LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

Heurística Randomizada para o Problema Multiobjetivo de Máquinas paralelas não-relacionadas

Miriam Luiza de Jesus Ribeiro
Universidade Federal de Minas Gerais
Avenida Antônio Carlos, 6627, CEP 31270-901, Belo Horizonte, MG, Brasil
miriamluiza.ribeiro@gmail.com

Meline Fonseca de Castro
Instituto Tecnológico de Aeronáutica
Praça Marechal Eduardo Gomes, 50 - Vila das Acacias, São José dos Campos - SP, Brasil
castromeline@gmail.com

Martín Gomez Ravetti
Universidade Federal de Minas Gerais
Avenida Antônio Carlos, 6627, CEP 31270-901, Belo Horizonte, MG, Brasil
martin.ravetti@dep.ufmg.br

RESUMO

Este trabalho tem como objetivo propor uma heurística randomizada para o problema de máquinas paralelas não relacionadas com o objetivo de minimizar simultaneamente o makespan e o atraso ponderado total. Diante disso, foram propostas uma heurística construtiva para a solução inicial do problema, uma estratégia de troca de pares entre as tarefas para visitar outras vizinhanças e por fim, duas buscas locais para aprimorar a solução corrente. Ao considerar cada objetivo de forma individual, e comparando-os aos lower bounds encontrados na literatura, foi possível concluir que a heurística apresenta bons resultados tanto para o desvio relativo percentual quanto para a eficiência computacional.

PALAVRAS CHAVE. Máquinas paralelas não-relacionadas, heurística, otimização multiobjetivo.
AD&GP - PO na Adm. e Gestão da Produção

ABSTRACT

This paper proposes a randomized heuristic for the unrelated parallel machines problem with the objective of minimizing simultaneously the makespan and the weighted total tardiness. Given that, it was proposed a constructive heuristic for the initial solution, a pairwise strategy between the jobs to find new neighbours, and lastly two local searches to improve the current solution. Considering each objective separately and comparing them with the lower bounds available in the literature, it was possible to conclude that the heuristic produces good results both for the relative percentage deviation and for the computational efficiency.

KEYWORDS. Unrelated parallel machines, heuristic, multiobjective optimization.
AD&GP - OR in Administration and Production Management

https://proceedings.science/p/106622?lang=pt-br

1. Introdução

Com um mercado cada vez mais competitivo, o sequenciamento da produção vem sendo uma ferramenta estratégica para as empresas, no qual busca garantir que a produção seja eficaz atendendo as necessidades do mercado da melhor forma possível, como por exemplo, entrega dos produtos nos prazos estipulados, utilização eficiente dos recursos e a redução dos custos de produção.

De acordo com Watanabe et al. [2005] , o sequenciamento da produção está entre os principais e mais difíceis problemas enfrentados pelas empresas de manufatura. Este problema busca identificar quais os melhores modos de ordenação do programa de produção nas máquinas de maneira a satisfazer vários objetivos simultaneamente.

Dentre os vários problemas de sequenciamento em máquinas paralelas, destacam-se os problemas de máquinas paralelas não-relacionadas. Neste ambiente, o problema de sequenciamento consiste em determinar a máquina que será mais apropriada para cada tarefa, e então, na determinação da melhor sequência de processameno das tarefas alocadas a cada máquina [Pinedo, 2016] [Lin et al., 2011].

Os trabalhos que abordam os problemas de máquinas paralelas não-relacionadas, optam por desenvolver algoritmos e metodologias que otimizem apenas uma função objetivo. São poucos os estudos que consideram a otimização multiobjetivo e principalmente a otimização do *makespan* e atraso total ponderado ao mesmo tempo. Porém, na vida real, é comum a existência de problemas de otimização que envolvem mais de um objetivo.

Portanto, o objetivo deste trabalho consiste em propor uma heurística randomizada para o problema de máquinas paralelas não-relacionadas no qual deseja minimizar o *makespan* e o atraso ponderado simultaneamente.

Este artigo está estruturado da seguinte forma. Na Seção 2 é apresentado alguns trabalhos da literatura que estão relacionados ao problema deste artigo. Na Seção 3 é realizado a definição do problema. Na Seção 4 é apresentado a heurística proposta. Por fim, na Seção 5 e na Seção 6 são descritos os experimentos e resultados realizados e as conclusões, respectivamente.

2. Estudos Relacionados

Existem diversos problemas de sequenciamento na literatura que começaram a ser publicados por volta dos anos 1950. Os primeiros trabalhos publicados foram de autores como W.E. Smith, S.M. Johnson, e J.R. Jackson [Ravetti, 2003]. Como retratado por Allahverdi et al. [2008], houve um crescente aumento nos trabalhos publicados envolvendo metaheurísticas como solução para um problema de sequenciamento em máquinas paralelas não relacionadas.

As heurísticas propostas por Ibarra [1977] são comumente utilizadas como base de comparação para o desenvolvimento de novas heurísticas para o problema de minimizar o *makespan*. Martello et al. [1997] propõe *lower bounds* baseados na relaxação lagrangeana para o problema.

