

Perspectivas multiobjetivo em ambientes de máquinas paralelas não-relacionadas

Antonio Sergio Gallo Junior¹, Vitor Jardim¹, Mario Alberto Mendoza Villalba¹, Flávia Barbosa², Priscila C. Berbert Rampazzo¹

¹ Faculdade de Ciências Aplicadas, Universidade Estadual de Campinas, R. Pedro Zaccaria, 1300, Limeira, São Paulo, Brasil, 13484-350

² Faculdade de Engenharia da Universidade do Porto, R. Dr. Roberto Frias, 378, Porto, Portugal, 4200-465

a289962@dac.unicamp.br, v289445@dac.unicamp.br,
m247931@dac.unicamp.br, flavia.barbosa@inesctec.pt,
priscila.rampazzo@fca.unicamp.br

RESUMO

Este trabalho aborda o problema de escalonamento em máquinas paralelas não-relacionadas (UPMSP) sob a perspectiva da otimização multiobjetivo, considerando a minimização simultânea do *makespan* e do atraso total. Foi utilizado o algoritmo NSGA-II, reconhecido por sua eficiência na obtenção de soluções não-dominadas em problemas com objetivos conflitantes. Adotou-se uma codificação baseada em um vetor de números reais, que facilita a implementação dos operadores evolutivos em problemas combinatórios. Os resultados experimentais demonstraram a existência de *trade-offs* entre os objetivos, com diferentes padrões de ocupação das máquinas a depender do critério priorizado. A heterogeneidade dos tempos de processamento, característica do UPMSP, acentua essa dinâmica, reforçando a importância de abordagens que forneçam múltiplas soluções alternativas para apoiar a tomada de decisão.

PALAVRAS CHAVE. Otimização Multiobjetivo. *Scheduling*. Metaheurísticas.

Tópicos: OMO, AD&GP, MH

ABSTRACT

This work addresses the scheduling problem in unrelated parallel machines (UPMSP) from the perspective of multiobjective optimization, considering the simultaneous minimization of makespan and total delay. The NSGA-II algorithm was used and recognized for its efficiency in obtaining non-dominated solutions in problems with conflicting objectives. A coding based on a vector of real numbers was adopted, facilitating the implementation of evolutionary operators in combinatorial problems. The experimental results demonstrated the existence of trade-offs between the objectives, with different patterns of machine occupation depending on the prioritized criterion. The heterogeneity of processing times, a characteristic of UPMSP, accentuates this dynamic, reinforcing the importance of approaches that provide multiple alternative solutions to support decision making.

KEYWORDS. Multiobjective Optimization. *Scheduling*. Metaheuristics.

Paper topics: OMO, AD&GP, MH

1. Introdução

O desenvolvimento das áreas de engenharias, computação e economia tem sido caracterizado pelo crescente emprego de modelos e métodos de otimização como paradigmas para representação e resolução de problemas de tomada de decisão. Problemas de Programação da Produção ou *Scheduling Problems* representam uma classe de problemas de tomada de decisão muito importante na otimização. Pertencem à esta classe [Brucker, 2006; Pinedo, 2010; Fuchigami e Rangel, 2014]: sequenciamento de tarefas em máquina única, sequenciamento de tarefas em máquinas paralelas (idênticas, uniformes ou não relacionadas), *flow shop* (máquinas em série, com fluxo de tarefas unidirecional), *job shop* (máquinas em série com fluxo de tarefas não unidirecional), e uma variedade de outros problemas de *scheduling*. As áreas que encontram aplicações a estes problemas são diversas: indústria, manufatura, gerenciamento de processos em um computador, alocação de navios em portos, aviação, transporte ferroviário, produção agrícola, entre outros. Podemos otimizar os processos, realizando o planejamento do sequenciamento de tarefas, o que resulta na melhoria do controle do fluxo de produção, cumprindo os prazos de entrega e programando as tarefas de maneira a melhor utilizar os recursos disponíveis [Qi et al., 2004].

O problema de sequenciamento de tarefas consiste em alocar um conjunto de tarefas, a um conjunto de recursos, satisfazendo restrições de alocação e otimizando uma função-objetivo. Dentre as possíveis funções-objetivo do problema [Leung, 2004], podemos destacar: a minimização do *makespan*, que consiste em minimizar o maior tempo de término em um conjunto de tarefas; a minimização do tempo de atraso total, que consiste determinar uma sequência, onde a soma de atrasos das tarefas seja mínima; a minimização do tempo de espera, ou seja, o tempo que a tarefa gasta desde que chega na máquina até começar a ser executada; e a minimização do adiantamento de tarefas que determina uma sequência com o menor valor do adiantamento total; a minimização da quantidade de tarefas finalizadas após o prazo de término, etc. Os seguintes parâmetros podem ser considerados no problema: tempo de chegada, prazo de execução, ordem de precedência das tarefas, tempo de processamento, tempo de preparação, entre outros.

