

# TCI - PROBLEMA DA MOCHILA MÚLTIPLA

Adriano Rodrigues Alves  
Julio Huang  
Luís Fernando Leite França  
Patrick Escorsi Silva

Setembro 2021

## 1 Introdução

O objetivo deste trabalho foi avaliar a qualidade das soluções geradas por heurísticas ingênuas para o problema da mochila múltipla. Para isso, nos foi fornecido um algoritmo exato do tipo *branch-and-bound* e, ao decorrer do trabalho, foram implementadas duas heurísticas, uma denominada heurística gulosa e uma outra heurística randômica.

Descreveremos neste documento os detalhes sobre o projeto e como foi feita a implementação das partes fundamentais do trabalho, como as heurísticas, os testes e quais foram os resultados obtidos.

## 2 Heurísticas

Nesta seção nós descreveremos as duas heurísticas implementadas para o desenvolvimento deste trabalho.

### 2.1 Heurística Gulosa

Para a heurística gulosa, haviam algumas opções, mas a ideia, no fundo, era fazer uma ordenação seguindo um critério que parecia ser mais vantajoso na hora de preencher as mochilas. Dito isso, realizamos a ordenação por meio da função *qsort*, usando uma função de comparação auxiliar chamada *Comparador*. Para o critério de comparação, nós utilizamos três opções: ordenando de forma crescente pelo **valor** de cada item, ordenando de forma decrescente pelo **peso** de cada item e ordenando de forma crescente pela **razão** entre o valor e o peso de cada item. Depois disso, a heurística percorre toda a lista de itens e verifica se o item cabe ou não dentro de alguma mochila. Ou a heurística coloca o item na primeira mochila que couber ou o item será descartado. Por descartado, nos referimos ao item não ser mais verificado. Ao final, após passar por toda a lista de itens, a heurística entrega um subconjunto dos melhores itens, além do valor total da mochila.

### 2.2 Heurística Aleatória

A heurística aleatória baseia-se em sortear aleatoriamente um item qualquer e ir preenchendo as mochilas, até elas encherem. Caso o item sorteado tenha um peso maior que o suportado pela mochila selecionada, o algoritmo tentará encontrar uma outra mochila que o suporte, mas se caso não encontrar, este item será trocado com o último item da lista por meio da função *troca*, e após isso, decrementamos o número de possíveis itens, fazendo com que este item não seja escolhido novamente pelo função *RandomInteger*. Ao final, a heurística entrega um subconjunto dos itens, além do valor total da mochila.

## 3 Testes e resultados

Nesta seção nós apresentaremos quais testes foram realizados para o desenvolvimento do trabalho e quais resultados obtivemos a partir destes testes.

### 3.1 T1

O primeiro teste foi feito para verificar a qualidade da solução PLI limitada a um tempo de 5 minutos, em comparação com a relaxação linear. Para verificarmos isso, calculamos o gap entre estas soluções e apresentaremos os resultados obtidos nesta comparação.

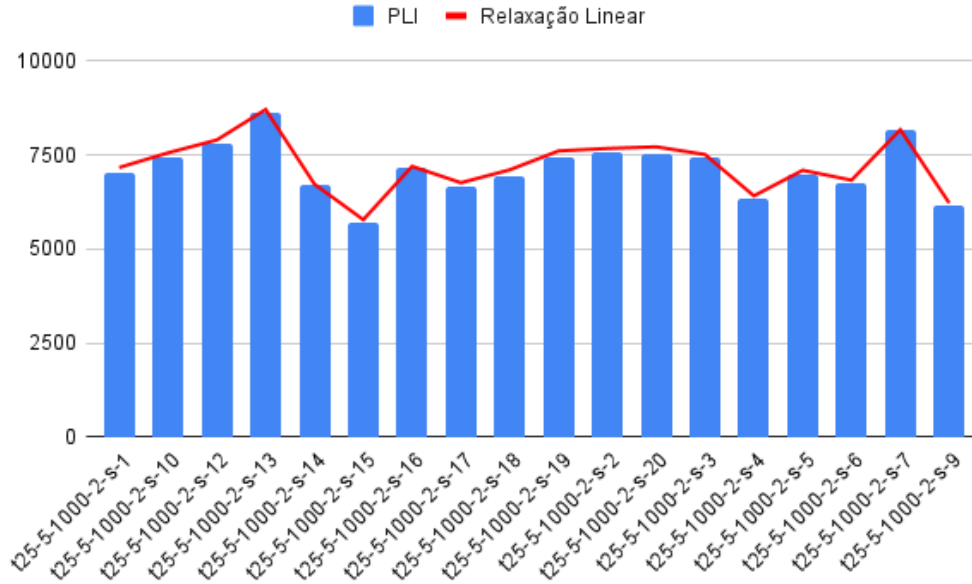


Figure 1: Comparação entre a solução PLI e a relaxação linear

No gráfico acima, apresentamos apenas os testes que possuem 25 itens. Nele, é possível verificar que o gap entre as duas soluções foi baixo, mas podemos observar isso mais claramente quando calculamos o gap médio de todos os testes realizados, que ficou em 1%, enquanto o gap mínimo foi de 0% e o máximo de 2,67%.