Alguns modelos clássicos na literatura, utilizam o método *Branch-and-Bound* [Roundy et al., 2005] para solução de problemas de sequenciamento. O modelo baseado na ideia de "dividir para conquistar", é utilizado por Liaw et al. [2003] para minimizar a soma ponderada dos atrasos em um problema de máquinas paralelas não-relacionadas. Liaw et al. [2003] também propõe *lower bounds* e *upper bounds* para o problema.

Com o objetivo de minimizar C_{max} + $\sum w_j T_j$ em um ambiente de máquinas paralelas, De Paula et al. [2007] utiliza a técnica de *variable neighborhood search* (VNS) e compara com três versões da metaheurística GRASP desenvolvida a partir de três algoritmos distintos de busca local proposto no artigo. O algoritmo VNS obteve melhores resultados para instâncias com mais de 60

https://proceedings.science/p/106622?lang=pt-br

tarefas e, ao considerar uma instância com 150 tarefas, o algoritmo VNS obteve soluções superiores ao do algoritmo GRASP em 95% dos casos.

Lin et al. [2011] compara o desempenho de várias heurísticas para cada objetivo individual com um algoritmo genético para o problema $R_m \mid\mid C_{max}, \sum w_j C_j, \sum w_j T_j$. Como forma de medir a robustez das heurísticas, foi o utilizado o teste da diferença mínima significativa (LSD). O algoritmo genético obteve melhores resultados para todos os três objetivos se comparado aos resultados das heurísticas.

Lin et al. [2013] estuda o problema multiobjetivo sobre máquinas paralelas não relacionadas. O objetivo é minimizar o *makespan*, minimizar o tempo total de conclusão ponderado e o atraso ponderado total, classificado como $R_m \mid\mid C_{max};\ \sum w_j C_j;\ \sum w_j T_j$. Esta pesquisa propõe duas heurísticas para cada par de critérios e um algoritmo genético para os três objetivos.

Nogueira et al. [2014] propõe três heurísticas baseadas na metaheurística GRASP para resolver o problema $R_m \mid s_{i,j,k} \mid \sum(\alpha_j E_j + \beta_j T_j)$. No trabalho de Faêda [2015] um algoritmo genético (GA) é proposto para resolver o problema de máquinas paralelas não relacionadas, com restrição de precedência e com tempos de preparação dependentes da sequência, referenciado por $R_m \mid prec, s_{ijk} \mid C_{max}$.

Nikabadi [2016] propôs um algoritmo híbrido baseado em um algoritmo genético multiobjetivo (MOGA) utilizando *simulated annealing* (SA) para um problema de sequenciamento de máquinas paralelas não relacionadas. Soper [2018] prova que uma sequência ótima pode ser encontrada em tempo polinomial para um problema de duas máquinas paralelas uniformes com o objetivo de minimizar o *makespan* de forma que seja realizada no máximo uma interrupção durante o sequenciamento.

O estudo de Vepsalainen [1987] se assemelha a um dos objetivos do problema deste trabalho. Este utiliza regras de despacho, chamada *Apparent Tardiness Cost* (ATC), em um ambiente *job-shop* baseadas nas regras de *Weighted Shortest Processing Time* (WSPT) e na regra de Menor Folga, na qual atribui a cada tarefa uma prioridade. Lin et al. [2011] adapta a regra ATC para um ambiente de máquinas paralelas não relacionadas, mesmo ambiente retratado neste trabalho.

Ao que se refere ao estudo de funções multiobjetivo semelhantes a proposta pode-se citar os trabalhos de De Paula et al. [2007] e Lin et al. [2013]. Ambos utilizam o mesmo objetivo em um mesmo ambiente de sequenciamento porém possuem restrições diferentes. De Paula et al. [2007] considera o tempo de *setup* dependente da sequência enquanto Lin et al. [2013] não considera nenhuma restrição. Portanto, o estudo de Lin et al. [2013] é o que mais se assemelha ao que será proposto neste trabalho.

3. Definição do Problema

O problema deste trabalho será baseado no sequenciamento de máquinas paralelas não relacionadas, considerado NP-difícil por Garey [1979] e terá como objetivo minimizar o *makespan* acrescido de uma penalidade pelo tempo do atraso ponderado, ou seja, uma tarefa poderá atrasar, porém isso irá acarretar em uma penalidade, levando em consideração os prazos de entrega (*due dates*) para as mesmas.

Seguindo a notação $\alpha \mid \beta \mid \gamma$, proposta por Blazewicz et al. [1983] e Graham et al. [1979] o problema pode ser definido como $R_m \mid\mid \alpha C_{max} + \beta \sum w_j T_j$. Assim, existem um conjunto de máquinas independentes, que são capazes de processar todas as tarefas mediante um certo tempo de processamento, que é dado dependendo tanto da máquina quanto da tarefa a ser processada. Existe também um tempo de entrega esperado para cada tarefa que possui um peso associado a ela, que demonstra a sua urgência quanto à entrega, e implica diretamente no objetivo final de minimizar o

https://proceedings.science/p/106622?lang=pt-br

atraso total ponderado; concomitantemente, é necessário reduzir o tempo total de processamento de todas as tarefas, de maneira que possam ser finalizadas no menor tempo possível.