O problema de alocação de tarefas em máquinas paralelas não-relacionadas – *Unrelated Parallel Machine Scheduling Problem* (UPMSP) – representa uma das variantes mais complexas dentro da família dos problemas de programação da produção. Ele ocorre em ambientes onde diversas máquinas estão disponíveis para processar um conjunto de tarefas, mas com características heterogêneas, ou seja, o tempo de processamento de uma tarefa pode variar dependendo da máquina que a executa. Esse problema pode ser representado na notação de Graham [Pinedo, 2010] como, por exemplo, $R|C_{max}$, onde R indica máquinas paralelas não-relacionadas e C_{max} é o critério de minimização do *makespan*.

A natureza do UPMSP é do tipo combinatória e classificada como NP-completa. Não há algoritmo determinístico capaz de resolver o problema de forma eficiente para todos os casos, o que exige o uso de estratégias alternativas para lidar com instâncias de maior escala [Vavasis, 2001]. Em cenários práticos, múltiplos critérios geralmente são conflitantes, exigindo abordagens que considerem simultaneamente diferentes perspectivas de desempenho. Nesse contexto, metaheurísticas evolutivas mostram-se adequadas por sua capacidade de explorar espaços de soluções complexos e gerar conjuntos eficientes de alternativas.

Dessa forma, este trabalho propôs explorar diferentes enfoques – operacional e estratégico – para o UPMSP, por meio da otimização evolutiva multiobjetivo, com o intuito de oferecer ao tomador de decisão um conjunto de soluções alternativas e de qualidade, atendendo às distintas necessidades dos processos de tomada de decisão. O NSGA-II se mostra uma escolha apropriada por ser uma metaheurística clássica, amplamente reconhecida por sua robustez na resolução de problemas com múltiplos objetivos conflitantes [Deb et al., 2002]. Sua estrutura baseada em elitismo

e diversidade permite uma exploração equilibrada do espaço de soluções. Uma representação das soluções por meio de vetores de números reais [Ting et al., 2014] contribui adicionalmente para a simplicidade e generalidade da abordagem, facilitando a aplicação de operadores evolutivos mesmo em problemas de natureza combinatória, como o UPMSP.

A Seção 2 apresenta o UPMSP e os objetivos considerados, com breve revisão bibliográfica. A Seção 3 descreve a abordagem proposta para a resolução do problema. A descrição dos experimentos computacionais são apresentados na Seção 4, cujos resultados e discussões são explicitados na Seção 5. Finalmente, a Seção 6 relata as conclusões do presente estudo e possibilidades de extensão em trabalhos futuros.

2. Apresentação do Problema

O estudo e a aplicação de problemas de *scheduling* têm suas origens no campo da otimização, especialmente no contexto de programação de produção e gerenciamento de operações. O conceito de *scheduling* refere-se ao processo de alocação de recursos limitados (como máquinas, trabalhadores ou tempo) a um conjunto de tarefas ou atividades, com o objetivo de otimizar um ou mais critérios de desempenho [Johnson, 1954].

O UPMSP consiste em sequenciar e alocar $|N|$ tarefas em $|M|$ máquinas. Cada tarefa i está associada uma janela de tempo $[r_i, d_i]$ e deve ser alocada a exatamente uma máquina j . E em um determinado instante de tempo, uma máquina só pode ser ocupada por uma única tarefa. A duração do processamento de uma tarefa depende da máquina onde ela é alocada. Os seguintes parâmetros foram considerados:

M : Conjunto de máquinas, $M = \{1, 2, \dots, |M|\}$ (1)

N : Conjunto de tarefas, $N = \{1, 2, \dots, |N|\}$ (2)

r_i : Tempo de chegada da tarefa i (3)

d_i : Prazo de entrega da tarefa i (4)

p_{ij} : Tempo de processamento da tarefa i na máquina j (5)

As variáveis de decisão, que representam a solução e deverão ser obtidas através da decodificação (Seção 3) são:

x_i = tempo de início da tarefa i no horizonte de planejamento. (6)

$z_{ij} = \begin{cases} 1 & \text{se a tarefa } i \text{ é alocada à máquina } j. \\ 0 & \text{caso contrário.} \end{cases}$ (7)

No contexto da Otimização Multiobjetivo, este trabalho considerou os dois objetivos: minimização do atraso total (Definição 8) e minimização do *makespan* (Definição 9).

Minimização do atraso total: $\sum_{i \in N} a_i$ (8)

$$a_i \geq x_i + \sum_j p_{ij} z_{ij} - d_i \quad \forall i \in N$$

$$a_i \geq 0 \quad \forall i \in N$$

Minimização do *makespan*: C_{max} (9)

$$C_{max} \geq x_i + \sum_j p_{ij} z_{ij} \quad \forall i \in N$$

$$C_{max} \geq 0$$

A consideração simultânea dos objetivos de minimização do atraso total e do *makespan* trazem diferentes perspectivas à tomada de decisão. A minimização do *makespan*, que corresponde ao tempo de conclusão da última tarefa, traz enfoque operacional, pois está diretamente relacionada ao uso eficiente dos recursos disponíveis e à maximização da produtividade do sistema. Por outro lado, a minimização do atraso total, definido como a soma dos atrasos individuais das tarefas em relação aos seus prazos, está mais alinhada a uma perspectiva estratégica, pois busca reduzir o impacto de descumprimentos de prazos sobre a satisfação do cliente, penalidades contratuais ou perda de reputação. A coexistência desses dois objetivos evidencia um conflito comum em ambientes reais e justifica a adoção de abordagens multiobjetivo para apoiar decisões mais equilibradas.

