ALGORITMA BRANCH-AND-BOUND UNTUK MENCARI JARAK TERPENDEK

Arya Tri Prabawa
NIM: 13506063

Teknik Informatika ITB
Jl. Ganesha 10, Bandung 40132
e-mail: aryaprabawa@yahoo.com

ABSTRAK

Algoritma *Branch and Bound* adalah algoritma pencarian dengan skema *Breadth First Search* yang meminimumkan pembentukan pohon solusi dengan sebuah fungsi pembatas. Fungsi pembatas ini mempertimbangkan nilai ongkos dari setiap langkah yang akan diambil. Dalam makalah ini, penulis menerapkan algoritma *Branch and Bound* tersebut pada pencarian solusi jarak terpendek.

Kata kunci: branch and bound, breadth first search, jarak terpendek.

1. PENDAHULUAN

Permasalahan pencarian jarak terpendek direpresentasikan dengan sebuah papan permainan di mana terdapat dua titik, yaitu titik mulai dan titik selesai, dan papan memiliki sejumlah penghalang sehingga kita harus mencari jarak terpendek yang dapat dilalui dari titik mulai menuju titik selesai. Sebelum mulai memaparkan proses pencarian solusi, terlebih dahulu penulis memberikan sedikit gambaran mengenai algoritma *Branch-and-Bound*.

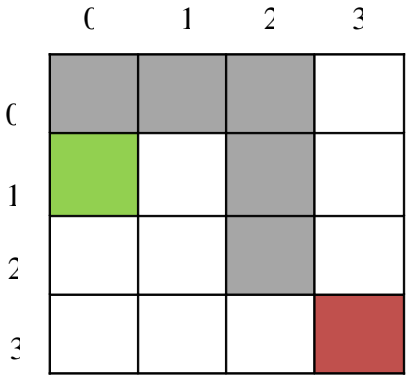
Pencarian solusi pada algortima *Branch and Bound* (B&B) menggunakan skema pencarian *Breadth First Search* (BFS). Dengan metode *BFS*, simpul akar pada pohon pencarian dibangkitkan pertama kali, kemudian semua simpul anak-anaknya, kemudian *successor* dari setiap simpul anak, dan seterusnya.

Algoritma B&B meminimumkan pembentukan pohon pada skema BFS. Pencarian ke simpul solusi dipercepat dengan memilih simpul hidup berdasarkan nilai ongkos (cost). Setiap simpul hidup diasosiasikan dengan sebuah ongkos yang menyatakan nilai batas (bound). Nilai batas yang dipergunakan dalam pembuatan aplikasi shortest-path finder ini adalah panjang lintasan dari sebuah simpul ke simpul solusi terdekat.

2. PENCARIAN SOLUSI

Langkah-langkah Algoritma *Branch and Bound* yang dipergunakan adalah sebagai berikut:

1. Simpul awal yang dibangkitkan adalah simpul *start*
2. Simpul memiliki 4 cabang, yaitu simpul kiri, kanan, atas dan bawah, catat simpul yang kini sedang dibangkitkan sebagai simpul pendahulu keempat simpul anak ini
3. Masukkan simpul cabang yang bukan tembok dan belum pernah dibangkitkan ke dalam list
4. Simpul yang terdapat dalam list dengan nilai biaya terendah dibangkitkan dan dihapus dari list
5. Ulangi langkah 2 sampai dengan 4 hingga simpul akhir (*finish*)
6. Dimulai dari simpul akhir, telusuri simpul-simpul pendahulu hingga sampai pada simpul awal
7. Simpul-simpul inilah lintasan terpendek yang dapat dilalui oleh robot

