Case modelo V_2

Felipe Sinnecker

Fevereiro 2024

1 Problema

O problema proposto é uma versão do job-shop problem (JSP). De maneira geral temos um conjunto de tarefas $J_1, \cdots, J_n$ a serem realizadas e um conjunto de máquinas $M_1, \cdot, M_m$ para realiza-las. Cada tarefa precisa ser alocada em uma máquina, que irá levar um determinado tempo para concluir a tarefa. O objetivo é organizar a ordem em que as tarefas serão alocadas a cada máquina, baseado em algum critério como custo ou tempo, de forma que o critério escolhido seja otimizado.

Para o modelo V_2 apresentado, estamos querendo saber em que ordem e os tempos em que os tecidos serão alocados nas máquinas de enfesto, assim como a ordem dos cortes, levando em consideração a capacidade de cada mesa e que os cortes devem acontecer na mesma sequência do enfesto por mesa. O objetivo é minimizar o tempo total para a etapa de corte a ser concluída.Temos 2 máquinas de enfesto E_1, E_2, 1 máquina de corte C_1 e 9 ordens de produção diferentes $OP1, OP2, OP3, OP4, OP5, OP6, OP7, OP8, OP9$. Os Dados foram utilizados para a modelagem foram os tempos de setup de cada máquina, o tempo de troca de mesa da máquina de corte, o comprimento da ordem na mesa e o tempo médio de enfesto e de corte para cada ordem:

Máquina	Tempo de setup	Tempo de troca
E1	2	-
E1	3	-
C1	4	5

Ordem de Produção	Comprimento na mesa	Tempo médio de enfesto	Tempo médio de corte
OP1	3	18	4
OP2	3	14	4
OP3	4	12	6
OP4	4	6	6
OP5	8	36	12
OP6	6	9	6
OP7	2	9	2
OP8	2	12	2
OP9	4	12	4

Baseado nos vídeos disponíveis assim como as informações fornecidas no enunciado, podemos assumir que a etapa de enfesto não pode ser dividida em pequenas etapas e realizada em mais de uma máquina, portanto uma vez que o enfesto de uma ordem começa devemos terminar antes de realizar outra. A etapa de corte só pode começar caso sua respectiva etapa de enfesto já tenha sido concluída, e a ordem de corte precisa respeitar a ordem de enfesto da mesa. Todas as ordens de produção devem ser concluídas. Podemos utilizar o mesmo modelo anterior $v1$ adaptando as variáveis.

2 Modelo

O modelo clássico de job-shop utiliza variáveis binárias para indicar a ordem em que as tarefas são executadas em cada máquina, sendo que existe uma tarefa extra 0, podemos considerar as tarefas a serem realizadas como de 1 a 9, esta tarefa extra representa o início da ordem de cada máquina, assim como o retorno a ela representa o encerramento das tarefas na máquina.

Dados e conjuntos

Temos os seguintes conjuntos para o problema:

$$\begin{aligned} J_0, &\quad \text{ordem de produção com a ordem extra} \\ J, &\quad \text{ordem de produção} \\ M, &\quad \text{máquinas} \end{aligned}$$

O conjunto J_0 representa tanto o conjunto J com as ordens de produção de 1 a 9, como a ordem extra 0. O conjunto M representa as máquinas 1,2 respectivamente (E_1, E_2). Para os dados fornecidos do problema temos:

$$\begin{aligned} st_m, &\quad m \in M &&\quad \text{tempo de setup de cada máquina } m \\ TE_i, &\quad i \in J &&\quad \text{tempo médio de enfesto de cada ordem } i \\ TC_i, &\quad i \in J &&\quad \text{tempo médio de corte de cada ordem } i \\ CO_i, &\quad i \in J &&\quad \text{comprimento sobre a mesa de cada ordem } i \\ \delta, & &&\quad \text{tempo de troca de mesa da máquina de corte} \\ M, & &&\quad \text{comprimento das mesas} \end{aligned}$$

Variáveis

Podemos modelar o problema utilizando variáveis binárias:

$$
\begin{aligned}
x_{i,j,m} &\in \{0,1\} \quad i,j \in J_0 \quad m \in M \\
y_{i,m} &\in \{0,1\} \quad i \in J \quad m \in M \\
z_{i,j} &\in \{0,1\} \quad i,j \in J_0 \\
w_{i,j} &\in \{0,1\} \quad i,j \in J_0 \\
d^1_{i,j} &\in \{0,1\} \quad i,j \in J \\
d^2_{i,j} &\in \{0,1\} \quad i,j \in J \\
d_{i,j} &\in \{0,1\} \quad i,j \in J
\end{aligned}
$$

