

Abordagens Exatas para o Problema de Escalonamento Robusto em Máquinas Paralelas

Heloisa Vasques da Silva¹, Alberto Locatelli², Silvio Alexandre de Araujo³, Manuel Iori²

¹UNESP, Avenida Eng. Luís Edmundo Carrijo Coube, 14-01, Vargem Limpa, Bauru-SP, Brasil

²DISMI, University of Modena and Reggio Emilia, via Amendola 2, 42122 Reggio Emilia, Itália

³UNESP, Rua Cristóvão Colombo, 2265, Jardim Nazareth, São José do Rio Preto-SP, Brasil

heloisa.vasques@unesp.br, alberto.locatelli@unimore.it,

silvio.araujo@unesp.br, manuel.iori@unimore.it

RESUMO

No Problema de Escalonamento de Máquinas Paralelas com minimização de makespan, existem um conjunto de tarefas com tempos de processamento e um conjunto de máquinas paralelas idênticas. Cada máquina pode processar no máximo uma tarefa por vez, e preempção não é permitida. O objetivo é escalonar todas as tarefas, minimizando o tempo máximo de conclusão. Em aplicações práticas, os tempos de processamento são frequentemente incertos. Este artigo aborda uma variante robusta do problema que considera tais incertezas. Adotamos o *budget uncertainty set*, limitando o número de tarefas que podem desviar de seus tempos de processamento nominais em cada máquina. Propomos duas abordagens exatas para resolver este Problema Robusto de Escalonamento de Máquinas Paralelas: uma formulação de fluxo em arcos e um procedimento de busca binária. Para avaliar o desempenho dos métodos propostos, os testamos em instâncias da literatura e comparamos os resultados com estudos anteriores, destacando as abordagens mais eficazes.

PALAVRAS CHAVE. Escalonamento de Máquinas Paralelas, Formulação de Fluxo em Arcos, Otimização Robusta.

OD – Otimização Discreta, MOI – Métodos de Otimização sob Incerteza (estocásticos e robustos)

ABSTRACT

In the Parallel Machine Scheduling Problem with makespan minimization, we are given a set of jobs with processing times and a set of identical parallel machines. Each machine can process at most one job at a time, and preemption is not allowed. The goal is to schedule all jobs while minimizing the maximum completion time. In practical applications, processing times are often uncertain. This paper addresses a robust variant of the problem that considers such uncertainties. We adopt a budgeted uncertainty set, limiting the number of jobs that may deviate from their nominal processing times on each machine. We propose two exact approaches to solve this Robust Parallel Machine Scheduling Problem: an arc-flow formulation and a binary search procedure. To evaluate the performance of the proposed methods, we tested them on benchmark instances from the literature and compared the results with previous studies, highlighting the most effective approaches.

KEYWORDS. Parallel Machine Scheduling. Arc Flow formulation. Robust Optimization.

DO – Discrete Optimization, OMU – Optimization Methods under Uncertainty (stochastic and robust)

1. Introdução

O escalonamento de tarefas em máquinas é essencial em diversas aplicações industriais, incluindo manufatura, geração de energia, transporte e saúde. Em aplicações do mundo real, os sistemas de escalonamento frequentemente enfrentam incertezas significativas devido a fatores como erros de previsão e variações na implementação. Uma fonte comum de incerteza é a duração real das tarefas a serem executadas. Considerando que os tempos de processamento dos trabalhos não são exatamente conhecidos com antecedência, mas geralmente pertencem a um intervalo, neste trabalho abordamos o problema de Escalonamento Robusto em Máquinas Paralelas (Robust Parallel Machine Scheduling — RPMS).

A Otimização Robusta visa gerar uma solução para um problema de otimização que permaneça viável independentemente da realização dos dados dentro do conjunto de incerteza [Bent-Tal et al., 2009]. Neste trabalho, adotamos o conjunto de incerteza de Bertsimas e Sim [2004], onde um parâmetro inteiro Γ é fornecido como número de trabalhos cujo tempo de processamento pode desviar de seus valores nominais.

Dadas m máquinas paralelas idênticas no conjunto de máquinas M ($i = 1, \dots, m$), que podem executar um trabalho por vez sem preempção, e um conjunto J ($j = 1, \dots, n$) de trabalhos, onde cada trabalho j possui um tempo de processamento nominal p_j e uma máxima variação $\hat{p}_j$. O objetivo é minimizar o tempo de conclusão máximo (makespan) de todos os trabalhos, independentemente de qual seja o tempo de processamento real de cada trabalho. Introduzindo a variável binária x_{ij} , que indica se o trabalho j é atribuído à máquina i , o RPMS pode ser formulado como:

$$\min C \quad (1)$$

$$\text{s.t.: } \sum_{j \in J} p_j x_{ij} + \max_{S \subset J: |S| \leq \Gamma} \sum_{j \in S} \hat{p}_j x_{ij} \leq C \quad \forall i \in M \quad (2)$$

$$\sum_{i \in M} x_{ij} = 1 \quad \forall j \in J \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in M, \forall j \in J. \quad (4)$$

O objetivo (1) minimiza o makespan. As restrições (2) garantem que o tempo total de processamento dos trabalhos atribuídos a uma máquina i não ultrapasse o makespan C . A restrição (3) assegura que cada trabalho seja atribuído exatamente a uma máquina.