Na literatura, vários modelos matemáticos ou abordagens alternativas foram propostos para este problema baseados em outros problemas clássicos de otimização, como corte e empacotamento, escalonamento, roteamento de veículos ou particionamento generalizado de conjuntos [Barbosa et al., 2016; Barbosa, 2018].

Um problema semelhante (Alocação de Navios em Portos) aos UPMSP foi resolvido por [Ting et al., 2014] com um algoritmo de Otimização por Enxame de Partículas (PSO), considerando um único objetivo. Os autores Sadeghi et al. [2014] argumentam que os problema reais de escalonamento envolvem diversos *trade-offs*, como o custo de uso de máquinas e tempo de processamento, tornando inviável a representação por um único objetivo.

De forma similar, Afzalirad e Rezaeian [2017] propuseram uma modelagem biobjetiva para o UPMSP com critérios como minimização de atrasos e balanceamento de carga entre máquinas, reforçando o aumento do interesse acadêmico por abordagens multiobjetivo. Considerando a característica NP-completa deste problema, os autores propõem duas meta-heurísticas clássicas, incluindo o NSGA-II e um algoritmo de otimização de colônia de formigas multiobjetivo (MOACO).

Lin et al. [2016] apresentam um algoritmo de alto desempenho voltado à resolução do problema de escalonamento em máquinas paralelas não-relacionadas sob uma perspectiva multi-objetivo. A proposta é baseada no algoritmo Pareto Greedy Iterated, com a introdução de uma estratégia de lista Tabu para melhorar sua eficiência. Os resultados indicam que a lista Tabu evitou movimentos de escalonamento pouco promissores.

No contexto do UPMSP, sobressaem os métodos alternativos, como algoritmos de busca local e metaheurísticas, incluindo algoritmos genéticos, *simulated annealing*, GRASP, dentre outras. Essas abordagens oferecem soluções de boa qualidade em tempo computacional aceitável, especialmente em cenários com grande número de tarefas e recursos [Durasević e Jakobović, 2023; Lin e Huang, 2021; Santos et al., 2019].

3. Abordagem Proposta

Os Algoritmos Genéticos (AGs) são metaheurísticas inspiradas nos mecanismos da evolução natural, como a seleção, o cruzamento e a mutação. Propostos por John Holland na década de 1970, os AGs tornaram-se ferramentas amplamente aplicadas na resolução de problemas de otimização complexos, especialmente aqueles classificados como NP-difíceis, para os quais métodos exatos se tornam inviáveis [Holland, 1975].

Esses algoritmos operam sobre uma população de soluções candidatas (indivíduos). A seleção favorece os indivíduos com maior aptidão, que então são combinados por meio do operador

de cruzamento, gerando novos indivíduos (descendentes). A mutação é aplicada aleatoriamente em partes dos cromossomos para introduzir diversidade genética e evitar a convergência prematura para ótimos locais. Ao final de cada geração, os indivíduos mais aptos são preservados ou substituem os menos aptos, mantendo o processo evolutivo em andamento [Eiben e Smith, 2016; Goldberg, 1989].

A abordagem multiobjetivo implementada neste trabalho foi baseado no *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) proposto por Deb et al. [2002], com a incorporação de um mecanismo de seleção elitista [Deb, 2001] fundamental para a preservação das melhores soluções ao longo das gerações. Esse mecanismo garante que soluções não-dominadas já obtidas não sejam perdidas durante o processo evolutivo [Zitzler et al., 2000]. O processo de classificação (*non-dominated ranking*) irá separar cada indivíduo em diferentes categorias (ou diferentes fronteiras) de acordo com os critérios de dominância. A distância de aglomeração (*crowding distance*), irá privilegiar pontos extremos ou pontos mais espalhados pelas diferentes fronteiras, preservando a diversidade no conjunto de soluções e evitando a concentração em regiões específicas do espaço de objetivos. Essas características tornam o NSGA-II particularmente adequado para problemas com múltiplos objetivos.

3.1. Codificação, Decodificação e Avaliação

A população inicial P_0 (da primeira geração) é formada por indivíduos aleatórios. A codificação de cada indivíduo segue a proposta de Ting et al. [2014], apresentada na Figura 1 (exemplo com 6 tarefas e 2 máquinas). A representação da solução é um vetor de ponto flutuante com tamanho $|N|$: a cada tarefa é atribuído um número real no intervalo $(0, |M|)$. A parte inteira representa a máquina na qual a tarefa está alocada; a parte fracionária é usada para sequenciar as tarefas na respectiva máquina. A adoção de uma codificação em vetor de números reais facilita a implementação de operadores genéticos e contribui para a simplicidade e flexibilidade do algoritmo.

