PERBANDINGAN PENERAPAN ALGORITMA BRUTE FORCE DAN ALGORITMA DEEP FIRST SEARCH DALAM PERSOALAN PEDAGANG KELILING

M. Auriga Herdinantio (13506056)

Program Studi Teknik Informatika Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
e-mail: if16056@students.if.itb.ac.id

ABSTRAK

Persoalan pedagang keliling (TSP) merupakan persoalan optimasi bagi pedagang keliling yang ingin berkunjung ke beberapa kota dan kembali ke kota asal keberangkatan dengan melewati perjalanan terpendek. Persoalan tersebut dapat dipecahkan dengan beberapa metode. Namun hingga saat ini belum ditemukan algoritma yang mangkus untuk menyelesaikannya.
Cara termudah untuk menyelesaikan masalah ini yaitu dengan mencoba semua kemungkinan rute dan mencari rute yang terpendek atau dinamakan algoritma Brute Force.
Sifat- sifat algoritma Brute Force yang seperti itu memberikan beberapa kelebihan maupun kekurangan. Adapun kelebihan yang dimilikinya adalah kehandalannya dalam menemukan solusi dari sebuah permasalahan , akan tetapi kelemahannya yaitu algoritma ini tidak mangkus, mempunyai kompleksitas yang terbesar dalam pemecahan suatu permasalahan dan tidak efisien untuk jumlah kemungkinan solusi yang banyak .
Namun, pada zaman yang serba praktis sekarang ini dibutuhkan algoritma yang dapat menyelesaikan TSP dengan cepat sehingga diperoleh solusi yang mendekati solusi optimal.
Seiring dengan waktu yang berjalan dan juga perkembangan ilmu pengetahuan dan teknologi, permasalahan pencarian lintasan terpendek ini telah terpecahkan dengan berbagai algoritma.
Makalah ini akan membandingkan penerapan algoritma Brute Force dengan algoritma Deep First Search pada persoalan pedagang keliling agar dapat diketahuibalgoritma mana yang lebih baik untuk menyelesaikan persoalan ini.

Kata kunci: TSP, brute force, DFS.

1. PENDAHULUAN

Kenaikan harga bahan bakar minyak membuat kalangan industri harus memutar otak. Kenaikan ini tentu saja membuat ongkos produksi semakin bertambah. Sesuai dengan hukum ekonomi yang mengatakan bahwa "Dengan pengorbanan yang sekecil-kecilnya didapat keuntungan yang sebesar-besarnya", maka ongkos produksi pun harus ditekan sekecil-kecilnya.
Oleh karena itu, cara apa pun harus dilakukan untuk menghemat ongkos produksi. Salah satunya adalah dengan pemakaian bahan bakar minyak yang seefesien mungkin. Untuk mengefisienkan penggunaan bahan bakar minyak diperlukan sebuah solusi atas pencarian rute yang terpendek sehingga memudahkan dalam pendistribusian barang dan kembali ketempatnya semula dengan rute yang terpendek (*travelling solving problem).*
TSP merupakan masalah klasik yang sampai saat ini belum ada algoritma yang cukup mangkus untuk menyelesaikannya. Salah satu algoritma pemecahan masalah TSP adalah algoritma Brute Force yang mempunyai kompleksitas O(n!) dan algoritma DFS (*Depth First Search*).
Diharapkan nantinya algoritma yang dihasilkan dapat dipakai dalam memberikan solusi rute yang paling efektif dalam hal jarak tempuh .

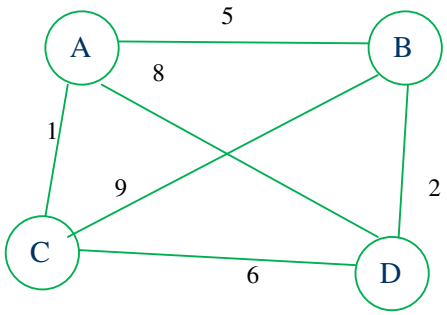
.

2. PEMBAHASAN

2.1. Algoritma Brute Force pada TSP

Adapun algoritma *Brute Force* pada dasarnya adalah alur penyelesaian suatu permasalahan dengan cara berpikir yang sederhana , tidak membutuhkan suatu permikiran yang cukup lama untuk dapat menyelesaikan sebuah permasalahan tertentu . Model penelitian yang akan dilakukan adalah mengimplementasikan penggunaan algoritma brute force pada permasalahan TSP dengan menggunakan konsep *exhaustive search*

Misal terdapat empat kota, yaitu kota A, kota B, kota C, dan Kota C. Jarak kota A-B adalah 5 km, kota A-C 9 km, kota A-D 8 km, kota B-C 9 km, kota B-D 2 km, kota C-D 6 km.
Permasalahan tersebut dapat dirangkum dalam sebuah graf sebagai berikut :