Neste trabalho será proposta uma heurística randomizada para resolver o problema de sequenciamento em máquinas paralelas não relacionadas, e cada objetivo será comparado ao seu *lower bound* com o intuito de testar a efiência das soluções encontradas pela heurística.

Diante disso, algumas suposições acerca do problema tratado devem ser consideradas:

- Cada máquina pode processar apenas um *job* por vez sendo que não é possível que este *job* seja processado em máquinas diferentes simultaneamente.
- Cada *job* possui um tempo de processamento para cada máquina e ambos estão disponíveis no tempo inicial do sequenciamento.
- Cada *job* pode ser processado em qualquer máquina.
- Não existem relações de precedências entre os *jobs*.
- Existe uma data de entrega (*due date*) que cada *job* precisa ser finalizado sujeito a uma penalização pelo atraso.
- Interrupções no processo (*preemptions*), tempos de *setups*, e quebras (*breakdown*) não serão consideradas.

4. Heurística Proposta

A heurística proposta consiste em uma heurística randomizada para resolver o problema multiobjetivo de máquinas paralelas não relacionadas. Para tanto, uma solução inicial é construída através de uma heurística construtiva baseada na regra *Apparent Tardiness Cost* (ATC) proposta por Vepsalainen [1987] e modificada para o problema de máquinas paralelas por Lin et al. [2011]. Esta regra é uma regra de despacho que consiste na combinação de três regras básicas de prioridades: WSPT (*weighted shortest processing time*), EDD (*earliest due date*) e MS (*minimal slack*). A regra ATC introduz um efeito para o peso do *job* e utiliza diferentes funções para estimar o atraso associado à cada *job*. Além disso, esta também estabelece uma prioridade para o *job* com o maior valor ATC. Para a formulação do algoritmo 1 foram utilizados os seguintes parâmetros e variáveis representados na tabela 1.

O algoritmo 1 cria na linha 1 uma lista com todas as tarefas ainda não sequenciadas e na linha 2 inicia o tempo de processamento total de cada máquina como zero. Em seguida, seleciona a primeira máquina disponível na linha 4 e calcula na linha 5 o valor da variável I_{i*j} para a máquina selecionada e para todas as tarefas ainda não sequenciadas. Para este cálculo, $\overline{p}_{i*}$ é o tempo médio de processamento de todas as tarefas considerando que sejam todas executadas na máquina i^* e k é um parâmetro usualmente fixado entre $1.5 \leqslant k \leqslant 4.5$ [Vepsalainen, 1987]. Para este problema foi considerado $k = 2$. Na linha 6 a tarefa a ser sequenciada é selecionada e nas linhas 8 e 10 a máquina na qual a tarefa será alocada é escolhida. Na condição em 8 é possível processar a tarefa em alguma máquina sem que haja atraso e em 10 a tarefa é processada, porém sujeita a um atraso. Em seguida na linha 12, a tarefa j^* é alocada na máquina i^{**}, na linha 13 o tempo de processamento total da máquina i^{**} é atualizado e na linha 14 a tarefa j^* sequenciada é excluída da lista U. Esse procedimento se repete até que todas as tarefas tenham sido alocadas em alguma máquina.

O segundo passo da heurística consiste na seleção de uma nova solução a partir da solução inicial obtida através da técnica de troca de pares. Este processo seleciona as cinco melhores

Algoritmo 1: Heuristica_construtiva

1 $U \leftarrow \{1, ..., n\}$;
2 $t_{i\,\forall\, i=(1,...,m)} \leftarrow 0$;
3 **while** *(*$U \neq \varnothing$*)* **do**
4 $\quad i^* \leftarrow$ *primeira máquina disponível*;
5 $\quad I_{i*j\,\forall\, j\in U} \leftarrow \frac{w_j}{p_{i*j}} \exp\left(-\frac{max(d_j - p_{i*j} - t_{i*}, 0)}{K\overline{p}_{i*}}\right)$;
6 $\quad j^* \leftarrow max_{j\in U} I_{i*j}$;
7 $\quad$ **if** $(t_i + p_{ij*} - d_{j*} \leqslant 0\ \forall\ 1 \leqslant i \leqslant m)$ **then**
8 $\quad\quad i^{**} \leftarrow$ *máquina i min*$\{p_{ij*} \mid t_i + p_{ij*} - d_{j*} \leqslant 0, 1 \leqslant i \leqslant m\}$;
9 $\quad$ **else**
10 $\quad\quad i^{**} \leftarrow$ *máquina i min*$\{t_i + p_{ij*} - d_{j*}\ \forall\ 1 \leqslant i \leqslant m\}$;
11 $\quad$ **end**
12 $\quad j^*$*é sequenciado na próxima posição disponível na máquina* i^{**};
13 $\quad t_{i**} \leftarrow t_{i**} + p_{i**j*}$;
14 $\quad U \leftarrow U \backslash \{\text{j*}\}$;
15 **end**

Tabela 1: Parâmetros e variáveis do Algoritmo 1.

