v51, p5029, doi

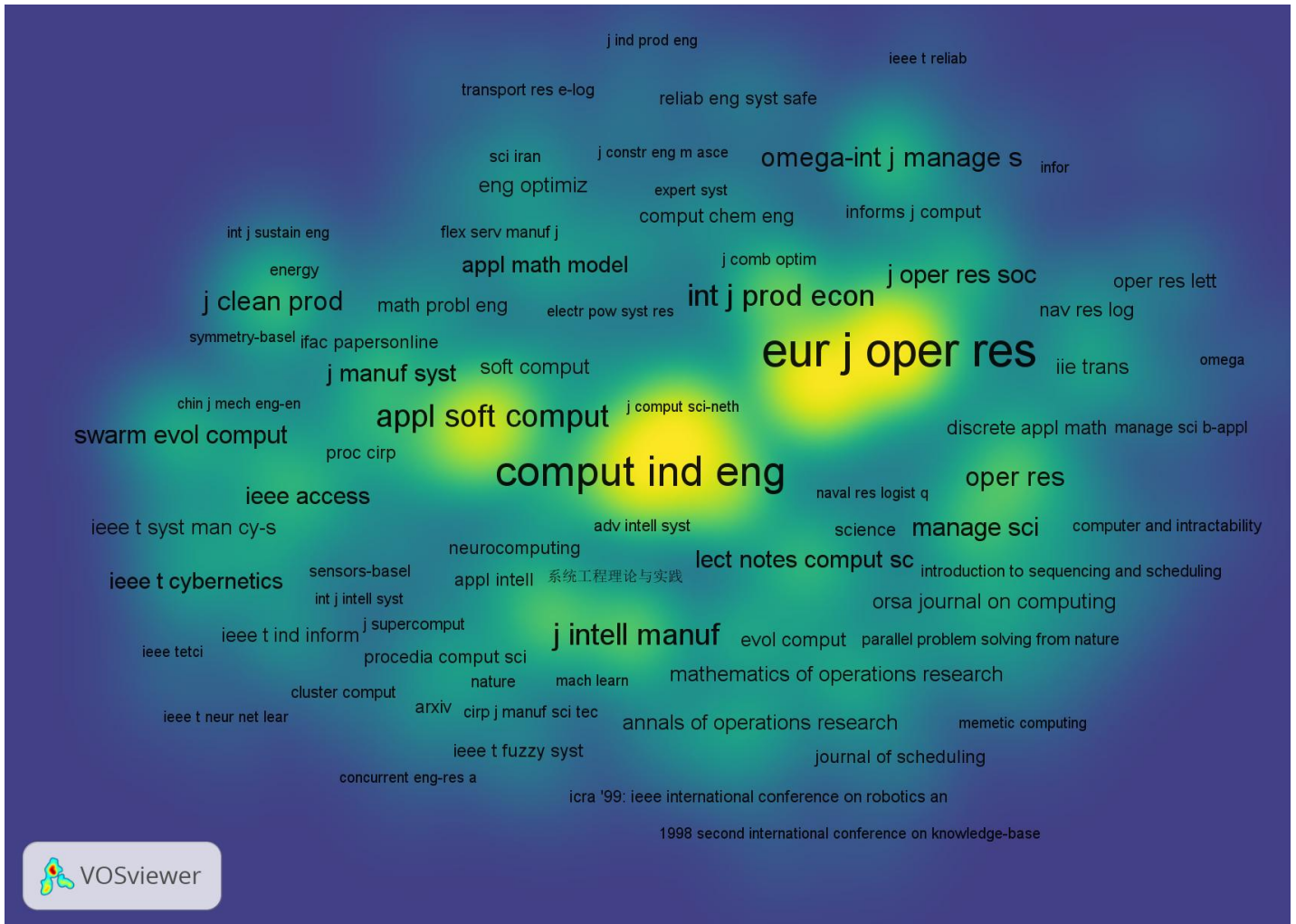


### Análise de Co-Citação por Fontes Citadas

- Vista de Redes

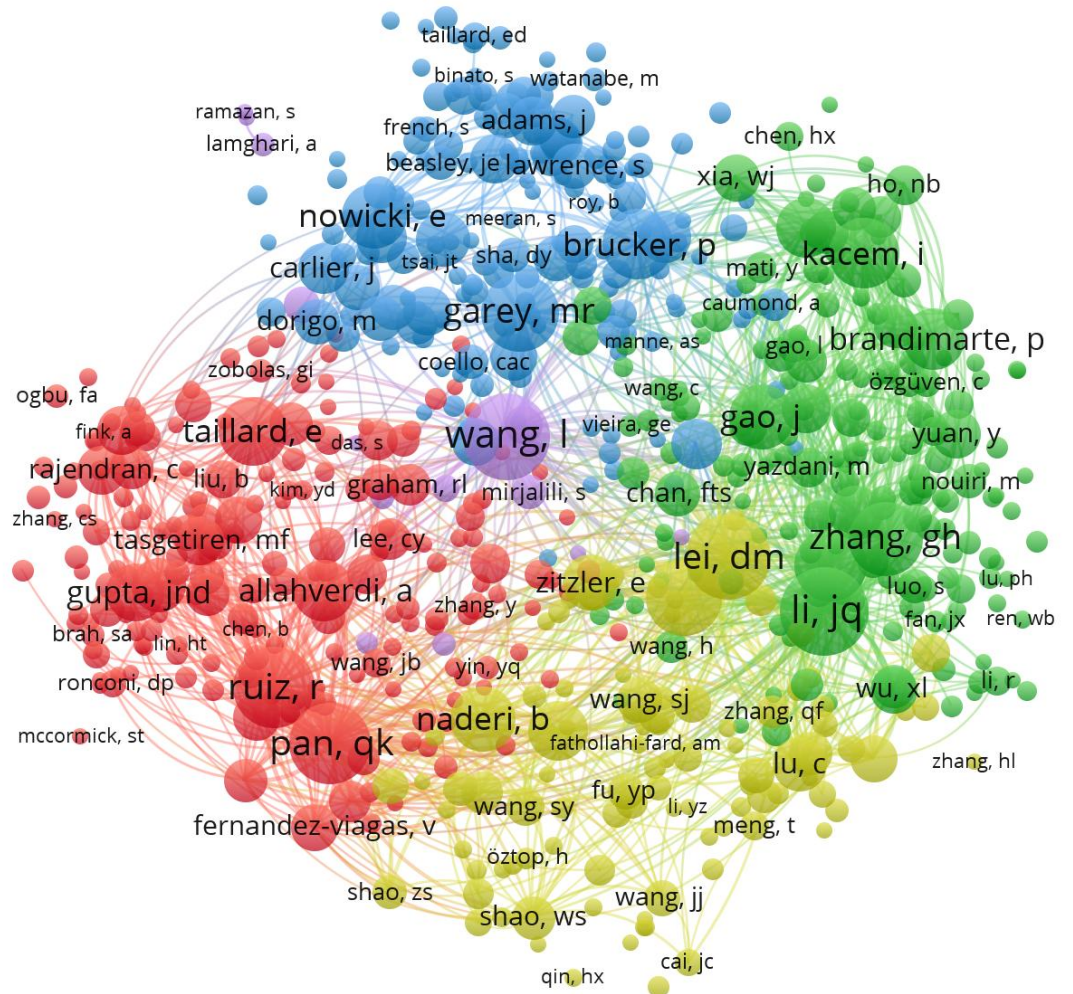


- **Densiométrica**