2. Abordagens de Solução

2.1. Uma formulação de fluxo em arcos para o RPMS

Formulações de fluxo em arcos modelam problemas por meio de uma rede, onde as variáveis de decisão representam fluxos não negativos nos arcos [de Lima et al., 2023]. Elas têm sido aplicadas com sucesso a uma variedade de problemas de otimização, incluindo problemas de corte e empacotamento [Delorme e Iori, 2020] e problemas de escalonamento [Kramer et al., 2019].

Para modelar o RPMS, definimos um grafo acíclico direcionado $\mathcal{N} = (\mathcal{V}, \mathcal{A})$, onde cada nó $v \in \mathcal{V}$ é da forma (d, j, γ) , com d representando o tempo de processamento acumulado, j o índice do trabalho, e γ o número de trabalhos com tempos desviados. O nó fonte é $(0, 0, 0)$ e o nó sumidouro é $(C, n + 1, \Gamma)$, com os trabalhos ordenados por ordem não crescente de desvio e C sendo um limitante superior. Há três tipos de arcos: $\hat{\mathcal{A}}$ representa a atribuição de um trabalho com tempo desviado, $\bar{\mathcal{A}}$ representa a atribuição com tempo nominal, e $\mathcal{A}'$ inclui arcos de perda e conexões para o nó sumidouro. Um caminho do nó fonte ao sumidouro codifica um escalonamento viável em uma máquina. Cada arco (v, v') possui uma variável de fluxo $\varphi_{(v, v')}$ associada. Variáveis binárias $x_{(v, v_{\text{sink}})}$ indicam se o arco correspondente contribui para o makespan. O modelo de fluxo

em arcos é então definido como:

$$\min C_{max} \quad (5)$$

$$\text{s.t.: } \sum_{(v,v') \in \mathcal{A}} \varphi(v,v') - \sum_{(v,v') \in \mathcal{A}} \varphi(v,v') = \begin{cases} -m & \text{se } v = v_{source} \\ m & \text{se } v = v_{sink} \\ 0 & \text{caso contrário} \end{cases} \quad \forall v \in \mathcal{V} \quad (6)$$

$$\sum_{\substack{(v,v') \in \mathcal{A} \cup \bar{\mathcal{A}}: \\ v' \in \mathcal{V}_j}} \varphi(v,v') \geq 1 \quad \forall j \in J \quad (7)$$

$$dx_{(v,v')} \leq C_{max} \quad \forall v = (d, n, \gamma) \in \mathcal{V}_n \quad (8)$$

$$x_{(v,v')} \leq \varphi(v,v') \quad \forall v' = v_{sink} \quad (9)$$

$$x_{(v,v')} \geq \frac{\varphi(v,v')}{m} \quad \forall v' = v_{sink} \quad (10)$$

$$x_{(v,v')} \in \{0, 1\}, \varphi(v,v') \in \mathbb{Z}_+ \quad (11)$$

O objetivo (5) minimiza o makespan. As restrições (6) asseguram a conservação de fluxo. A restrição (7) garante que todos os trabalhos sejam escalonados. As restrições (8)–(10) definem o makespan e vinculam os fluxos às variáveis binárias.

2.2. Um procedimento de busca binária

Há uma relação de dualidade conhecida entre o problema escalonamento para minimizar o makespan e o problema de empacotamento (BPP) [Coffman et al., 1978; Dell’Amico et al., 2008], onde o BPP pode ser visto como um problema de escalonamento em máquinas paralelas cujo objetivo é alocar todos os trabalhos minimizando o número de máquinas necessárias, sem ultrapassar um makespan predefinido C . Utilizando essa relação entre o BPP e o problema de escalonamento em máquinas paralelas, propomos um algoritmo de busca binária para resolver exatamente o RPMS, onde, a cada iteração, uma formulação de fluxo em arcos para o BPP Robusto é resolvida. O Algoritmo 1 descreve esse procedimento. Iniciando com a capacidade $C = \lfloor (LB + UB)/2 \rfloor$, a função $Check_{RBPP}(C)$ tenta resolver um BPP Robusto. Se o modelo for viável, o limitante superior UB é atualizado para C , caso contrário, o limitante inferior LB é atualizado para $C + 1$.

Algorithm 1 Binary Search algorithm

```

1: Input: Instance
2:  $LB \leftarrow LowerBound(Instance)$ 
3:  $UB \leftarrow UpperBound(Instance)$ 
4: while  $LB < UB$  do
5:    $C \leftarrow \lfloor (LB + UB)/2 \rfloor$ 
6:   if  $Check_{RBPP}(C)$  is True then
7:      $UB \leftarrow C$ 
8:   else
9:      $LB \leftarrow C + 1$ 
10:  end if
11: end while
12: return  $UB$ 

```

3. Resultados e discussões

As abordagens propostas foram implementadas em Python, utilizando o Gurobi 11.0.1 como solver de programação inteira. Os algoritmos são denominados AF-RPMS, referente ao modelo (5)–(11), e BS-AF, correspondente ao Algoritmo 1. O tempo limite foi de 1800 segundos para o AF-RPMS e 240 segundos para cada RBPP no BS-AF. Foram testadas 480 instâncias descritas em Song e Leus [2022]. As instâncias variam conforme o número de tarefas (n), número de máquinas (m), intervalo dos tempos de processamento (I) e o parâmetro Γ .

A Tabela 1 resume os resultados dos métodos propostos comparados ao DS-B&P de Song e Leus [2022], considerado o estado da arte. Este procedimento realiza uma busca dicotômica utilizando um algoritmo branch-and-price em cada iteração. A tabela mostra o número de instâncias

resolvidas até a otimalidade e o gap médio, organizados por parâmetro. Embora o DS-B&P tenha obtido mais soluções ótimas na maioria dos casos, o AF-RPMS e o BS-AF foram eficazes, resolvendo todas as instâncias com menores tempo de processamento.

Instâncias com maiores tempo de processamento são mais difíceis pois levam à redes maiores e maiores gaps iniciais. Além disso, instâncias com menos máquinas também apresentaram maior complexidade. No geral, AF-RPMS teve melhor desempenho, resolvendo mais instâncias até a otimalidade do que BS-AF. Ambos os métodos superaram o estado da arte nas instâncias com menores tempos de processamento. Contudo, apresentaram limitações em cenários com poucos recursos (menos máquinas e mais tarefas). Assim, estudos futuros devem focar na redução do tamanho da rede nos métodos propostos.

Tabela 1: Comparação dos métodos por parâmetros

		DS-B&C		AF-RPMS		BS-AF	
		opt	gap(%)	opt	gap(%)	opt	gap(%)
I	$[1, 20]$	236	0.02	240	0.00	240	0.00
	$[1, 100]$	228	0.04	200	0.31	169	0.06
n	40	120	0.00	120	0.00	120	0.00
	60	120	0.00	120	0.00	107	0.02
	80	118	0.04	105	0.23	96	0.05
	100	106	0.08	95	0.39	86	0.05
m	5	152	0.04	133	0.32	110	0.05
	10	152	0.06	147	0.15	139	0.03
	25	160	0.00	160	0.00	160	0.00
Γ	$\lceil n/m \rceil$	233	0.04	222	0.05	201	0.03
	$\lceil n/2m \rceil$	231	0.02	218	0.26	208	0.03
		464	0.01	440	0.16	409	0.03

Referências

- Ben-Tal, A., El Ghaoui, L., e Nemirovski, A. (2009). *Robust optimization*, volume 28. Princeton university press.
- Bertsimas, D. e Sim, M. (2004). The price of robustness. *Operations Research*, 52(1):35–53.
- Coffman, E. G., Jr, Garey, M. R., e Johnson, D. S. (1978). An application of bin-packing to multi-processor scheduling. *SIAM Journal on Computing*, 7(1):1–17.
- de Lima, V. L., Iori, M., e Miyazawa, F. K. (2023). Exact solution of network flow models with strong relaxations. *Mathematical Programming*, 197:813–846.
- Dell’Amico, M., Iori, M., Martello, S., e Monaci, M. (2008). Heuristic and exact algorithms for the identical parallel machine scheduling problem. *INFORMS Journal on Computing*, 20(3): 333–344.
- Delorme, M. e Iori, M. (2020). Enhanced pseudo-polynomial formulations for bin packing and cutting stock problems. *INFORMS Journal on Computing*, 32(1):101–119.
- Kramer, A., Dell’Amico, M., e Iori, M. (2019). Enhanced arc-flow formulations to minimize weighted completion time on identical parallel machines. *European Journal of Operational Research*, 275:67 – 79.
- Song, G. e Leus, R. (2022). Parallel machine scheduling under uncertainty: Models and exact algorithms. *INFORMS Journal on Computing*, 34(6):3059–3079.