Parâmetros:		**Variáveis:**	
m	Número de máquinas paralelas.	j^*	*Job* sequenciado na próxima posição disponível na máquina i^{**}.
n	Número total de *jobs*	i^*	Primeira máquina disponível.
U	Conjunto de *jobs* não sequenciados.	i^{**}	Máquina na qual o *job* j^* é sequenciado.
w_j	Penalidade de atraso do *job* j.	I_{i*j}	Índice de prioridade.
p_{i*j}	Tempo de processamento dos *jobs* j na máquina i^*.	t_i	Soma dos tempos de processamento dos *jobs* que já foram sequenciados nas máquinas i.
p_{ij*}	Tempo de processamento do *job* j^* nas máquinas i.	t_{i*}	Soma dos tempos de processamento dos *jobs* que já foram sequenciados na máquina i^* .
p_{i**j*}	Tempo de processamento do *job* j^* na máquina i^{**}.	t_{i**}	Soma dos tempos de processamento dos *jobs* que já foram sequenciados na máquina i^{**}.
d_j^*	Data de entrega do *job* j.	$\overline{p}_{i*}$	Tempo médio de processamento de todas as tarefas considerando que sejam todas executadas na máquina i^*.
d_{j*}	Data de entrega do *job* j^*.		
K	Parâmetro fixado como sendo 2.		

soluções aprimorantes obtidas com a técnica de troca de pares e retorna uma das soluções escolhida de forma aleatória. Entretanto, caso não exista solução aprimorante, este retornará a solução obtida através da heurística construtiva (algoritmo 1).

Os próximos passos da heurística, representado no algoritmo 2, consistem em buscas locais para aprimorar o resultado obtido através da troca de pares. A função *Busca Random*, na linha 5, utiliza uma busca local na qual é selecionada um *job* e uma máquina aleatória, sendo que o *job* não pode estar sequenciado na máquina sorteada. Em seguida, o *job* é excluído da máquina a qual este pertence e busca-se a melhor solução através de todas as possibilidades de inserção do *job* na máquina selecionada. Este processo ocorre até que o critério de parada seja alcançado.

A quarta etapa da heurística, na linha 7 do algoritmo 2, consiste em uma nova busca local, com o objetivo de se obter uma melhor solução através da troca de todos os *jobs* sequenciados na máquina com o maior tempo de processamento total e o menor tempo de processamento total. Diante disso, todas as possibilidades de troca entre as duas máquinas são testadas para que se

https://proceedings.science/p/106622?lang=pt-br

obtenha a solução com melhor custo, isto é realizado pela função *Busca Local*.

A última etapa da heurística utiliza novamente a função *Busca Random*, com o diferencial de que a solução inicial será dada pela função *Busca Local*.

O algoritmo 2 representa os passos da Heurística Randomizada proposta para o problema de máquinas paralelas não relacionadas, através da utilização de todas as funções mencionadas acima. Cada uma das dez iterações, linhas 2 à 13 consiste em criar uma solução inicial (linha 3); sortear uma dentre as melhores soluções aprimorantes advindas da troca de pares em cada máquina (linha 4); realizar uma busca local aleatória com o intuito de retirar tarefas de uma máquina ao passo que insere estas tarefas em outra máquina de forma a melhorar o custo da solução corrente (linha 5); utilizar uma busca local exaustiva para liberar máquinas ocupadas e ocupar máquinas ociosas (linha 7); e por fim realizar novamente uma busca local aleatória (linha 9). Ao final, o melhor custo obtido é retornado (linha 14).

Algoritmo 2: Heuristica_randomizada

```
1  iteracao ← 0;
2  while iteracao < 10 do
3  |  sequenciamento_1 ← Heuristica_construtiva(arquivo);
4  |  sequenciamento_2 ← Pair_wise(sequenciamento_1);
5  |  sequenciamento_3 ← Busca_random(sequenciamento_2);
6  |  while aprimora custo do
7  |  |  sequenciamento_4 ← Busca_local(sequenciamento_3);
8  |  end
9  |  sequenciamento_5 ← Busca_random(sequenciamento_4);
10 |  if aprimora custo then
11 |  |  custo_final ← funcao_objetivo(sequenciamento_5);
12 |  end
13 end
14 return custo_final;
```

5. Experimentos e Resultados

Os resultados computacionais obtidos através da Heurística Randomizada proposta neste trabalho serão mostrados nesta seção. O código foi implementado na linguagem C++ e os testes foram realizados em uma máquina com processador Intel Core i5 de 1.70 GHz, memória RAM de 6,00 GB e sistema operacional Windows 64 bits. Para execução, foi utilizada a versão 5.1.0 do compilador GCC.

