

O uso do Mixed-Integer e Constraint Programming para a resolução do Job Shop Scheduling Problem

Alvaro Neuenfeldt Júnior

Recibido: 13 de Febrero de 2019 / Aceptado: 26 de Abril de 2019

Resumo

Independente do processo industrial, o estudo sobre a alocação de recursos produtivos é atualmente um dos temas mais abordados cientificamente, principalmente quando envolve a busca pela gestão de forma mais eficiente e dinâmica às exigências do mercado ao qual a empresa está inserida. Para tanto, o presente artigo tem por objetivo comparar o desempenho de três técnicas de otimização ao contexto do Job Shop Scheduling Problem (JSSP), sendo uma baseada em modelagens Mixed-Integer Programming (MIP) e outras duas resolvidas com base nos conceitos do Constraint Programming (CP), por meio da utilização de 82 instâncias disponibilizadas pela biblioteca digital OR Library. Como resultados, foi possível verificar que a versão padrão do CP disponibilizado pelo software CPLEX Optimization Studio é o mais eficiente para encontrar soluções ótimas, conforme benchmark realizado com estudos anteriormente publicados cientificamente.

Palavras-Chaves

Job shop schedule problem; Mixed-integer programming; Constraint programming; Linear optimization.

1. Introdução

O fato de todo sistema produtivo necessitar o processamento de um conjunto de tarefas em máquinas é um fato concensual na literatura atual, de modo que a sincronização (ou sequenciamento) do tipo Job Shop Scheduling Problem (JSSP) tem por pressuposto a busca por melhorar a eficiência produtiva em relação ao tempo necessário para que os produtos finais sejam entregues, de maneira a evitar punições e multas por atrasos ou avanços incontrolados no fluxo do sistema de manufatura (Applegate et al., 1991; Araújo et al., 2008; Matteo et al., 2008; Pezzella et al., 2008).

O JSSP pode ser encontrado por meio de diversas características, adotando números iguais ou diferentes de tarefas, produtos ou ordens de produção, que possuem distintas rotas e consumos de matéria-prima e insumos, e necessitam ser processados (ou alocados) em máquinas ou postos operacionais. A medida do seu nível de ajustamento pode ser realizada por meio de indicadores como, por exemplo, o tempo necessário para cumprir todas as tarefas agendadas ou a minimização do tempo desperdiçado entre máquinas e tarefas. Variantes do problema notoriamente encontradas na literatura estabelecem o uso de restrições de disponibi-

lidade de produtos, além da possibilidade de não inclusão de buffers intermediários entre tarefas ou se algum da adição de algum tipo de bloqueio físico ou de fluxo de materiais entre tarefas, conforme o Flow Shop Scheduling Problem (FSSP) (Aggoune, 2004; Andrés e Maheut, 2018; Bautista e Alfaro, 2018).

A resolução do problema envolve a simulação de uma série de rotinas, sendo uma das metodologias mais exploradas a Mixed-Integer Programming (MIP), que tem por finalidade encontrar a solução ótima do problema através da submissão de restrições as variáveis de decisão, em relação a uma função objetivo (no caso de minimização). Já o Constraint Programming (CP) busca resolver o problema por meio da redução dos domínios das variáveis em relação as restrições predispostas, tendo como principal vantagem a tendência em buscar a solução ótima de forma mais eficaz ao MIP, através da eliminação de ramos cujos valores são inviáveis, a fim de evitar extensivo aprofundamento desta exploração.

Assim, a presente pesquisa tem por objetivo comparar o desempenho de três técnicas de otimização ao contexto do JSSP, através de modelagens MIP e CP, identificando qual possui maiores condições de ajustar com melhor qualidade as soluções geradas para cada instância. Para tanto, o restante do artigo está organizado em seções, de modo que a seguir é proposta a definição conceptual do problema. A seção 3 conduz para a verificação dos procedimentos metodológicos para a obtenção, na seção 4, dos resultados obtidos. Por fim, pretende-se com a seção 5 demonstrar algumas conclusões aferidas sobre o estudo.



Alvaro Neuenfeldt Júnior *
alvaroj.eng@gmail.com
<http://orcid.org/0000-0002-6492-6342>

*Núcleo de Inovação e Competitividade. Programa de Pós-Graduação em Engenharia de Produção. Universidade Federal de Santa Maria. Avenida Roraima, 1000, prédio 7, sala 302. 97105440 Santa Maria. Brasil.

2. Definição do problema

O JSSP tradicional envolve o cumprimento de uma série de operações que podem ser divididos em tarefas, a serem executadas em um número finito de máquinas. A solução consiste em determinar o ordenamento ideal em que cada tarefa deve ser alocada nestas máquinas, sem que haja sobreposição entre elas. Como exemplo, três máquinas são utilizadas para manufaturar três ordens de operação, com tempos de processamento distintos, onde o valor do período necessário para que todas estas sejam concluídas é equivalente ao valor total da última tarefa manufaturada (makespan).

