V Simpósio de Iniciação Científica 2022
Ciência, Tecnologia e Inovação no Brasil

Problema do empacotamento bidimensional

Charbel Daher Boulos(IC)[1], Pedro Hernique Bianco Del Hokama (PQ)[2], Mário César San Felice(PQ)[3]
[1] Instituto de Engenharia de Sistemas e Tecnologia da Informação - Universidade Federal de Itajubá
[2] Instituto de Matemática e Computação - Universidade Federal de Itajubá
[3] Departamento de Computação - Universidade Federal de São Carlos

Palavras-chave: Algoritmo, Contêiner, Eficiência, Empacotamento, Problema.

Introdução

Diversos problemas da realidade encontram obstáculos durante a busca de uma possível solução. Como, por exemplo, uma empresa de entrega de encomendas deve traçar o caminho que passe por todos os destinos visando minimizar o custo da gasolina. Ou também, uma empresa deseja transportar um conjunto de itens e a forma mais eficiente para isso é utilizando um contêiner, então o objetivo desta empresa é empacotar o número máximo de itens nesse contêiner. Estes são exemplos de problemas que podem ser resolvidos com otimização combinatória.

Neste projeto será abordado o problema do empacotamento 2D ou bidimensional. A base deste problema é encontrar uma solução para o empacotamento de um conjunto de itens em um contêiner de duas dimensões. O objetivo desta pesquisa é abordar conceitos e desenvolver lógicas e algoritmos que encontrem a solução desse problema de uma maneira mais eficaz.

Metodologia

O problema do empacotamento 2D tem como entradas um contêiner C com altura H e largura W, um conjunto de itens I o qual cada item tem sua altura h_i e sua largura w_i. O objetivo do problema é encontrar a melhor solução de empacotamento, ou seja, deseja-se empacotar o maior número de itens do conjunto I respeitando os limites de C e a regra da sobreposição, onde um item i só poderá ser empacotado em um intervalo que ambas dimensões não tenham outro item j empacotado.

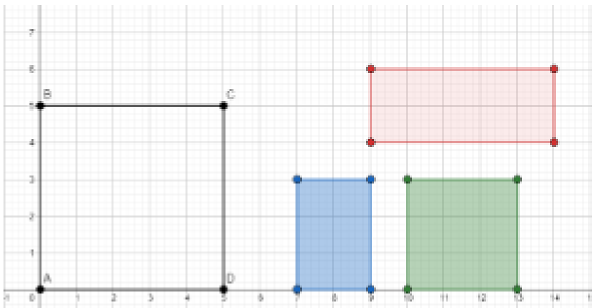

Figura 1 - Exemplo de instância do problema de empacotamento 2D

Na figura 1, é mostrado um exemplo de instância do problema de empacotamento 2D onde o contêiner C possui uma altura igual a 5 e uma largura igual a 5 também. Neste exemplo o conjunto de itens I possui 3 itens e cada um com sua respectiva altura e largura.

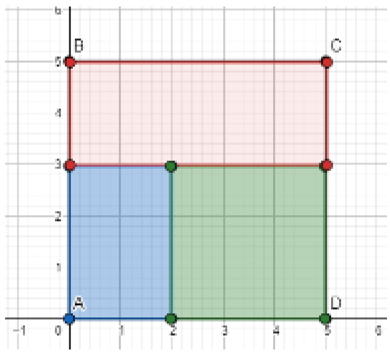

Figura 2 - Exemplo de uma solução possível para o exemplo da figura 1

Para resolver problemas de otimização é possível utilizar variados métodos como Programação Linear Inteira, programação por restrições, entre outros. Neste projeto será usado a programação por restrições ou CSP (do inglês Constraint solving programming) que trabalha com um conjunto de variáveis as quais cada uma tem um conjunto finito de possíveis soluções, denominadas por domínio (nomenclatura D). Assim, com esse tipo de programação é possível com uma ou mais restrições reduzir o domínio de cada variável, ou seja, restringi-las de determinados valores.

No caso do problema do empacotamento 2D, com as especificações dadas anteriormente, serão utilizadas duas variáveis para cada item *i*, *Xi* que representa a posição do item *i* no eixo x, ou seja, no eixo da largura e *Yi* que será a posição do item *i* no eixo y que representa o eixo da altura. O domínio dessas variáveis será D(Xi) = {0, ..., W − wi} para largura e D(Yi) = {0, ..., H − hi} para altura. Estes domínios garantem que nenhum item ultrapasse os limites de C. E também, para garantir a não sobreposição dos itens, será adicionado a seguinte restrição para cada par de item *i* e *j*:

$$[X_i + w_i \leq X_j] \text{ ou } [X_j + w_j \leq X_i] \text{ ou } [Y_i + h_i \leq Y_j] \text{ ou } [Y_j + h_j \leq Y_i]$$