Gambar 1 : Graf yang akan dianalisis

TSP pada gambar 1 akan coba diselesaikan dengan algoritma Brute Force. Diinginkan solusi rute terpendek dari state awal ke state tujuan dengan ketentuan hanya melewati sebuah node / kota sekali dan kembali ke state awal /kota keberangkatan.
Pemodelan kasus ini dapat dimodelkan sebagai graf dengan node yang merupakan representasi kota dan bobot antar node merepresentasikan jarak antar kota satu dengan kota yang lainnya .
Kasus diatas tidak lain hanyalah representasi sirkuit hamilton dengan pencarian sesuai dengan bobot paling minimum . Graf yang merepresentasikan hamilton tersebut disebut sebagai hamiltonian .
Adapun pemecahan masalah untuk graf diatas dapat diselesaikan sebagai berikut :
Enumerasikan langkah sejumlah (n-1) ! Jadi dengan node = n = 4 , maka didapatkan jumlah pilihan = (4 -1)! = 3! = 3 X 2 X 1 = 6. Dari graf diatas , maka didapat permutasi langkah yang dihasilkan seperti ini :

Graf yang ditempuh	Jarak
A-B-C-D-A	28
A-D-B-C-A	20
A-C-B-D-A	20
A-D-C-B-A	28
A-C-D-B-A	14
A-B-D-C-A	14

Tabel 1. Tabel hasil enumerasi dari Gambar 1

Dari tabel diatas dapat dilihat bahwa sebetulnya rute yang ditelusuri dapat diefisienkan dengan cara menghilangkan sejumlah rute yang sebetulnya merupakan pencerminan dari rute yang lain. Hal ini dapat dibuktikan dengan bobot yang dihasilkan hanyalah = 3 jenis , yaitu = 28, 14, 20. Oleh karena itu rute yang sama diambil salah satu saja , maka kita dapat menghilangkan rute – rute sebagai berikut :
- Untuk bobot 28 = A-D-C-B-A
- Untuk bobot 20 = A-C-B-D-A
- Untuk bobot 14 = A-C-D-B-A

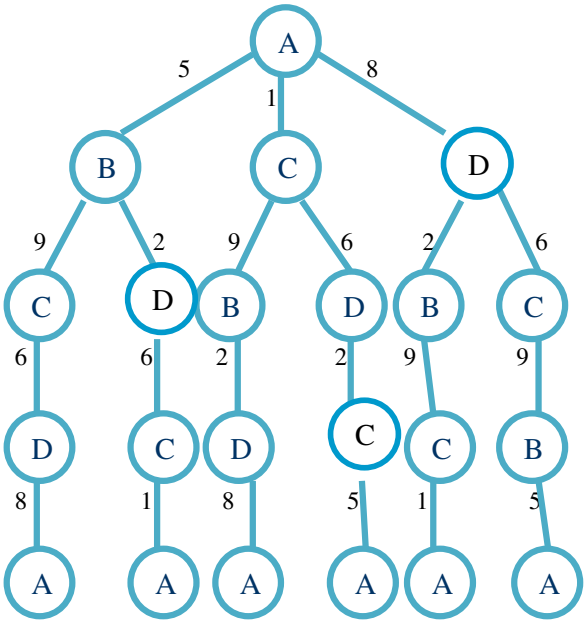
Jadi dengan penghilangan rute node yang sepadan tersebut , diharapkan jumlah kompleksitas dapat dikurangi sejumlah ((n-1) !)/ 2 . Maka kompleksita yang didapatkan = (3 !)/ 2 = 3 . Tetapi sayang , walaupun rute ini sudah di efisienkan , besarnya komplestitas ternyata masih besar.
Dari langkah – langkah yang sudah kita lakukan diatas , maka didapatkan bobot yang terkecil adalah
A-B-D-C-A = 14.
Adapun penyelesaian masalah TSP dengan algoritma Brute Force dapat kita simpulkan sebagai berikut:
1. Mengenumerasi semua Sirkuit Hamilton dari graf lengkap TSP,
2. Menghitung bobot setiap sirkuit Hamilton yang ditemukan pada langkah 1,
3. Memilih sirkuit Hamilton yang mempunyai bobot terkecil.

Karena algoritma ini menghitung bobot untuk setiap Sirkuit Hamilton yang mungkin terjadi, maka kompleksitasnya sebesar jumlah Sirkuit Hamilton untuk graf lengkap bersimpul n yang dimulai dari sebuah simpul, yakni permutasi dari n buah simpul = n1* ... * 1 = (n1)!.
Maka, kompleksitasnya adalah O(n!).

