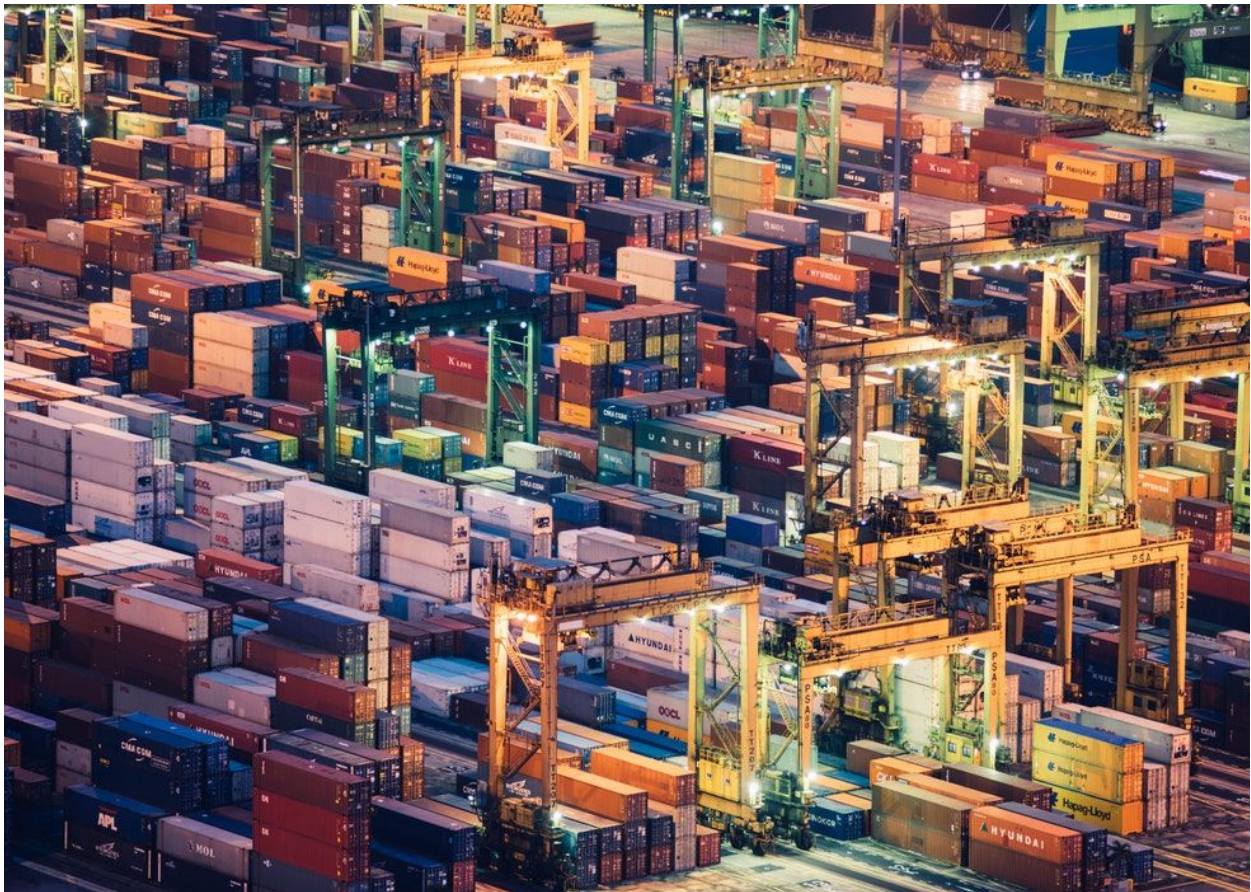


O Problema do Importador

Um desafio logístico no universo da computação



Por Fábio Melo e Victor Morais

Introdução

Otimizar o uso de espaço. Diminuir gastos, Maximizar Lucros. Estas são só algumas das motivações que impulsionam a sociedade capitalista desde a revolução industrial. Uma das maiores ferramentas que temos para a resolução deste problema é o estudo da logística: um campo do saber no qual um dos mais importantes desafios é “qual é a melhor forma de organizar o espaço que tenho e com isso maximizar o lucro?” Enfim, sem mais delongas: o que é o **Problema do Importador**?

