Jurnal Gradien Vol. 12 No. 2 Juli 2016 : 1171-1174
ISSN 0216-2393

Travelling Salesperson Problem dengan Pendekatan Heuristik

Ulfasari Rafflesia

Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Bengkulu, Indonesia

Diterima 23 April 2016; Disetujui 15 Juni 2016

Abstrak – *Travelling Salesperson Problem* (TSP) adalah permasalahan seorang *salesman travelling* yang harus melakukan kunjungan ke sejumlah kota tepat satu kali dalam menjajakan produknya. Tujuan TSP adalah mencari rute perjalanan semua kota dengan total bobot minimum. Beberapa pendekatan *Branch and Bound* telah dikembangkan untuk memecahkan persoalan *Travelling Salesman Problem* (TSP). Pada saat mencari solusi TSP dengan banyak kota, metode *Branch and Bound* membutuhkan waktu yang cukup lama sehingga diperlukan pendekatan heuristik untuk mencari solusi TSP tersebut. Heuristik adalah metode yang digunakan untuk memecahkan masalah dengan *trial and error* ketika pendekatan algoritma dinilai tidak praktis dan tidak efisien untuk menghasilkan suatu solusi. Penelitian ini bertujuan untuk mencari solusi dari TSP dengan pendekatan heuristik yaitu *Cheapest-Insertion Heuristics (CIH).* Hasil penelitian menunjukkan bahwa metode *Cheapest-Insertion Heuristics (CIH)* menghasilkan sebuah tur yang optimal.

Kata Kunci: **TSP, Heuristik, *Cheapest-Insertion Heuristics* (CIH), Solusi Optimal**

1. Pendahuluan

Traveling Salesman Problem (TSP) merupakan masalah klasik yang mencoba mencari rute terpendek yang dapat dilalui *salesman* yang ingin mengunjungi beberapa kota tanpa harus mendatangi kota yang sama lebih dari satu kali. Persoalan ini menggunakan representasi graf dan riset operasi untuk memodelkan persoalan yang diwakili sehingga lebih mudah penyelesaiannya. Tujuan TSP adalah mencari rute perjalanan semua kota dengan total bobot minimum. Pada perkembangannya, ternyata TSP merupakan persoalan yang telah banyak diaplikasikan pada berbagai persoalan dunia nyata. TSP melibatkan suatu algoritma yang mengharuskan untuk mencari kemungkinan semua solusi yang ada.
Beberapa contoh masalah yang dapat diselesaikan dengan menggunakan pendekatan TSP adalah **[1]** :

1) Pencarian rute bis sekolah untuk mengantarkan siswa
2) Pencarian rute truk pengantar parcel
3) Pengambilan tagihan telepon

Selain contoh di atas, TSP juga bisa digunakan untuk menyelesaikan persoalan penjadwalan produksi [**2**]. Ada banyak algoritma untuk memecahkan masalah ini, diantaranya dengan menggunakan *Linier Programming (LP), Genetic Algorithm (GA), Nearest Neighbourhoud Heuristics (NNH) dan Cheapest Insertion Heuristics (CIH)* [3]. Beberapa pendekatan *Branch and Bound* telah dikembangkan untuk memecahkan persoalan TSP. Ketika menggunakan metode *Branch and Bound* untuk memecahkan TSP dengan banyak kota akan memerlukan perhitungan waktu yang cukup lama. Salah satu cara yang sering digunakan untuk mempercepat pencarian solusi masalah TSP adalah dengan metode heuristik, atau heuristik.

Heuristic adalah seni dan ilmu menemukan. Heuristic adalah suatu metoda coba-coba yang terpaksa diambil ketika pendekatan algoritmik bersifat tidak praktis **[4]**. Heuristic tidak selalu dapat memecahkan masalah, tetapi sering kali memecahkan masalah dengan cukup baik dan lebih cepat dari pencarian solusi secara lengkap karena teknik heuristic bertujuan untuk mengurangi jumlah pencarian solusi. Metode heuristic yang cukup efektif untuk menyelesaikan masalah TSP adalah *Nearest Neighbor Heuristics (NNH)* dan *the Cheapest-Insertion Heuristics (CIH)* **[5]**.