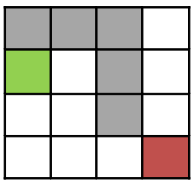


Gambar 1. Contoh papan permainan

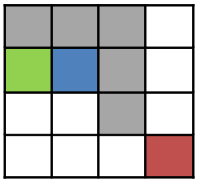
Sebagai ilustrasi, diberikan sebuah permasalahan seperti gambar di atas. Pencarian akan dimulai dari simpul *start* (1,0). Simpul anak yang bukan tembok dan belum dibangkitkan adalah simpul kanan (1,1) dan simpul bawah (2,0). Kedua simpul ini dimasukkan ke dalam list. Karena nilai biaya keduanya sama, maka dipilih salah satu berdasarkan urutan masuk ke dalam list, di mana dalam aplikasi ini urutannya adalah kanan, bawah, kiri, lalu atas. Simpul (1,1) dibangkitkan dan dihapus dari list, kemudian simpul (2,1) dimasukkan ke dalam list. Simpul (2,0) dengan nilai biaya terkecil dalam list dibangkitkan dan dihapus dari list, simpul (2,1) dan (3,0) dimasukkan ke dalam list. Simpul (2,1) dibangkitkan, simpul (3,1) dimasukkan ke dalam list. Simpul (3,0) dibangkitkan, simpul (3,1) dimasukkan ke dalam list. Selanjutnya simpul (3,1) dibangkitkan, simpul kanan (3,2) dimasukkan ke dalam list. Simpul (3,2) dibangkitkan, simpul akhir (3,3) dimasukkan ke dalam list. Pencarian selesai.

Perlu diingat bahwa setiap kali sebuah simpul dimasukkan ke dalam list, simpul yang sedang dibangkitkan dicatat sebagai simpul pendahulunya. Dengan menelusuri simpul-simpul pendahulu mulai dari simpul akhir sampai dengan simpul awal, maka lintasan terpendek yang dapat dilalui oleh robot mulai dari simpul awal adalah simpul (2,0), (3,0), (3,1), (3,2) sebelum sampai pada simpul akhir.

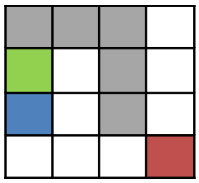
Untuk lebih memperjelas, berikut ini penulis sertakan gambar-gambar pembentukan pohon solusi.



Gambar 2. Keadaan awal (pohon solusi level-0)

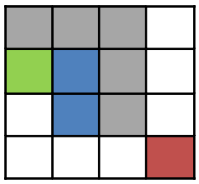


(1)

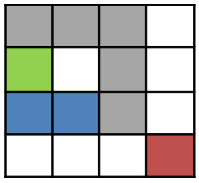


(2)

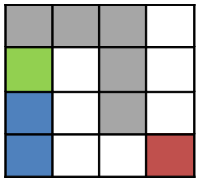
Gambar 3. Pohon solusi level-1



(1)

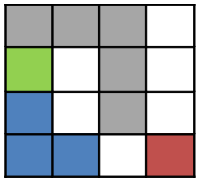


(2.1)



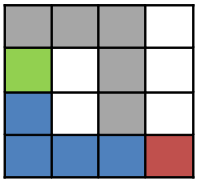
(2.2)

Gambar 4. Pohon solusi level-2



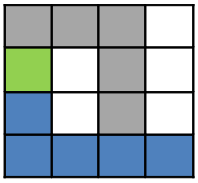
(2.2)

Gambar 5. Pohon solusi level-3



(2.2)

Gambar 6. Pohon solusi level-4



(2.2)

Gambar 7. Pohon solusi level-5

3. KESIMPULAN

Branch and Bound adalah algoritma pencarian solusi dengan skema *Breadth First Search*, di mana pembentukan pohon pencariannya diminimumkan dengan pemberian nilai biaya pada setiap simpul. Dengan demikian, pencarian akan diteruskan hanya pada simpul dengan nilai biaya terkecil. Penerapannya pada kasus pencarian jarak terpendek ini berhasil memberikan solusi optimal berupa lintasan terpendek yang dapat dilalui odari titik mulai menuju titik selesai.

REFERENSI

[1] Munir, Rinaldi. “Strategi Algoritmik”. Informatika ITB. 2006.