UNIVERSIDADE FEDERAL DE VIÇOSA - UFV - CAMPUS FLORESTAL

INSTITUTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO PRÁTICO 02:

"O ENSOPADO PERFEITO"

JOÃO ROBERTO MELO DOS SANTOS(3883)
MATHEUS NASCIMENTO PEIXOTO (4662)
MATHEUS NOGUEIRA MOREIRA (4668)

FLORESTAL – MG
2025

Sumário

1 - Introdução

Problemas de otimização combinatória estão presentes em diversas áreas da ciência e da engenharia, exigindo soluções eficientes para selecionar, dentre muitas possibilidades, a melhor combinação de elementos que satisfaça restrições específicas. No entanto, devido à complexidade computacional envolvida, principalmente quando o espaço de busca é muito grande ou contém restrições rígidas, métodos exatos se tornam inviáveis em tempo hábil. Nesse contexto, técnicas heurísticas e meta-heurísticas têm se mostrado alternativas viáveis e eficazes.

Este trabalho tem como objetivo aplicar a meta-heurística **Simulated Annealing (SA)** para resolver uma variante do problema da mochila (knapsack problem), no qual cada item possui um valor de sabor e um peso, estando sujeito a restrições de peso total e incompatibilidade entre pares de itens. A abordagem consiste em gerar uma solução inicial viável e, a partir dela, explorar vizinhanças utilizando critérios probabilísticos de aceitação para buscar soluções melhores, mesmo que temporariamente piores, a fim de evitar ótimos locais.

Durante os experimentos, diferentes instâncias do problema foram avaliadas, cada uma com um conjunto distinto de itens, restrições de peso e relações de incompatibilidade. Os resultados das execuções serão analisados ao longo desse documento, avaliando valores mínimos, máximos, médias, desvios-padrão, comparações com o BKV (Best Known Value) fornecido e tempo computacional.

2 - Desenvolvimento

Ao longo deste capítulo serão discutidas as metodologias utilizadas para o desenvolvimento desse projeto bem como as decisões da equipe para a obtenção dos resultados.

Todo o projeto foi realizado utilizando a linguagem de programação Python.

2.1 - Formulação Matemática do Problema

O problema abordado neste trabalho pode ser modelado como uma variante do problema da mochila 0-1, com restrições adicionais de incompatibilidade entre itens. O objetivo é maximizar o sabor total dos ingredientes selecionados, respeitando um limite máximo de peso (W) e evitando a combinação de ingredientes mutuamente incompatíveis.
Formalmente:

- Sejam:
  - n o número de ingredientes disponíveis;
  - w_i o peso do ingrediente i, para $i = 1, 2, \ldots, n$;
  - f_i o valor de sabor do ingrediente i;
  - $x_i \in \{0, 1\}$ a variável que indica se o ingrediente i foi selecionado (1) ou não (0);
  - $I \subseteq \{(i, j) | i \neq j\}$ o conjunto de pares de ingredientes incompatíveis;
  - W o limite máximo de peso.

- Função objetivo:

$$Maximizar \sum_{i=1}^{n} f_i * x_i$$

- Sujeito a:

$$\sum_{i=1}^{n} w_i * x_i \leq W$$
$$x_i + x_j \leq 1 \ \forall (i,j) \in I$$
$$x_i \in \{0,1\} \ \forall i = 1, \ldots, n$$

2.2 - Abordagem Utilizada

Para resolver esse problema, foi utilizada a metaheurística **Simulated Annealing (SA)**, devido à sua capacidade de escapar de ótimos locais e encontrar boas soluções em espaços de busca complexos.

2.2.1 - Representação da Solução

A solução é representada por um vetor binário $x \in \{0,1\}^n$, onde cada posição indica se o respectivo ingrediente foi incluído ou não na seleção final.

2.2.2 - Geração da Solução Inicial

A solução inicial é construída de forma gulosa aleatória. Os itens são embaralhados e inseridos na solução, desde que:

- Não ultrapassem o limite de peso;
- Não violem nenhuma restrição de incompatibilidade com itens já incluídos.