Considerado como um problema NP-Difícil, o JSSP tem uma probabilidade maior de atingir uma solução ótima quando as instâncias são consideradas de pequeno tamanho, devido ao menor esforço computacional demandado para a resolução completa do algoritmo, independente se forem utilizados métodos cuja base está formatada em CP ou MIP (Martin e Shmoys, 1996; Naderi et al., 2009; Yilmaz et al., 2014; Neuenfeldt Júnior et al., 2019), considerando um número finito de M máquinas ($j=\{1,\dots,k,\dots,n\}$) e O_{ji} operações ($O_{j1},\dots,O_{jh},\dots,O_{jm}$), subdivididas em J tarefas ($i=\{1,\dots,h,\dots,m\}$) a serem alocadas de maneira uniforme por turnos, possuindo tempos de processamento $p(i)\in\mathbb{N}$, sem que haja possibilidade de ocorrer interrupções (Vaessens et al., 1994; Runarsson et al., 2012). A função objetivo é definida pelo valor do makespan (C_{max}), calculado a partir da medida do tempo necessário para completar todas as tarefas. Quanto menor o seu valor, maior a tendência de que a modelagem esteja ajustada eficazmente (Kashan et al., 10; Yilmaz et al., 2014).

Sob o contexto dos métodos MIP para a resolução do JSSP, uma das abordagens mais notórias na literatura é o modelo disjuntivo (MIP-DJM) apresentado por Manne (1960), que possui como premissa o uso de uma variável binária z_{jik} para rejeitar (0) ou aceitar (1) a possibilidade de precedência de uma tarefa j em relação a outra k, a uma máquina específica i, de maneira que $j < k$, conforme a função de minimização de C_{max} apresentada em [1].

$$\text{Minimizar } C_{max} \quad (1)$$

sujeito a:

$$x_{ji} \geq 0; \quad \forall j \in J; i \in M \quad (2)$$

$$x_{ji} \geq x_{jh} + p_{jh}; \quad \forall j \in J; i, h \in M; h \geq 1; h = i - 1 \quad (3)$$

$$x_{ji} \geq x_{ki} + p_{ki} - Vz_{jik}; \quad \forall j, k \in J; i \in M \quad (4)$$

$$x_{ki} \geq x_{ji} + p_{ji} - V(1 - z_{jik}); \quad \forall j, k \in J; \quad (5)$$

$$i \in M \quad C_{max} \geq x_{jm} + p_{jm}; \quad \forall j, k \in J; \quad (6)$$

$$i \in M; i = m \quad z_{jki} \in \{0, 1\}; \quad \forall j, k \in J; i \in M \quad (7)$$

A restrição [2] assegura que o tempo inicial (x_{ji}) de cada tarefa j alocada em i é não-negativo, [3] garante que a regra de precedência seja respeitada entre processos (p_{ji} e p_{ki}) durante a simulação, [4] e [5] não permitem que duas tarefas sejam alocadas ao mesmo tempo em uma mesma máquina, sendo V é um número com grande módulo, [6] impede que a medida do makespan seja superior ao ciclo de processamento da última tarefa de cada máquina e [7] garante que z_{jki} deva assumir um valor binário.

A formulação em CP está proposta através da resolução do problema com duas técnicas. A primeira considera a adaptação do modelo MIP disjuntivo adotado anteriormente (CP-DJM), de forma a tornar viável a sua utilização ao contexto. Assim, a função objetivo de minimização C_{max} é apresentada em [8].

$$\text{Minimizar } C_{max} \quad (8)$$

sujeito a:

$$x_{ji} \geq 0; \quad \forall j \in J; i \in M \quad (9)$$

$$x_{ji} \geq x_{jh} + p_{jh}; \quad \forall j \in J; i, h \in M; h \geq 1; h = i - 1 \quad (10)$$

$$C_{max} \geq x_{jm} + p_{jm}; \quad \forall j, k \in J; i \in M; i = m \quad (11)$$

$$x_{ji} \geq x_{ki} + p_{ki} // x_{ki} \geq x_{ji} + p_{ji}; \quad \forall j, k \in J; i \in M; k < j \quad (12)$$

As restrições [9], [10] e [11] estão definidas de forma semelhante as restrições [2], [3] e [6] descritas originalmente no formato MIP-DJM, [12] descreve a formatação com que a escolha do sequenciamento ocorre, considerando como referência a relação de precedência existente entre j e k. É possível afirmar que este formato, em termos matemáticos, não muda substancialmente ao anteriormente proposto, havendo somente a dispensa pela utilização da variável binária z_{jki} .