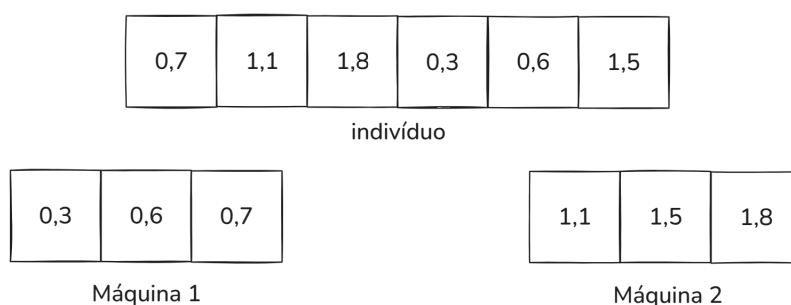


Figura 1: Codificação do indivíduo.

Os indivíduos são decodificados nas variáveis de decisão: x_i (Eq. 6) e z_{ij} (Eq. 7). As funções de avaliação adotadas neste estudo visam otimizar diferentes critérios de desempenho associados ao sequenciamento de tarefas: a minimização do atraso total (Eq. 8) considera o somatório dos atrasos individuais; a minimização do *makespan* (Eq. 9) objetiva reduzir a data de conclusão da última tarefa, sendo um critério amplamente utilizado na literatura por seu impacto direto na eficiência operacional e no tempo de ocupação dos recursos. Associa-se, a cada indivíduo k , os valores de f_1^k (atraso total, conforme Definição 8) e f_2^k (*makespan*, conforme Definição 9).

3.2. Rotina principal

O algoritmo é inicializado a partir da leitura de um arquivo que contém os parâmetros das instâncias obtidas através do gerador de dados, explicado na Seção 4: número de tarefas (N), número de máquinas (M), uma matriz contendo o tempo de processamento (p_{ij}) da tarefa i na máquina j , um vetor de tempo de chegada (r_i) da tarefa i e um vetor contendo prazo final de entrega (d_i) da tarefa i . São definidos os parâmetros específicos do NSGA-II: a quantidade de rodadas (R), o número de gerações (G) e a probabilidade de *crossover* (p_c).

A probabilidade de mutação definida depende da quantidade de máquinas (M) em cada instância e possui uma probabilidade de ocorrência proporcional a $|M|$. A mutação altera um gene do indivíduo para um valor real de 0 a M gerado por uma distribuição uniforme Deb et al. [2002]. O operador de *crossover* aplicado é o operador de n pontos. No *crossover* de n pontos, n posições aleatórias são escolhidas ao longo dos cromossomos dos pais. Os segmentos entre essas posições são alternadamente copiados de um pai ou de outro, para formar os filhos. O número de pontos é proporcional a $|N|$

Chen et al. [2012] realizaram um estudo teórico no qual concluem que populações demasiadamente grandes podem aumentar o custo computacional sem ganho adicional nos resultados; alternativamente, alterações nos operadores de seleção, *crossover* e mutação são apresentados como alternativa para obter melhores soluções. Segundo Reeves [2010], em AGs, a população inicial com tamanho mínimo igual a L , com indivíduos compostos por alelos discretos pode ser calculada conforme a Equação 10. Essa abordagem foi adotada para a definição do tamanho da população para cada instância de teste.

$$P_r = \left(1 - \left(\frac{1}{2}\right)^{|M|-1}\right)^L \quad (10)$$

onde P_r é a probabilidade de 99,99 % de sobrevivência do esquema; $|M|$ é o número de máquinas; e L é a quantidade indivíduos da população inicial.

A cada geração $g = 1, \dots, G$ do algoritmo, a seguinte sequência básica é realizada: reprodução com herança genética, introdução de variações aleatórias, promoção de competição e seleção de indivíduos de uma dada população. A população passa pelo processo de decodificação e avaliação (Seção 3.1).

Seja F a imagem, no espaço dos objetivos, do conjunto de soluções (população), com $F = \{\mathbf{f}^1, \mathbf{f}^2, \dots, \mathbf{f}^L\}$, onde cada vetor de objetivos é dado por: $\mathbf{f}^k = (f_1^k, f_2^k)$, sendo os objetivos de minimização. A solução $\mathbf{f}^{k_1}$ domina $\mathbf{f}^{k_2}$ se, e somente se:

$$\left(f_1^{k_1} \leq f_1^{k_2} \text{ e } f_2^{k_1} \leq f_2^{k_2}\right) \text{ e } \left(f_1^{k_1} < f_1^{k_2} \text{ ou } f_2^{k_1} < f_2^{k_2}\right).$$

Ou seja, o indivíduo k_1 não é pior que o indivíduo k_2 em nenhum dos objetivos e é melhor em pelo menos um. Essa definição estabelece uma nova ordem parcial utilizada no processo de seleção, no qual os novos indivíduos são selecionados baseados na classificação da população (*non-dominated ranking*), conforme exemplo da Figura 2, com $F = F_1 \cup F_2 \cup F_3$.