2.2.3 - Estrutura de Vizinhança

A vizinhança é gerada por duas estratégias aleatórias:

- Inversão de um único bit (incluir ou excluir um item);
- Troca de um item incluído por um item excluído.

2.2.4 - Critério de Aceitação

A solução vizinha é aceita se for melhor ou, caso contrário, com uma probabilidade

$$p = e^{\frac{\Delta}{T}},$$

onde Δ é a diferença de valor entre a nova e a atual solução e T é a temperatura atual.

2.2.5 - Critério de Parada

O algoritmo pára quando a temperatura atinge um valor mínimo definido como final_temp. A temperatura é atualizada em cada ciclo multiplicando-se por um fator de resfriamento alpha.

2.3 - Implementação

A implementação foi feita em Python 3, utilizando as bibliotecas random, math, matplotlib, numpy e time.

2.3.1 - Estrutura geral do algoritmo:

1. Leitura das instâncias a partir de arquivos .dat.
2. Geração de uma solução inicial viável.
3. Aplicação da metaheurística Simulated Annealing.
4. Avaliação das soluções.
5. Execução repetida 30 vezes por instância para cálculo de estatísticas.
6. Geração de arquivos de saída com:
7. Melhor solução encontrada;
8. Valores médios, mínimos, máximos e desvio padrão;
9. Gráfico Boxplot com a dispersão dos resultados.

2.4 - Configurações Utilizadas

Os parâmetros definidos para o Simulated Annealing foram:

- Temperatura inicial: 1000.0
- Temperatura final: 0.1
- Fator de resfriamento (alpha): 0.98
- Iterações por temperatura: 200
- Número de execuções por instância: 30

Essas configurações foram escolhidas com base em experimentação empírica, equilibrando desempenho e tempo de execução.

2.5 - Término da Execução

Após finalizada as 30 execuções de cada instância, 3 arquivos são gerados para cada uma. O arquivo “epXX_best_solution.txt”, sendo ‘XX’ o número da instância, apresenta a melhor solução, mostrando o melhor sabor e os ingredientes selecionados. Em “epXX_results.txt” fica gravada a tabela de resultados para aquela instância. E, em “epXX_boxplot.png” é apresentado o boxplot da solução.

3 - Resultados Obtidos

Neste capítulo poderá ser visualizado, por meio das **Tabelas 1 e 2**, os resultados que foram obtidos ao final das execuções.

3.1 - Primeira Tabela

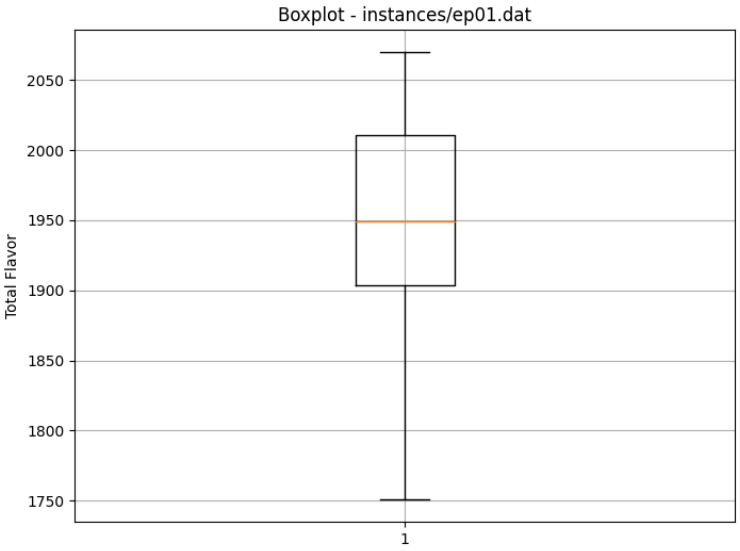
A tabela exibe os valores mínimos, máximos, médios e os desvios-padrão para cada instância, revelando padrões no desempenho do algoritmo.

Os resultados demonstram uma variação consistente entre os valores mínimo e máximo em todas as instâncias, com para ep01 (1751 a 2070) e ep02 (896 a 1287), que apresentaram as maiores amplitudes. As médias obtidas mostram que o algoritmo conseguiu manter desempenho estável em todas as execuções, com valores concentrados próximos aos ótimos conhecidos.

