MATEMATIKA, 1990, Jilid 6, no.1, hlm 49-67,

PENGOPTIMUMAN MASALAH PERJALANAN JURUJUAL

Omar bin Bakar dan Ismail bin Mohd.
Jabatan Matematik
Universiti Pertanian Malaysia
Serdang, Selangor, Malaysia

Abstrak

Di dalam makalah ini akan dibentangkan perbandingan antara 4 kaedah bagi menyelesaikan Masalah Perjalanan Jururjual. Kaedah berkenaan ialah,

1. Kaedah Pemasukkan Terjauh,
2. Kaedah Batas dan Ranting,
3. Kaedah Dua-optimum dan
4. Kaedah Tiga-optimum.

Satu kajian kes berkaitan dengan rangkaian secara amali juga dibentangkan bagi memperlihatkan keberkesanan keempat-empat kaedah di atas.

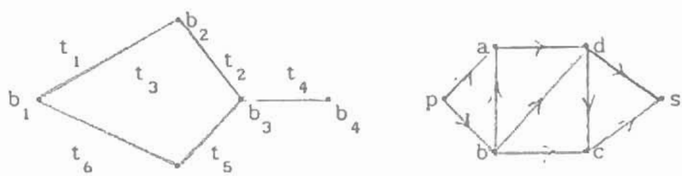
1. PENGENALAN

Masalah Perjalanan Jurujual merupakan suatu masalah klasik yang telah lama wujud. Masalah pertama ditimbulkan oleh seorang ahli matematik Irish bernama Sir Rowan Hamiltonian (1859), yang bertujuan mencari jarak yang terdekat atau masa yang tersingkat bagi sesuatu operasi atau kerja yang dilakukan. Sebenarnya setakad ini, belum ada lagi algoritma yang benar-benar berkesan bagi menyelesaikan Masalah Perjalanan Jurujual. Algoritma yang dikemukakan hanya mampu mencapai penyelesaian hampiran sahaja.

2. ASAS MATEMATIK

Takrif 2.1

$G = (B,T)$ ialah graf yang terdiri daripada set bucu $B = \{ b_1, b_2, \ldots, b_n \}$ dan set tepi $T = \{ t_1, t_2, \ldots, t_n \}$ (lihat Rajah 1(a)).



(a) Graf tak berarah (b) Graf berarah

Rajah 1

Takrif 2.2

Graf tak berarah ialah graf yang tepi-tepinya tidak ditandakan dengan anak panah (lihat Rajah 1(a)). ∎

Takrif 2.3

Graf Berarah ialah graf yang setiap tepinya ditandakan dengan arah anak panah (lihat Rajah 1(b)). ∎

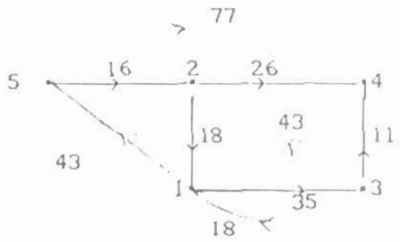
Dalam makalah ini rangkaian dirujukkan kepada graf berarah ataupun graf tak berarah. Nod dirujukkan kepada bucu dan bucu ini pula mewakili tempat atau keadaan-keadaan tertentu.

Nod sumber atau nod pendahulu dilambangakan dengan huruf S dan boleh mengambil salah satu dari nilai 1,2,...,n, dengan n bilangan nod dalam rangkaian.

Matriks pemberat bagi n nod dalam rangkaian ialah matriks $W = (w_{ij})_{n \times n}$ $(i,j = 1,2,\ldots,n)$ dengan w_{ij} $(i,j=1,\ldots,n)$ adalah pemberat tepi i,j dan memenuhi

$$w_{ij} = \begin{cases} 0 & (i = j) \\ \infty & (\text{nod } i \text{ dan nod } j \text{ tidak berhubungan}) \end{cases}$$

$(i,j=1,\ldots,n)$.



Rajah 2 : Rangkaian Berarah dengan 6 nod

Bagi Rajah 2, matriks pemberatnya *adalah*

$$(w_{ij})_{n \times n} = \begin{pmatrix} 0 & \infty & 35 & \infty & 43 \\ 18 & 0 & \infty & 26 & \infty \\ 18 & 43 & 0 & 11 & \infty \\ \infty & \infty & \infty & 0 & \infty \\ \infty & 16 & \infty & 77 & 0 \end{pmatrix}$$

Algoritma kaedah yang dibicarakan dalam makalah ini ditulis dalam bentuk tatatanda kod pseudo seperti yang dimaksudkan dalam (Mohd, 1989).