4. Experimentos Computacionais

O algoritmo foi implementado através da linguagem de programação Python no ambiente Google Colaboratory (versão gratuita). Para a avaliação do desempenho do método proposto, conduziram-se experimentos com quatro instâncias distintas, apresentadas na Tabela 1.

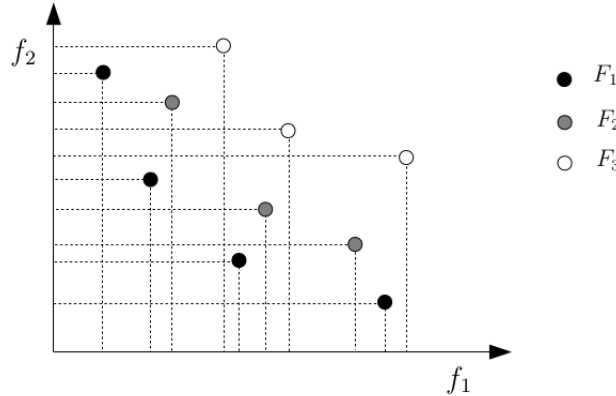


Figura 2: Classificação da população com minimização de f_1 e f_2

$ N $	$ M $	p	r	d	Classe
30	7	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	C1
40	7	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 2, 5)$	C1
100	7	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	C2
150	10	$f(x, 5, 2)$	$f(x, 2, 5)$	$f(x, 5, 2)$	C2

Tabela 1: Instâncias para teste do NSGA-II

Seguindo a mesma proposta dos trabalhos Silva et al. [2014] e Barbosa et al. [2022], desenvolvemos um gerador de dados utilizando Distribuição Beta devido à sua versatilidade. A Distribuição Beta é descrita pela função densidade de probabilidade apresentada na Equação 11:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 \mu^{\alpha-1}(1-\mu)^{\beta-1} d\mu} \quad 0 \leq x \leq 1, \alpha > 0, \beta > 0 \quad (11)$$

Os dados referentes aos tempos de processamento (p_{ij}) foram gerados por $\alpha = 5$ e $\beta = 2$, curva assimétrica à esquerda, para aumentar a probabilidade de resultar em valores mais altos. Os valores de tempos de chegada (r_i) foram gerados com $\alpha = 2$ e $\beta = 5$, curva de assimétrica à direita, com o objetivo acumular tarefas a serem processadas. No caso dos prazos de entrega (d_i), os valores de α e β foram alternados: $\alpha = 2$ e $\beta = 5$ para classe C1; $\alpha = 5$ e $\beta = 2$ para classe C2.

Os parâmetros α e β e a definição dos intervalos para a geração dos dados de tempo de processamento (p_{ij}), tempo de chegada (r_i) e prazo de entrega (d_i) das tarefas foram escolhidos conforme indicação de complexidade de resolução para estas classes, segundo o trabalho de Barbosa et al. [2022].

5. Resultados e Discussão

Os parâmetros de entrada do NSGA-II foram definidos empiricamente: probabilidade de *crossover* (p_c) igual a 0,7; número de gerações (G) igual a 250; L igual a 17 (conforme Equação 10). A natureza estocástica dos algoritmos evolutivos faz com que seja necessária a realização de várias execuções para validação dos resultados. Para cada instância foram realizadas $R = 30$ rodadas de execução independentes.

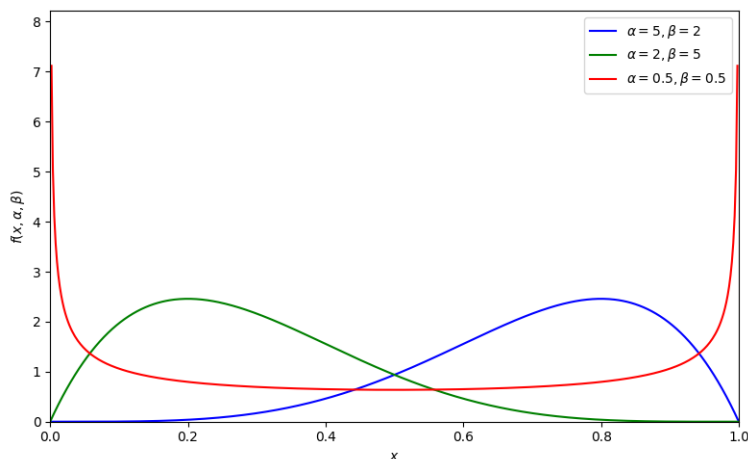


Figura 3: Codificação do indivíduo.