A Proposta: Importação como um problema de otimização combinatória.

Uma Definição

Supondo que uma empresa de logística tenha um conjunto n de objetos com valores e tamanhos distintos, e um conjunto finito de caixas, cada uma delas possuindo um tamanho máximo finito. queremos enviá-las do exterior para uma central de distribuição, mas tal processo conta com as seguintes restrições:

- Cada caixa possui um custo fixo.
- Quando uma caixa é interceptada pela alfândega, é cobrado uma taxa calculada com base no valor total dos itens contidos nela.
- Caso o valor total dos itens contidos na caixa seja menor que um certo valor m definido, é isenta a taxa.

Nesse projeto, iremos trabalhar com heurísticas que tem como objetivo maximizar os possíveis lucros no empacotamento destas caixas.

Complexidade do Problema

O Problema descrito neste artigo é intrinsecamente um problema de decisão baseado no problema clássico do Empacotamento (Bin Packing). e assim como ele, faz parte da família dos **problemas NP**: não há como possivelmente resolvê-lo em tempo polinomial, podendo apenas ser calculados apenas em **aproximações à uma resposta ótima**.

E Como Provamos a satisfabilidade de suas soluções?

Supondo que exista uma solução S para o problema, composta por um conjunto de caixas, cada uma composta por um conjunto de itens. podemos verificar sua validade utilizando a seguinte condição:

- Verificamos se o somatório dos tamanhos dos itens de cada caixa é menor ou igual ao tamanho máximo de caixa pré-definido pela função.

Caso isso seja verdadeiro, podemos afirmar que a solução é **válida e satisfaz** as condições do problema, mas não que tal solução se aproxima da **ótima**, isto é, a melhor forma de descobrir a melhor solução é fazendo múltiplas permutações de possíveis soluções heurísticas e tentar refina-la através de meta-heurísticas.

Reduzindo o Problema: Bin Packing

Por ser uma variação de tal problema, podemos reduzir o problema proposto em um problema do empacotamento (bin packing) unidimensional, que é **reconhecidamente definido pela literatura como NP-Hard**, e definido de tal forma:

“ Supondo que tenhamos n objetos de tamanhos distintos, e vários recipientes que possuem uma capacidade V , coloque cada um dos objetos nos recipientes, de forma que o total de recipientes usados seja o menor possível. ”

Podemos reduzir um conjunto de soluções do problema da importador em um conjunto de soluções do problema do empacotamento, de forma que ela continue sendo satisfeita, pois ambas se tratam de conjuntos de conjuntos de itens, que são resolvidas da mesma forma.

Instâncias do Problema:

Por ser um problema original, definido neste artigo, não há referências de tal na literatura científica. para este projeto, desenvolvemos um programa que gera instâncias aleatórias baseadas nos seguintes preceitos:

- Instâncias com objetos de tamanho e valores aleatórios, dentro de limites especificados pela função.
- Instâncias gerados com base no **princípio de pareto**: onde 80% dos itens correspondem aos com menores valor e peso, e os 20% restantes correspondem aos que possui a maior razão entre valor e peso. (ou vice-versa)
- As instâncias possuem quantidades de itens distintos, para que possamos testar as performance dos algoritmos heurísticos
- Alguns destes exemplos gerados podem ser encontrados em:
<https://gist.github.com/fabio-melo/2914ba3921e8bee4c1e0a760b1f51092>

Referências Bibliográficas:

- How to prove that a problem is NP complete?, disponível em: <https://stackoverflow.com/questions/4294270/how-to-prove-that-a-problem-is-np-complete>
- "Strong" NP-Completeness Results: Motivation, Examples, and Implications, disponível em: <https://dl.acm.org/citation.cfm?doid=322077.322090>
- Foto (royalty-free) por chuttersnap, Unsplash, disponível em: <https://unsplash.com/photos/xewrfLD8emE>