

# ALGORITMO DE BUSCA TABU APLICADO NA SOLUÇÃO DO PROBLEMA DE MÚLTIPLAS MOCHILAS

MÜLLER, M. R. <sup>1</sup>; FRANCO, E. M. C. <sup>2</sup>

<sup>1,2</sup> Universidade Estadual do Oeste do Paraná - UNIOESTE/Pós-Graduação em Engenharia de Sistemas Dinâmicos e Energéticos - PGESEDE.

**marcos\_ricardo@live.com; emfra.unioeste@gmail.com**

## Resumo

Em um primeiro momento o trabalho apresenta considerações teóricas a respeito do algoritmo de Busca Tabu, e outras técnicas de solução para o problema de Múltiplas Mochilas. Depois disso, um algoritmo baseado na metaheurística de Busca Tabu é desenvolvido e utilizado para solucionar o problema de múltiplas mochilas com duas mochilas, as principais características deste método são apresentadas, assim como detalhes sobre o algoritmo desenvolvido, os testes realizados e os resultados obtidos. Por fim, são abordadas algumas peculiaridades do algoritmo, assim como propostas implementações futuras e outros aspectos a serem considerados e estudados.

**Palavras-chave:** Metaheurísticas, Múltiplas Mochilas, Busca Tabu.

## Abstract

A specialized tabu search algorithm to solve the multiple knapsack problem is presented. The memory characteristics of this method are exploited to adjust to the behavior of the problem. Details about the computational implementation of the algorithm are also presented. The algorithm is tested on three different well known test systems with two knapsacks, finding the best known configuration with few iterations and short computational times.

**Keywords:** Metaheuristics, Multiple knapsacks, Tabu Search.

## INTRODUÇÃO

O problema da mochila (knapsack problem, em inglês) é um problema de otimização combinatória, ou seja, caracteriza-se como um problema de otimização sobre um conjunto finito, de maneira que envolve uma função objetivo e um conjunto de restrições. É um dos 21 problemas NP-completos apresentado por Karp em [1].

O nome está relacionado ao tipo de problema, onde é necessário alocar objetos em uma mochila de capacidade limitada, observando que, o conjunto de objetos apresenta pesos e valores diversos, de tal forma que definir o subconjunto de objetos cujo somatório dos pesos não exceda a capacidade da mochila e ao mesmo tempo maximize o valor total, compreende o objetivo e a solução do problema da mochila.

O problema das múltiplas mochilas é uma variação do problema da mochila simples, e consiste basicamente em preencher um determinado número de mochilas com objetos únicos com pesos e valores conhecidos, que somem o maior valor possível sem exceder a capacidade das mochilas, maximizando uma função objetivo e atendendo um conjunto de restrições.

Em diversas situações existe a preocupação em se obter resultados rapidamente, ou não se deseja implementar um algoritmo do tipo força bruta, pela necessidade de diminuir custo computacional, inviabilidade ou por outros motivos, abrindo mão da garantia de respostas ótimas globais, se apresentam

as heurísticas. Segundo Cordenonsi [2] “os métodos heurísticos englobam estratégias, procedimentos e métodos aproximativos com o objetivo de encontrar uma boa solução, mesmo que não seja a ótima, em um tempo computacional razoável”.

Para possibilitar que as heurísticas de busca local possam ser guiadas, a fim de fugir de ótimos locais, tentando vencer segmentos que apresentem piora nos resultados após passar por um ótimo local (escarpa), e/ou soluções vizinhas que não apresentem melhora nem piora para a solução atual (platô), mas que podem anteceder soluções melhores, as metaheurísticas são utilizadas [2].

A literatura apresenta diversas metaheurísticas, sendo algumas delas mais conhecidas, dentre as quais pode-se citar os algoritmos genéticos [3], Sistemas de colônias de formigas [4], Simulated Annealing [5], Busca Tabu [6], Greedy Randomized Adaptive Search Procedure - GRASP [7], Variable Neighborhood Search - VNS [8] e Redes Neurais [9]. Para resolver o problema das mochilas, alguns trabalhos utilizam metaheurísticas [10]-[12].

A Busca Tabu é uma metaheurística que guia um algoritmo de busca local, de maneira a permitir a exploração contínua dentro de um espaço de busca. A Busca Tabu surgiu a partir de uma solução para problemas de programação inteira, sendo que, posteriormente foi descrita para uso em problemas de otimização combinatória em geral [13].