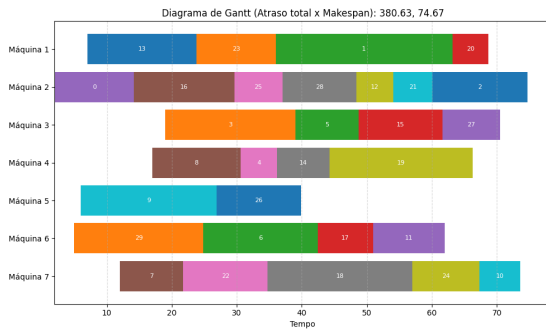
A Tabela 2 apresenta os pontos extremos do conjunto de soluções obtido para diferentes instâncias do UPMSP, considerando como objetivos a minimização do atraso total e a minimização do *makespan*. Observa-se o *trade-off* entre os dois objetivos, evidenciado pelas variações entre os valores dos pontos extremos. Por exemplo, na instância $|N| = 30$ tarefas, a minimização do atraso (de 380.63 para 180.01) implica em um aumento no atraso total (de 74.67 para 101.02). Em instâncias da classe C2, é possível obter atrasos iguais a zero ao custo de um *makespan* maior, indicando uma concentração tardia das tarefas. Esses resultados podem ser visualizados nas Figuras 4, 5, 6 e 7.

Esses resultados reforçam a importância da otimização multiobjetivo, uma vez que a escolha de uma solução depende da perspectiva adotada: operacional, com foco na utilização eficiente dos recursos e minimização do *makespan*; ou estratégica, com ênfase no cumprimento de prazos e minimização do atraso total. A existência de tais extremos evidencia a necessidade de disponibilizar ao tomador de decisão um conjunto diverso de soluções não-dominadas, permitindo maior flexibilidade e alinhamento com os objetivos organizacionais específicos.

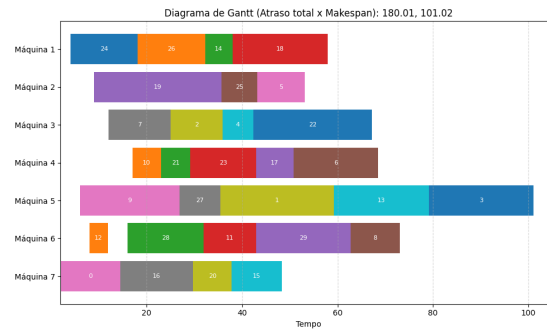
Classe	$ N $	$ M $	Ponto Extremo 1	Ponto Extremo 2
C1	30	7	(380.63, 74.67)	(180.01, 101.02)
C1	40	7	(638.48, 91.69)	(203.71, 107.00)
C2	100	7	(129.49, 162.86)	(0.00, 213.00)
C2	150	10	(405.54, 228.57)	(0.00, 242.49)

Tabela 2: Pontos extremos obtidos.

A análise dos pontos extremos revela diferenças significativas na forma como as máquinas são ocupadas, a depender do objetivo priorizado no ponto extremo. Quando o objetivo é a minimização do *makespan* (Ponto Extremo 1), o algoritmo tende a distribuir as tarefas de forma mais equilibrada entre as máquinas, buscando reduzir o tempo de conclusão da última tarefa. Isso resulta em um uso mais uniforme dos recursos disponíveis, uma vez que o gargalo de produção precisa ser evitado. Por outro lado, ao priorizar a minimização do atraso total (Ponto Extremo 2), a alocação favorece tarefas com prazos mais apertados, mesmo que isso leve a uma concentração de carga em determinadas máquinas, ou até mesmo à ociosidade em outras, desde que os atrasos se-

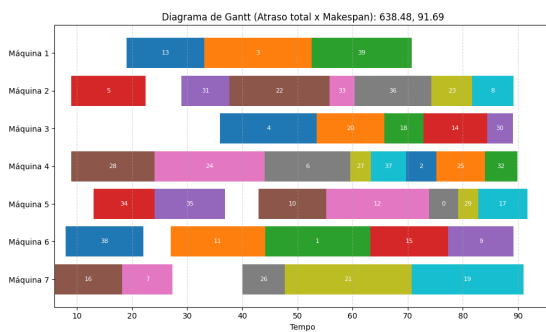


Ponto Extremo 1

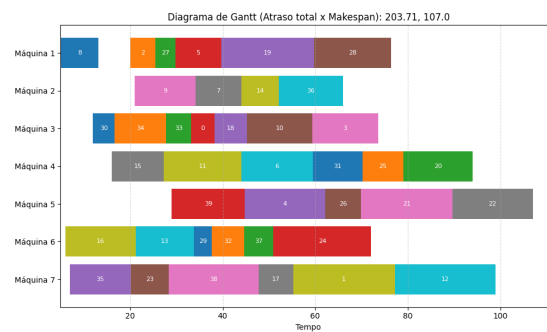


Ponto Extremo 2

Figura 4: Instância $|N| = 30$, $|M| = 7$.