As instâncias utilizadas foram geradas através do *software* R de forma aleatória com base em Lin et al. [2011]. Para o tempo de processamento, foram gerados valores aleatórios através de uma distribuição uniforme variando entre [1, 100]. O peso de cada tarefa foi obtido de forma aleatória através de uma distribuição uniforme variando entre [1, 10]. Os valores das datas de entrega foram obtidos da forma proposta por Potts [1982] através da distribuição uniforme variando entre [P(1-T-R/2), P(1-T+R/2)], no qual $P = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}}{m}$, R é um parâmetro chamado fator médio do atraso e T é um parâmetro chamado variação relativa dos *due dates*. Para as instâncias do problema foram utilizados os valores de R = 0.9 e T = 0.6.

Ao todo 60 instâncias foram geradas para o problema. Destas, 30 representam problemas com instâncias pequenas, e possuem sempre 20 tarefas e 4 máquinas. As outras 30 instâncias representam problemas com instâncias grandes, e possuem sempre 100 tarefas e 10 máquinas.

https://proceedings.science/p/106622?lang=pt-br

Não foi encontrado na literatura nenhum *lower bound* (LB) do problema multiobjetivo estudado. Porém, existem na literatura *lower bounds* para cada função objetivo separadamente. Sendo assim, a análise dos resultados será feita através da comparação individual de cada função objetivo com o *lower bound* referente, ou seja, os valores de α e β da função multiobjetivo, $\alpha C_{max} + \beta \sum w_j T_j$, será:

- α = 1 e β = 0 para avaliar o *makespan* com o LB proposto por Martello et al. [1997] através de uma relaxação lagrangeana;
- α = 0 e β = 1 para avaliar o atraso ponderado com o LB proposto por Liaw et al. [2003] baseado na solução de um problema de atribuição;

A métrica adotada para realizar a comparação da heurística, tem o objetivo de verificar o quão próximo o resultado se encontra do *lower bound*. Deste modo, calcula-se para cada instância o desvio relativo percentual (RPD) da solução encontrada pela heurística randomizada (HR) em relação ao LB.

$$Desvio\ Relativo\ Percentual\ (RPD) = \frac{(HR - LB)}{LB} \times 100 \quad (1)$$

O cálculo dos *lower bounds* encontrados na literatura foram implementados na linguagem de modelagem AMPL e resolvidos através do pacote computacional CPLEX 12.8.

5.1. Resultados Computacionais

Para ter um maior controle na geração dos dados, cada instância foi executada para cinco sementes diferentes, e o valor utilizado para o cálculo do RPD será a média dos resultados.

Para avaliar a performance da heurística randomizada, foram elaborados dois *boxplots* para analisar os valores do RPD referente ao *makespan* e referentes ao atraso ponderado total (Figura 1).

Avaliando inicialmente o *makespan*, é possível inferir que a heurística randomizada apresenta melhores resultados para as instâncias menores, pois estas possuem a mediana igual a 2,17%, enquanto para as instâncias grandes a mediana é de 22,91%. Além disso, a heurística apresenta menor variabilidade para as pequenas instâncias em comparação as instâncias maiores.

Quando analisados os valores do desvio relativo percentual para o problema do atraso ponderado total, a HR em relação ao tamanho da instância é mais robusta. Para as pequenas instâncias o valor do RPD para metade dos resultados é igual a 0%, já para as grandes instâncias metade dos resultados estão abaixos de 2,69%.

Independente do tamanho das instâncias, a heurística proposta apresenta resultados muito próximos em relação ao *lower bound* da literatura, demonstrando assim, que a heurística possui uma excelente performance para o problema do atraso ponderado total. É possível perceber também, que para as instâncias menores, a heurística possui uma maior variabilidade quando se comparado com as grandes instâncias.

Foi elaborado a tabela 2 para avaliar o comportamento do tempo computacional da heurística em relação a variação do α e β da função multiobjetivo. Os valores de α e β variam entre 0 e 1, sendo que $\alpha + \beta = 1$. Para tanto, os valores de α foram variados de 0 a 1 com incremento de 0.1 em cada iteração. Assim, ocorreram ao todo 11 combinações distintas referentes aos valores de α e β, para cada uma das cinco sementes utilizadas.

Desta forma, a tabela 2 informa que o tempo computacional para pequenas instâncias é em média 4.22 segundos com um desvio padrão de 0.20, enquanto que para as grandes instâncias