Di antara heuristics untuk TSP, Metode *Nearest Neighbor Heuristic* (NNH) adalah metode heuristik yang paling sederhana. Metode NNH pertama kali diperkenalkan pada tahun 1983. Pemecahan masalah dilakukan dengan memulai titik awal kemudian mencari titik terdekat atau jarak paling minimum. Kemudian dari titik tersebut dicari lagi titik yang belum dikunjungi dengan jarak yang paling minimum, iterasi berhenti setelah semua titik dilewati dan kembali ke titik awal. NNH merupakan algoritma yang mudah untuk diimplementasikan dan mudah untuk dieksekusi, tetapi tidak menjamin solusi yang dihasilkan optimal.

Metode *Cheapest Insertion Heuristic* (CIH) membangun suatu tour dari sikel-sikel kecil dengan bobot minimal dan secara berturut-turut ditambah dengan titik baru. Pemilihan titik baru tersebut dilakukan bersamaan dengan pemilihan sisi sehingga didapatkan nilai penyisipan minimum. Selanjutnya titik baru tersebut disisipkan di antara dua titik yang membentuk sisi yang telah terpilih.

2. Landasan Teori

Traveling Salesman Problem (TSP)

Permasalahan matematika tentang *Traveling Salesman Problem* dikemukakan pada tahun 1800 oleh Matematikawan Irlandia William Rowan Hamilton dan Matematikawan Inggris Thomas Penyngton. Diskusi mengenai awal studi dari Hamilton dan Kirkman dapat ditemukan di Graph Theory tahun 1736-1936 oleh N. L. Biggs, E. K. LLoyd, dan R. J. Wilson, di Oxford, tahun 1976. Bentuk umum dari TSP pertama dipelajari oleh para Matematikawan mulai tahun 1930. Diawali oleh Karl Menger di Vienna dan Harvard. Setelah itu permasalahan TSP dipublikasikan oleh Hassler Whitney dan Merrill Flood di Princeton.

TSP adalah problem menemukan rute dengan lintasan terpendek pada sejumlah n-tempat dimana setiap tempat dikunjungi tepat satu kali sebelum kembali ke awal keberangkatan. Dengan demikian TSP merupakan problem menemukan sirkuit Hamilton. Model TSP didefinisikan oleh dua buah data, yaitu:

a. Jumlah tempat, katakan n
b. Jarak c_{ij} antara tempat i dan j.

($c_{ij} = \infty$ jika tempat i dan j tidak terhubung)
Jumlah maksimum rute pada n-tempat adalah $(n-1)!$

Model *TSP* didefinisikan dengan sejumlah n buah kota dan c_{ij} yang merupakan jarak antara kota i dan kota j, seseorang ingin membuat suatu lintasan tertutup dengan mengunjungi setiap kota satu kali. Tujuannya adalah memilih lintasan tertutup yang total jaraknya paling minimum diantara pilihan dari semua kemungkinan lintasan.

Model dari TSP adalah sebagai berikut **[6]**:

Fungsi Tujuan:

$$Minimum\ z = \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$c_{ij} = \infty \text{ untuk } i=j$$

Dengan batas :

$$\sum_{i=0} x_{ij} = 1\ ; i = 0,1,\dots,n$$

$$\sum_{j=0} x_{ij} = 1\ ; i = 0,1,\dots,n$$

$$x_{ij} = [0,1]$$

dengan :

n : jumlah kota/ lokasi /pelanggan yang akan dikunjungi (n tidak termasuktempat asal (base), yang diindekskan dengan i = 0).

c_{ij} : biaya/ jarak dari kota i ke kota j

A : sepasang arc / edge (i,j) yang ada. Note bahwa (i,j) yang dimaksud adalah arc yang ada dari node i ke node j.

Variabel:

$$x_{ij} = \begin{cases} 1,\ jika\ tempat\ j\ terhubung \\ \qquad dari\ tempat\ i \\ 0,\ lainnya \end{cases}$$

Metode Heuristik

Metode heuristik adalah subbidang dari kecerdasan buatan yang digunakan untuk melakukan pencarian dan optimasi. Metode heuristik berdasarkan strategi pencarian pintar pada pemecahan masalah dengan komputer, dengan menggunakan beberapa pendekatan **[**7**]**.

Dua tujuan dasar dalam pemecahan masalah optimisasi pada ilmu komputer adalah mencari algoritma yang cepat menyelesaikan masalah dan memperoleh hasil yang optimal. Metode heuristik adalah metode yang menghilangkan salah satu atau dua dari tujuan tersebut. Misalnya, pada pemecahan masalah optimisasi, dihasilkan solusi yag cukup optimal, tetapi secara manual, belum tentu solusi yang lebih optimal dapat diperoleh karena kompleksnya permasalahan yang ada. Atau, solusi yang didapat dihasilkan dengan waktu yang sangat cepat, namun secara manual masih dapat ditemukan hasil yang lebih optimal.