Em um segundo momento houve a escolha pela utilização do solver padrão para o JSSP adotada no software CPLEX Optimization Studio, que foca na redução dos domínios das variáveis de decisão, explorando os ramos em um processo discreto, de modo a verificar quais são satisfatórias a partir da descrição dos intervalos de processamento de cada tarefa ($Itvs_{jm}$) e sequência de máquinas adotada ($Mchs_{jm}$), conforme mostra a proposta de minimização de C_{max} descrita pela função objetivo [13].

$$\text{Minimizar } C_{max} \quad (13)$$

sujeito a:

$$noOverlapMchs_{ji}; \quad \forall j \in J; i \in M \quad (14)$$

$$endBeforeStart(Itvs_{ji}; Itvs_{jr}); \quad \forall j \in J; i, h \in M; h = i - 1 \quad (15)$$

A restrição [14] serve para prevenir a existência de intervalo sem uma sequência de overlapping, para aumentar as chances de obter a medida da mínima distância na sequência entre dois intervalos de tarefas consecutivos, e [15] restringe o momento em que ocorre o posicionamento do início de uma tarefa, a fim de precaver que não existam dois processos com inícios em tempos iguais para uma máquina.

Quanto ao estado da arte relativa a pesquisas científicas conduzidas para a resolução dos mais diferentes tipos de JSSP, recentemente revisões de literatura foram conduzidas para identificar a evolução das técnicas utilizadas para a resolução do JSSP ao decorrer dos anos (Çaliş e Bulkan, 2015; Chaudhry e Khan, 2016; Ku e Beck, 2016; Sharma e Jain, 2016; Rossit et al., 2018; Zhang et al., 2019).

3. Procedimentos metodológicos

A metodologia de pesquisa foi proposta de maneira a viabilizar a realização do comparativo entre as técnicas de otimização para o JSSP. Assim, primeiramente houve a definição das instâncias utilizadas na simulação, onde optou-se pelo uso dos 82 dados disponibilizados na biblioteca digital OR Library, por serem capazes de expor amplamente os fenômenos geralmente observados ao contexto. Em um segundo momento, estes foram separados em doze grupos, através da semelhança entre características e número de máquinas e tarefas intrínsecos de cada, conforme mostra a Tabela 1.

Tabela 1 . Características gerais dos doze grupos de instâncias adotados

Grupo	Tarefas	Máquinas	Tempo médio de processamento	Número de instâncias	Relação tarefas/máquinas
A	6	6	5.47	1	1.0
B	10	5	50.66	5	2.0
C	10	10	52.57	18	1.0
D	15	5	52.92	5	3.0
E	20	5	51.85	6	4.0
F	15	10	51.51	5	1.5
G	15	15	52.04	5	1.0
H	20	10	51.17	15	2.0
I	20	15	24.88	3	1.3
J	20	20	29.51	4	1.0
K	30	10	51.70	5	3.0
L	50	10	50.29	10	5.0
Total				82	

Em maiores detalhes, observou-se que 43 instâncias estão vinculadas a apenas três grupos, C, H e L, contidos em diferentes características de complexidade. Já o tempo médio de processamento é aproximadamente equivalente a 50 unidades de tempo em praticamente todos os casos, com alguma exceções, verificadas em A (5.47), I (24.88) e J (29.52), sendo que em conjunto os dois últimos possuem maiores frequências de conflitos para a utilização das máquinas em uma mesma operação.

Sobre os técnicas de otimização adotadas, o uso da versão MIP disjuntiva (MIP-DJM) proposta por Manne (1960) justifica-se devido a sua vasta difusão científica, sendo reconhecida como uma das mais eficientes na busca por soluções ótimas, além de possibilitar uma adaptação para que seja aplicada sob os procedimentos da programação em CP solver (CP-DJM). Em paralelo, a abordagem CP padrão vinculada ao CPLEX Optimization Studio (CP-CPLEX) foi escolhida por atender satisfatoriamente as características e requisitos do contexto JSSP ao qual o estudo foi proposto.

Após o estabelecimento dos valores para os limites inferior e superior do MIP-DJM, e das soluções para as técnicas CP-DJM e CP-CPLEX de cada instância, foi possível a verificação do nível de ajustamentos destas, por meio de um benchmark com valores ótimos já predispostos cientificamente em outras pesquisas, de maneira a verificar quais são capazes de retornar menores níveis de diferenciação, podendo classificar assim os considerados como mais eficientes. Em conjunto, foi proposta a identificação do número de soluções exatas que cada técnica é capaz de gerar, dentro do tempo de simulação equivalente a $t=1200s$ e um número ilimitado de falhas para o CP solver.