2.2 Algoritma DFS pada TSP

Pencarian solusi dengan DFS dicirikan dengan ekspansi simpul-simpul terdalam terlebih dahulu. Mula-mula simpul akar dibangkitkan pertama kali, kemudian simpul pada aras kedua, lalu sebuah simpul pada aras ketiga, begitu seterusnya.
Jadi pencarian mengikuti sebuah lintasan tunggal mulai dari simpul akar terus menurun ke bawah ke simpul-simpul pada aras di bawahnya. Jika pencarian mencapai simpul yang tidak mempunyai succesor, pencarian dilakukan pada lintasan lainnya.
Pada algoritma DFS untuk memecahkan permasalahan TSP ini apabila sebelumnya sudah ada nilai minimum yang lebih kecil dari nilai lintasan yang akan diekspansi. bobot lintasan tersebut tidak akan dihitung.

Misalnya, untuk kasus graf pada Gambar 1yang dimodelkan dalam pohon seperti ini:



Gambar 2. Pohon ruang status yang memodelkan graf pada Gambar 1

Terdapat lintasan **A-B-D-C-A** yang terpilih sebagai lintasan minimum sementara (bobotnya 14), maka simpul **A-D-B-C** tidak akan diekspansi lagi karena bobotnya tidak lebih kecil dari bobot minimum sementara (19 > 14).
Adapun penyelesaian masalah TSP dengan algoritma DFS dapat kita simpulkan sebagai berikut:
1. Bangun sebuah pohon yang cabangnya berupa simpul pada graf,
2. Lakukan metode DFS pada tiap cabang sampai semua simpul dipilih (tidak ada yang dipilih dua kali),
3. Hitung bobotnya,
4. Lakukan langkah ke2 dan ke3 sampai seluruh simpul asal telah dipilih. Apabila pada waktu membangkitkan simpul anak ternyata tidak lebih kecil dari minimum sementara, maka simpul tersebut dimatikan (tidak diekspansi lebih lanjut).

III. ANALISIS

Algoritma brute force rata – rata hampir dapat dipakai untuk menyelesaikan permasalahan –permasalahan yang terjadi disekitar kita . Untuk kasus TSP , pemakaian algoritma brute force menjamin penyelesaian solusi dengan baik. Dalam kasus TSP implementasi algoritma brute force dikenal dengan sebutan *exhaustive search* .

Kelemahan dari metode ini adalah pencarian solusi harus membangkitkan sebanyak (n-1) ! / 2 . Untuk jumlah node yang sedikit , pemakaian brute force sangat efisien , tetapi untuk jumlah node yang banyak , maka algoritma ini menjadi sangat tidak efisien.
Pada umumnya, algoritma DFS lebih mangkus daripada algoritma Brute Force pada penyelesaian masalah TSP, karena tidak semua kemungkinan solusi harus dienumerasikan.
Kasus terbaik untuk algoritma DFS ini terjadi ketika solusi pertama yang ditemukan adalah solusi terbaik,dan pembangkitan simpul selanjutnya selalu tidak lebih kecil dari bobot solusi pertama tersebut. Hal ini dimungkinkan jika permasalahan TSP yang diselesaikan memiliki bobot yang tidak seimbang, dimana beberapa lintasan antara dua simpul memiliki bobot yang sangat kecil dibandingkan yang lain.
Kasus terburuk untuk algoritma DFS ini mempunyai kompleksitas yang sama dengan algoritma Brute Force karena jika graf TSP yang diselesaikan cukup seimbang, maka hampir semua simpul akan dibangkitkan.

IV. KESIMPULAN

Penggunaan algoritma Brute Force di dalam masalah pencarian cukup efisien untuk yang node-nya sedikit , tetapi untuk yang node-nya banyak , algoritma ini menjadi sangat tidak mangkus , karena harus menelusuri seluruh kemungkinan rute yang ada.
Penyelesaian *Travelling Salesperson Problem* (TSP) dengan menggunakan DFS dalam kasus rata-rata lebih mangkus dibandingkan dengan algoritma Brute Force karena hanya sirkuit-sirkuit yang mengarah ke solusi yang dibangkitkan.
Oleh karena itu, untuk kasus-kasus TSP tertentu, algoritma DFS lebih baik daripada algoritma Brute Force.

REFERENSI

[1] http://en.wikipedia.org
diakses pada tanggal 16 Mei 2008 pukul 19:40
[2] Munir, Rinaldi, “Diktat Kuliah IF2251 Strategi Algoritmik”, Program Studi Teknik Informatika, 2006.