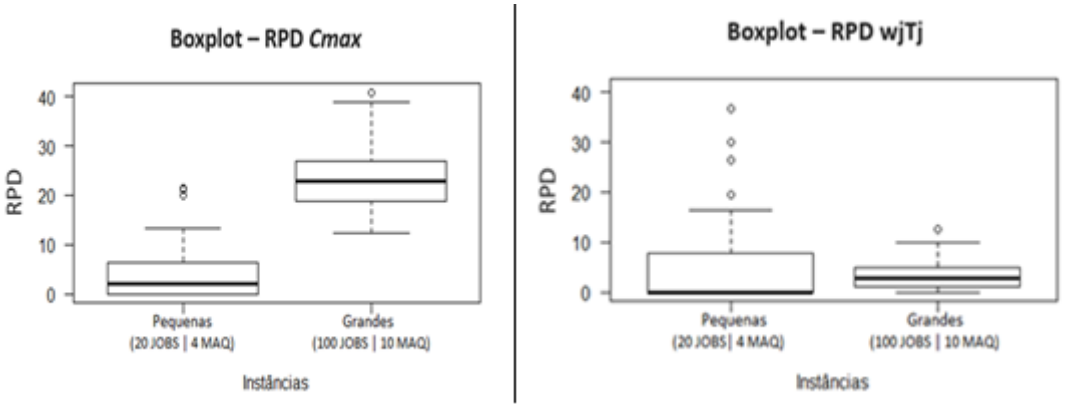


Figura 1: *Boxplot* do RPD - C_{max} e do w_jT_j. Fonte: Elaborada pelos autores.

o tempo é superior, com uma média de 85.58 segundos com um desvio de 5.10. Ou seja, pode ser observado que mesmo para o grupo de grandes instâncias a HR apresenta um bom desempenho computacional.

Tabela 2: Tempo médio computacional (segundos)

Instâncias	**Médias**	**Desvio Padrão**
Pequenas (20 *jobs* | 4 Maq)	4.22	0.20
Grandes (100 *jobs* | 10 Maq)	85.58	5.10

A solução da HR para o problema multiobjetivo de máquinas paralelas não relacionadas não apresenta uma única solução, mas sim, um conjunto de soluções que devem ser analisadas pelo tomador de decisão para compreender qual o melhor cenário necessário para a situação em que a empresa está vivenciando. Sendo assim, é necessário criar uma curva de pareto que demonstra o conjunto de soluções eficientes pertencentes ao problema. Portanto, para análise das soluções obtidas através da curva de Pareto as instâncias foram separadas em dois grupos: grupo de instâncias A e B. Para uma das instâncias referente ao grupo de instâncias A foi elaborado a tabela 3 e a figura 2 e para um exemplo do grupo de instâncias do grupo B foi elaborado a tabela 4 e a figura 3.

Para o grupo de instância A percebe-se que as soluções obtidas para o atraso ponderado possuem os mesmos valores, ou seja, a solução dominante será aquela que apresentará o menor valor para o Cmax. Já as instância do grupo B apresentam valores diferentes tanto para o atraso ponderado quanto para o *makespan*, obtendo assim, um conjunto de soluções.

Este conjunto de soluções eficientes auxiliará o tomador de decisão na escolha do melhor cenário dentre todos propostos pelo otimizador. Assim, como proposto no método a *posteriori*, a escolha é feita após o decisor ter conhecimento de todo o conjunto de soluções. Se para um determinado momento empresarial o atraso é considerado um fator mais importante do que o tempo final de processamento de todos os pedidos, então o tomador de decisão deverá priorizar soluções para as quais os valores de β seja superior aos valores de α.

Um exemplo prático para esta situação, considerando a unidade de tempo do problema em dias, ocorre quando a empresa precisa entregar uma ordem de produção em no máximo 62 dias. Assim, a penalidade para que se atinja esse objetivo será de no mínimo 43 unidades, conforme é

Tabela 3: Dados curva de pareto - Instância: Grupo A

α	β	αCmax	$\beta\sum$wjTj	Z	Cmax	$\sum$wjTj
0	1	0.00	108.00	108.00	134.00	108.00
0.1	0.9	13.40	97.20	110.00	134.00	108.00
0.2	0.8	20.28	86.40	106.00	101.40	108.00
0.3	0.7	30.60	75.60	105.60	102.00	108.00
0.4	0.6	40.08	64.80	104.20	100.20	108.00
0.5	0.5	53.10	54.00	107.00	106.20	108.00
0.6	0.4	61.56	43.20	104.40	102.60	108.00
0.7	0.3	70.70	32.40	102.60	101.00	108.00
0.8	0.2	83.52	21.60	104.60	104.40	108.00
0.9	0.1	90.90	10.80	101.00	101.00	108.00
1	0	100.40	0.00	100.40	100.40	108.00