3. MASALAH JURUJUAL

Syarat mengenai Masalah Perjalanan Jurujual adalah ringkas, iaitu jurujual berkenaan dikehendaki melawat suatu rangkaian n bandar dan memastikan dirinya melawat setiap bandar hanya sekali dan kembali ke bandar asalnya. Dengan kata lain jurujual berkenaan dikehendaki melawat suatu rangkaian n nod dan melengkapkan satu Kitaran Hamiltonian dengan jumlah kos terkecil yang mungkin.

Oleh sebab itu kita mestilah berusaha mendapatkan suatu kaedah penyelesaian bagi mempastikan jumlah kos adalah minimum. Dalam bahagian-bahagian berikut kita akan mengemukakan beberapa kaedah penyelesaian untuk tujuan ini.

4. KAEDAH PEMASUKAN TERJAUH

Proses penyelesaian Kaedah Pemasukkan Terjauh ini dilakukan secara lelaran dan boleh dirumuskan dalam bentuk langkah seperti yang berikut.

Langkah 1

Tentukan nod permulaan S

Tentukan (Jarak) = unsur-unsur baris S bagi matriks **W**

Langkah 2

Pilih nod f iaitu $f \in (V - V_T)$ dengan

V = set semua nod dalam rangkaian $\{1,2,\ldots,n\}$

V_T = set nod dalam kitaran semasa dan f ialah nod terjauh dari V_T

Langkah 3

Periksa kos pemasukan f dari nod i ke nod j iaitu

$$C_{ij} = w_{if} + w_{fj} - w_{ij}, \qquad (i,j \in V_T,\ f \in V)$$

Pilih satu kos pemasukan terkecil, katakan C_{th} iaitu $C_{th} = \min \{C_{ij}\}$.

Langkah 4

Kemaskinikan set kitaran semasa V_T iaitu dengan memasukkan nod f di antara nod t dan nod h.

Langkah 5

Tentukan jumlah kos iaitu,

Jumlah kos = hasiltambah kos pemasukan bagi setiap nod dalam kitaran V_T

Langkah 6

Jika set V_T = set V maka kita telah melengkapkan satu Kitaran Hamiltonian dengan penyelesaiannya sebagai laluan = { V_T} dan jumlah kos = jumlah kos semasa. Jika tidak, kemaskinikan set (jarak) iaitu nilai minimum bagi baris S dan f bagi matriks pemberat W, dan kembali ke Langkah 2.

Algoritma Pemasukkan Terjauh.

Data : $n \in N$, $W = (w_{ij})_{nxn}$ dengan $w_{ij} > 0$ $(i,j=1,\ldots,n)$ dan S=1 sebagai nod punca.

1. V_T := { S } ! V_T set kitaran semasa
2. E_T := { (S,S) }
3. kos := 0 ! Nilai awal kos semasa = 0
4. $J_u := w_{su}$ $(u \in V - V_T)$! J = jarak nod u dari nod s.
5. *selagi* $|V_T|$ < n *buat* ! $|V_T|$ ialah bilangan kardinal set V_T

   5.1. f := nod dalam $(V-V_T)$ dengan nilai terbesar bagi J_f

   5.2. *untuk* setiap tepi $(i,j) \in E_T$ *buat*

      5.2.1. $C_{ij} = w_{if} + w_{fj} - w_{ij}$! C_{ij} ialah kos pemasukan ! nod f

      5.2.2. (t,h) := tepi E_T dengan nilai terkecil C_{th}.

