Busca Bidirecional

Daniel B. de Salles, Guilherme A. D. Maciel, Halliday G. C. dos Santos, Iago C. Andrade, Vinicius N. Targa

[1]Departamento de Computação
Universidade Federal de Ouro Preto (UFOP) – Ouro Preto, MG – Brazil

daniel.bortot@aluno.ufop.edu.br, guilherme.maciel@aluno.ufop.edu.br,

halliday.santos@aluno.ufop.edu.br, iago.andrade@aluno.ufop.edu.br,

vinicius.targa@aluno.ufop.edu.br

Abstract. *This report aims to describe the bidirectional search algorithm, a uninformed search algorithm, often capable of solving any solvable problem, but not in the most efficient way. This analysis also characterizes this graph search algorithm according to concepts such as its NP-completeness and asymptotic complexity.*

Resumo. *Este relatório visa descrever o algoritmo de busca bidirecionada, um algoritmo de busca sem informação, por vezes capaz de resolver qualquer problema solucionável, mas não da maneira mais eficiente. Esta análise também caracteriza este tipo de busca de acordo com conceitos como sua NP-completeza e sua complexidade assintótica.*

1. Descrição

A busca bidirecional é uma estratégia de busca cega, ou seja, é um tipo de busca que não possui nenhuma informação adicional sobre estados, além daquelas fornecidas na definição do problema.

Ao contrário das buscas heurísticas, esta estratégia apenas possui capacidade de gerar nós sucessores e de discernir um estado objetivo de um estado não objetivo, e não conseguem identificar o quão ”promissores” os estados não objetivos são.

2. Características

O algoritmo da busca bidirecional visa executar duas buscas simultâneas, uma partindo do estado inicial e outra partindo do objetivo, de forma que as duas buscas se encontrem num ponto intermediário.

Sua implementação se dá aplicando uma verificação para averiguar se a borda das duas buscas se encontram; caso isso ocorra, uma solução foi encontrada. Observa-se que a primeira solução encontrada não necessariamente é a solução ótima, e pode-se aplicar uma busca adicional para investigar se não há atalhos no espaço.

Ademais, a busca bidirecional é completa, ou seja, sempre encontra a solução se ela existe para um fator de ramificação b finito e quando ambos os sentidos usam busca em largura. Outrossim, a busca é ótima se os custos dos passos são idênticos e ao usar busca em largura para ambos os sentidos.

Imaginando um grafo com um fator de ramificação b, ou seja, cada nó se ramificando em b nós, e levando em consideração um profundidade d, tem-se como motivação que $b^{d/2} + b^{d/2}$ é muito menor que b^d. Ou seja, a partir da figura 1, observa-se que a área dos dois círculos pequenos é menor que a área de um único suposto círculo grande, com centro em 'Start' e que chega até o objetivo 'Goal'[Norvig and Russel 2013].

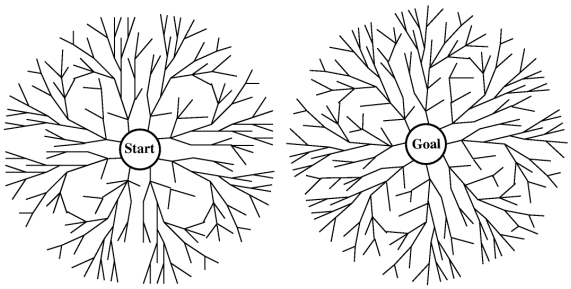

Figure 1. Esquema da ramificação dos nós inicial e objetivo, onde a busca bidirecional está prestes a ter sucesso, ou seja, as ramificações em questão possuem uma fronteira em comum.

Para contextualizar, a busca em largura possui complexidade $b + b^2 + b^3 + ... + b^d = O(b^d)$. Tendo como exemplo um fator de ramificação b = 10 e profundidade d = 8, uma busca em largura geraria 111.111.110 nós, enquanto a busca bidirecional geraria 22.220 nós. Assim, sua complexidade de tempo é de $O(b^{d/2})$, assumindo que as duas buscas são executadas simultaneamente e em tempo constante. Além disso, sua complexidade de espaço também é de $O(b^{d/2})$, uma vez que para as duas buscas se encontrarem, a fronteira de uma delas deve ser armazenada em memória.

Contudo, apesar de tal otimização de tempo e espaço serem excepcionais em teoria, ressalta-se que a busca inversa pode não ser tão simples quanto parece. Na presença de mais de um estado objetivo ou diante de um objetivo que se trata de uma descrição abstrata, a busca bidirecional é difícil de ser realizada.

3. Aplicações

O algoritmo em questão pode ser utilizado para averiguar a existência de um caminho entre dois pontos em um grafo, sendo capaz de resolver labirintos (visto que a entrada e a saída do labirinto são conhecidas), bem como identificar um caminho possível entre duas cidades (a cidade de partida e de chegada também são conhecidas).

Seu funcionamento enquanto busca não informada possibilita encontrar o caminho mais curto em grafos de não ponderados ou ponderados de peso único (por exemplo, considerando a distância entre cidades)[J.A. Pavlik and Jacobson 2021].

A busca bidirecional também pode ser utilizada na otimização de algoritmos como o Djsktra (Djsktra Birecional)[Vaira and Kurasova 2011] e de algoritmos de caminho mais curto (em grafos de peso múltiplo)[Ma and Liang 2013]. Além disso, também se observa sua aplicação na resolução das ações de um agente inteligente para alcançar uma meta[Melo 2018], questão notável no contexto de inteligência artificial.

4. Pseudocódigo

Um possível pseudocódigo para o algoritmo de busca bidirecional é apresentado na figura 2.

```
Entrada: Um grafo, nó inicial e nó final

      fronteira inicial = [nó inicial]
      fronteira final = [no final]

Enquanto nenhuma das fronteiras for vazia:
      Busca em largura: fronteira inicial
      Busca em largura: fronteira final
            Se um nó repete em ambas as fronteiras:
                  Retorna caminho encontrado
Fim Enquanto

Retorna caminho inexistente
```

Figure 2. Pseudocódigo do algoritmo de busca bidirecional

References

J.A. Pavlik, E. S. and Jacobson, S. (2021). Two new bidirectional search algorithms. *Comput Optim Appl*, 80:377–409.

Ma, H. and Liang, R. (2013). Using bidirectional search to compute optimal shortest paths over multi-weight graphs. In *2013 International Conference on Information Science and Cloud Computing Companion*, pages 66–71. IEEE.

Melo, V. T. (2018). Aplicação de busca bidirecional em planejamento como verificação simbólica de modelos. In UFC, editor, *Ciência da Computação - Quixadá - Monografias*. Universidade Federal do Ceará.

Norvig, P. and Russel, S. (2013). *Inteligência Artificial*. Grupo GEN, 3rd edition.

Vaira, G. and Kurasova, O. (2011). Parallel bidirectional dijkstra's shortest path algorithm. *Databases and Information Systems VI, Frontiers in Artificial Intelligence and Applications*, 224:422–435.