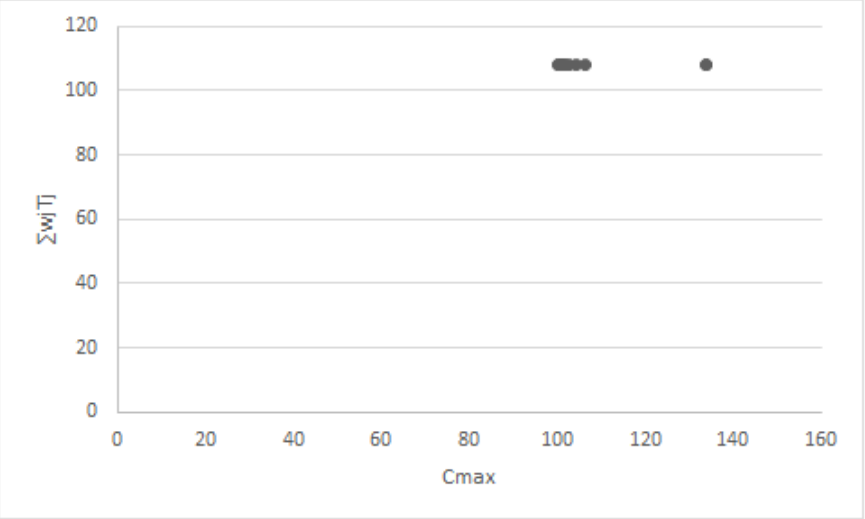


Figura 2: Curva de Pareto - Instância: Grupo A. Fonte: Elaborada pelos autores.

mostrado na tabela 3. Porém, pode ser que a empresa consiga postergar a entrega do pedido em 8 dias, o que reduziria a penalidade em aproximadamente 25%.

6. Conclusão

O problema de máquinas paralelas não relacionadas vem sendo muito estudado nos últimos tempos, onde diversos autores contribuíram com a elaboração de algoritmos que pudessem resolver o problema em um tempo computacional viável e apresentando soluções de boa qualidade. Porém, quando este problema integra mais de um objetivo a quantidade de conteúdo disponível é muito inferior, trazendo dificuldades para solucionar diversos problemas reais, pois estes, apresentam na maioria das vezes mais de um objetivo.

Sendo assim, existe tanto um interesse acadêmico, quanto empresarial, para obter novos métodos de resolução para estes tipos de problemas. Assim, este trabalho propôs uma heurística para resolver o problema de máquinas paralelas não relacionadas visando a minimização do tempo máximo de conclusão do sequenciamento (*makespan*) e do atraso ponderado total.

Tabela 4: Dados curva de pareto - Instância: Grupo B

α	β	αCmax	$\beta\sum$wjTj	Z	Cmax	$\sum$wjTj
0	1	0.00	5374.00	5374.00	206.00	5374.00
0.1	0.9	18.72	4783.14	4801.20	187.20	5314.60
0.2	0.8	37.76	4302.24	4339.60	188.80	5377.80
0.3	0.7	56.58	3759.28	3815.20	188.60	5370.40
0.4	0.6	72.48	3221.28	3293.20	181.20	5368.80
0.5	0.5	92.40	2684.90	2776.80	184.80	5369.80
0.6	0.4	96.48	2159.28	2255.20	160.80	5398.20
0.7	0.3	116.20	1619.64	1735.40	166.00	5398.80
0.8	0.2	128.96	1084.96	1213.40	161.20	5424.80
0.9	0.1	141.30	542.40	683.40	157.00	5424.00
1	0	154.40	0.00	154.40	154.40	5919.80

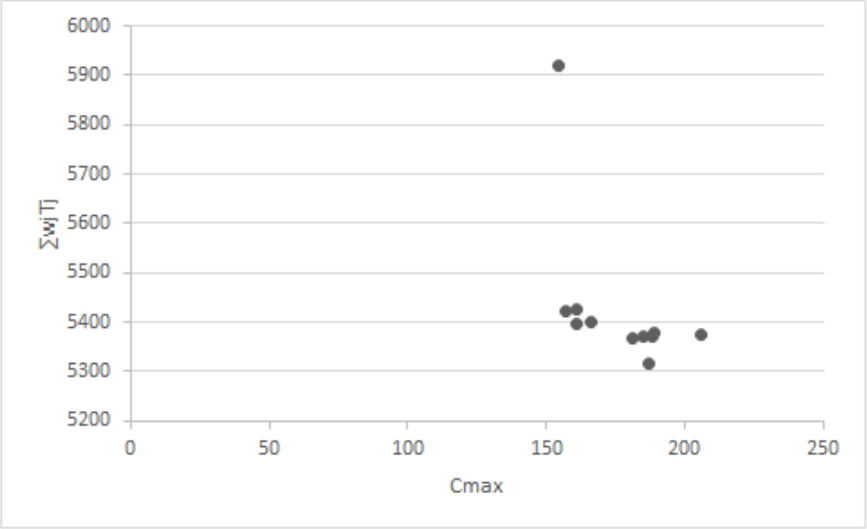


Figura 3: Curva de Pareto - Instância: Grupo B. Fonte: Elaborada pelos autores.

A heurística randomizada elaborada neste trabalho apresentou ótimos resultados, principalmente para os problemas de atraso ponderado, indicando um RPD médio de 3.53% para as pequenas instâncias e um RPD médio de 7.33% para as grandes instâncias. Apesar dos resultados para o problema de minimização do *makespan* para grandes instâncias não terem sido muito satisfatórios, com um RPD médio de 22.91%, quando se leva em consideração a influência das duas funções objetivo, ocorre o equilíbrio do RPD.