Jadi, hasil yang diperoleh belum tentu yang paling optimal. Tetapi penggunaan metode heuristik yang umum tetap diterapan di dunia nyata. Karena terdapat beberapa masalah, di mana hanya metode heuristik yang memungkinkan untuk memperoleh solusi yang optimal dalam waktu yang sangat singkat

Cheapest-Insertion Heuristics (CIH)

Algoritma CIH adalah Algoritma *Insertion* yang pada setiap penambahan kota baru yang akan disisipkan ke dalam *subtour* mempunyai bobot penyisipan paling minimal. Bobot penyisipan diperoleh dari persamaan $c(i,k,j) = c(i,k) + c(k,j) - c(i,j)$. Algoritma ini memberikan rute perjalanan yang berbeda tergantung dari urutan penyisipan kota-kota pada *subtour* yang bersangkutan.

Algoritma TSP CIH:

1. Inisialisasi, mulai dengan parsial tur dengan hanya satu kota i, dipilih secara acak.
2. Seleksi, Temukan kota k, i, dan j (i dan j titik ekstrem yang berada di parsial tur dan k tidak berada di tur tersebut) sehingga $c_{ik} + c_{kj} - c_{ij}$ adalah minimum.
3. Insersi, sisipkan k di antara i dan j.
4. Jika semua kota sudah disisipkan, maka kembali ke langkah 2.

3. Hasil dan Pembahasan

Metode *Branch and Bound* digunakan untuk menentukan rute terpendek yang harus ditempuh oleh seorang kurir ekspedisi untuk mengirim sejumlah paket ke lima kota. Jarak antara satu kota ke kota lainnya (dalam km) dapat dilihat pada tabel berikut:

Tabel 1. Jarak antar lokasi (km):

Lokasi	**1**	**2**	**3**	**4**	**5**
1	0	120	210	160	50
2	120	0	285	190	75
3	210	285	0	110	300
4	160	190	110	0	196
5	50	75	300	196	0

Dengan menggunakan metode *Branch and Bound,* rute yang dilalui oleh kurir ekspedisi untuk mengirimkan paket adalah kota 1 – kota 5 – kota 2 - kota 4 - kota 3 – kota 1 dengan panjang lintasan minimum adalah 643 km.

Penyelesaian TSP dengan Metode *Cheapest-Insertion Heuristics (CIH)*

Penyelesaian dengan mengggunakan metode CIH. Perjalanan dengan menggunakan metode CIH dimulai dari kota 1, dari kota 1 perjalanan terdekat adalah menuju kota 5 artinya perjalanan dari kota1 ke kota 5, sehingga bisa ditulis subtur (1,5)–(5,1). Kemudian (1,5) dapat disisip dan diganti dengan (1,2)-(2,5), (1,3)-(3,5) atau (1,4)-(4,5). Kemudian (5,1) dapat juga diganti dengan (5,2)-(2,1),(5,3)-(3,1) atau (5,4)-(4,1). Perhitungan yang digunakan untuk menentukan lintasan yang harus diganti oleh (1,5)-(5,1) dapat diberikan oleh tabel 2 berikut:

Tabel 2. Menentukan Arc (1,5)-(5,1) yang akan diganti

Arc yang akan diganti	Arc yang akan ditambahkan	Tambahan Jarak
(1,5)*	$(1,2) - (2,5)$	$c_{12} + c_{25} - c_{15} = 145$
(1,5)	$(1,3) - (3,5)$	$c_{13} + c_{35} - c_{15} = 460$
(1,5)	$(1,4) - (4,5)$	$c_{14} + c_{45} - c_{15} = 306$
(5,1)*	$(5,2) - (2,1)$	$c_{52} + c_{21} - c_{51} = 145$
(5,1)	$(5,3) - (3,1)$	$c_{53} + c_{31} - c_{51} = 460$
(5,1)	$(5,4) - (4,1)$	$c_{54} + c_{41} - c_{51} = 306$

*: menunjukkan pergantian yang benar

Tabel 2 memperlihatkan tambahan jarak terkecil diperoleh apabila arc (1,5) diganti dengan (1,2) dan (2,5) sehingga diperoleh subtur baru yaitu (1,2)-(2,5)-(5,1).