Ao longo do projeto foram abordados algumas formas de fazer com que este domínio inicial das variáveis fossem reduzidos, denominadas Padrões. Um Padrão é um conjunto de pontos formado por diferentes critérios, porém, com o mesmo objetivo que é encontrar uma solução para empacotar todos os itens no contêiner. Os Padrões analisados neste projeto foram:

- **Padrão Normal**

Cada item *i* possui seu conjunto *Ni* de pontos encontrados pelo Padrão Normal, onde para encontrar esses pontos basta listar os pontos mais à esquerda que este item pode ser empacotado, ou também:

$$N_i = \left\{ x = \sum_{j \in I \setminus \{i\}} \omega_j \xi_j : 0 \leq x \leq W, \xi_j \in \{0,1\}, \text{para } j \in I \right\}$$

Portanto, todos os pontos que o item analisado possa ser encaixado, sem sobrepor outro item e respeitando as margens do contêiner, são adicionados ao conjunto.

Sendo assim, quando for encontrado todos os Padrões e os unindo em um só conjunto, obtém-se o Padrão Normal:

$$N = \{x \in N_i : x \leq W - \omega_{min}, \forall i \in I\}.$$

- **Padrão Boschetti**

O Padrão Boschetti é uma redução do Padrão visto anteriormente, portanto, esse método visa aumentar a eficiência para encontrar a solução por meio da redução de pontos, logo, B ⊆ N. Para encontrar os pontos do Padrão Boschetti é preciso seguir a mesma linha do Padrão Normal, porém, agora para formar o conjunto *B* de um item *i* é analisado se é possível empacotar os outros itens entre 0 e *H* - *hi* para altura e entre *W* - *wi* para largura.

$$\beta_i = \left\{ x = \sum_{j \in I \setminus \{i\}} \omega_j \xi_j : 0 \leq x \leq W - \omega_i, \xi_j \in \{0,1\}, \text{para } j \in I \setminus \{i\} \right\}$$

Feito este procedimento com cada item, basta fazer a união deles e, assim, obtendo o conjunto geral de pontos do Padrão Boschetti.

- Padrão Meet in the Middle

O Padrão Meet in the Middle ou MIM é a redução do Padrão Boschetti, ou seja, |M| ≤ |B| (BARTÁK, 1999), mas essa prova foge ao escopo deste projeto. O conjunto de uma dimensão deste Padrão, diferente dos anteriores, é formado pela junção de outros dois conjuntos, além de utilizar um parâmetro de limite denominado pela letra *t* (do inglês, threshold). A partir deste limite traçado, é obtido o conjunto do Padrão da esquerda *Lit* e o conjunto do Padrão da direita deste limite *Rit*. Os elementos destes conjuntos são obtidos da seguinte maneira:

$$L_{it} = \left\{ x = \sum_{j \in I \setminus \{i\}} \omega_j \xi_j : 0 \leq x \leq \min\{t-1, W-\omega_i\}, \xi_j \in \{0,1\}, \text{para } j \in I \setminus \{i\} \right\}$$

$$R_{it} = \left\{ W - \omega_i - x : x = \sum_{j \in I \setminus \{i\}} \omega_j \xi_j, 0 \leq x \leq W - \omega_i - t, \xi_j \in \{0,1\}, \text{para } j \in I \setminus \{i\} \right\}$$

Encontrando os dois conjuntos se obtém o conjunto de cada item para largura que é a união do conjunto da esquerda e da direita, removendo os itens repetidos:

$$M_{itW} = L_{it} \cup R_{it}.$$

Enfim, para encontrar o conjunto geral da largura, basta fazer a união de todos os conjuntos gerais de cada item:

$$M_{tW} = \bigcup_{i \in I} M_{itW}$$

Para encontrar os conjuntos da altura será utilizado o mesmo limite e a mesma lógica, assim, acima do limite terá o conjunto *Uit* (equivalente ao da direita) e abaixo o conjunto *Dit* (equivalente ao da esquerda). Da mesma forma que foi feita na largura, para altura será feito a união dos conjuntos de cima e debaixo para encontrar o conjunto geral da altura.

$$M_{itH} = D_{it} \cup U_{it}$$

Assim:

$$M_{tH} = \bigcup_{i \in I} M_{itH}$$

Então, para se ter o conjunto do Padrão Meet in the Middle basta fazer a união dos conjuntos gerais de largura e de altura removendo os itens repetidos. Logo, se tem:

$$M_t = M_{tW} \cup M_{tH}$$

Por fim, para analisar os métodos apresentados de forma prática, foram desenvolvidos algoritmos para encontrar os conjuntos de cada Padrão.