Outras medidas foram adotadas para a determinação da qualidade de ajustamento das técnicas adotadas. No caso do MIP, houve a verificação do erro médio (gap) entre os limites superior e inferior para cada instância. Enquanto isso, como

o CP retorna apenas uma solução, a verificação interna deste tipo de modelo foi desenvolvida com base em três critérios: tempo para gerar a última solução viável, total de memória necessária e qualidade de ajustamento em relação aos limites encontrados para o MIP.

Por fim, todas os experimentos foram desenvolvidos através de uma unidade de processamento Intel Core i5 450M 2.40 GHz, com um total de 4 GB de memória, por meio da utilização do CPLEX Optimization Studio v12.4.

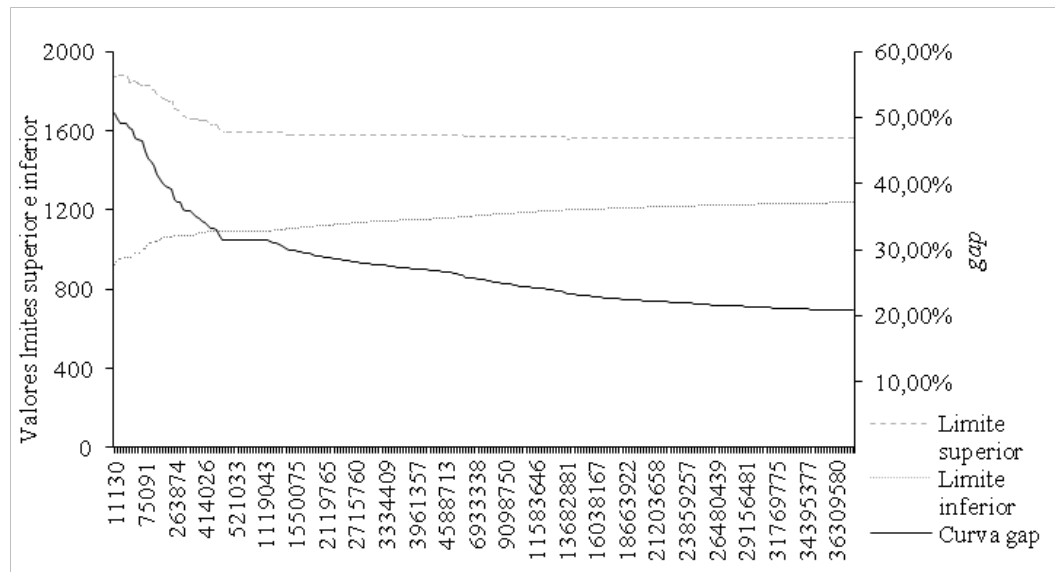
4. Resultados e discussões

A presente seção tem por objetivo mostrar os resultados obtidos por meio da simulação dos modelos escolhidos para resolver o JSSP, comparando a performance de cada um conforme os requisitos supracitados anteriormente.

A fim de iniciar a discussão sobre o tema, a Figura 1 mostra um exemplo de comportamento obtido ao realizar o ajustamento das curvas superior e inferior quando da aplicação do modelo MIP para a instância abz5, onde foi possível observar que houve uma redução do erro médio entre ambas de 20.80% ($t=1200s$), com maior redução verificada nos primeiros $t=36s$ (22.96%). A partir deste momento até o final da simulação, o declive foi de apenas 6.80%, com valores superior e inferior de, respectivamente, 1558 e 1234.

Para o comportamento dos resultados conforme cada grupo de instâncias, de acordo com a Figura 2, a primeira característica observada foi o fraco ajustamento dos valores limites das soluções para os casos considerados como de grande dimensão, e que por consequência não foram capazes de retornar bons ajustamentos do erro. Em contrapartida, encontrou-se ao total 8 soluções ótimas, de maneira que todas pertencem apenas a instâncias de baixo grau de complexidade (grupos A: 1, B: 5 e C: 2).

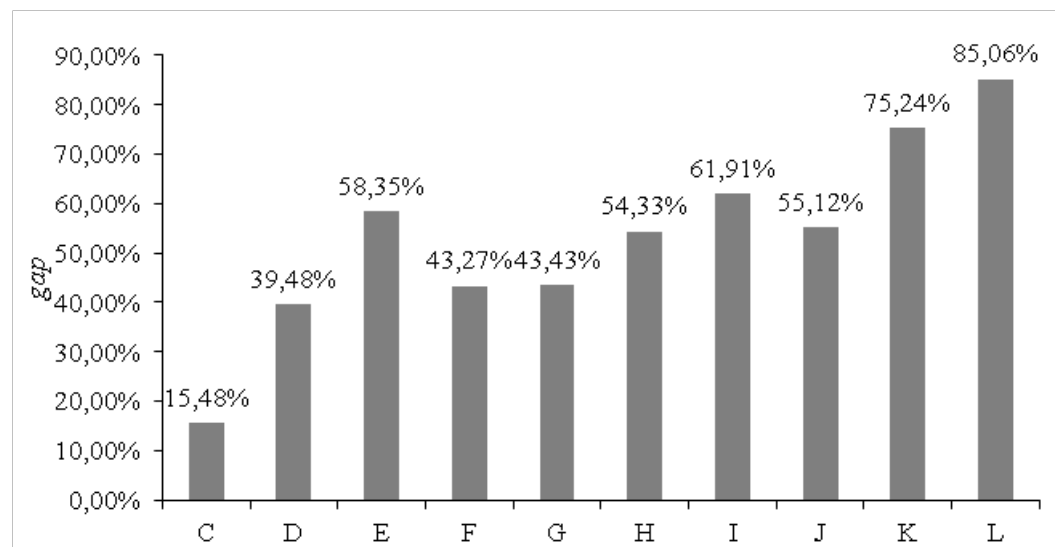
Figura 1 Comportamento da curva para o erro médio verificado na instância abz5



