**Título: Desafio de Otimização: Bin Packing Unidimensional**

Alunos: Caio Varandas do Carmo, Carlos Eduardo Fernandes Araujo

Introdução:

O problema de Bin Packing Unidimensional é um clássico desafio de otimização combinatória que visa alocar um conjunto de itens, cada um com um peso específico, em um número mínimo de mochilas, sem ultrapassar a capacidade máxima de cada mochila. Este problema é amplamente estudado devido à sua aplicabilidade em diversas áreas, como logística, gerenciamento de estoques e design de sistemas computacionais.

Esss trabalho se concentra na aplicação de metaheurísticas para resolver o problema do Bin Packing, através da utilização de variações de metaheuristicas e adaptações de algoritmos genéticos. A escolha dessas abordagens é motivada pela necessidade de encontrar soluções eficientes e eficazes em um espaço de busca considerável, onde soluções exatas se tornam inviáveis devido à complexidade computacional.

Descrição do Hardware e Software Utilizados:

Sistema Operacional: Linux Mint 5.4.0-122-generic

Versão do Cinnamon: 5.2.7

Kernel do Linux: 5.4.0-122-generic

RAM: 8 GB

CPU: Intel© Core™ i7-3770S CPU @ 3.10GHz × 4

Disco: 244.0 GB

Placa de Vídeo: Intel Corporation Xeon E3-1200 v2/3rd Gen Core processor Graphics Controller

Compilador: Visual Studio Code (VSCode)

O desenvolvimento do código foi realizado utilizando o editor de código VSCode, com Python como linguagem de programação principal, devido à sua flexibilidade e ao suporte a bibliotecas para a manipulação de algoritmos de otimização.

Instâncias Utilizadas:

Foram usadas algumas instancias retiradas da internet abaixo estão algumas:

Quantidade de itens: 200

Capacidade de cada caixa: 100000

Peso de cada item: 34991 34941 34922 34866 34849 34771 34768 34748 34544 34358 34254 34155 34098 34076 34055 34048 34029 33990 33871 33780 33750 33654 33612 33581 33430 33260 33197 33155 33115 33007 32989 32795 32708 32394 32384 32309 32193 32039 32038 32008 31995 31961 31946 31865 31839 31829 31692 31633 31354 31169 31141 31006 30929 30843 30842 30807 30741 30514 30395 30387 30341 30296 30287 30284 30140 30135 30063 29975 29933 29859 29735 29730 29703 29525 29518 29423 29378 29234 29218 29178 29092 29089 28947 28647 28574 28550 28547 28471 28461 28299 28267 28252 28251 28159 28009 28003 27967 27852 27811 27664 27508 27413 27409 27184 27162 27113 27099 27048 27041 26733 26506 26362 26183 25997 25976 25897 25856 25784 25700 25668 25641 25522 25490 25433 25408 25322 25299 25237 25091 25057 25015 24990 24974 24939 24834 24777 24743 24625 24555 24449 24367 24340 24329 24126 24085 24050 24020 23999 23989 23974 23928 23837 23836 23565 23491 23422 23417 23205 23195 23156 23092 22712 22644 22417 22392 22281 22239 22212 22067 22045 22042 22003 21866 21851 21849 21713 21674 21608 21607 21594 21401 21296 21239 21180 21128 21059 20954 20948 20947 20813 20755 20725 20693 20585 20513 20431 20338 20310 20296 20081

Quantidade de itens: 1000

Capacidade de cada caixa: 150

Peso de cada item: 42 69 67 57 93 90 38 36 45 42 33 79 27 57 44 84 86 92 46 38 85 33 82 73 49 70 59 23 57 72 74 69 33 42 28 46 30 64 29 74 41 49 55 98 80 32 25 38 82 30 35 39 57 84 62 50 55 27 30 36 20 78 47 26 45 41 58 98 91 96 73 84 37 93 91 43 73 85 81 79 71 80 76 83 41 78 70 23 42 87 43 84 60 55 49 78 73 62 36 44 94 69 32 96 70 84 58 78 25 80 58 66 83 24 98 60 42 43 43 39 97 57 81 62 75 81 23 43 50 38 60 58 70 88 36 90 37 45 45 39 44 53 70 24 82 81 47 97 35 65 74 68 49 55 52 94 95 29 99 20 22 25 49 46 98 59 98 60 23 72 33 98 80 95 78 57 67 53 47 53 36 38 92 30 80 32 97 39 80 72 55 41 60 67 53 65 95 20 66 78 98 47 100 85 53 53 67 27 22 61 43 52 76 64 61 29 30 46 79 66 27 79 98 90 22 75 57 67 36 70 99 48 43 45 71 100 88 48 27 39 38 100 60 42 20 69 24 23 92 32 84 36 65 84 34 68 64 33 69 27 47 21 85 88 59 61 50 53 37 75 64 84 74 57 83 28 31 97 61 36 46 37 96 80 53 51 68 90 64 81 66 67 80 37 92 67 64 31 94 45 80 28 76 29 64 38 48 40 29 44 81 35 51 48 67 24 46 38 76 22 30 67 45 41 29 41 79 21 25 90 62 34 73 50 79 66 59 42 90 79 70 66 80 35 62 98 97 37 32 75 91 91 48 26 23 32 100 46 29 26 29 26 83 82 92 95 87 63 57 100 63 65 81 46 42 95 90 80 53 27 84 40 22 97 20 73 63 95 46 42 47 40 26 88 49 24 92 87 68 95 34 82 84 43 54 73 66 32 62 48 99 90 86 28 25 25 89 67 96 35 33 70 40 59 32 94 34 86 35 45 25 76 80 42 91 44 91 97 60 29 45 37 61 54 78 56 74 74 45 21 96 37 75 100 58 84 85 56 54 71 52 79 43 35 27 70 31 47 35 26 30 97 90 80 58 60 73 46 71 39 42 98 27 21 71 71 78 76 57 24 91 84 35 25 77 96 97 89 30 86 81 39 75 66 85 36 60 56 50 75 75 37 87 95 21 99 42 57 31 37 42 40 69 91 45 97 84 90 52 43 68 53 37 65 79 73 92 87 20 20 73 42 52 20 24 76 71 72 21 21 82 92 78 87 50 41 31 73 89 59 88 40 71 69 45 57 49 68 84 32 69 77 92 98 57 39 32 23 99 91 48 21 70 43 73 69 65 57 67 28 84 42 61 92 82 34 74 55 60 69 26 25 67 77 67 79 47 84 50 21 87 83 44 88 78 53 78 37 47 52 32 88 85 82 55 41 60 66 78 72 34 64 20 60 100 62 80 34 68 38 32 32 37 82 98 90 58 97 56 34 70 39 56 69 36 20 99 84 53 27 88 53 42 45 42 31 54 60 55 27 36 31 39 91 45 97 26 80 41 56 70 97 48 87 23 32 75 100 97 51 78 78 21 72 72 79 46 30 48 27 95 48 67 58 46 92 21 82 91 40 56 24 94 44 91 92 81 24 84 44 83 37 98 85 88 95 29 35 100 55 48 27 20 66 62 52 88 59 97 91 81 81 86 48 43 60 72 88 90 48 38 60 53 55 90 48 55 57 59 25 51 22 43 31 52 89 96 58 63 27 46 43 30 44 71 66 64 28 83 88 42 92 95 36 24 62 44 82 59 31 96 44 61 78 72 62 76 65 22 41 27 85 80 72 100 29 27 43 83 32 33 53 95 99 20 23 72 50 50 27 89 53 75 81 34 27 69 48 84 37 69 54 51 49 49 54 100 55 45 83 61 96 91 37 53 76 50 66 70 87 92 35 53 95 47 56 55 86 32 99 83 88 41 63 77 60 66 53 79 81 96 34 99 47 74 87 44 77 52 99 69 64 94 38 69 61 98 40 84 89 49 64 53 41 34 85 35 55 61 68 100 75 98 36 44 57 24 60 45 48 60 94 71 70 64 62 93 20 69 37 63 61 26 54 89 46 54 50 32 71 62 40 26 59 62 27 60 50 74 34 40 70 56 23 66 57 43 45 65 25 82 82 37 66 47 44 94 23 24 51 100 22 25 51 95 58 97 30 79 23 53 80 20 65 64 21 26 100 81 98 70 85 92 97 86 71 91 29 63 34 67 23 33 89 94 47 100 37 40 58

Descrição Detalhada da Metaheurística e Variações:

Fundamentação Teórica:

Durante os estudos sobre o problema Bin Packing e sobre Heurísticas e Metaheurísticas, foram utilizados artigos nacionais e internacionais para investigar as diferentes abordagens disponíveis. O objetivo era identificar qual metaheurística seria a mais adequada para o problema em questão, além de explorar possibilidades de hibridização para potencializar os resultados. Esse embasamento teórico levantou possibilidades de uso para metaheurísticas como a Simulated Anneling, Tabu Search, GRASP (Greedy Randomized Adaptive Search Procedure) e Genetic Algorithm (GA). Heuristicas como First Fit Decreasing (FFD), First Fit, Next Fit ou Best Fit, visando criar soluções iniciais eficientes para as metaheurísticas.

Implementação e Adaptações:

Inicialmente, optou-se pelo uso da metaheuristica Simulated Anneling (SA), que se destacou em muitos artigos como uma abordagem eficaz para o problema de Bin Packing, devido à sua capacidade de explorar soluções de forma ampla e evitar mínimos locais. Para garantir uma solução inicial, foi utilizada a heurística First Fit Decreasing (FFD), cujo objetivo era criar uma solução inicial e representar as mochilas e os itens contidos nelas. No entanto, essa abordagem foi rapidamente descartada, pois ao executar o código, a FFD sempre resultava em uma resolução muito próxima do ótimo. Como consequência, ao rodar o Simulated Annealing (SA), o algoritmo não conseguia encontrar outra solução, permanecendo com o que havia sido obtido pela FFD.

Diante disso, o grupo decidiu por mudar para a heurística Next Fit, que possui o mesmo objetivo da FFD, criando uma solução inicial e atribuindo os itens às mochilas correspondentes. Essa alteração visou aumentar a capacidade do Simulated Annealing de explorar diferentes áreas do espaço de soluções, potencialmente levando a resultados mais variados e, consequentemente, melhores.

Para encontrar a próxima solução, foi implementada uma abordagem que envolve a remoção de um item aleatório de uma das mochilas. Em seguida, verifica-se se esse item pode ser inserido em outra mochila, também escolhida aleatoriamente. Essa estratégia visa explorar novas combinações de distribuição dos itens entre as mochilas, potencialmente resultando em soluções mais eficientes.

Durante o processo de desenvolvimento, foi testada a aplicação da metaheurística Tabu Search, com o objetivo de evitar a repetição de combinações já exploradas, armazenando movimentos em uma lista tabu. Entretanto, essa abordagem resultou em um aumento significativo no tempo de execução do código, sem apresentar melhorias suficientes para justificar seu uso, o que levou ao seu descarte.

Variações Testadas:

A variação mais notável foi a tentativa de combinar o Simulated Annealing com uma versão do Tabu Search para restringir movimentos repetidos. Essa combinação visava aprimorar a exploração do espaço de busca, mas a complexidade adicional da lista tabu e o tempo elevado de execução tornaram essa abordagem inviável para os experimentos.

Justificativa para as Escolhas:

A escolha pelo Simulated Annealing baseou-se em sua capacidade de evitar mínimos locais, uma característica fundamental para um problema onde soluções subótimas podem ser facilmente alcançadas. A mudança para a heurística Next Fit como geradora de soluções iniciais se deu para melhorar a variabilidade das soluções, permitindo uma melhor exploração por parte do Simulated Annealing.

O Tabu Search foi considerado inicialmente para evitar que itens fossem reposicionados nas mesmas mochilas, permitindo ao SA explorar outras combinações. No entanto, os testes mostraram que a complexidade adicional prejudicava a eficiência do algoritmo.

Parâmetros:

**Temperatura inicial:** Para otimizar a busca por uma melhor solução eficaz, foi necessário ajustar a temperatura inicial. Os valores usados foram:

Caso de teste 1: 1000

Caso de teste 2: 100000

**Temperatura final:** O valor da temperatura final também foi ajustado para melhorar os resultados. Os parâmetros definidos foram:

Caso de teste 1: 0.00001

Caso de teste 2: 0.0000001

**Alpha:** O fator de resfriamento (alpha) teve que ser variado para encontrar uma solução satisfatória, sendo mantido constante em ambos os casos de teste:

Caso de teste 1: 0.9999

Caso de teste 2: 0.9999

**Solução Inicial:** Gerada pela heurística Next Fit.

Resultados:

Texto

Descrição gerada automaticamenteCaso de teste 1:

Após 1.72 segundos de execução, a metaheuristica retornou um total de 60 caixas.

Observou-se que, embora o resultado frequentemente se aproximasse de 60, ele variou entre 60 e 65 caixas em execuções sucessivas.

Texto

Descrição gerada automaticamenteCaso de teste 2:

Com um tempo de execução de 21.56 segundos, a solução finalizou em 433 caixas.

Os resultados oscilaram entre 433 e 522 caixas em diferentes execuções.

Análise dos resultados:

A escolha da metaheurística Simulated Annealing, combinada com a heurística Next Fit, mostrou-se eficiente para a busca de soluções em ambos os casos de teste. No entanto, a análise dos resultados revela que, em casos de teste maiores (com 1000 itens ou mais), o tempo necessário para encontrar uma solução adequada aumentou significativamente em comparação com os testes menores. Isso sugere que a complexidade do problema tende a crescer exponencialmente com o aumento do número de itens, refletindo a dificuldade de otimização que a metaheurística enfrenta em cenários mais amplos.

Portanto, é recomendável considerar a implementação de estratégias adicionais de otimização ou ajustes nos parâmetros para melhorar a eficiência do algoritmo em situações de maior escala. Além disso, a análise dos resultados sugere uma necessidade de equilibrar a busca por soluções ótimas com a viabilidade do tempo de execução, especialmente em problemas que envolvem grandes conjuntos de dados.