      5.2.3. $E_T := E_T \cup \{(t,f),(f,h)\} - \{(t,h)\}$

      5.2.4. $V_T := V_T \cup f$! mengemaskinikan set kitaran ! semasa

      5.2.5. kos := kos + C_{th}

      5.2.6. $J_x := \text{minim}\ \{J_x, w_{fx}\}$ $(x \in (V - V_T))$

6. *kembali.* ∎

Contoh 4.1

Misalkan matriks pemberat berbentuk

$$(w_{ij})_{n\times n} = \begin{pmatrix} 0 & 3 & 93 & 13 & 33 & 9 \\ 4 & 0 & 77 & 42 & 21 & 16 \\ 45 & 17 & 0 & 36 & 16 & 28 \\ 39 & 90 & 80 & 0 & 56 & 7 \\ 28 & 46 & 88 & 33 & 0 & 25 \\ 3 & 88 & 18 & 46 & 92 & 0 \end{pmatrix}$$

Pilih S = 1, maka

Jarak = (-,3,93,13,33,9)

yang merupakan baris 1 bagi matriks pemberat w.

Lelaran 1

Nod terjauh ialah 3 dengan nilai 93. Kos pemasukan nod 3 ialah

$$C_{11} = w_{13} + w_{31} - w_{11} = 93 + 45 - 0 = 138$$

Maka, kos semasa = 138 dan kitaran semasa (1 3 1). Jarak = (-,3,-,13,16,9).

Lelaran 2

Nod terjauh ialah nod 5 dengan nilai 16.

Kos pemasukannya ialah

$$C_{13} = w_{15} + w_{53} - w_{13} = 33 + 88 - 93 = 28$$

$$C_{31} = w_{35} + w_{51} - w_{31} = 16 + 28 - 46 = -1$$

Kos pemasukkan terkecil ialah C_{31} = -1, maka

kos semasa = 138 + (-1) = 137

Masukkan nod 5 di antara nod 3 dan nod 1, maka

kitaran semasa = (1 → 3 → 5 → 1).

Jarak = (-, 3, -, 13, -, 9).

Lelaran 3

Nod terjauh ialah nod 4 dengan nilai 13

Kos pemasukan nod 4 ialah,

$$C_{13} = w_{14} + w_{43} - w_{13} = 13 + 80 - 93 = 0$$

$$C_{35} = w_{34} + w_{45} - w_{35} = 36 + 56 - 16 = 76$$

$$C_{51} = w_{54} + w_{41} - w_{51} = 33 + 39 - 28 = 44$$

Kos pemasukan terkecil ialah $C_{13} = 0$, maka

kos semasa = 137 + 0 = 137.

Masukkan nod 4 antara nod 1 dan nod 3, maka

kitaran semasa = $(1 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 1)$.

Jarak = (-, 3, -, -, -, 7).

Lelaran 4

Nod terjauh ialah 6 dengan nilainya 7.

Kos pemasukan nod 6 ialah,

$$C_{14} = w_{16} + w_{64} - w_{14} = 9 + 46 - 13 = 42$$

$$C_{43} = w_{46} + w_{63} - w_{43} = 7 + 18 - 80 = -55$$

$$C_{51} = w_{36} + w_{65} - w_{35} = 28 + 92 - 16 = 104$$

$$C_{51} = w_{56} + w_{61} - w_{51} = 25 + 3 - 28 = 0$$

Kos pemasukan terkecil ialah $C_{43} = -55$, maka

kos semasa = 137 + (-55) = 82.

Masukkan nod 6 di antara nod 4 dan nod 3, maka

$$\text{kitaran semasa} = (1 \to 4 \to 6 \to 3 \to 5 \to 1).$$

Jarak = (-, 3, -, -, -, -).

Lelaran 5

Nod yang dipilih ialh nod 2.

Kos pemasukannya ialah,

$$C_{14} = w_{12} + w_{24} - w_{14} = 3 + 42 - 13 = 32$$

$$C_{46} = w_{42} + w_{26} - w_{46} = 90 + 16 - 7 = 99$$

$$C_{63} = w_{62} + w_{23} - w_{63} = 88 + 77 - 18 = 147$$

$$C_{35} = w_{32} + w_{25} - w_{35} = 17 + 21 - 16 = 22$$

$$C_{51} = w_{52} + w_{21} - w_{51} = 46 + 4 - 28 = 22$$

Kos pemasukan terkecil ialah $C_{35} = C_{51} = 22$. Pilih salah satu, katakan kita pilih C_{35}, maka

$$\text{jumlah kos} = 82 + 22 = 104.$$

Masukkan nod 2 di antara nod 3 dan nod 5, maka diperoleh kitaran optimum

$$\text{kitaran} = (1 \to 4 \to 6 \to 3 \to 2 \to 5 \to 1).$$

5. Kaedah Batas dan Ranting

Penyelesaian kaedah Batas dan Ranting, berasaskan kepada penurunan Matriks Pemberat W. Nilai yang tersusut merupakan jumlah kos semasa. Kemudian satu nod dipilih untuk mendapatkan kitaran semasa. Seterusnya turunkan satu matra bagi matriks pemberat W.

Ulang proses di atas sehinggalah kita melengkapkan satu Kitaran Hamiltonian.

Algoritma Batas dan Ranting memerlukan beberapa subaturcara seperti yang berikut.

penyusutan(p;A:r)

! $A = (a_{ij})_{p \times p}$

! r ialah nilai penyusutan matriks A.

1. r := 0.0 ! Nilai awal penyusutan matriks
2. *untuk* i := 1 *ke* p *buat*
   2.1. $b_t := \underset{1 \le j \le p}{\text{minim}} \{a_{ij}\}$
   2.2. *jika* $b_t > 0$ *buat*
      2.2.1. $a_{ij} := a_{ij} - b_t$ (j:=1,...,p)
      2.2.2. $r := r + b_t$
3. *untuk* j := 1 *ke* p *buat*
   2.1. $c_s := \underset{1 \le j \le p}{\text{minim}} \{a_{ij}\}$
   2.2. *jika* $c_s > 0$ *buat*
      2.2.1. $a_{ij} := a_{ij} - c_s$ (i:=1,...,p)
      2.2.2. $r := r + b_s$
4. *kembali* ■

tepiterbaik(A,p,inf:semua,t,s)

! A, p lihat subaturcara penyusutan

! t indeks baris dan s indeks lajur dengan semua ialah nilai

! tepi terbaik yang sepadan dengan unsur a_{ts}.

! inf integer yang melebihi nilai pemberat terbesar yang

! merupakan unsur matriks A.

1. semua := - inf
2. *untuk* i := 1 *ke* p *buat*
   2.1. *untuk* j := 1 *ke* p *buat*
      2.1.1. *jika* $a_{ij} = 0$ *buat*

2.1.1.1. $b_t := \underset{1 \le i \le p}{\text{minim}} \{a_{ij}\}$ $(b_t \neq 0)$

2.1.1.2. $c_s := \underset{1 \le i \le p}{\text{minim}} \{a_{ij}\}$ $(c_s \neq 0)$

2.1.1.3. jumlah := $b_t + c_s$

2.1.1.4. *jika* jumlah > semua *buat*

2.1.1.4.1. semua := jumlah

2.1.1.4.2. t := i

2.1.1.4.3. s := j

3. *kembali*. ■

jelajah(ntepi, n; kos, jumberat, A)

! ntepi menandakan bilangan tepi yang termasuk dalam

! penyelesaian separa.

! p, A seperti dalam subaturcara penyusutan

! kos ialah biaya penyelesaian separa

! jumberat ialah jumlah pemberat penyelesaian separa semasa.

1. kos := kos + *penyusutan*(n;A:r)

2. *jika* kos < jumberat *buat*

2.1. *jika* ntepi = n-2

maka

2.1.1. tambahkan dua tepi terakhir kepada A

2.1.2. jumberat := kos

2.1.3. rekodkan penyelesaian baru

selainnya

2.1.4. *tepiterbaik*(A,n,inf:semua,t,s)

2.1.5. baba := kos + semua

2.1.6. Abaru := A - lajur s - baris t ! buang baris t
! dan lajur s

2.1.7. *jelajah*(ntepi+1,n;kos,jumberat,Abaru)

2.1.8. A := Abaru + lajur s + baris t ! tambah baris t
! dan lajur s

2.1.9. *jika* baba < jumberat *buat*

2.1.9.1. a_{ts} := inf

2.1.9.2. *jelajah*(ntepi,n;kos,jumberat,A)

2.1.9.3. a_{ts} := 0

3. *kembali*. ■

ALGORITMA BATAS DAN RANTING

Data input :

$W = (w_{ij})_{nxn}$ matriks pemberat

n, inf $\in I^{+}$ dengan inf $> \max_{i,j=1,\dots,n} (w_{ij})$

Data output :

B jumlah pemberat optimum.

