

# Hashing na Solução de Problemas Combinatórios Difíceis

Lucila Maria de Souza Bento\*

Vinícius Gusmão Pereira de Sá

Departamento de Ciência da Computação, Instituto de Matemática, UFRJ,

21941-916, Rio de Janeiro, RJ

E-mail: lucila@ufrj.br, vigusmao@dcc.ufrj.br,

**Jayme Luiz Szwarcfiter**

Instituto de Matemática, NCE e COPPE, UFRJ,

21941-916, Rio de Janeiro, RJ

E-mail: jayme@nce.ufrj.br.

## RESUMO

Problemas de otimização combinatória possuem grande importância prática, aparecendo em diversas aplicações. Muitos são difíceis de se resolver, pois requerem a busca de uma solução num espaço exponencialmente grande, sem que algoritmos eficientes sejam conhecidos, ou mesmo possíveis. Neste trabalho, é apresentada uma técnica que, utilizando *hashing* como coadjuvante à busca exaustiva (*backtracking*), oferece ganhos consideráveis de tempo e espaço em muitas situações. Os problemas CARGA DA Balsa e QUEBRA-CABEÇA DO 15 serão abordados para ilustrar seu uso.

**Palavras-chave:** *busca exaustiva, hashing, problemas combinatórios*

## 1 Introdução

Quando inexistem algoritmos eficientes para tal, a busca pela solução ótima de um problema de otimização combinatória é em geral implementada como uma busca exaustiva em um grafo implícito [4], cujos nós correspondem a soluções viáveis — ou estados, ou configurações válidas — do problema em questão. Suas arestas correspondem a escolhas possíveis numa sequência de decisões que permite obter todos os estados ou configurações atingíveis a partir do ponto de partida, o estado inicial. Em outras palavras, uma aresta  $xy$  indica que existe um passo, uma tomada de decisão na sequência pré-estabelecida, que nos leva diretamente da configuração ou estado  $x$  à configuração ou estado  $y$ . O que se deseja é encontrar um nó do grafo que maximiza ou minimiza determinada função, ou que constitui um “estado-alvo”, atendendo certa condição de parada. Esta é a técnica conhecida como *backtracking*.

Ao realizar uma busca exaustiva é necessário ter atenção quanto à sequência de configurações, pois, se não forem tomados os devidos cuidados, configurações iguais, simétricas ou equivalentes podem ser visitadas mais de uma vez, acarretando perda de tempo ou laços intermináveis. Por esta razão, é preciso demarcar, ao longo da busca, que nós do grafo já foram visitados. O que nos leva invariavelmente a um problema de espaço, pois o conjunto de todas as configurações possíveis — atingíveis ou não, mas como sabê-lo? — pode ser enorme, e por conseguinte a quantidade de memória alocada para a desejada demarcação.

Diante desses questões, é possível propor uma abordagem que promove uma redução considerável de tempo e espaço, como ilustrado na solução dos problemas definidos a seguir.

---

\*Bolsista CNPq

## 2 Definição dos problemas

Como se sabe, balsas são embarcações usadas para transportar veículos através de cursos d'água. Normalmente possuem largura suficiente para formar duas pistas com veículos de diferentes tamanhos, ocupando todo o seu comprimento. Os veículos aguardam pela embarcação em uma fila única, onde sua ordem na fila determina sua ordem de embarque. O problema CARGA DA Balsa consiste em determinar como carregar a balsa com o maior número possível de veículos, onde cada veículo pode ocupar a pista da direita ou a da esquerda. Se não houver espaço livre para o veículo à frente da fila, a balsa está cheia e os veículos restantes devem aguardar a próxima balsa [3]. Este problema guarda semelhanças com três conhecidos problemas NP-difíceis [2]: EMPACOTAMENTO, PARTIÇÃO e ESCALONAMENTO DE TAREFAS.

O QUEBRA-CABEÇA DO 15 consiste em uma grade  $4 \times 4$  de quadrados numerados de 1 a 15 com um quadrado faltando. O objetivo do jogo é, a partir de uma ordem aleatória, ordenar os quadrados numerados, trocando a posição do espaço vazio com um dos quadrados que seja seu vizinho na vertical ou na horizontal, de modo que, na configuração final do jogo, o espaço vazio ocupe a última posição, e tenha sido realizado o menor número possível de movimentos [1].

## 3 Sobre as Soluções Propostas

A técnica conhecida como hashing consiste no mapeamento de chaves de um espaço grande em um espaço relativamente menor. O mapeamento é realizado pela *função hash*, que gera um código, ou endereço-base, usado para localizar a chave numa tabela. Esta localização requer tempo médio constante, utilizando espaço apenas linear na quantidade de chaves armazenadas [5]. Nos algoritmos propostos para solução dos problemas CARGA DA Balsa e QUEBRA-CABEÇA DO 15 emprega-se hashing para a demarcação dos nós visitados ao longo do backtracking.

No problema da Balsa, duas configurações  $x$  e  $y$  são equivalentes se, em  $x$ , os carros embarcados ocupam exatamente a mesma porção de cada pista que é ocupada pelos carros embarcados em  $y$ , a menos de simetria. Uma vez visitada uma dessas configurações, não é necessário visitar qualquer outra. Isto reduz o número de nós da árvore de busca, que seria exponencial na quantidade de carros, para um número que é pseudo-polinomial no tamanho da balsa. Para demarcar as configurações visitadas utilizando uma matriz, seria necessário pré-alocar memória para todos os possíveis pares de espaços ocupados, ao passo que, evidentemente, apenas alguns deles aparecerão de fato durante a busca — tal o ganho da abordagem com hashing.

No problema do 15, o número de configurações possíveis é enorme ( $16!$ ), mas o número de configurações que precisarão ser demarcadas é, em geral, bem menor, pois o número máximo de movimentos necessários para ir de uma posição solucionável qualquer à posição final é limitado.

## Referências

- [1] E. Kasner and J.R. Newman, *Mathematics and the Imagination*. Redmond, WA: Tempus Books, pp. 177-180, 1989.
- [2] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications, Inc., New York, 1998.
- [3] S. S. Skiena and M.A. Revilla, *Programming Challenges*, Springer, New York, 2003.
- [4] J. L. Szwarcfiter, *Grafos e Algoritmos Computacionais*, Editora Campus Ltda., Rio de Janeiro, 2ª ed., 1986.
- [5] J. L. Szwarcfiter and L. Markenzon, *Estruturas de Dados e Seus Algoritmos*, LTC, Rio de Janeiro, 2ª ed., 1994.