Além disso,também foram gerados os boxplots para as instâncias ep01 a ep10 (Figuras 01 a 05) que ilustram a distribuição dos valores de sabor obtidos em 30 execuções do algoritmo Simulated Annealing.

Tabela 1

Instância	Mínimo	Máximo	Média	Desvio-Padrão
ep01	1751	2070	1939.13	92.56
ep02	896	1287	1069.00	79.77
ep03	2490	2590	2528.00	22.14
ep04	2430	2500	2464.87	15.65
ep05	2380	2436	2409.60	14.32
ep06	4370	4438	4393.60	15.01
ep07	4240	4280	4260.63	11.78
ep08	4549	4629	4583.33	22.96
ep09	4287	4346	4315.83	15.37
ep10	4207	4247	4224.40	10.64



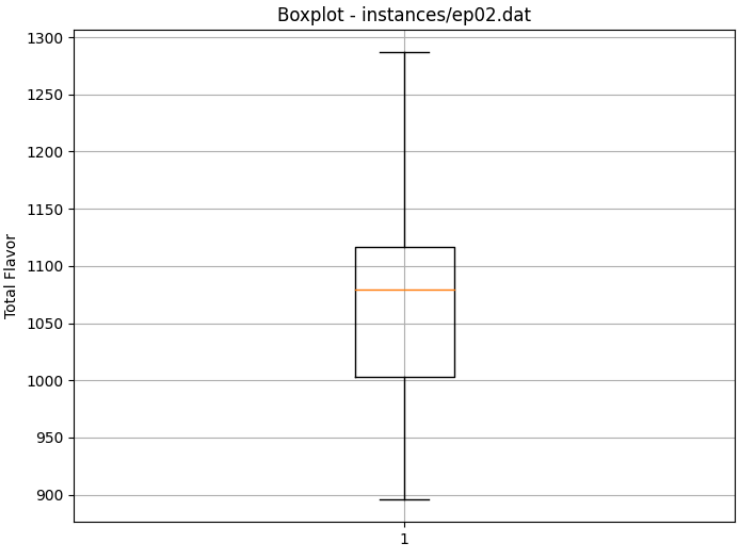
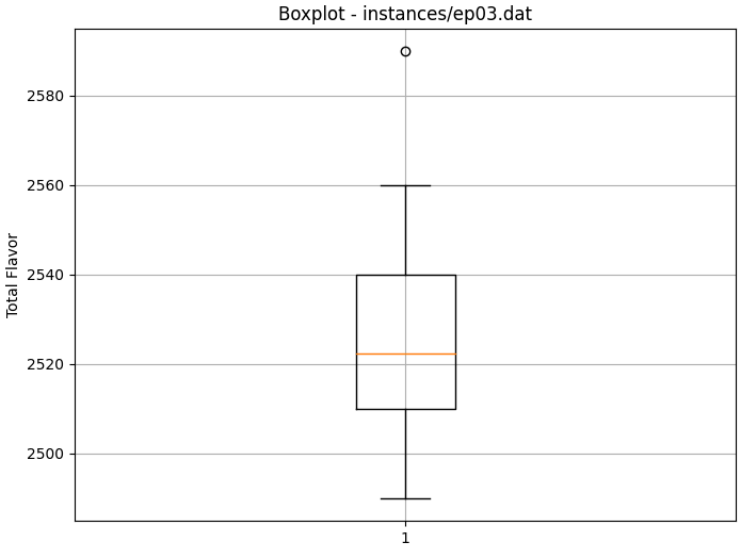