## O PROBLEMA

O problema das múltiplas mochilas (multiple knapsack problem, do inglês) é uma generalização do problema da mochila, ou problema da mochila simples, e, pode ser caracterizado como um problema de programação inteira ou de programação linear. O problema consiste em  $m$  mochilas, de capacidades  $L_1, L_2, \dots, L_m$ , e um conjunto de  $n$  itens de pesos  $p_1, p_2, \dots, p_n$ , e com valores  $v_1, v_2, \dots, v_n$  associados. De acordo com esses valores, o objetivo é alocar itens nas  $m$  mochilas sem repeti-los, respeitando o peso máximo de cada mochila, encontrando ainda, o maior valor possível. A variável de decisão binária  $x_{ij}$  receberá o valor 1 (um) se o item  $j$  for alocado ao recipiente  $i$ , caso contrário, será 0 (zero). A modelagem matemática do problema é apresentada a seguir:

$$\text{Maximizar} = \sum_{i=1}^m \sum_{j=1}^n v_j x_{ij} \quad (1)$$

$$\text{Sujeito a: } \sum_{j=1}^n p_j x_{ij} \leq L_i \quad i = 1, \dots, m \quad (2)$$

$$\sum_{i=1}^m x_{ij} \leq 1 \quad j = 1, \dots, n \quad (3)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \quad (4)$$

A função objetivo apresentada (1), busca maximizar o valor de cada mochila em função de um vetor binário  $x_{ij}$ . A restrição (2) garante que a capacidade máxima de cada mochila não seja excedida pela solução encontrada, sendo a restrição (3), a responsável por evitar que um mesmo item seja alocado em mais de uma mochila.

## O FUNCIONAMENTO

Originalmente o algoritmo de Busca Tabu procura a cada iteração um ótimo local, por meio da seleção do melhor vizinho  $s'$  da vizinhança  $V(s)$ , da corrente solução  $s$ . Para que possa percorrer todo o espaço de busca, mesmo que  $f(s')$  seja pior que  $f(s)$ ,  $s'$  será admitido como a solução corrente. Para evitar que ocorra retorno para soluções previamente encontradas, o algoritmo utiliza uma lista tabu (proibidos), possibilitando travar a 'mudança' de um determinado atributo por um número  $x$  de iterações (tempo tabu). Mesmo que uma 'mudança' esteja travada pela lista tabu, ela pode ser admitida desde que a solução consiga satisfazer um determinado critério de aspiração. Os atributos são ajustados de maneira a permitir a convergência mais rápida e a melhores soluções. O procedimento chega ao fim quando algum critério de parada é atendido.

## CÁLCULO DA FUNÇÃO OBJETIVO

O algoritmo desenvolvido utiliza uma construção inteligente para solução inicial, garantindo que cada item seja utilizado no máximo em uma mochila, e que, as capacidades das mochilas sejam respeitadas. No caso da construção de soluções iniciais randômicas, é possível aplicar penalização para as respostas que não atendam as restrições de item único e capacidade das mochilas. O pseudocódigo apresentado a seguir, demonstra a aplicação de penalização para configurações que excedam a capacidade das

mochilas:

```
%Caso o peso máximo seja excedido, é calculada a penalização.
if maxpeso-(solucao*pesoc')<0
    penalizacao=max(pesoc)+(maxpeso-(solucao*pesoc'))^2;
% Caso não exceda, a penalização recebe 0 (zero)
else
    penalizacao=0;
end
% A Função Objetivo recebe a solução * valor, acrescido da penalização.
FO=(solucao*valorc')-penalizacao;
```

## TÉCNICA DE SOLUÇÃO

Inicialmente foi desenvolvido um algoritmo baseado na meta heurística conhecida como Busca Tabu (Tabu Search, em inglês). A codificação foi realizada no ambiente interativo para computação numérica MATLAB® versão 7.10.0, nome que também designa a linguagem utilizada.

O algoritmo se baseia em criar uma solução inicial, após isso, passa pelo melhoramento da função objetivo evitando ótimos locais, atualiza a lista tabu, efetua o rebalanceamento entre as mochilas e a retirada melhorada de itens repetidos, e por fim, após todas as iterações serem realizadas, apresenta a solução encontrada.

## TESTES E RESULTADOS

Os problemas um, cinco e seis apresentados em [15] foram utilizados para testar o algoritmo desenvolvido. Após a execução do algoritmo para cada um dos problemas, os parâmetros relativos ao número máximo de iterações, e a duração da memória Tabu foram ajustados, encontrando assim, as respostas ótimas dos três problemas. Os problemas foram solucionados pelo algoritmo desenvolvido no

mesmo computador, por três vezes cada, havendo apenas pequena variação no tempo de execução, sendo realizada a média aritmética dos mesmos, e dispostos juntamente com os parâmetros utilizados, conforme apresentado na Tabela 1.