Das 94 instâncias de testes, obtivemos 18 instâncias que foram resolvidas na otimidade, gastando em média 4,0597 segundos para serem resolvidos.

### 3.2 T2

No segundo teste verificamos a qualidade dos limitantes superiores dados pela relaxação linear. Tanto o gráfico, quanto os *gaps* ficaram muito parecidos com os valores encontrados no **T1**. O *gap* médio ficou em 1,01%, o total de instâncias que o *gap* foi zero foi 18 e o tempo médio gasto pela relaxação linear foi de 0,0161805 segundos.

### 3.3 T3

O terceiro teste foi cometido com objetivo de verificar a qualidade das soluções geradas pelas heurísticas gulosa e aleatória em comparação com a solução encontrada pela relaxação linear.

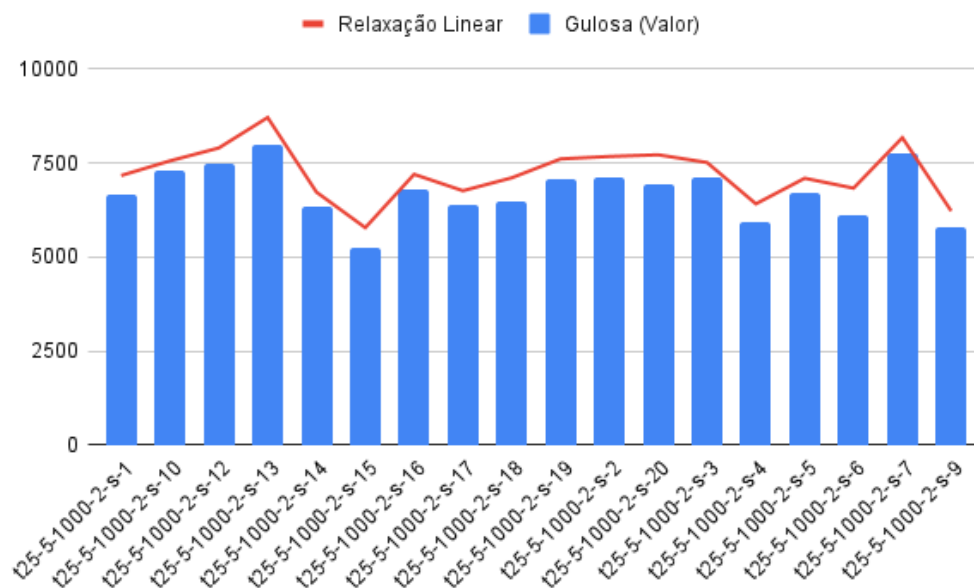


Figure 2: Comparação entre a Relaxação Linear e a Heurística Gulosa

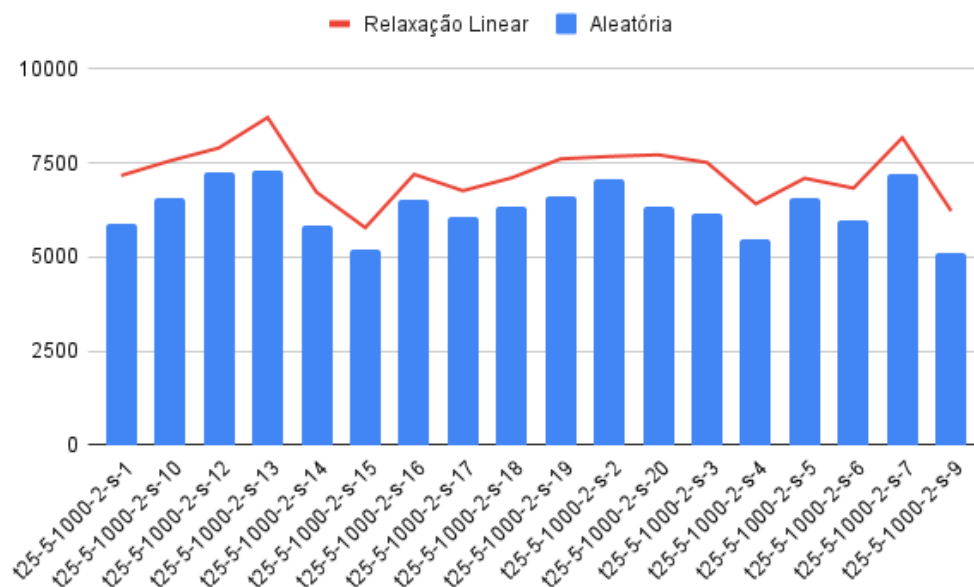


Figure 3: Comparação entre a Relaxação Linear e a Heurística Aleatória

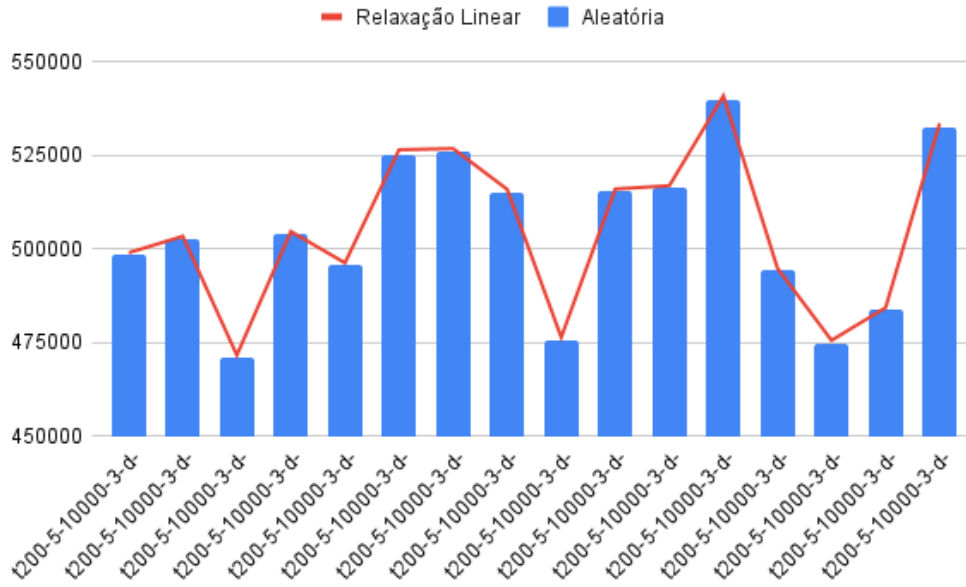


Figure 4: Comparação entre a Relaxação Linear e a Heurística Aleatória, com entrada de 200 itens

A partir de uma análise visual do gráfico, pode ser visto que a heurística Gulosa teve um desempenho levemente superior em relação à Aleatória, chegando mais perto do resultado da relaxação do que a outra. Apesar dessa diferença entre as duas, ambas tiveram um desempenho satisfatório, chegando bem perto do resultado da Relaxação em alguns casos, em especial, vale citar em entradas com 200 itens, onde a aleatório foi realmente muito bem, tanto em relação à heurística relaxada, quanto à gulosa.

O gap médio de todos os testes realizados na heurística aleatória ficou em 9,54%, o gap mínimo foi de 0,1% e o máximo de 18,11%. Na heurística gulosa o gap médio foi de 5,18%, o mínimo de 0,0% e o máximo de 10,52%.

Dentre as 94 instâncias de testes, a heurística aleatória obteve o melhor resultado em 11, enquanto a heurística gulosa gerou a melhor solução em 51 instâncias.

A média de tempo gasto para resolver uma instância de teste pela heurística aleatória é de 0,000019 segundos, já na heurística gulosa foi de 0,000036.

## 4 Conclusões

O desenvolvimento desse trabalho auxiliou na absorção e no aprofundamento do conhecimento referente aos problemas NP-difícil e a todo o processo para encontrar uma possível solução razoável para este tipo de problema.

A partir da análise feita pelo grupo, foi chegado à um consenso que ambas as heurísticas implementadas pelo grupo foram bem sucedidas em seus objetivos, mas cada uma com seus pontos fortes e fracos. No caso da gulosa, ela se saiu melhor com entradas de até 125 itens, porém com entradas maiores que essa, a aleatória teve resultados superiores, como demonstrado na figura 4, em que a aleatória chegou a resultados muito próximos em comparação com os da relaxação linear.