Uma Heurística Baseada em Programação Linear Inteira para o Problema do Empacotamento de Soma Mínima

Rafael C. S. Schouery[1*]

[1]Instituto de Computação – Universidade Estadual de Campinas (Unicamp)
Av. Albert Einstein, 1251 – Cidade Universitária, Campinas – SP, 13083-852

rafael@ic.unicamp.br

Abstract. *We consider the Min-Sum Packing Problem, in which, given a set of items with their respective weights, we want to pack them into bins of limited capacity to minimize the sum, over all items, of the index of the bin where the item was packed. We present a heuristic based on Integer Linear Programming that, even with a time limit of just one minute, was able to find solutions with an average gap of only 1.15% for approximately 64% of the considered instances. For the remaining instances, where it was not possible to calculate an upper bound, the heuristic was able to improve the solution by 13% on average when compared with the initial heuristic used.*

Resumo. *Consideramos o Problema do Empacotamento de Soma Mínima, em que, dado um conjunto de itens com os seus respectivos pesos, desejamos empacotá-los em recipientes de capacidade limitada com o objetivo de minimizar a soma, sobre todos os itens, do índice do recipiente onde o item foi empacotado. Apresentamos uma heurística baseada em Programação Linear Inteira que mesmo com um tempo limite de um minuto foi capaz de encontrar soluções com um gap médio de 1,15% para 64% das instâncias consideradas. Para as instâncias restantes, onde não foi possível calcular um limitante superior a tempo, a heurística foi capaz de melhorar a solução em 13% em média quando comparado com a heurística inicial utilizada.*

1. Introdução

O Problema do Empacotamento é um clássico problema de otimização combinatória onde são dados como entrada um inteiro W e um conjunto $I = [n]$ de itens em que w_i é o peso do item i. O objetivo é encontrar uma partição $\{B_1, B_2, \ldots, B_k\}$ de I com k mínimo tal que $\sum_{i \in B_j} w_i \leq W$ para todo $j \in [k]$.

Trata-se de um problema de grande importância para a indústria e para o setor de serviços dada sua aplicabilidade em problemas de corte de materiais ou empacotamento de produtos. Por exemplo, cada elemento da partição pode ser visto como uma barra de ferro de comprimento W que é cortada em diversos pontos para produzir barras menores. Nesse sentido, o objetivo é usar a menor quantidade de barras para atender a demanda de produção ou, equivalentemente, minimizar a perda de material causada pelo corte.

[1]Este trabalho foi parcialmente financiado pelo processo 2015/11937–9, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e pelos processos 311039/2020–0, 425340/2016–3 e 425806/2018–9, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Já o Problema do Empacotamento de Soma Mínima é uma variante do Problema do Empacotamento em que o objetivo não é minimizar k, mas sim minimizar $\sum_{j=1}^{k} j \cdot |B_j|$. Assim, se considerarmos que cada barra leva um tempo constante t para ser cortada, desejamos minimizar o tempo médio de produção dos itens, já que um item em B_j levará tempo $j \cdot t$ para ser produzido. Desta forma, o Problema do Empacotamento de Soma Mínima é interessante quando não estamos tão preocupados com a perda de material, mas sim com o tempo de produção ou entrega dos itens produzidos.

O Problema do Empacotamento de Soma Mínima foi introduzido por [Epstein et al. 2018], que estudaram o problema do ponto de vista de algoritmos de aproximação e de algoritmos online. Em particular, os autores apresentaram um esquema de aproximação polinomial para o problema, e provaram que alguns algoritmos clássicos do Problema do Empacotamento não apresentam razão de aproximação assintótica finita. Por fim, provaram que o algoritmo Next Fit Increasing tem uma razão de aproximação absoluta de 2 e uma razão de aproximação assintótica de 1.6188 e projetaram um algoritmo misto entre Next Fit Increasing e First Fit Decreasing com razão de aproximação assintótica de 1.5604. Este trabalho é o único a abordar o Problema do Empacotamento de Soma Mínima na literatura até o presente momento.