Figura 01: Boxplot instância 01 e 02



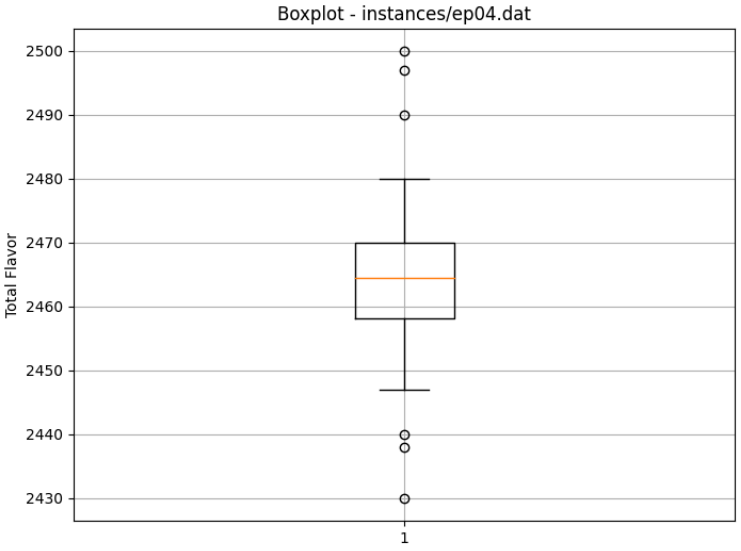
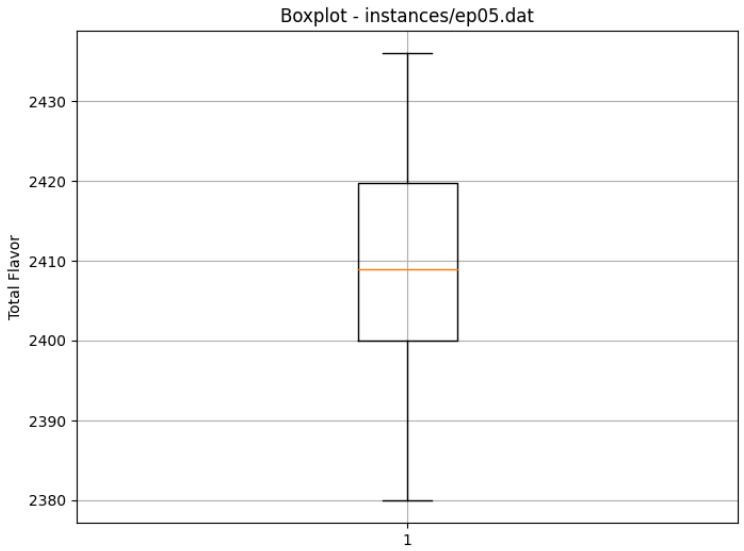


Figura 02: Boxplot instância 03 e 04



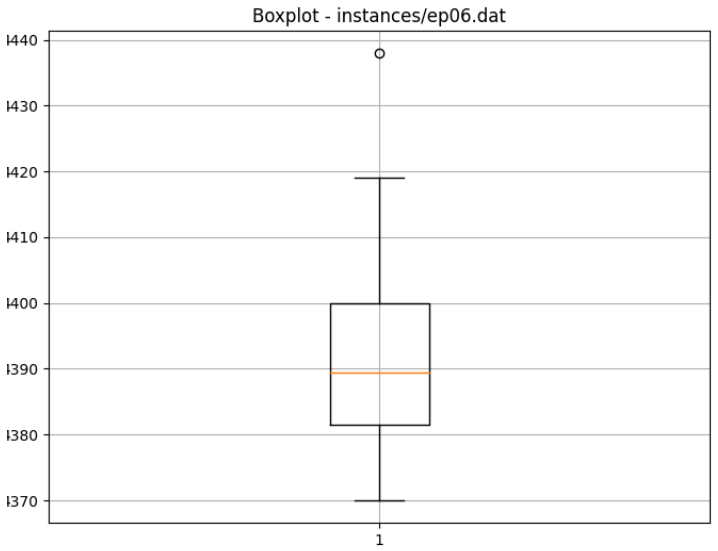
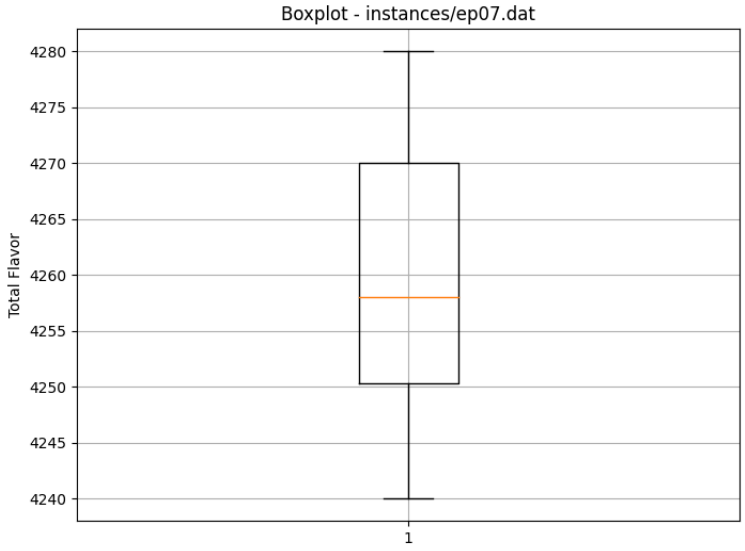


