SEPARAÇÃO DE CORTES DO PROBLEMA DE ESCALONAMENTO DE TAREFAS EM MÁQUINAS PARALELAS USANDO ALGORITMO GENÉTICO

Daniel Oliveira, Artur Pessoa
daniel_oliveira@id.uff.br, artur@producao.uff.br
Universidade Federal Fluminense

LOGIS
Núcleo de Logística Integrada e Sistemas

1. Introdução

Seja um conjunto J de n tarefas a serem processadas em m máquinas paralelas idênticas. Cada tarefa tem uma duração p_j, um deadline d_j, um peso w_j e está associada a um custo $f_j(C_j) = w_jT_j$, denominado atraso ponderado (*weighted tardiness*). O atraso é definido como $T_j = max\{0, C_j - d_j\}$, sendo C_j o período de término da tarefa j. Cada máquina pode executar somente uma tarefa por vez e cada tarefa deve ser processada em somente uma máquina, sem pausas ou troca de tarefas antes da conclusão. O problema de escalonamento consiste em encontrar um sequenciamento de todas as tarefas nas máquinas, que minimize a soma dos custos individuais. Segundo a literatura, tal problema é codificado como $P||\Sigma w_jT_j$.

É utilizada uma formulação estendida para o problema, chamada de arco-tempo-indexada, onde cada variável binária $x^t_{i,j}$ indica o término da tarefa i e início da tarefa j, no período t, em uma mesma máquina. A tabela 1 apresenta dados de uma instância exemplo do problema.

Tarefa(j)	Duração (p_j)	Peso (w_j)	Deadline (d_j)
1	6	2	5
2	4	2	10
3	3	3	11
4	6	1	7
5	5	4	10
6	6	2	5
7	4	2	10
8	8	3	11

Tabela 1: Dados da Instância Exemplo

Uma solução ótima da relaxação linear do problema é dada pelo conjunto de variáveis em (1).

$$\begin{array}{lllll} x^0_{0,1} = 1 & x^6_{1,2} = 0,5 & x^{10}_{2,8} = 0,5 & x^{13}_{3,6} = 0,5 & x^{19}_{8,0} = 0,5 \\ x^0_{0,5} = 0,5 & x^6_{1,5} = 0,5 & x^{10}_{7,3} = 0,5 & x^{16}_{2,4} = 0,5 & x^{22}_{4,0} = 0,5 \\ x^0_{0,6} = 0,5 & x^6_{6,7} = 0,5 & x^{11}_{5,8} = 0,5 & x^{18}_{8,4} = 0,5 & x^{24}_{4,0} = 0,5 \\ x^5_{5,7} = 0,5 & x^9_{7,3} = 0,5 & x^{12}_{3,2} = 0,5 & x^{19}_{6,0} = 0,5 & \end{array} \quad (1)$$

As figuras 1 e 2 apresentam representações gráficas da solução dada em (1).