Excluídas estas situações, observou-se que em C (15.48%) o modelo foi capaz de retornar, senão soluções ótimas, boas convergências para a maioria dos seus elementos, sendo possível afirmar que o MIP-DJM possui potencial para encontrar soluções ideais em outras instâncias similares, desde

que sejam disponibilizados períodos de simulação superiores a t=1200s. por exemplo, testes adicionais foram realizados utilizando tempos de processamento superiores, e mostraram que em t=4800s a qualidade das instâncias C melhorou em 8,38%.

Figura 2 Resultados médios do gap em relação a cada um dos grupos de instâncias



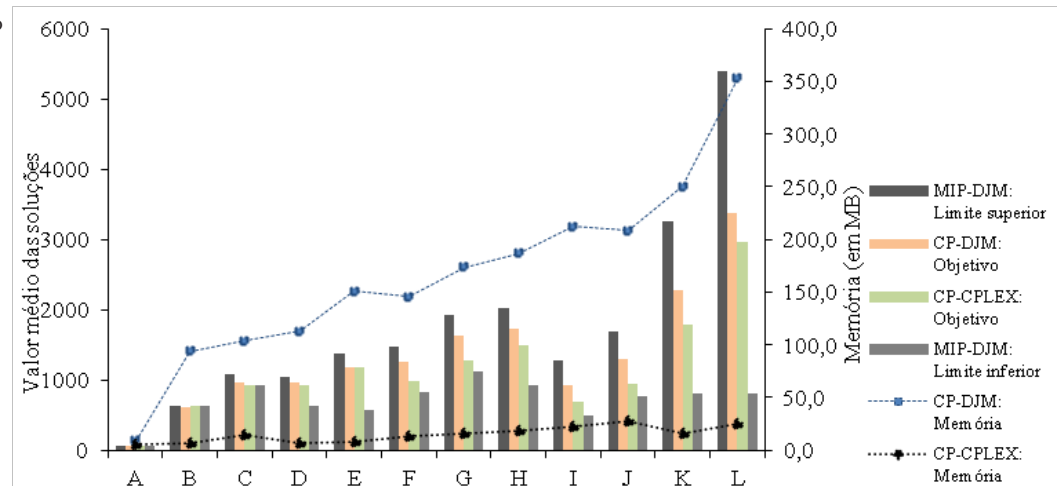
Porém, em instâncias cujas características são consideradas mais complexas o perfil apresentado não se perpetuou, como foi possível notar a partir de D (39.48%), onde o gap é 24% superior a C. O fato pode ser explicado devido ao maior tempo médio de processamento das atividades de D em comparação a C, além de possuir um maior número de tarefas que coincidam para ser processadas em uma máquina ao mesmo tempo, em oposição ao argumento de que foi proposta apenas a substituição de cinco destas em detrimento ao mesmo número de atividades.

No extremo oposto, as instâncias K e L apresentaram os maiores valores para o erro médio entre os limites superior e inferior, devido principalmente a extensão do número de elementos necessários a serem percorridos durante a simulação. Ainda, estas são definidas, conforme a Tabela 1, como as de maior complexidade, de maneira que o método não consegue ao menos aproximar os limites e, assim, concluir o Branch-and-Bound, fato indesejável quando considera-se a possibilidade da aplicação deste tipo de estudo em situações industriais verificadas na prática.

A próxima análise foi proposta de forma a verificar o nível de declive do erro médio obtido, onde notou-se que a redução está mais diretamente vinculada às características intrínsecas de cada instância, como tempo necessário para o processamento das operações, o número de interseções para que duas tarefas sejam executadas em cada máquina e a relação entre o número de tarefas em proporção ao total de máquinas disponibilizadas para manufatura.