1. B := inf
2. *jelajah*(0,n;0,B,W)
3. *berhenti.* ■

6. Kaedah Dua-Optimum

Bagi kaedah ini penyelesaian berasaskan kepada Kitaran Hamiltonian awal iaitu H = {1,2,...,n} dengan n ialah bilangan nod dalam rangkaian.

Pilih dua tepi, katakan x_i, x_j dan gantikan dengan tepi yang baru, katakan y_p, y_q. Ujikan pengurangan nilai kos iaitu

$$\delta = [w(x_i) + w(x_j)] - [w(y_p) + w(y_q)]$$

Jika $\delta \leq 0$ maka kita telah mendapat penyelesaiannya. Jika tidak ulang proses di atas.

ALGORITMA DUA-OPTIMUM

Data

n = bilangan nod dalam rangkaian

$W = (w_{ij})_{nxn}$, $\forall$ i,j = 1,2,...,n

1. H := { $x_1, x_2, x_3, \dots, x_n$ } ! perjalan semasa
2. *ulang*

   2.1. ∂_{maks} := 0 ! umpukkan nilai awal $\partial_{maks} = 0$

   2.2. *untuk* i := 1 *ke* n-2 *buat*

      2.2.1. *untuk* j := (i+2) ke n *atau* (n-1) *buat*

         2.2.1.1. *jika* {$w(x_i) + w(x_j)$} -

$\{w(y_p) + w(y_q)\} > \partial_{maks}$ *buat*

2.2.1.1.1. $\partial_{maks} = \{w(x_i) + w(x_j)\} - \{w(y_p) + w(y_q)\}$

2.2.1.1.2. rekod nilai i dan j

2.3. *jika* $\partial_{maks} > 0$ *buat*

2.3.1. $H := H - (x_i, x_j) \cup (y_p, y_q)$

sehingga

2.4. $\partial_{maks} = 0$

3. *kembali.* ■

7. Kaedah Tiga-Optimum

Proses penyelesaian Kaedah Tiga-optimum sama seperti Dua-optimum. Tapi di sini ia melakukan pertukaran 3 tepi. Ujian pengurangan kos dilakukan dengan

$$\delta = [w(x_i) + w(x_j) + w(x_u)] - [w(y_p) + w(y_q) + w(y_r)]$$

Jika $\delta < 0$ maka penyelesaian Tiga-optimum di capai, jika tidak ulang proses di atas.

Syarat : Bagi Kaedah 2-optimum dan Tiga-optimum matriks pemberat mestilah simetri iaitu $w_{ij} = w_{ji}$ $(i,j = 1,2,\ldots,n)$.

8. Perbandingan Kaedah

Kita kemukakan beberapa contoh penyelesaian dalam rangka penentuan kaedah yang terbaik.

Contoh 1

Seorang pengeluar ais krim mengeluarkan 6 jenis perasa ais krim pada setiap hari dengan menggunakan satu mesin. Ia memerlukan masa penyediaan mesin (katakan w_{ij}), termasuk masa membasuh apabila jenis pengeluaran bertukar dari perasa i ke perasa j. Dari pengalamannya, masa penyediaan mesin untuk setiap pertukaran perasa diketahui. Tujuan pengeluar tersebut ialah untuk meminimumkan masa penyediaan mesin supaya dapat meningkatkan kuantiti pengeluaran. Masa penyediaan mesin dirumuskan dalam bentuk matriks W (unit masa dalam minit).

$$W_{6\times6} = \begin{bmatrix} 0 & 7 & 20 & 21 & 12 & 23 \\ 27 & 0 & 13 & 16 & 46 & 5 \\ 53 & 15 & 0 & 25 & 27 & 6 \\ 16 & 2 & 35 & 0 & 47 & 10 \\ 31 & 29 & 5 & 18 & 0 & 4 \\ 28 & 24 & 1 & 17 & 5 & 0 \end{bmatrix}$$

Keputusan

i. Dengan menggunakn Kaedah Pemasukkan Terjauh

laluan = (1 5 3 6 4 2 1)

Jumlah kos = 69

ii. Dengan menggunakan Kaedah Batas dan Ranting

laluan = (1 2 6 5 3 4 1)

Jumlah kos = 63

iii. Dengan menggunakan Kaedah Dua-optimum dan kaedah tiga-optimum tidak boleh diselesaikan.