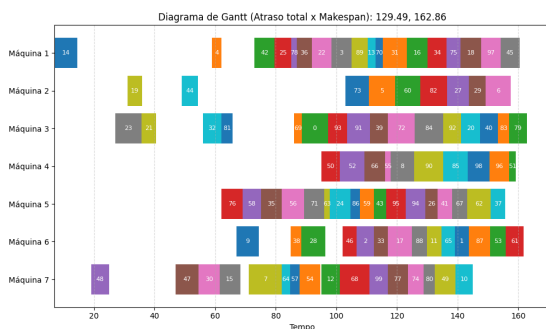


Ponto Extremo 1

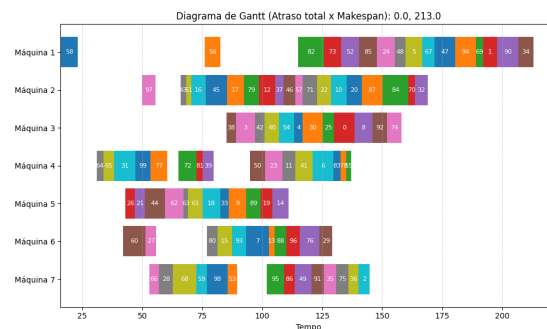


Ponto Extremo 2

Figura 5: Instância $|N| = 40$, $|M| = 7$.



Ponto Extremo 1



Ponto Extremo 2

Figura 6: Instância $|N| = 100$, $|M| = 7$.

jam minimizados. Essa diferença entre as soluções obtidas nas diferentes perspectivas é ainda mais acentuada no contexto de máquinas paralelas não-relacionadas, onde os tempos de processamento variam de acordo com a máquina na qual a tarefa é alocada. Neste tipo de problema, o algoritmo pode optar por alocar uma tarefa em uma máquina mais lenta e piorar o resultado do *makespan*, se isso for vantajoso do ponto de vista de minimização do atraso. Assim, a heterogeneidade dos tempos de processamento influencia diretamente as decisões de alocação, reforçando os efeitos dos

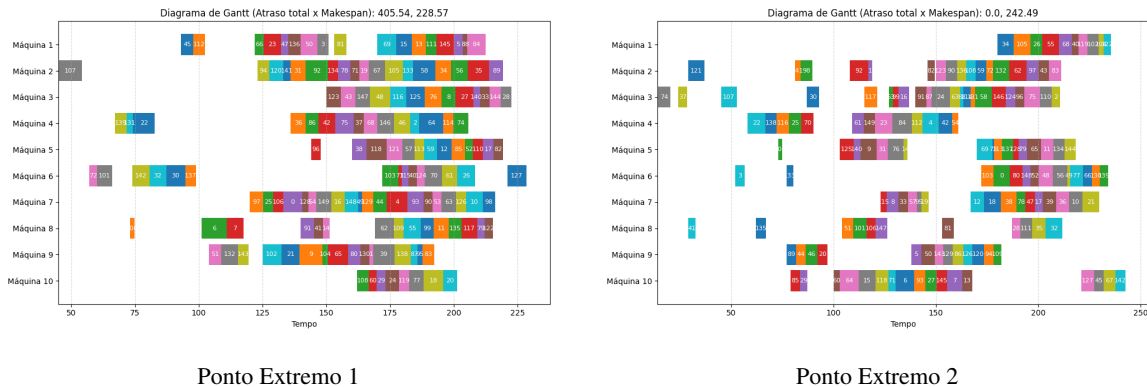


Figura 7: Instância $|N| = 150$, $|M| = 10$.

trade-offs entre os objetivos e justificando o uso de abordagens multiobjetivo para fornecer soluções alternativas com diferentes critérios operacionais e estratégicos.

6. Conclusão

Este trabalho abordou o problema de sequenciamento de tarefas em ambientes de máquinas paralelas não-relacionadas sob a perspectiva da Otimização Multiobjetivo, com aplicação do algoritmo NSGA-II. Foram considerados dois critérios: a minimização do *makespan* e a minimização do atraso total. Tais objetivos representam diferentes prioridades na gestão da produção – o *makespan* como métrica de eficiência global e aproveitamento dos recursos, e o atraso total como indicador da qualidade do atendimento a prazos. Os resultados obtidos com as instâncias de teste indicam a existência de conflitos entre os objetivos, operacional (*makespan*) e estratégico (atraso total), reforçando a necessidade de abordagens que permitam ao tomador de decisão explorar o espaço de soluções alternativas.

A representação das soluções foi realizada por meio de um vetor de números reais, o qual codifica de forma compacta e eficiente a estrutura combinatória do problema. Tal modelagem facilitou a aplicação dos operadores genéticos clássicos, além de contribuir para a generalidade e flexibilidade da implementação.

Este trabalho investigou a aplicação de um método a um problema multiobjetivo e suas implicações para a tomada de decisão de um gestor. Nesta direção, para trabalhos futuros, recomenda-se avaliar e comparar os desempenhos do método apresentado com os de metaheurísticas conhecidas, a exemplo do GRASP, MOACO, PSO, *Simulated Annealing*. Desta forma, busca-se explorar a robustez e a relevância do método em relação aos *benchmarks*.

Como direções para trabalhos futuros, sugere-se a inclusão de novos objetivos que reflitam outras dimensões relevantes do problema. Adicionalmente, destaca-se a importância de aprofundar a análise dos conflitos entre os objetivos considerados. No que se refere à eficiência do método, há espaço para o desenvolvimento de operadores específicos para o problema de *scheduling* – estratégias de inicialização da população, diferentes operadores de *crossover* e procedimentos de busca local – com o intuito de aprimorar a convergência e a diversidade do conjunto de soluções.