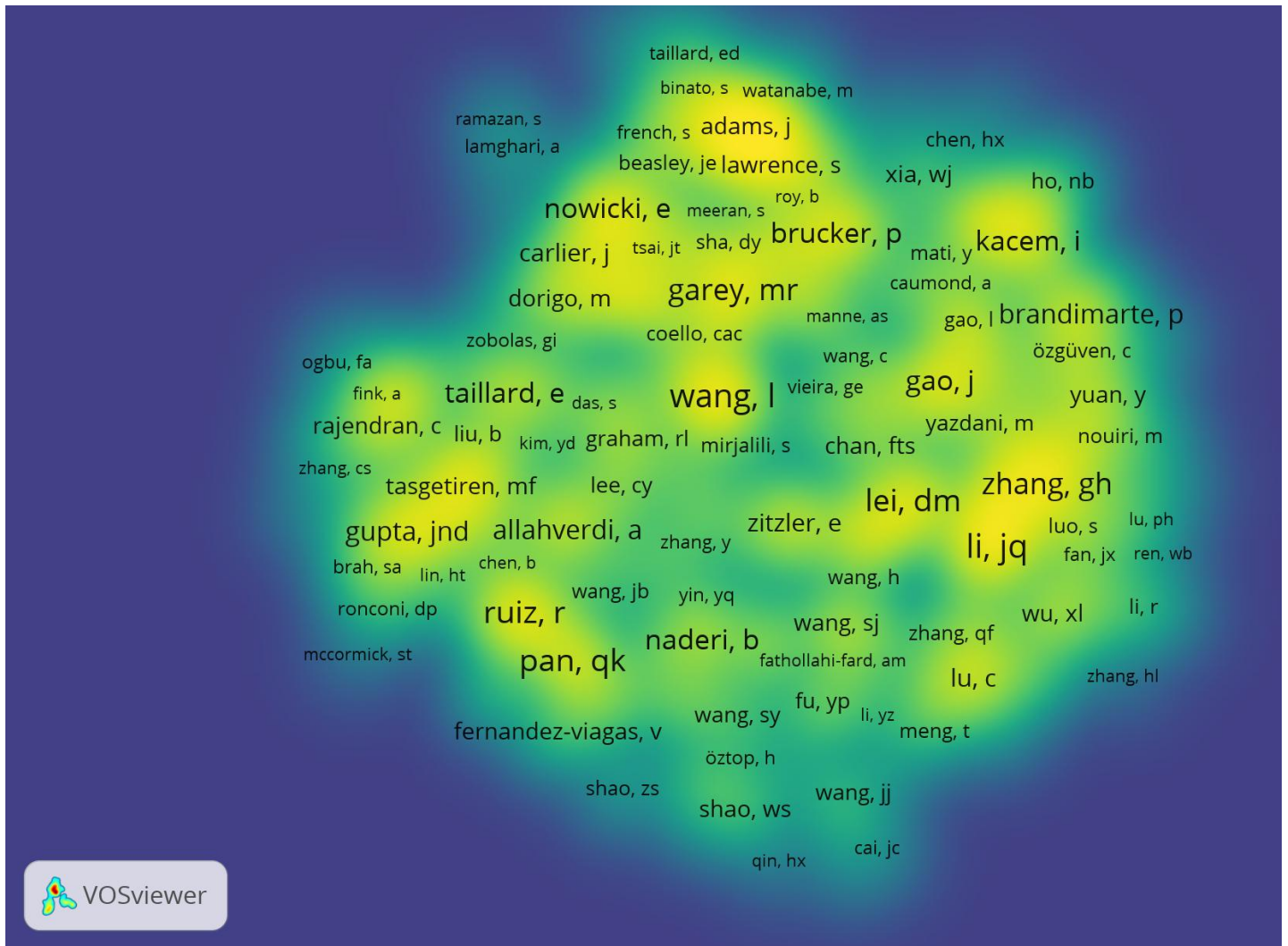


### Análise de Co-Citação por Autores Citados

- Vista de Redes

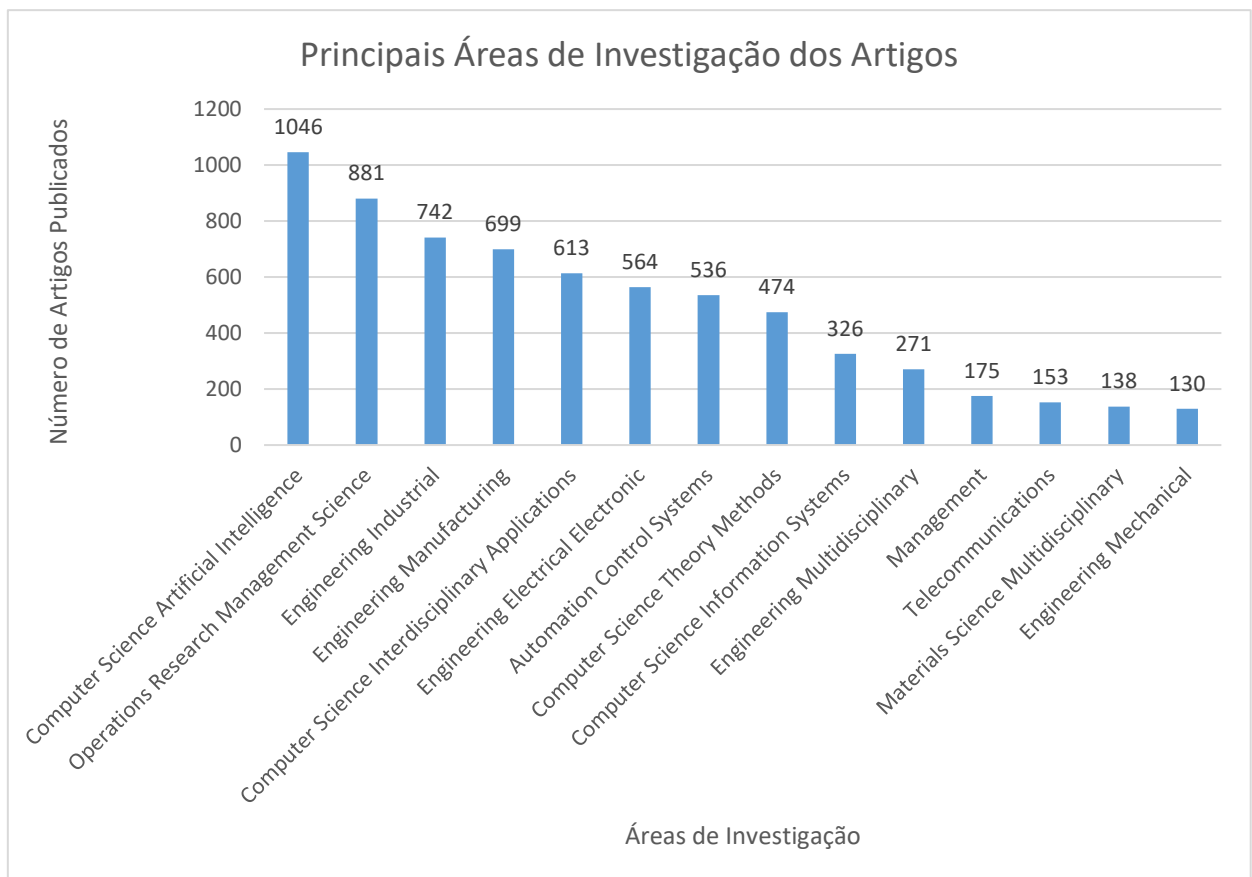


- **Densiométrica**



# Apêndice D

## Principais Áreas de Investigação dos Artigos provenientes da Query





# Apêndice E

## Evolução Temporal do Número de Publicações dos Artigos provenientes da Query