A variável $x_{i,j,m}$ indica que a etapa de enfesto da ordem i antecede a da ordem j na máquina m. A variável $y_{i,m}$ indica qual mesa recebeu a ordem i. A variável $z_{i,j}$ indica que ordem i antecede a ordem j na etapa de corte. A variável $w_{i,j}$ indica que a etapa de corte mudou de mesa para realizar o próximo corte. A variável $d^1_{i,j}$ indica que a ordem j começa a etapa de enfesto antes da etapa de corte de i terminar. A variável $d^1_{i,j}$ indica que o corte da ordem j termina depois do enfesto da ordem i começar. A variável $d_{i,j}$ indica que as ordens i e j estão ao mesmo tempo na mesa durante alguma etapa. Temos as variáveis inteiras:

$$
\begin{aligned}
E_i &\in Z \quad i \in J_0 \\
C_i &\in Z \quad i \in J_0 \\
T &\in Z
\end{aligned}
$$

As variáveis E_i representam o tempo final da etapa de enfesto de cada ordem i, as variáveis C_i representam o tempo final da etapa de corte de cada ordem i e a variável T serve para modelar o maior tempo de corte $\max_i\{C_i\}$.

Função Objetivo

A função objetivo é dado pelo valor do maior tempo, logo é o valor da variável Z:

$$\min Z$$

Restrições

Para o conjunto de restrições temos:

$$E_0 = 0 \tag{1}$$

$$C_0 = 0 \tag{2}$$

$$x_{i,i,m} = 0, \qquad \forall i \in J_0 \quad \forall m \in M \tag{3}$$

$$z_{i,i} = 0, \qquad \forall i \in J_0 \tag{4}$$

$$w_{0,i} = 0, \qquad \forall i \in J \tag{5}$$

$$\sum_{m \in M} y_{i,m} = 1, \qquad \forall i \in J \tag{6}$$

$$\sum_{j \in J_0} x_{i,j,m} = y_{i,m}, \qquad \forall i \in J \quad \forall m \in M \tag{7}$$

$$\sum_{i \in J_0} x_{i,j,m} = y_{j,m}, \qquad \forall j \in J \quad \forall m \in M \tag{8}$$

$$\sum_{j \in J} x_{0,j,m} \leq 1, \qquad \forall m \in M \tag{9}$$

$$d^1_{i,j} \geq (C_i - E_j + TE_j)/\theta - (2 - y_{i,m} - y_{j,m})\theta, \qquad \forall i,j \in J \quad m \in M \tag{10}$$

$$d^2_{i,j} \geq (C_j - E_i + TE_i)/\theta - (2 - y_{i,m} - y_{j,m})\theta, \qquad \forall i,j \in J \quad m \in M \tag{11}$$

$$d_{i,j} \geq d^1_{i,j} + d^2_{i,j} - 1.5, \qquad \forall i,j \in J \tag{12}$$

$$\sum_{j \in J} CO_j d_{i,j} \leq M, \qquad \forall i \in J \tag{13}$$

$$w_{i,j} \leq z_{i,j}, \qquad \forall i,j \in J_0 \tag{14}$$

$$\sum_{j \in J_0} z_{i,j} = 1, \qquad \forall i \in J \tag{15}$$

$$\sum_{i \in J_0} z_{i,j} = 1, \qquad \forall j \in J \tag{16}$$

$$\sum_{j \in J} z_{0,j} \leq 1 \tag{17}$$

$$\sum_{m \in M} x_{i,j,m} + \sum_{k \in J_0} w_{i,k} \geq z_{i,j}, \qquad \forall i,j \in J_0 \tag{18}$$

$$E_j \geq E_i + st_1 + TE_j - (1 - x_{i,j,1})\theta, \qquad \forall i \in J_0 \quad \forall j \in J \tag{19}$$

$$E_j \geq E_i + st_2 + TE_j - (1 - x_{i,j,2})\theta, \qquad \forall i \in J_0 \quad \forall j \in J \tag{20}$$

$$C_j \geq C_i + st_3 + \delta w_{i,j} + TC_j - (1 - z_{i,j})\theta, \qquad \forall i \in J_0 \quad \forall j \in J \tag{21}$$

$$C_i \geq E_i + st_3 + TC_i, \qquad \forall i \in J \tag{22}$$

$$T \geq C_i, \qquad \forall i \in J \tag{23}$$

$$E_i, C_i \geq 0, \qquad \forall i \in J \tag{24}$$

Temos várias restrições, podemos separar por grupos:

Restrições de início

Essas restrições representam aspectos da etapa inicial do processo. As restrições (1) e (2) garantem que os tempos de enfesto e corte da variável auxiliar sejam 0, sendo ela a primeira da ordem sempre. A restrição (3) e (4) garante que não possamos partir de uma ordem para ela mesma em qualquer etapa. A restrição (5) faz com que máquina de corte não mude de mesa antes do primeiro corte.

Restrições de ordem de enfesto

Essas restrições garantem que cada ordem seja executada em uma das máquinas, além de garantir que uma sequência seja respeitada. A restrição (6) garante que cada ordem seja executada em somente uma das máquina. As restrições (7) e (8) garantem que cada ordem tenha apenas um sucessor e um predecessor na máquina em que está sendo executada. A restrição (9) garante que a ordem extra de cada máquina tenha apenas um sucessor, dando início a sequência de tarefas em cada máquina.

Restrições de capacidade

A restrição (10) faz com que $d^1_{i,j}$ seja 1 caso qualquer ordem j, que esteja na mesma mesa de i, comece a etapa de enfesto antes da etapa de corte de i terminar. A restrição (11) faz com que $d^2_{i,j}$ seja 1 caso qualquer ordem j, que esteja na mesma mesa de i, termine a etapa de corte depois da etapa de enfesto de i começar. A restrição (12) garante que $d_{i,j}$ seja 1 caso tenha ambas as ordens estejam na mesma durante as tarefas, caso a etapa de corte de i termine depois de j começar o enfesto e o corte de j comece depois de i começar sua etapa de enfesto então existe uma sobreposição na mesa. A restrição (13) garante que a capacidade da mesa seja respeitada ao longo do processo.

Restrições de tempo

A restrições (19) e (20) utilizam um valor θ grande para garantir que caso a ordem j seja a sucessora da ordem i na máquina m a tarefa é realizado somente após a anterior, respeitando os tempos de setup e de enfesto, quando $x_{i,j,m} = 0$ temos que a restrição (24) vale. A restrição (21) garante que só tenha um corte por vez, respeitando o tempo de troca de mesa δ, setup e corte anterior, quando $z_{i,j} = 0$, ou seja o corte i não antecede j vale a restrição (24). A restrição (22) impede que a etapa de corte comece antes da etapa de enfesto chegar ao fim. A restrição (23) representa o valor máximo dos C_i e restringe a variável da função objetivo ao $\max_i C_i$.

3 Implementação Gurobi

O modelo foi implementado com a linguagem de programação julia, utilizando o pacote JuMP para escrever o modelo. O solver escolhido para resolver foi o

Gurobi. Os dados do problema foram inseridos de forma manual. Este modelo é bem mais complicado que suas versões anteriores, além das restrições de capacidade, temos as restrições de ordem de corte que devem seguir as de enfesto. Podemos também estabelecer um lower bound para o problema, considerando que só temos uma máquina de corte o tempo total para realizar todos os cortes junto ao tempo de setup, sem considerar a mudança de mesa totaliza 82 minutos, se juntarmos a isso o menor tempo de enfesto possível de 6 minutos na mesa mais rápida, temos ainda mais 8 minutos para que a etapa de corte pudesse começar, assim podemos estabelecer um lower bound de no mínimo 90.

Resultado

Temos que o resultado dado pelo Gurobi foi a sequência de enfesto [OP4,OP6,OP3,OP2,OP1] na máquina E_1 e seus tempos [8,19,41,57,77], [OP9,OP8,OP7,OP5] na máquina E_2 e seus tempos [15,30,48,90]. A sequência de cortes [OP4,OP6,OP9,OP8,OP7,OP3,OP2,OP1,OP5] na máquina $C1$ possui tempos finais [18,29,42,48,54,69,77,85,106] respectivamente, com 106 sendo o valor ótimo do problema. O gurobi demorou mais de uma hora para chegar no valor 106, tomando ainda mais tempo para aumentar o lower bound de 90.

4 Implementação Metaheurística

Para resolver o modelo v_2 podemos utilizar a mesma técnica de simulated annealing para resolver, fazendo uma pequena mudança na estrutura de vizinhança. As soluções continuam representadas como vetores. Neste caso cada solução s é um vetor de vetores, onde cada vetor representa uma máquina e contem a ordem de tarefas a ser executada. Temos 2 máquinas de enfesto e 1 máquina de corte, então temos 3 vetores, o terceiro com uma sequência contendo todas as ordens e os dois primeiros dividem as ordens entre si:

$$
\begin{aligned}
s &= [V_1, V_2] \\
V_i &= [v_1^i, \cdots, v_m^i], \quad v_j^i \in \mathcal{N}
\end{aligned}
$$

Tome o exemplo fornecido,$s =$ [[1,4,5],[2,3,6,7,8,9],[1,4,2,3,6,7,5,8,9]], a máquina E_1 realiza as ordens [OP1,OP4,OP5] e a máquina E_2 [OP2,OP3,OP6,OP7,OP8,OP9]. A máquina de corte C_1 realiza as ordens na sequência [1,4,2,3,6,7,5,8,9]. Podemos utilizar esses dados para calcular a função objetivo e os tempos de Enfesto e Corte também. Podemos ter soluções que não respeitem as restrições de capacidade de mesa, neste caso aplicamos uma penalidade ao valor final.

Vizinhança

Podemos aplicar o mesmo tipo de operação do modelo v0 para os vetores de enfesto, assim como as operações do modelo v1 no vetor de Cortes. Porém para facilitar, todas as mudanças devem obedecer as restrições de ordem de tarefa, ou seja, corte deve respeitar a ordem de enfesto. Assim torna-se bem mais difícil aplicar as operações, mudar a ordem de enfesto das máquinas acarreta na mudança da ordem de corte também. Para mudar a ordem de corte devemos respeitar a ordem de enfesto, então as mudanças ficam bem mais restritas

Resultados

O modelo v2 acabou sendo muito complicado para implementar todas as mudanças duranta o tempo de prova.