Figura 03: Boxplot instância 05 e 06



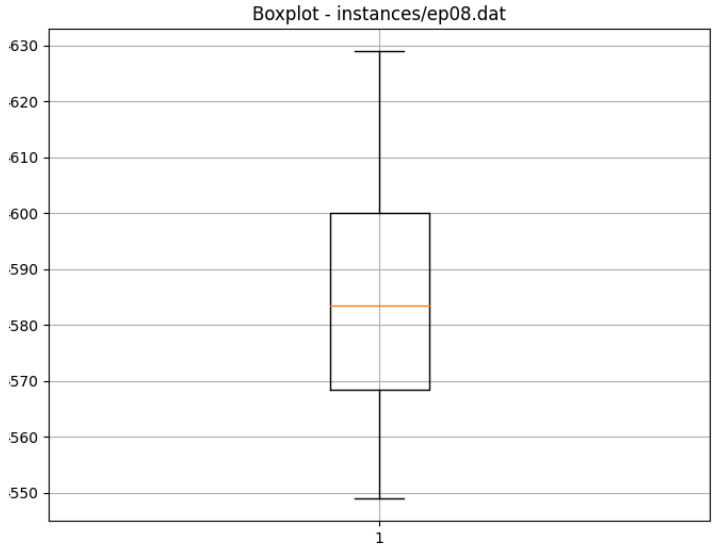


Figura 04: Boxplot instância 07 e 08

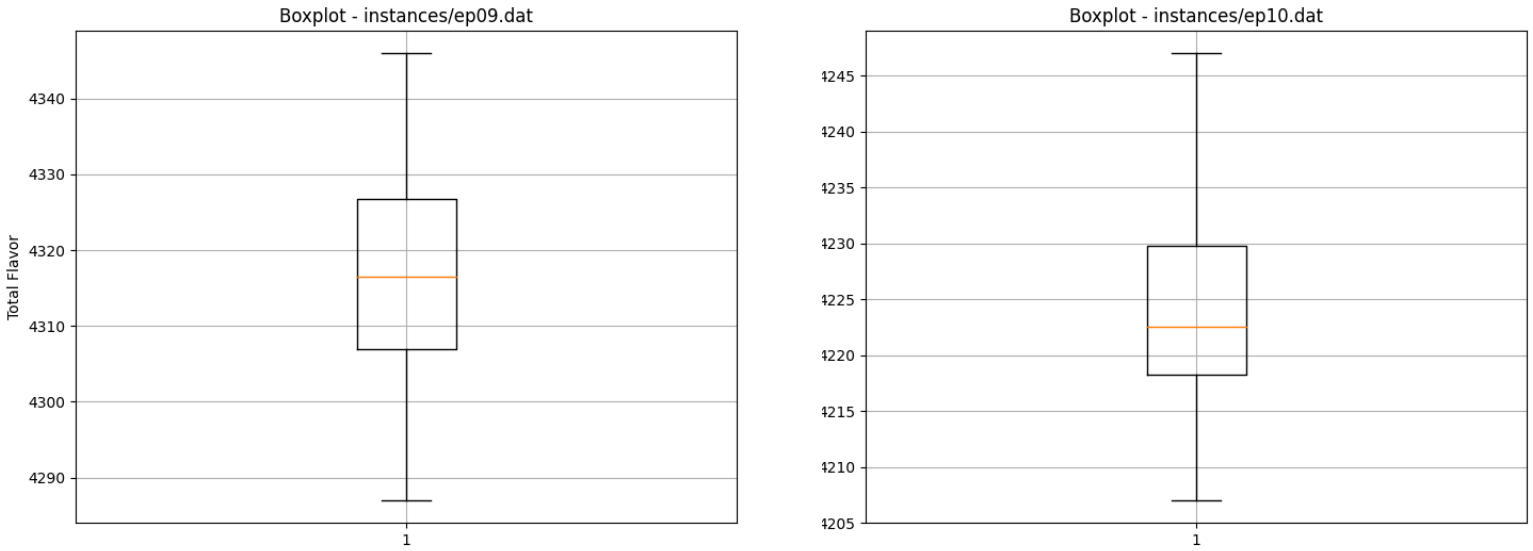


Figura 05: Boxplot instância 09 e 10

3.2 - Segunda Tabela

Nessa tabela estão sendo exibidos os resultados da aplicação do Simulated Annealing ao problema da mochila com restrições, contendo: o valor da solução inicial (SI), obtida por um método guloso aleatório; o valor da solução final (SF) após otimização;o desvio percentual em relação ao Best Known Value (BKV); e o tempo computacional por instância; o desvio percentual da solução final em relação à inicial

$$DP = 100 * ((SF - SI)/100)$$

Os resultados demonstram a eficácia do Simulated Annealing em aprimorar significativamente as soluções iniciais. O caso mais notável é a instância ep02, onde o valor saltou de 610 para 1287 (aumento de 110,98%), superando inclusive o BKV em 6,60%. Mesmo nas instâncias com soluções iniciais mais próximas do ótimo (ep05 e ep10), o algoritmo obteve melhorias adicionais, reduzindo os desvios para 7,16% e 3,26% respectivamente.

Tabela 2

Instância	SI	SF	Desvio % SI	Desvio % SF	Tempo
ep01	1233	2070	-67.88%	2.27%	180.88 s
ep02	610	1287	-110.98%	6.60%	180.27 s
ep03	2378	2590	-8.92%	9.12%	558.55 s
ep04	2419	2500	-3.35%	8.42%	459.64 s
ep05	2388	2436	-2.01%	7.16%	389.13 s
ep06	4207	4438	-5.49%	5.37%	573.85 s