Porém, a principal influência está vinculada a quantidade absoluta de máquinas e tarefas que devem ser alocadas, sendo as maiores reduções observadas para instâncias de menor tamanho, pois ao decorrer da simulação o algoritmo MIP tem uma maior capacidade de percorrer um número maior de nós no Branch-and-Bound, demandando uma menor utilização de recursos computacionais. Por consequência, ao contexto adotado observou-se quedas superiores e inferiores do erro para C (5.7%) e L (1.5%), respectivamente, desconsiderando instâncias cuja as soluções encontradas foram ótimas (A e B).

Figura 3 Valores médios para a função objetivo e total de memória utilizada para as aplicações em CP



Para os valores encontrados na função objetivo em comparação com os limites superior e inferior encontrados no modelo MIP, é possível afirmar que o CP-DJM tem um ajuste de melhor qualidade para aproximações com os limites superiores. Fato inverso está apresentado para o CP-CPLEX, por possuir uma aproximação mais forte com as soluções inteiras, demonstrando o seu melhor nível de ajuste em comparação ao CP-DJM, principalmente em instâncias de menor tamanho.

Ao total o CP-CPLEX atingiu a solução ótima em 33 dos 82 casos, de forma a abranger desde grupos de instâncias pequenas até as classificadas como de alta complexidade, como na totalidade do casos existentes em K e em 5 opor-

Quando a aplicação em CP, primeiramente foi verificado que o modelo disjuntivo adaptado CP-DJM exige, em média, 10 vezes mais memória para a simulação, independente do número de restrições e variáveis processadas, bem como do tamanho de cada instância, quando comparado ao CP-CPLEX, o que faz com que esta técnica seja capaz de reproduzir um número superior de soluções, conforme o gráfico apresentado na Figura 3.

Em relação ao tempo necessário para encontrar a solução mais ajustada, dentro do tempo de simulação determinado, tem-se que em nove grupos de instâncias o CP-DJM demanda um maior tempo para estabilizar um resultado mais preciso, o que pode refletir uma maior necessidade de se dispor tempos de simulação mais extensos para que haja ocorrências de soluções ótimas, em detrimento as aproximações, com uma eficiência 37% superior em prol do CP-CPLEX.

tunidades para L. Com uma performance muito inferior, o CP-DJM encontrou o valor ideal da função objetivo em apenas 5 situações, sendo estas originadas nos grupos A e B. Quanto a memória demandada, mais uma vez o CP-CPLEX se mostrou em média 6 vezes mais eficiente ao CP-DJM, apesar deste possuir um comportamento constante de utilização da memória, independente das características e do tamanho das instâncias.

A etapa posterior da pesquisa consistiu em realizar a comparação do nível de ajustamento das funções objetivo de cada uma das três técnicas adotadas em relação a 46 exemplos de resultados ótimos verificados cientificamente por outros autores, conforme mostram os dados da Tabela 2.

Tabela 2 Comparação dos modelos MIP e CP com soluções ótimas para a função objetivo

Instância	Grupo	Solução ótima	CP-DJM		CP-CPLEX		MIP-DJM		Referência
			Objetivo	Diff.	Objetivo	Diff.	Límite inferior	Diff.	
abz5	C	1234	1379	0.12	1234	0.00	1180	0.04	Klemmt (2009)
abz6	C	943	1062	0.13	943	0.00	902	0.04	Klemmt (2009)
abz7	I	655	892	0.36	670	0.02	460	0.30	Klemmt (2009)
ft06	A	55	55	0.00	55	0.00	55	0.00	Klemmt (2009), Gao (2011)
ft10	C	900	917	0.02	938	0.04	1007	0.12	Applegate (1991), Vaessens (1994), Gao (2011)
ft20	E	1165	1141	0.02	1165	0.00	545	0.53	Gao (2011)
la01	B	666	666	0.00	666	0.00	666	0.00	Klemmt (2009), Gao (2011)
la06	D	926	1064	0.15	926	0.00	718	0.22	Gao (2011)
la11	E	1222	1165	0.05	1222	0.00	610	0.50	Klemmt (2009)
la21	F	1046	1221	0.17	1046	0.00	891	0.15	Klemmt (2009)
la26	H	1218	1657	0.36	1218	0.00	807	0.34	Klemmt (2009), Gao (2011)
la31	K	1784	2113	0.18	1784	0.00	815	0.54	Gao (2011)
la37	G	1397	1712	0.23	1397	0.00	1210	0.13	Vaessens (1994)
orb01	C	1035	987	0.05	1099	0.06	944	0.09	Applegate (1991)
...									
Dif. total				6.31		0.37		10.82	

Com uma diferença de apenas 0.37 pontos, CP-CPLEX foi capaz de retornar com melhor eficiência e muita superioridade as demais técnicas, atingindo a solução ótima inclusive em instâncias consideradas como de alto grau de complexidade, como as inseridas no grupo K (Ia31 a Ia35), podendo esta ser considerada como a melhor opção disponibilizada ao contexto do JSSP explorado no presente estudo.