Resultados e discussão

Após a análise e desenvolvimento destes algoritmos discutidos anteriormente, oito instâncias distintas foram postas à prova a fim de chegar a uma conclusão sobre a teoria abordada neste projeto, sendo obtidos os seguintes resultados:

- Ordenação pelo item de maior área:

	DSP	DB	λ	DMIM	τ
Instância01	0.757	1.665	119.95	0.722	-4.62
Instância02	0.833	0.842	1.08	0.937	12.48
Instância03	1.376	1.144	-16.86	1.245	-9.52
Instância04	1.222	1.171	-4.17	0.839	-31.34
Instância05	0.997	0.927	-7.02	0.949	-4.81
Instância06	0.911	0.776	-14.82	0.79	-13.28
Instância07	1.506	1.515	0.60	1.561	3.65
Instância08	0.48	0.487	1.46	0.407	-15.21

- Ordenação pelo item mais alto:

	DSP	DB	λ	DMIM	τ
Instância01	0.866	0.647	-25.29	0.653	-24.60
Instância02	1.046	0.722	-30.98	0.862	-17.59
Instância03	1.254	0.653	-47.93	0.643	-48.72
Instância04	1.759	1.065	-39.45	0.907	-48.44
Instância05	1.1	0.832	-24.36	0.864	-21.45
Instância06	0.916	0.634	-30.79	0.591	-35.48
Instância07	1.621	1.325	-18.26	1.497	-7.65
Instância08	0.684	0.476	-30.41	0.456	-33.33

- Ordenação pelo item mais comprido:

	DSP	DB	λ	DMIM	τ
Instância01	5.664	1.877	-66.86	2.162	-61.83
Instância02	0.523	0.574	9.75	0.583	11.47
Instância03	1.401	1.258	-10.21	1.318	-5.92
Instância04	0.728	0.649	-10.85	0.716	-1.65
Instância05	0.705	0.574	-18.58	0.647	-8.23
Instância06	0.673	0.597	-11.29	0.65	-3.42
Instância07	1.249	1.21	-3.12	1.391	11.37
Instância08	0.531	0.523	-1.51	0.532	0.19

A coluna representada por “DSP” apresenta o tempo, em segundos, que o algoritmo sem padrões encontrou a solução para cada instância. Já a coluna “DB” representa o tempo, em segundos, do algoritmo com o domínio encontrado pelo Padrão Boschetti. Isto também se aplica para a coluna “DMIM”, porém para o algoritmo com o domínio encontrado pelo Padrão Meet in the Middle.

Em relação às colunas representadas por *tau* e *lambda* informam, respectivamente, a comparação do tempo da solução no domínio do Padrão MIM com a solução sem Padrão, e a comparação do tempo da solução no domínio do Padrão Boschetti com a solução de domínio inicial, sendo ambas calculadas em porcentagem. Portanto, os resultados maiores do que zero são consequências de instâncias em que a solução com domínio inicial foi melhor que as soluções com Padrões. Já os resultados menores que zero são as instâncias em que as soluções mais eficientes foram as encontradas pelos Padrões.

Conclusões

Portanto, após compreender o que são Padrões, desenvolver diferentes tipos e testá-los na prática, foi possível concluir que os Padrões realmente trazem uma mais eficiência na busca pela solução do problema de empacotamento 2D. Ademais foi visto que o conjunto de pontos formado pelo Padrão Meet in the Middle é uma redução do conjunto formado pelo Padrão Boschetti, ou seja, $|M| \leq |B|$ e isto também foi provado, sendo que, por mais que tenha tido um certo equilíbrio entre eles nas tabelas de resultado, o Padrão MIM saiu-se melhor no geral, pois na ordenação pela área o Padrão MIM superou a solução sem Padrão em 6 de 8 instâncias, já o Padrão Boschetti superou em apenas 4. Na ordenação pelo item mais alto, ambos Padrões superaram todas as instâncias sem Padrão. Na ordenação pelo item mais comprido o resultado foi semelhante, no entanto, foram apenas 6 instâncias superadas por ambos Padrões.

Agradecimento

Ao Prof. Dr. Pedro Hokama e ao Prof. Dr. Mário César San Felice pela orientação. Ao Programa Institucional de Bolsas de Iniciação Científica (PIBIC) UNIFEI/UNIÃO pelo apoio financeiro.

Referências

BARTÁK, Roman. Constraint programming: In pursuit of the holy grail. In: **Proceedings of the Week of Doctoral Students (WDS99)**. Prague: MatFyzPress, 1999. p. 555-564.

CÔTÉ, Jean-François; IORI, Manuel. The meet-in-the-middle principle for cutting and packing problems. **INFORMS Journal on Computing**, v. 30, n. 4, p. 646-661, 2018.

CONSTRAINT propagation. *In*: **ILOG CPLEX Optimization Studio**. [*S. l.*], 25 nov. 2021. Disponível em: https://www.ibm.com/docs/en/icos/20.1.0?topic=optimizer-constraint-propagation. Acesso em: 14 set. 2022.