2. Modelo Matemático e Heurística

Nossa formulação segue o modelo de [Gilmore and Gomory 1961] que considera o Problema do Empacotamento como um problema de cobertura por conjuntos que representam todas as possíveis formas de empacotar itens respeitando a capacidade do recipiente.

Para evitar simetrias, neste trabalho nós generalizamos o Problema do Empacotamento de Soma Mínima para considerar que cada item $i \in I$ tem uma demanda d_i indicando o número de vezes que tal item precisa ser empacotado. Assim, em cada recipiente empacotamos um multiconjunto de itens. Formalmente, seja P um multiconjunto de itens tal que $a_{i,P}$ é a quantidade de vezes que o item i está presente em P. Temos que P é um *padrão de empacotamento* se e somente se $\sum_{i \in I} a_{i,P} \cdot w_i \leq W$. Ademais, denotamos por $\mathcal{P}$ o conjunto de todos os padrões de empacotamento.

Desta forma, podemos escrever a seguinte formulação para o Problema do Empacotamento de Soma Mínima, em que B é um número inteiro suficientemente grande[1], que indica o número máximo de recipientes que podem ser usados e a variável $x_{j,P}$ que indica se no recipiente $j \in [B]$ utilizamos o padrão de empacotamento $P \in \mathcal{P}$.

$$
\begin{aligned}
\text{(GG)} \quad & \text{minimize} \sum_{j=1}^{B} \sum_{P \in \mathcal{P}} j \cdot |P| \cdot x_{j,P} && && (1)\\
& \text{sujeito a} \quad \sum_{j=1}^{B} \sum_{P \in \mathcal{P}} a_{i,P}\, x_{j,P} \geq d_i && \forall\, i \in I && (2)\\
& \sum_{P \in \mathcal{P}} x_{j,P} \leq 1 && \forall\, j \in [B] && (3)\\
& x_{j,P} \in \{0,1\} && \forall\, j \in [B] && (4)
\end{aligned}
$$

[1]Em nosso experimentos, tomamos $B = \lfloor \sum_{i \in I} w_i / \lceil W/2 \rceil \rfloor + 1$, por ser um limitante superior no número de recipientes em uma solução ótima.

em que a restrição 2 garante que todos os itens são empacotados, a restrição 4 garante que cada índice tem apenas um padrão associado e a função objetivo 1 contabiliza quantos itens têm um índice j.

Infelizmente essa formulação pode conter um número exponencial de variáveis. Porém, podemos gerar as variáveis apenas quando necessário, através de um processo chamado de geração de colunas. Para tanto, analisamos o dual da relaxação linear:

$$
\begin{aligned}
\text{(D)} \quad & \text{maximize} \sum_{i \in I} d_i \alpha_i - \sum_{j=1}^{B} \beta_j && && (5)\\
& \text{sujeito a} \quad \sum_{i \in I} a_{i,P}\, \alpha_i - \beta_j \leq j \cdot |P| && \forall\, j \in [B], \forall\, P \in \mathcal{P} && (6)\\
& \alpha_i \geq 0 && \forall\, i \in [i] && (7)\\
& \beta_j \geq 0 && \forall\, j \in [B] && (8)
\end{aligned}
$$

Note que, dada uma solução não necessariamente viável (α, β), o problema de encontrar uma restrição violada do conjunto (6) para um j fixo é equivalente a encontrar a solução ótima do Problema da Mochila onde o valor do item i é $\alpha_i - j$ e verificar se o valor desta solução é maior do que β_j, o que pode ser feito utilizando o clássico algoritmo de programação dinâmica para o Problema da Mochila.

Desta forma, propomos a seguinte heurística:

1. Inicie (D) com o conjunto das restrições encontradas pela heurística FFDS.
2. Calcule uma solução ótima do dual considerando apenas as restrições geradas.
3. Se existir uma restrição violada, adicione ao conjunto de restrições.
4. Se não tiver atingido 90% do tempo limite, volte para o passo 2.
5. Calcule uma solução inteira de (GG) restrito às variáveis correspondentes a restrições geradas nos passos anteriores com o tempo limite restante.

Em nossa heurística utilizamos a heurística First-Fit Decreasing Sorted (FFDS) no primeiro passo, descrita a seguir.