Em um segundo momento, novamente houve o destaque para uma modelagem em CP, no caso o CP-DJM, com uma pontuação da diferença equivalente a 6.31, sendo capaz de encontrar a solução ótima para instâncias pequenas, assim como o MIP-DJM, porém atingindo uma redução do erro melhor ajustada para os demais elementos da amostragem, a um custo de memória semelhante ao MIP mas superior ao CP-CPLEX.

Porém, apesar de ser uma medida de eficiência relevante para os testes realizados a memória consumida para o processamento das instâncias não é um fator crítico para identificar a qualidade de um método. O maior consumo de memória (350 MB) foi para as instâncias com relação tarefas/máquinas iguais a 5.0, que ainda sim é um valor muito distante dos 4 GB de memória disponibilizados pela unidade de processamento utilizada. Dessa forma, a medida da memória consumida no processamento é importante para verificar uma tendência de consumo caso seja necessário resolver através de MIP ou CP instâncias com relação tarefas/máquinas superiores as utilizadas neste estudo.

6. Conclusões

A presente pesquisa possibilitou a comparação envolvendo o uso de técnicas de otimização baseadas nos conceitos MIP e CP para o contexto de aplicação do JSSP, em seu formato tradicional. Primeiramente, foi possível observar na aplicação MIP que o tamanho das instâncias, o tempo de processamento médio requerido por cada tarefa, a tarefas/máquinas e o número de conflitos para utilização das máquinas influenciam diretamente na performance para a redução do gap.

Em específico aos modelos CP, tem-se que a abordagem padrão adotada no CPLEX Optimization Studio (CP-CPLEX) mostrou-se mais eficiente em comparação a abordagem disjuntiva adaptada, considerando os três indicadores (memória utilizada, tempo para gerar a última solução viável e qualidade de ajustamento em relação aos limites encontrados para o MIP) utilizados para informar o nível de desempenho de ambas.

Foi possível notar que nenhuma técnica se mostrou capaz de encontrar eficazmente a solução ótima para o JSSP, havendo um destaque especial apenas para a aplicação CP-CPLEX, que possibilitou retornar 33 soluções ótimas, em

detrimento ao CP-DJM (5) e MIP-DJM (8), sendo que para os dois últimos a obtenção de valores ótimos se restringiu apenas para grupos consideradas de pequenas dimensões (A, B e C). Ainda, verificou-se que, mesmo disponibilizando um período de simulação superior, dificilmente a eficiência observada sofreria alterações substanciais, visto o comportamento da curva de declive para o gap observado em MIP-DJM.

De forma a englobar os resultados obtidos para as duas formas de modelagens propostas, e com base na comparação dos valores ótimos encontrados na literatura para a função objetivo, conclui-se que o CP-CPLEX possui uma capacidade de ajustamento superior as demais técnicas, independente do tamanho e características das instâncias em análise. Cabe ressaltar que para a afirmação proposta ser validada com maior amplitude, é necessário que pesquisas futuras considerem uma amostragem de instâncias maior para verificação, apesar do autor acreditar que o perfil de resultados estabelecido no presente estudo dificilmente sofrerá alterações substanciais com o incremento de mais amostras a análise.

Como limitações da pesquisa, os testes realizados com as técnicas de MIP e CP foram conduzidas com um número limitado de tarefas e máquinas, de modo a possibilitar que o CPLEX (e qualquer outro solver comercial) seja capaz de resolver as instâncias, mesmo que obtendo soluções de baixa qualidade. Para instâncias com maior número de tarefas e máquinas, os tempo de processamento poderiam ser ampliados para permitir que soluções de melhor qualidade pudessem ser obtidas. Além disso, foi constatado que uma relação tarefas/máquinas superior a 5.0 é suficiente para que dificuldades no processamento das instâncias utilizando o CPLEX é verificado.

Para trabalhos futuros, espera-se a comparação dos métodos apresentados na presente pesquisa com aplicações voltadas para a resolução do JSSP sob o enfoque de heurísticas e metaheurísticas, de maneira a verificar a diferença de desempenho e o número de soluções ótimas que estas são capazes de gerar. Ainda, pode ser proposta a utilização de outras instâncias e/ou softwares para a simulação, a fim de demonstrar o nível de eficiência e influência do CPLEX para a obtenção dos resultados.

7. Referências

- Aggoune, R. (2004). Minimizing the makespan for the flow shop scheduling problem with availability constraints. *European Journal of Operational Research*, 153(3), 534-543.
- Andrés, C., e Maheut, J. (2018). Secuenciación con Almacenes Limitados. Una Revisión de la Literatura. *Dirección y Organización*, 66, 17-33.