Tabel selanjutnya adalah tabel yang menyimpan kota yang bisa disisipkan dalam *subtour* (1,2)-(2,5)-(5,1) beserta tambahan jaraknya, seperti pada tabel 3 berikut:

Tabel 3. Menentukan Arc (1,2)-(2,5)-(5,1) yang akan diganti

Arc yang akan diganti	Arc yang akan ditambahkan	Tambahan Jarak
(1,2)	$(1,3) - (3,2)$	$c_{13} + c_{32} - c_{12} = 375$
(1,2)*	$(1,4) - (4,2)$	$c_{14} + c_{42} - c_{12} = 230$
(2,5)	$(2,3) - (3,5)$	$c_{23} + c_{35} - c_{25} = 510$
(2,5)	$(2,4) - (4,5)$	$c_{24} + c_{45} - c_{25} = 311$
(5,1)	$(5,3) - (3,1)$	$c_{53} + c_{31} - c_{51} = 460$
(5,1)	$(5,4) - (4,1)$	$c_{54} + c_{41} - c_{51} = 306$

*: menunjukkan pergantian yang benar

Tabel 3 memperlihatkan tambahan jarak terkecil diperoleh apabila pada arc(1,2) disisipkan dengan arc (1,4) dan arc (4,2). Subtur (1,2)-(2,5)-(5,1)dapat digantikan dengan *subtour* baru dari hasil penyisipan yaitu (1,4)-(4,2)-(2,5)-(5,1)

Tabel dilanjutkan dengan mencari lagi subtur baru yang menyimpan kota yang bisa disisipkan dalam *subtour* beserta tambahan jaraknya, seperti pada tabel 4 berikut:

Tabel 4. Menentukan Arc (1,4)-(4,2)-(2,5)-(5,1) yang akan diganti

Arc yang akan diganti	Arc yang akan ditambahkan	Tambahan Jarak
(1,4)*	$(1,3) - (3,4)$	$c_{13} + c_{34} - c_{14} = 160$
(4,2)	$(4,3) - (3,2)$	$c_{43} + c_{32} - c_{42} = 205$
(2,5)	$(2,3) - (3,5)$	$c_{23} + c_{35} - c_{25} = 510$
(5,1)	$(5,3) - (3,1)$	$c_{53} + c_{31} - c_{51} = 460$

Pada Tabel 4 terlihat bahwa tambahan jarak terkecil diperoleh apabila: Arc(1,4) diganti dengan arc (1,3) dan arc (3,4). Maka diperoleh *subtour* baru yaitu (1,3)-(3,4)-(4,2)-(2,5)-(5,1)

Karena semua kota sudah termasuk dalam *subtour* terakhir maka diperoleh tur minimum adalah (1,3)-(3,4)-(4,2)-(2,5)-(5,1) dengan jarak minimum sebesar:

$$c_{13} + c_{34} + c_{42} + c_{25} + c_{25} = 635\ km$$

Berdasarkan hasil perhitungan diperoleh panjang lintasan minimum adalah 643 km dengan rute adalah kota1 – kota 5 – kota 2 – kota 4-kota 3 – kembali ke kota 1.

4. Kesimpulan

Solusi yang diperoleh dengan menggunakan CIH merupakan suatu penyelesaian tur yang optimal. Solusi optimal ini sama dengan hasil yang diperoleh dengan menggunakan metode *Branch and Bounch*.

Daftar Pustaka

[1] Winston, W.L., 1994. *Operation Research Aplication and Algorithms*. Dexbury Press. California.

[2] Pop, Petrica Claudiu, *et al*. 2011. "Heuristic algorithms for solving the generalized vehicle routing problem." *International Journal of Computers Communications & Control* 6.1:158-165.

[3] Vitra, I., 2004. *Perbandingan metode-metode dalam algoritma genetika untuk Travelling Salesman Problem.* Proceedings Seminar Nasional Aplikasi. Teknologi Informasi

[4] Winston. W.L., 1995. *Introduction to Mathematical Programming Applications and Algorithms, 2nd Edition*, Belmout California.

[5] Kusrini dan Istiyanto, J.E., 2007, P*enyelesaian Travelling Salesman Problem dengan Algoritma Cheapest Insertion Heuristic dan Basis Data*, JURNAL INFORMATIKA, Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra, Vol. 8, No. 2, Nopember 2007, 109-114.

[6] Taha, H.A. 1996. *Riset Operasi. Jilid I*. Binarupa Aksara. Jakarta.

[7] Pearl, J, 1984, "*Heuristics: Intelligent Search Strategies for Computer Problem Solving*", Addison-Wesley Publishing Company. Canada.