Considere, por simplicidade, que cada item i tem d_i cópias a serem empacotadas, de forma que podemos falar do empacotamento de uma certa cópia de um certo item. No First-Fit Decreasing Sorted, os itens são ordenados em ordem decrescente de peso, e as copias dos itens são empacotadas uma-a-uma no primeiro recipiente em que ela couber (abrindo um novo recipiente caso necessário). Por fim, ao final do algoritmo, ordenamos os recipientes em ordem decrescente da quantidade de itens, potencialmente melhorando o valor da solução.

Note que a heurística proposta não é um algoritmo exato para o Problema do Empacotamento de Soma Mínima pois pode ser necessário gerar novos padrões de empacotamento ao subdividir o problema em um algoritmo de *branch-and-bound*, o que não é feito pela heurística pois levaria a um tempo de execução maior. De toda forma, esta é uma heurística que apresentou bons resultados na prática, como apresentamos a seguir.

3. Experimentos Computacionais

Os experimentos computacionais foram executados em uma máquina Intel(R) Xeon(R) E5–2630 com 64 GB de memória RAM rodando Ubuntu Linux 18.04.2 LTS. Toda a implementação foi feita em linguagem C++ e o *solver* Gurobi foi utilizado.

Em termos de instâncias, utilizamos a classe de instâncias “Randomly Generated Instances” fornecidas por Delorme et al. [Delorme et al. 2016] sendo esta parte da BPPLib, uma conhecida biblioteca de instâncias para o Problema de Empacotamento [Delorme et al. 2022].

O algoritmo teve apenas um minuto para executar em cada instância. Das 3840 instâncias da classe, o algoritmo foi incapaz de terminar a execução da primeira fase (geração de restrições – passos 1 a 4) em 1384 destas (aproximadamente 36%). Nesse caso o algoritmo encontra uma solução pelo menos tão boa quanta a da heurística FFDS, podendo ser melhor dependendo dos padrões gerados mas sem nenhuma garantia de qualidade já que um limitante dual não foi calculado. Em nossos experimentos, a melhora média em relação a heurística FFDS foi de 13% nestas instâncias.

Das 2456 instâncias onde a primeira fase terminou, em 1668 das instâncias (43,4% do total) ele foi capaz de terminar a execução do Branch-and-Bound a tempo e no restante (20,5%) o Branch-and-Bound foi interrompido por causa do limite do tempo (um minuto menos o tempo necessário para gerar os padrões de empacotamento). Entre as instâncias onde foi possível calcular um limitante dual, o gap[2] médio observado foi de 1,15% com um desvio padrão de 0,021. O maior gap observado foi de 29,74%, porém 75% das instâncias tiveram um gap não superior a 1,36%.

4. Conclusão

Neste texto apresentamos uma heurística baseada em geração de colunas para o Problema do Empacotamento de Soma Mínima. Os resultados indicam que a heurística foi capaz de encontrar, rapidamente, soluções com um gap pequeno em 64% das instâncias consideradas. Nas instâncias restantes, onde um limitante superior é desconhecido, foi possível melhorar a solução inicialmente dada em 13% em média.

Desta forma, os próximos objetivos do trabalho são: otimizar o processo de geração de padrões potencialmente utilizando outros algoritmos para o Problema da Mochila ou reutilizando padrões que foram encontrados para um determinado índice; fortalecer a formulação utilizando cortes; comparar a heurística proposta com um algoritmo genético; implementar um algoritmo de Branch-and-Cut-and-Price para resolver o problema de forma exata; e explorar outras formulações baseadas em geração de coluna.

Referências

Delorme, M., Iori, M., and Martello, S. (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, 255(1):1–20.

Delorme, M., Iori, M., and Martello, S. (2022). BPPLIB — A Bin Packing Problem Library. `http://or.dei.unibo.it/library/bpplib`. Accessed: 2022-03-15.

Epstein, L., Johnson, D. S., and Levin, A. (2018). Min-sum bin packing. *Journal of Combinatorial Optimization*, 36:508–531.

Gilmore, P. C. and Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849–859.

[2]Calculado como $(s - b)/s$ onde s é o valor da solução e b é o limitante dual.