Em relação aos objetivos tratados neste trabalho, o *makespan* é utilizado em casos nos quais há uma maior preocupação na redução do tempo total de processamento do último *job* concluído no sistema, o que afeta diretamente os custos operacionais incorridos durante a produção. Ao passo que o atraso ponderado total está diretamente relacionado ao nível de atendimento ao cliente.

Ao apresentar um conjunto de soluções que atendem aos diferentes objetivos, é disponibilizado ao gestor, alternativas para auxiliar a sua tomada de decisão. Assim, é possível escolher àquela solução que mais se adapta a realidade da empresa. Em termos práticos, a diminuição do

makespan pode incorrer em um aumento no atraso total, o que nem sempre representa um cenário promissor em um ambiente empresarial, por isso, é importante considerar simultaneamente esses dois objetivos.

Portanto, a maior vantagem em se utilizar uma metodologia multiobjetivo é a possibilidade de se obter um modelo que se adapta à diferentes situações práticas, vivenciadas cotidianamente em um ambiente empresarial.

Para os trabalhos futuros, pretende-se avaliar a contribuição de métodos que auxiliem a redução do RPD para o *makespan*, devido a menor eficiência da heurística para este tipo de problema se comparado ao problema do atraso ponderado. Outra sugestão para trabalhos futuros, é avaliar a inclusão de outras estratégias de vizinhança. Por fim, para os trabalhos futuros, sugere uma busca para encontrar *lower bounds* para a função multiobjetivo.

Referências

Allahverdi, A., Ng, C., Cheng, T. E., e Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European journal of operational research*, 187(3):985–1032.

Blazewicz, J., Lenstra, J. K., e Kan, A. R. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete applied mathematics*, 5(1):11–24.

De Paula, M. R., Ravetti, M. G., Mateus, G. R., e Pardalos, P. M. (2007). Solving parallel machines scheduling problems with sequence-dependent setup times using variable neighbourhood search. *IMA Journal of Management Mathematics*, 18(2):101–115.

Faêda, F. M. (2015). Métodos de resolução do problema de sequenciamento em máquinas paralelas não-relacionadas com restrições de precedência e tempos de preparação. *Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Viçosa*.

Garey, M. R. (1979). A guide to the theory of np-completeness. *Computers and intractability*.

Graham, R. L., Lawler, E. L., Lenstra, J. K., e Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Annals of discrete mathematics*, volume 5, p. 287–326. Elsevier.

Ibarra, C. E., Oscar H e Kim (1977). Heuristic algorithms for scheduling independent tasks on nonidentical processors. *Journal of the ACM (JACM)*, 24(2):280–289.

Liaw, C.-F., Lin, Y.-K., Cheng, C.-Y., e Chen, M. (2003). Scheduling unrelated parallel machines to minimize total weighted tardiness. *Computers and Operations Research*, 30(12):1777–1789.

Lin, Y.-K., Fowler, J. W., e Pfund, M. E. (2013). Multiple-objective heuristics for scheduling unrelated parallel machines. *European Journal of Operational Research*, 227(2):239–253.

Lin, Y.-K., Pfund, M. E., e Fowler, J. W. (2011). Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems. *Computers & Operations Research*, 38(6):901–916.

Martello, S., Soumis, F., e Toth, P. (1997). Exact and approximation algorithms for makespan minimization on unrelated parallel machines. *Discrete applied mathematics*, 75(2):169–188.

Nikabadi, R., M e Naderi (2016). A hybrid algorithm for unrelated parallel machines scheduling. *International Journal of Industrial Engineering Computations*, 7(4):681–702.

Nogueira, J. P., Arroyo, J. E. C., Villadiego, H. M. M., e Gonçalves, L. B. (2014). Hybrid grasp heuristics to solve an unrelated parallel machine scheduling problem with earliness and tardiness penalties. *Electronic Notes in Theoretical Computer Science*, 302:53–72.

Pinedo, M. L. (2016). *Scheduling: theory, algorithms, and systems*. Springer.

Potts, L., Ci N e Van Wassenhove (1982). A decomposition algorithm for the single machine total tardiness problem. *Operations Research Letters*, 1(5):177–181.

Ravetti, M. (2003). *Problemas de seqüenciamento com máquinas paralelas e tempos de preparação dependentes da seqüência*. PhD thesis, Dissertação (Mestrado em Ciência da Computação)–Belo Horizonte-MG, Universidade Federal de Minas Gerais–UFMG.

Roundy, R., Chen, D., Chen, P., Cakanyildirim, M., Freimer, M. B., e Melkonian, V. (2005). Capacity-driven acceptance of customer orders for a multi-stage batch manufacturing system: models and algorithms. *Iie Transactions*, 37(12):1093–1105.

Soper, V. A., Alan J e Strusevich (2018). Schedules with a single preemption on uniform parallel machines. *Discrete Applied Mathematics*.

Vepsalainen, T. E., Ari PJ e Morton (1987). Priority rules for job shops with weighted tardiness costs. *Management science*, 33(8):1035–1047.

Watanabe, M., Ida, K., e Gen, M. (2005). A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem. *Computers & Industrial Engineering*, 48(4):743–752.