ep07	4170	4280	-2.64%	3.60%	611.84 s
ep08	4210	4629	-9.95%	7.79%	803.21 s
ep09	4260	4346	-2.02%	4.86%	627.86 s
ep10	4195	4247	-1.24%	3.26%	603.76 s

4 - Conclusão

Este trabalho aplicou a meta-heurística Simulated Annealing (SA) para resolver uma variante do Problema da Mochila com restrições de incompatibilidade, visando maximizar o valor de sabor dos itens selecionados sem violar as restrições de peso ou combinações proibidas. A abordagem mostrou-se eficiente na exploração do espaço de busca, obtendo soluções de alta qualidade mesmo em cenários complexos.

Os resultados demonstraram uma melhoria significativa em relação às soluções iniciais geradas de forma gulosa aleatória, com redução consistente do desvio percentual em relação ao Best Known Value (BKV). A análise estatística, baseada no desvio padrão e nos boxplots, confirmou a baixa dispersão dos resultados, reforçando a robustez e consistência do algoritmo.

A escolha dos parâmetros como temperatura inicial, fator de resfriamento e número de iterações, foi crucial para equilibrar exploração e exploitation, permitindo que o SA escapasse de ótimos locais sem comprometer excessivamente o tempo de execução.

Por fim, o trabalho contribuiu para a consolidação dos conceitos teóricos vistos em sala de aula, além de demonstrar a aplicabilidade prática do Simulated Annealing em problemas de otimização combinatória com restrições.