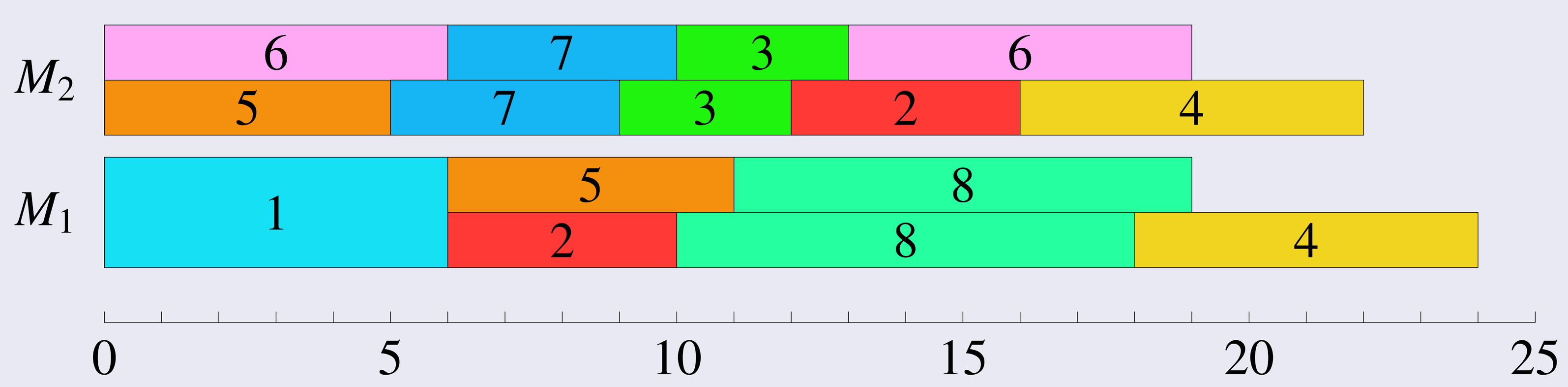

Figura 1: Gráfico Gantt da solução fracionária

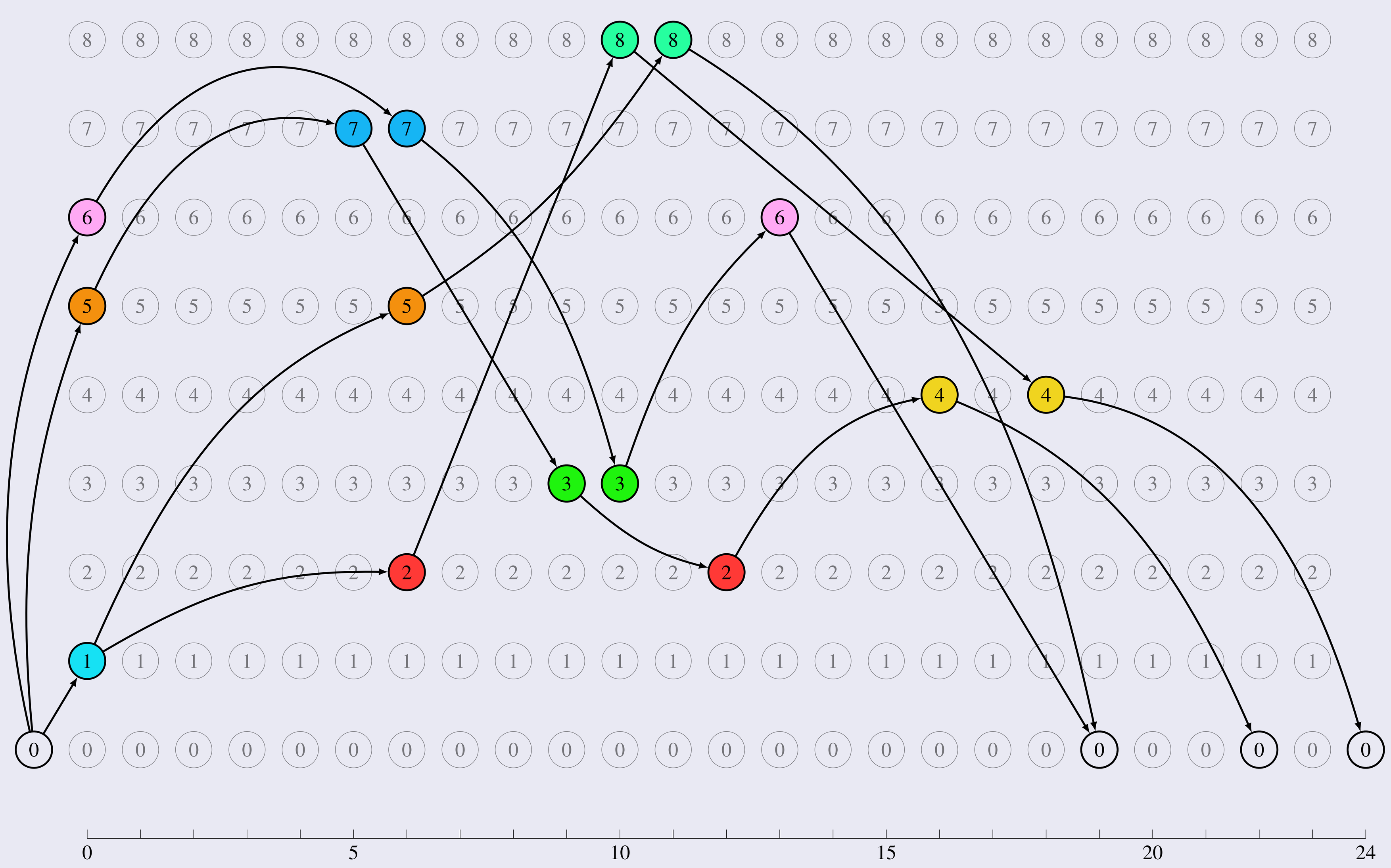

Figura 2: Grafo da solução fracionária

2. Desigualdades Válidas

Devido ao grande número de variáveis, Pessoa et al. (2010) desenvolveram um algoritmo de *Branch-cut-and-price* para sua resolução. Como o número de desigualdades válidas também é muito grande, são consideradas somente aquelas que são violadas pela relaxação linear.