- Araújo, J. A., del Olmo, R., e Laviós, J. J. (2013). Subasta combinatoria para la programación dinámica en sistemas de fabricación distribuidos. *Dirección y Organización*, 51, 55-64.
- Applegate, D., e Cook, W. (1991). A computational study of the job-shop scheduling problem. *Orsa Journal on Computing*, 3(2), 149-156.
- Bautista, J., e Alfaro, R. (2018). Mixed integer linear programming models for Flow Shop Scheduling with a demand plan of job types. *Central European Journal of Operations Research*, 2018, 1-19.
- Chaudhry, I. A., & Khan, A. A. (2016). A research survey: review of flexible job shop scheduling techniques. *International Transactions in Operational Research*, 23(3), 551-591.
- Çaliş, B., & Bulkan, S. (2015). A research survey: review of AI solution strategies of job shop scheduling problem. *Journal of Intelligent Manufacturing*, 26(5), 961-973.
- Eroglu, D. Y., e Ozmutlu, H. C. (2014). Mip models and hybrid algorithms for simultaneous job splitting and scheduling on unrelated parallel machines. *The Scientific World Journal*, 2014.
- Gao, L., Zhang, G., Zhang, L., e Li, X. (2011). An efficient memetic algorithm for solving the job shop scheduling problem. *Computers & Industrial Engineering*, 60(4), 699-705.
- Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1(1), 61-68.
- Kashan, A. H., Karimi, B., e Jolai, F. (2010). An effective hybrid multi-objective genetic algorithm for bi-criteria scheduling on a single batch processing machine with non-identical job sizes. *Engineering Applications of Artificial Intelligence*, 23(6), 911-922.
- Klemmt, A., Horn, S., Weigert, G., e Wolter, K-J. (2009). Simulation-based optimization vs. mathematical programming: A hybrid approach for optimizing scheduling problems. *Robotics and Computer-Integrated Manufacturing*, 25(6), 917-925.
- Ku, W-Y., e Beck, J. C. (2014). Revisiting off-the-shelf mixed integer programming and constraint programming models for job shop scheduling. *University of Toronto Technical Reports*, 2014(1), 1-21.
- Ku, W. Y., & Beck, J. C. (2016). Mixed integer programming models for job shop scheduling: A computational analysis. *Computers & Operations Research*, 73, 165-173.
- Manne, A. S. (1960). On the job-shop scheduling problem. *Operations Research*, 8(2), 219-223.
- Martin, P., e Shmoys, D. B. (1996). A new approach to computing optimal schedules for the job-shop scheduling problem. Pages 389-403 of: *Integer Programming and Combinatorial Optimization*. Springer.
- Mateo, M. R. I., e Pascual, R. C. (2009). Un procedimiento GRASP para la programación de pedidos en máquinas paralelas con tiempos de preparación. *Dirección y Organización*, 39, 71-78.
- Naderi, B., Zandieh, M., e Fatemi Ghomi, S. M. T. (2009). Scheduling job shop problems with sequence dependent setup times. *International Journal of Production Research*, 47(21), 5959-5976.
- Neuenfeldt Júnior, A., Silva, E., Gomes, A. M., Soares, C., e Oliveira, J. F. (2019). Data mining based framework to assess solution quality for the rectangular 2D strip-packing problem. *Expert Systems with Applications*, 118(2019), 365-380.
- Pezzella, F., Morganti, G., e Ciaschetti, G. (2008). A genetic algorithm for the exible job-shop scheduling problem. *Computers & Operations Research*, 35(10), 3202-3212.
- Rossit, D. A., Tohmé, F., & Frutos, M. (2018). The non-permutation flow-shop scheduling problem: a literature review. *Omega*, 77, 143-153.
- Runarsson, T. P., Schoenauer, M., e Sebag, M. (2012). Pilot, rollout and monte carlo tree search methods for job shop scheduling. Pages 160-174 of: *Learning and intelligent optimization*. Springer.
- Sharma, P., & Jain, A. (2016). A review on job shop scheduling with setup times. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 230(3), 517-533.
- Vaessens, R. J. M., Aarts, E. H. L., e Lenstra, J. K. (1996). Job shop scheduling by local search. *Journal on Computing*, 8(3), 302-317.
- Yilmaz, E., Duygu, O., Cenk, H., e Ozmutlu, S. (2014). Genetic algorithm with local search for the unrelated parallel machine scheduling problem with sequence-dependent set-up times. *International Journal of Production Research*, 52(19), 5841-5856.
- Zhang, J., Ding, G., Zou, Y., Qin, S., & Fu, J. (2019). Review of job shop scheduling research and its new perspectives under Industry 4.0. *Journal of Intelligent Manufacturing*, 30(4), 1809-1830.