Tabela 1 - Testes e resultados obtidos

Problema	Nº de Iterações	Parâmetro de Memória	Tempo (seg)	Desvio Padrão (seg)	Resposta Encontrada	Resposta Ótima
1	17	6	0.087174	0.031432	333	333
5	9	4	0.083009	0.027644	345	345
6	129	11	0.135215	0.039793	452	452

## COMENTÁRIOS FINAIS

A técnica de solução se mostrou satisfatória com os problemas submetidos, foi capaz de encontrar a melhor resposta conhecida nos três problemas. Para um trabalho futuro o algoritmo pode ser modificado, e por meio das penalizações e da memória tabu, procurar impedir a construção de soluções que não atendam de imediato as restrições de limite individual das mochilas e da não repetição de itens.

Da forma que foi implementado o algoritmo é capaz de chegar a soluções boas e ótimas com um número baixo de iterações, provavelmente devido às melhorias de rebalanceamento das mochilas e de retirada de itens. É necessário observar ainda que, a solução inicial inteligente é capaz de montar soluções boas na maioria dos casos, mas uma implementação randômica poderia trazer diferentes características que poderiam influenciar nos resultados de acordo com os diferentes números de iterações empregados, por exemplo.

Por fim, o trabalho permite observar que com a implementação da metaheurística de busca tabu, diversos problemas com diferentes características

puderam ser resolvidos, tendo suas soluções ótimas encontradas, sem a necessidade que um algoritmo de força bruta precisasse ser moldado para cada um dos problemas.

## AGRADECIMENTOS

A Fundação Parque Tecnológico Itaipu - FPTI, pela bolsa, a UNIOESTE e ao PGESDE.

## REFERÊNCIAS

- [01] KARP, R. M. 1972. "Reducibility among combinatorial problems". In Complexity of Computer Computations: Proc. of a Symp. New York - NY: Plenum Press, 1972, pp. 85-103.
- [02] CORDENONSI, A. Z. 2008. "Ambientes, Objetos e Dialogicidade: uma estratégia de ensino superior em heurísticas e meta-heurísticas". Tese - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008.
- [03] GOLDBERG, D. E. 1989. "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley, 1989.
- [04] DORIGO, M. 1991. "Optimization, Learning and Natural Algorithms". Phd thesis. Dipartimento di Elettronica, Politecnico di Milano, IT, 1991.
- [05] KIRKPATRICK, S.; GELLATI, C.D.; VECCHI, M. 1983. "Optimization by Simulated Annealing". Science, v. 220, pp. 671-680, 1983.
- [06] GLOVER, F.; LAGUNA, M. 1993. "Tabu Search". In: REEVES, Colin (Ed.). Modern Heuristic Techniques. Oxford: Blackwell, 1993. pp. 70-150.
- [07] FEO, T.; RESENDE, M. 1994. "A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set". Operations Research, v. 42, pp. 860-879, 1994.
- [08] MLADENOVIC, N; Hansen, P. 1997. "Variable neighborhood search". Computers and Operations Research 24 (11): 1097-1100.
- [09] POTVIN, J.Y. (1993). "The Traveling Salesman Problem: A Neural Network Perspective". ORSA Journal on Computing, v. 5, pp. 328-348, 1993.
- [10] FUKUNAGA, A. S. 2008. "A new grouping genetic algorithm for the Multiple Knapsack Problem," IEEE Congress on Evolutionary Computation, pp. 2225-2232, Jun. 2008.
- [11] KHURI, S; Bäck, T; Heitkötter, J. 1994. "The Zero / One Multiple Knapsack Problem and Genetic Algorithms," in ACM Symposium on Applied Computing, 1994, pp. 188-193.
- [12] XIAO-HUA, X; An-bao, W; Ai-bing, N. 2010. "Competitive Decision Algorithm for 0-1 Multiple Knapsack Problem," II Int. Workshop on Education Technology and Computer Science, pp. 252-255.
- [13] ADAMS, J., Balas, E., e Zawack, D. 1988. "The shifting bottleneck procedure for job shop scheduling". Management Science, 34:57-73, 1988.
- [14] MANSANO, E. M. 2008. "Algoritmos Busca Tabu Paralelos Aplicados ao Planejamento da Expansão da Transmissão de Energia Elétrica". Dissertação de Mestrado. Universidade Estadual Paulista "Julio de Mesquita Filho" - Faculdade de Engenharia - Campus de Ilha Solteira. 2008.
- [15] \_\_\_\_\_. 2009. "KNAPSACK\_MULTIPLE - Data for the 01 Multiple Knapsack Problem". Disponível em: < [http://people.sc.fsu.edu/~jburkardt/datasets/knapsack\\_multiple/knapsack\\_multiple.html](http://people.sc.fsu.edu/~jburkardt/datasets/knapsack_multiple/knapsack_multiple.html) >. Acesso em 6 de Novembro de 2012.