Dado a variedade de aplicações do problema em ambientes de serviços e manufatura, a geração de soluções alternativas e eficientes por meio da Otimização Multiobjetivo é uma abordagem promissora. É importante fornecer ao tomador de decisão um conjunto de soluções, permitindo escolhas que considerem os pontos de vista estratégico e operacional.

Referências

- Afzalirad, M. e Rezaeian, J. (2017). A realistic variant of bi-objective unrelated parallel machine scheduling problem: NSGA-II and MOACO approaches. Applied Soft Computing, 50:109–123. ISSN 1568-4946.
- Barbosa, F. (2018). Recent advances on the berth allocation problem. PhD thesis, Universidade Estadual de Campinas.
- Barbosa, F., Moretti, A. C., Azevedo, A. T., et al. (2016). A brief survey of the berth allocation problem. Pesquisa Operacional para o Desenvolvimento, 8(1):39–56.
- Barbosa, F., Rampazzo, P. C. B., de Azevedo, A. T., e Yamakami, A. (2022). The impact of time windows constraints on metaheuristics implementation: a study for the discrete and dynamic berth allocation problem. Applied Intelligence, 52(2):1406–1434.
- Brucker, P. (2006). Scheduling Algorithms. Springer-Verlag B. H., New York, 5 edition.
- Chen, T., Tang, K., Chen, G., e Yao, X. (2012). A large population size can be unhelpful in evolutionary algorithms. Theoretical Computer Science, 436:54–70.
- Deb, K. (2001). Multi-objective Optimization Using Evolutionary Algorithms. John Wiley Sons, United Kingdom, 1ªed. edition.
- Deb, K., Pratap, A., Agarwal, S., e Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. ISSN 1941-0026.
- Durasević, M. e Jakobović, D. (2023). Heuristic and metaheuristic methods for the parallel unrelated machines scheduling problem: a survey. Artificial Intelligence Review, 56(4):3181–3289. ISSN 1573-7462.
- Eiben, A. E. e Smith, J. E. (2016). Introduction to Evolutionary Computing. Springer Berlin, Heidelberg, 2 edition.
- Fuchigami, H. Y. e Rangel, S. (2014). Uma análise de estudos de casos em sequenciamento da produção. In XLVI Simpósio Brasileiro de Pesquisa Operacional, p. 159–170, Salvador - BA. SBPO.
- Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition. ISBN 0201157675.
- Holland, J. H. (1975). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press, Oxford, England.
- Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly, 1(1):61–68.
- Leung, J. Y.-T. (2004). Handbook of Scheduling: Algorithms, Models, and Performance Analysis. Chapman and Hall/CRC, United States, 4ªed. edition.
- Lin, D.-Y. e Huang, T.-Y. (2021). A hybrid metaheuristic for the unrelated parallel machine scheduling problem. Mathematics, 9(7). ISSN 2227-7390.

- Lin, S.-W., Ying, K.-C., Wu, W.-J., e Chiang, Y.-I. (2016). Multi-objective unrelated parallel machine scheduling: A tabu-enhanced iterated pareto greedy algorithm. International Journal of Production Research, 54(4):1110–1121.
- Pinedo, M. L. (2010). Scheduling: Theory, Algorithms and Systems. Springer, New York, 4 edition.
- Qi, X., Yang, J., e Yu, G. (2004). Handbook of Scheduling: Algorithms, Models, and Performance Analysis, volume 1, chapter Scheduling Problems in the Airline Industry, p. 50.1–50.20. CRC Press, Berlin, 1 edition.
- Reeves, C. R. (2010). In Gendreau, M. e Potvin, J.-Y., editors, Handbook of Metaheuristics, volume 146 of International Series in Operations Research Management Science, chapter 5, p. 109–139. Springer, New York.
- Sadeghi, S., Ariafar, S., Ghanbari, M., e N., I. (2014). An optimization approach to solve a multi-objective unrelated parallel machine scheduling problem. Applied Mechanics and Materials, 564: 585–589.
- Santos, H. G., Toffolo, T. A., Silva, C. L., e Vanden Berghe, G. (2019). Analysis of stochastic local search methods for the unrelated parallel machine scheduling problem. International Transactions in Operational Research, 26(2):707–724.
- Silva, E., Oliveira, J. F., e Wascher, G. (2014). 2dcpackgen: A problem generator for two-dimensional rectangular cutting and packing problems. European Journal of Operational Research, 237(3):846 – 856. ISSN 0377-2217.
- Ting, C.-J., Wu, K.-C., e Chou, H. (2014). Particle swarm optimization algorithm for the berth allocation problem. Expert Systems with Applications, 41(4):1543–1550.
- Vavasis, S. A. (2001). Complexity theory, p. 300–304. Springer US, Boston, MA. ISBN 978-0-306-48332-5.
- Zitzler, E., Deb, K., e Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation, 8(2):173–195. ISSN 1941-0026.