**Trabalho de Pesquisa Operacional - Bin Packing com Heurísticas Básicas**

Carlos Eduardo Ferreira - 2022101225

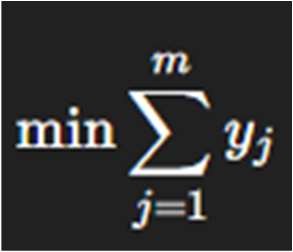
A -

Formulação baseada em itens e bins (Modelo de Gilmore-Gomory) Variáveis:

* X*ij* ∈ {0,1}: 1 se o item *i* é colocado no bin j, 0 caso contrário.
* Y*j* ∈ {0,1}: 1 se o bin *j* é usado, 0 caso contrário. Dados:
* *n*: número de itens.
* W*i*: peso (ou tamanho) do item *i*.
* *C*: capacidade do bin.
* *m*: número máximo de bins (pode ser igual a *n* inicialmente, no pior caso, cada item em um bin).

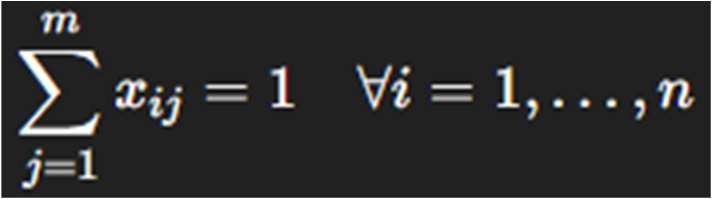
Função objetivo:

Minimizar o número de bins utilizados:

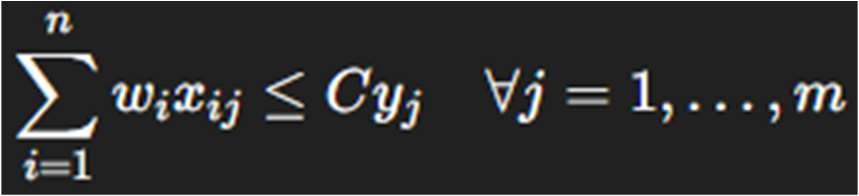


Sujeito a:

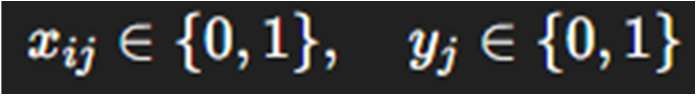
1. Cada item deve ser colocado em exatamente um bin:



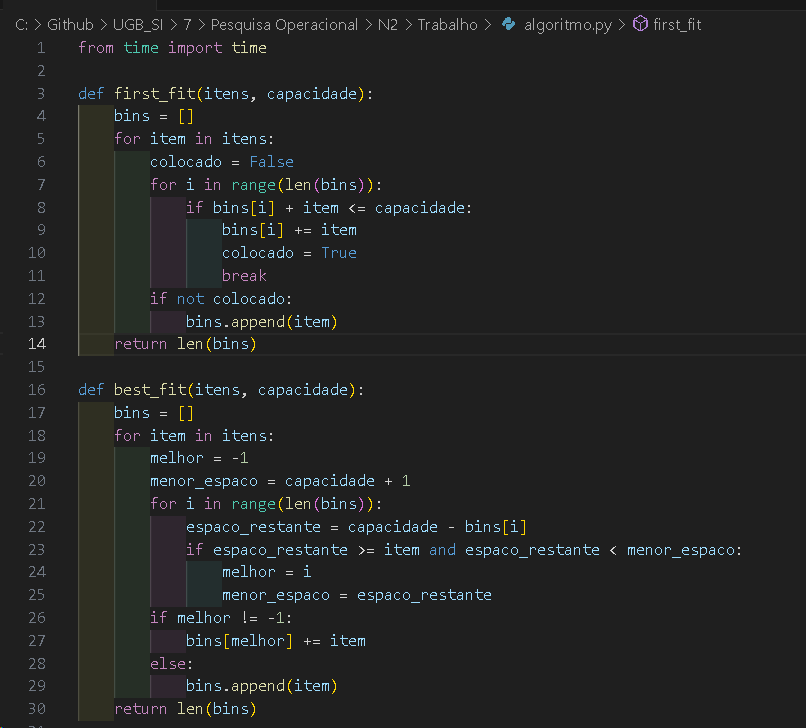
1. A soma dos pesos dos itens em cada bin não pode ultrapassar a capacidade:

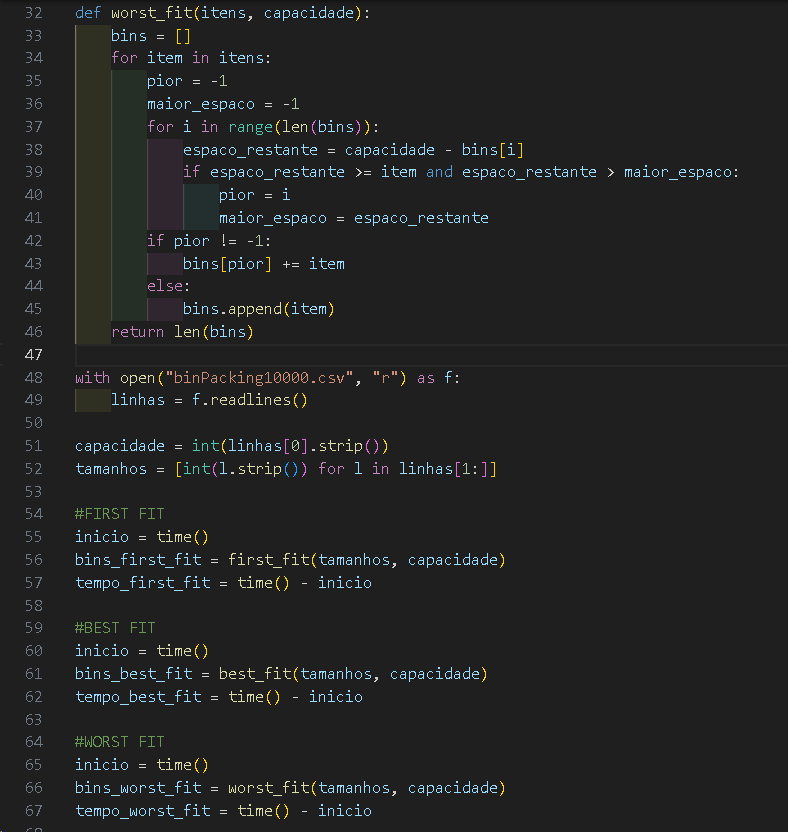


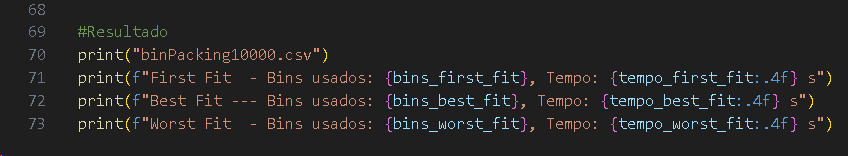
1. Variáveis binárias:



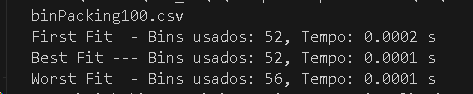
B -



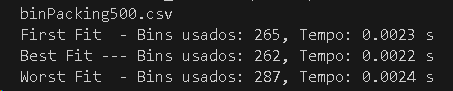




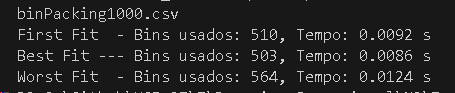
* + Resultado bin Packing 100:



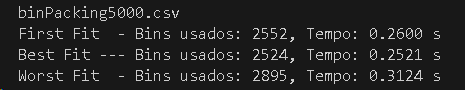
* + Resultado bin Packing 500:



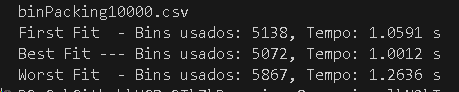
* + Resultado bin Packing 1000:



* + Resultado bin Packing 5000:

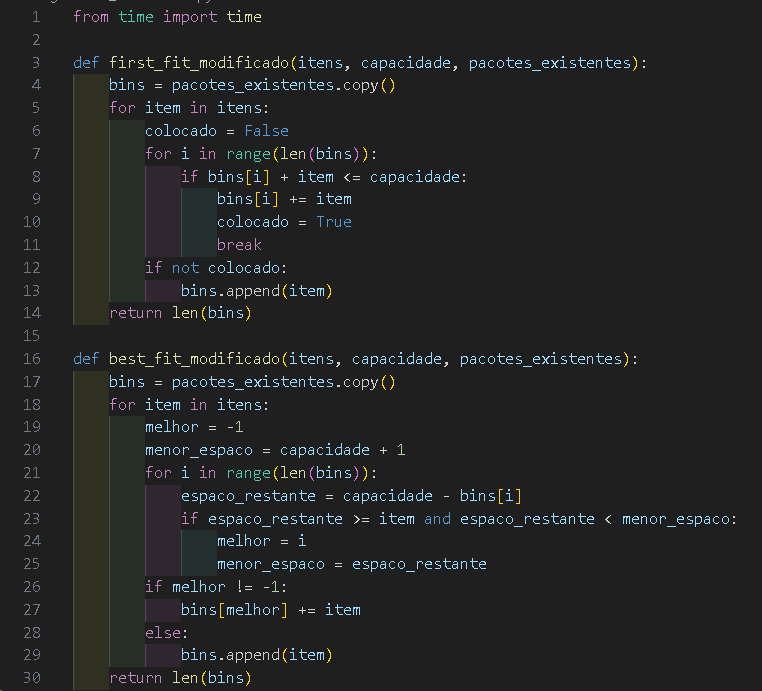


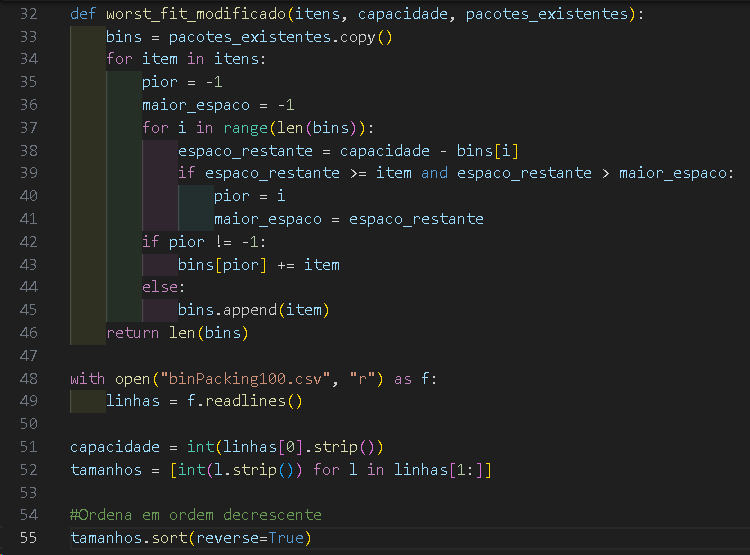
* + Resultado bin Packing 10000:

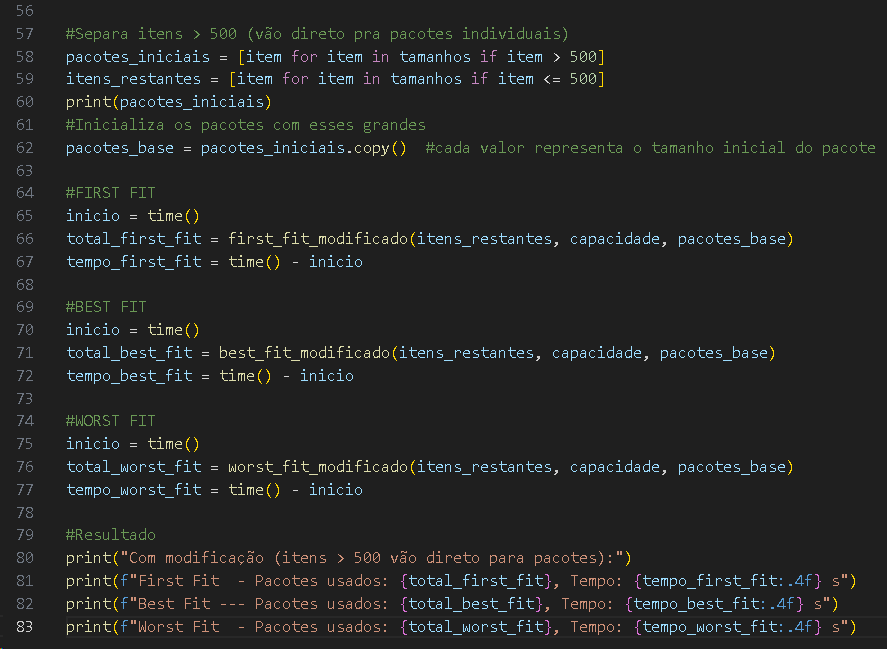


C -

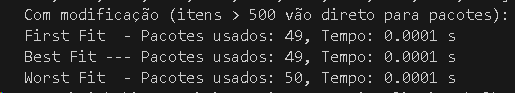
Código modificado:



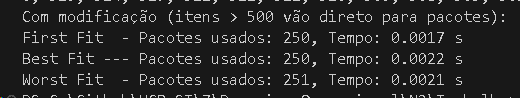




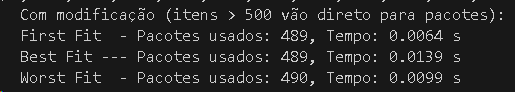
* + bin Packing 100:



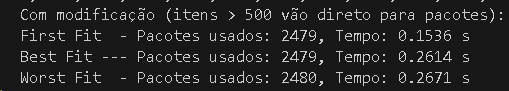
* + bin Packing 500:



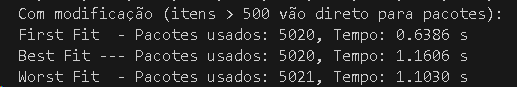
* + bin Packing 1000:



* + bin Packing 5000:



* + bin Packing 10000:



D -

Durante os testes com as heurísticas de Bin Packing — First Fit, Best Fit e Worst Fit — foram utilizados cinco arquivos com diferentes quantidades de itens (100, 500, 1000, 5000 e 10000). As análises aconteceram em duas etapas: primeiro com os algoritmos em sua forma original e, depois, com uma modificação proposta. Essa modificação consistia em ordenar os itens do maior para o menor e separar, logo de início, os itens com tamanho superior a 500 em pacotes próprios.

Tempo de Execução

No geral, o tempo de execução foi bastante baixo para todas as heurísticas, mesmo nos testes com 10.000 itens. Isso mostra que, em termos de desempenho, os algoritmos são bem eficientes.

O First Fit se destacou como o mais rápido, graças à sua lógica simples. Em seguida, vieram o Worst Fit e o Best Fit, com pequenas diferenças entre eles. Quando a modificação foi aplicada (com o pré-processamento dos itens maiores), o tempo aumentou um pouco, o que era esperado, já que há etapas extras antes da alocação final.

Eficiência no Uso dos Pacotes

Quando se trata de economizar espaço — ou seja, usar o menor número possível de pacotes —, o Best Fit foi o que teve o melhor desempenho na maioria dos testes. Já o Worst Fit não se saiu tão bem, pois acaba espalhando os itens de forma menos eficiente.

A modificação aplicada teve impactos diferentes dependendo do cenário. Em conjuntos com muitos itens grandes, o número de pacotes iniciais já era alto, o que dificultou ganhos significativos. Por outro lado, nos arquivos menores, a estratégia funcionou bem e ajudou, principalmente no caso do Best Fit, a reduzir o número total de pacotes usados.

Conclusão

O Best Fit se mostrou a heurística mais equilibrada, entregando bons resultados tanto em desempenho quanto em eficiência, especialmente quando utilizada com a modificação proposta. Já o First Fit, apesar de ser mais simples, continua sendo uma ótima escolha por sua rapidez e por entregar resultados próximos.

No fim das contas, a melhor escolha vai depender do tipo de dados que se está lidando especialmente da quantidade de itens grandes e do que se valoriza mais: a velocidade do processamento ou a economia no uso dos pacotes.

Tabela de comparativos de heurísticas:

