

Relatório Técnico: Análise de Algoritmos Aproximativos para o Problema da Mochila

Disciplina: Análise e Projeto de Algoritmos

Tema: Classes de Problemas (P, NP, NP-C, NP-H) e Algoritmos de Aproximação

Integrantes: Antonio Heitor Gomes Azevedo, Deivison Ryan Brito Tavares, Gustavo Yuji Virgolino Nishimura, Heitor Yasuo Yamamoto

1. Introdução

Este trabalho visa explorar a aplicação de algoritmos de aproximação para a resolução de problemas classificados como intratáveis. O problema selecionado pela equipe foi o Problema da Mochila (Knapsack Problem - KP), na sua variação 0/1, um clássico da otimização combinatória. O objetivo é maximizar o valor total dos itens carregados em uma mochila sem exceder sua capacidade de peso.

Dada a natureza computacionalmente custosa para encontrar a solução exata em instâncias grandes, implementamos e comparamos duas meta-heurísticas: Particle Swarm Optimization (PSO) e Simulated Annealing (SA).

2. Fundamentação Teórica e Complexidade

Para compreender a dificuldade do Problema da Mochila, é necessário situá-lo dentro da Teoria da Complexidade Computacional.

2.1. Classes de Problemas

- **P (Polynomial Time)**: Classe de problemas que podem ser resolvidos por uma máquina determinística em tempo polinomial em relação ao tamanho da entrada.
- **NP (Nondeterministic Polynomial Time)**: Classe de problemas cuja solução, se fornecida, pode ser verificada em tempo polinomial. Note que a classe P está contida na classe NP.
- **NP-Completo (NP-C)**: São os problemas mais difíceis da classe NP. Se encontrarmos um algoritmo polinomial para resolver qualquer problema NP-C, resolvemos todos os problemas em NP.
- **NP-Difícil (NP-Hard)**: Problemas que são pelo menos tão difíceis quanto os problemas NP-C. Eles não precisam necessariamente estar em NP (ex: problemas de otimização, onde verificar a solução ótima é tão difícil quanto encontrá-la).

2.2. O Problema da Mochila na Hierarquia

O Problema da Mochila, em sua versão de decisão ("Existe um subconjunto de itens com valor $\geq K$ e peso $\leq W$?"), é NP-Completo. A versão de otimização (maximizar o valor), tratada neste trabalho, é classificada como NP-Difícil. Isso justifica o uso de algoritmos

aproximativos, pois algoritmos exatos (como força bruta de complexidade $O(2^n)$) tornam-se inviáveis à medida que o número de itens (n) cresce.

2.3. Redução Polinomial

A Redução Polinomial ($A \leq_p B$) é uma técnica onde transformamos uma instância de um problema A em uma instância de um problema B em tempo polinomial. A NP-Compleitude da Mochila é frequentemente provada através da redução a partir do problema da Cobertura Exata (Exact Cover) ou Partição. Isso demonstra matematicamente que resolver a Mochila eficientemente implicaria em resolver todos os problemas da classe NP.

3. Metodologia e Implementação

Os algoritmos foram implementados em linguagem Python 3, utilizando bibliotecas padrão (random, math, statistics, matplotlib.pyplot, time).

3.1. Configuração dos Algoritmos

Para garantir a reprodutibilidade dos testes, foram fixados os seguintes parâmetros baseados na literatura e em testes preliminares:

- Particle Swarm Optimization (PSO):
 - População: 30 partículas.
 - Iterações: 200.
 - Coeficiente de Inércia (w): 0.7.
 - Coeficientes Cognitivo ($c1$) e Social ($c2$): 1.5.
- Simulated Annealing (SA):
 - Temperatura Inicial (T_0): 100.
 - Resfriamento (α): 0.99.
 - Iterações: 3000.

3.2. Análise Assintótica (Big-O)

Analisamos a complexidade teórica das implementações, onde " n " é o número de itens, " P " é o número de partículas e " I " é o número de iterações.

1. **PSO:** A complexidade é dominada pela atualização de todas as partículas em cada iteração.
Complexidade: $T(n)$ aproximadamente $O(I * P * n)$
Justificativa: Cada partícula percorre o vetor de itens (n) para atualizar velocidade e posição.
2. **Simulated Annealing:** A complexidade depende da geração de vizinhos e avaliação da função objetivo.
Complexidade: $T(n)$ aproximadamente $O(I * n)$
Justificativa: Em cada iteração, avaliamos uma solução de tamanho n .

4. Análise Experimental e Resultados

Os experimentos foram realizados em três instâncias de tamanhos diferentes (n=10, 30, 100). Foram executadas 100 rodadas independentes para cada instância para garantir significância estatística.

4.1. Tabela Comparativa de Resultados

Instância	Métrica	PSO (População)	SA (Recozimento)	Diferença
Pequena	Melhor Valor	344	344	0%
(n=10)	Média	344.0	331.77	PSO +3.7%
	Tempo Médio (s)	0.065s	0.012s	SA 5x mais rápido
Média	Melhor Valor	1001	952	PSO +5.1%
(n=30)	Média	946.29	856.30	PSO +10.5%
	Tempo Médio (s)	0.158s	0.019s	SA 8x mais rápido
Grande	Melhor Valor	3174	3170	Empate Técnico
(n=100)	Média	2928.32	2459.50	PSO +19%
	Desvio Padrão	67.64	271.41	PSO mais estável
	Tempo Médio (s)	0.525s	0.048s	SA 10x mais rápido

4.2. Discussão Crítica

- **Qualidade da Solução:** O PSO mostrou-se superior em consistência. Na instância grande, embora os melhores valores tenham sido próximos (3174 vs 3170), a média do PSO foi quase 20% superior à do SA. O baixo desvio padrão do PSO indica

robustez, enquanto o SA variou muito (Desvio de 271.4), indicando frequente aprisionamento em ótimos locais.

- **Desempenho de Tempo:** O Simulated Annealing confirmou sua vantagem de complexidade $O(I * n)$, sendo até 10 vezes mais rápido que o PSO. No entanto, essa velocidade custou a qualidade média da solução.
- **Convergência:** Para problemas NP-Difíceis onde a precisão é crítica, o custo extra de tempo do PSO (0.5s) é desprezível e justifica seu uso. O SA seria preferível apenas em sistemas de tempo real com restrições severas de processamento (milissegundos).

5. Conclusão

O projeto demonstrou que algoritmos de aproximação são ferramentas vitais para lidar com problemas das classes NP-Completo e NP-Difícil. Confirmamos experimentalmente a teoria: não buscamos a solução exata garantida (o que custaria tempo exponencial), mas sim uma solução "boa o suficiente" em tempo polinomial.

O PSO apresentou o melhor equilíbrio geral para o Problema da Mochila nas configurações testadas, oferecendo soluções de alta qualidade com estabilidade, enquanto o SA destacou-se pela velocidade extrema, mas com menor confiabilidade nos resultados médios.

6. Referências Bibliográficas

CORMEN, T. H. et al. Introduction to Algorithms. 3rd Edition. MIT Press, 2009.

GAREY, M. R.; JOHNSON, D. S. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.

KENNEDY, J.; EBERHART, R. "Particle Swarm Optimization". Proceedings of ICNN'95 - International Conference on Neural Networks, 1995.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. "Optimization by Simulated Annealing". Science, 1983.