Seja S um subconjunto de tarefas, $P(S)$ é definido como a soma dos tempos de processamento p_j das tarefas contidas em S, $\delta^+(S)$ e $\delta^-(S)$ são definidos como o conjunto de arcos com vértices $i \in S$ e $j \notin S$ e vice-versa, respectivamente. Sendo $0 < r < 1$ um multiplicador arbitrário, a primeira família de desigualdades válidas é representada por:

$$\sum_{(i,j)^t \in \delta^+(S)} \lceil rt \rceil \, x^t_{i,j} - \sum_{(i,j)^t \in \delta^-(S)} \lfloor rt \rfloor \, x^t_{i,j} \geq \lceil rp(S) \rceil \quad (2)$$

Definimos as variáveis agregadas y^t e z^t, para um dado S, sendo $J_0 = J \cup \{0\}$, como:

$$y^t = \sum_{j \in S} \sum_{i \in (J_0 \setminus S)} x^t_{j,i} \qquad z^t = \sum_{j \in S} \sum_{i \in (J_0 \setminus S)} x^t_{i,j} \quad (3)$$

Para $m \geq 2$ e $t \in \{1, \ldots, \lfloor (P(S) - 1)/m \rfloor + 1\}$ a família de desigualdades (4) também é válida para qualquer solução inteira do problema $P||\Sigma w_jT_j$.

$$\sum_{q=t}^{t_1} y^q + \sum_{q=t_1+1}^{T} 2\, y^q - \sum_{\substack{q=max\{t_1,\\ T-P(S)+m(t-1)+1\}}}^{T-1} z^q \geq 2, \quad t_1 = P(S) - t - (m-2)(t-1) \quad (4)$$

3. O Algoritmo de Separação

Foi desenvolvido um algoritmo genético para a separação das famílias de cortes (2) e (4). Por separação de cortes, entende-se a busca por uma desigualdade que seja violada por dada solução. Esse procedimento é geralmente utilizado para fortalecer a relaxação linear de um problema de programação inteira, na esperança de se encontrar uma solução inteira ao resolver o problema de programação linear.

Cada cromossomo é composto por um conjunto de tarefas S e um parâmetro r ou t. Uma população inicial de n soluções é gerada a partir da ordenação de tarefas pelo período de conclusão médio $\sum_t \sum_j t\, x^t_{i,j}$ e, para $i = 1, \ldots, n$, são escolhidas as i primeiras tarefas para compor S, aplicando-se busca local em seguida.

O resultante do operador crossover consiste na interseção dos conjuntos S, acrescido de elementos aleatórios que pertencem aos parentes. Caso os cromossomos parentes não tenham interseção, é escolhida uma tarefa aleatória de cada, e busca-se o conjunto de tarefas que constitui o menor caminho entre as duas no grafo da solução. Outro aspecto do crossover é a eliminação de eventuais subgrafos desconexos, para gerar somente conjuntos S conexos. São realizadas 20 operações de crossover com busca local por iteração do algoritmo.

O operador busca local realiza inclusões ou exclusões de 1 tarefa em S e, para cada S, testa todos os multiplicadores r relevantes, tal que $\lceil rP(S) \rceil - rP(S) > 0,99$, ou todos os valores t relevantes. O algoritmo é finalizado arbitrariamente após 100 iterações, e os 50 melhores cortes violados por mais de 0.05 são utilizados.

4. Resultados

Foram realizados testes em 39 instâncias do problema, usando o algoritmo genético proposto e o algoritmo guloso utilizado por Pessoa et al. (2010). Para a comparação, o benchmark utilizado foi o gap de integralidade obtido após a inserção dos cortes no nó raiz. Os resultados foram:

- Melhora média de 13% em 20 instâncias.
- Resultado igual em 14 instâncias.
- Piora média de 14% em 5 instâncias.

5. Conclusões

Apesar de animadores, os resultados obtidos ainda estão abaixo do esperado. Existem diversas alternativas para se buscar melhorias. Atualmente, as mais promissoras são:

- Busca por novas famílias de desigualdades válidas.
- Teste de outras regras de branching.
- Teste de outras Meta-heurísticas para a separação de cortes.

Referências

Pessoa, A., Uchoa, E., de Aragão, M. P., e Rodrigues, R. (2010). Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. *Mathematical Programming Computation*, 2(3-4):259–290.

Potts, C. N. e Wassenhove, L. N. V. (1985). A Branch and Bound Algorithm for the Total Weighted Tardiness Problem. *Operations Research*.

Uchoa, E., Fukasawa, R., Lysgaard, J., Pessoa, A., de Aragão, M. P., e Andrade, D. (2006). Robust branch-cut-and-price for the Capacitated Minimum Spanning Tree problem over a large extended formulation. *Mathematical Programming*, 112(2):443–472.