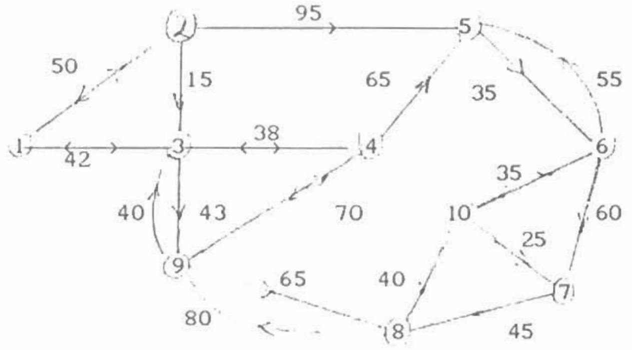
Contoh 2

Dalam contoh ini saya mengukur masa perjalanan dengan motorsikal di antara 10 lokasi (nod) tertentu di UPM. Masa dicatit dalam saat. Nod berkenaan ialah,

label nod	lokasi
1.	Pintu masuk 1 (utama)
2.	Pintu masuk 2
3.	Bangunan Pentadbiran
4.	Simpang 3 DKD
5.	Pusat Komputer
6.	Simpang 4 Biokimia
7.	Simpang 4 Sains Tanah

8.	Simpang 3 Fakulti Veterinar
9.	Dewan Besar
10.	Perpustakaan

Diperoleh rangkaian Berarah dengan 10 nod seperti dalam Rajah 3.



Rajah 3

Hendaklah diingat bahawa

i. Nombor dalam bulatan menunjukkan label nod.

ii. Nombor pada garisan menandakan masa perjalanan dalam saat.

$$(W)_{10\times 10} = \begin{bmatrix} 0 & 50 & 42 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ 50 & 0 & 15 & \infty & 95 & \infty & \infty & \infty & \infty & \infty \\ 42 & 15 & 0 & 38 & \infty & \infty & \infty & \infty & 43 & \infty \\ 75 & \infty & 38 & 0 & 65 & \infty & \infty & \infty & 70 & \infty \\ \infty & \infty & \infty & 240 & 0 & 35 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 55 & 0 & 60 & \infty & \infty & 35 \\ \infty & \infty & \infty & \infty & \infty & 55 & 0 & 45 & \infty & 25 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 80 & 40 \\ \infty & \infty & 40 & 70 & \infty & \infty & \infty & 65 & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & 35 & 25 & 40 & \infty & 0 \end{bmatrix}$$

Keputusan

i Dengan menggunakan Kaedah Pemasukan Terjauh

laluan = (1 2 3 9 8 10 7 6 5 4 1)

Jumlah kos = 663

ii. Dengan menggunakan Kaedah Batas dan Ranting

laluan = (1 2 5 6 7 10 8 9 4 3)

Jumlah kos = 535

iii. Kaedah Dua-optimum dan kaedah Tiga-Optimum tidak boleh digunakan

Contoh 3

Pertimbangkan matriks pemberat W berikut dan dapatkan penyelesaian bagi Masalah Perjalanan Jurujual.

$$W_{5x5} = \begin{pmatrix} 0 & 38 & 31 & 44 & 8 \\ 45 & 0 & 15 & 48 & 38 \\ 43 & 32 & 0 & 26 & 47 \\ 38 & 19 & 14 & 0 & 44 \\ 4 & 25 & 33 & 15 & 0 \end{pmatrix}$$

Keputusan

i. Dengan menggunakan Kaedah Pemasukkan Terjauh,

laluan = (1 3 4 2 5 1)

Jumlah kos = 118

ii. Dengan menggunakan Kaedah Batas dan Ranting,

laluan = (1 5 4 2 3 1)

Jumlah kos = 100

iii. Kaedah Dua-optimum dan kedah Tiga-Optimum tidak boleh digunakan.

Contoh 4

Pertimbangkan matriks pemberat W di bawah dan selesaikan untuk mendapat Kitaran Hamiltonian.penyelesaian bagi Masalah

$$(W)_{6x6} = \begin{pmatrix} 0 & 3 & 93 & 13 & 33 & 9 \\ 4 & 0 & 77 & 42 & 21 & 16 \\ 45 & 17 & 0 & 36 & 16 & 28 \\ 39 & 90 & 80 & 0 & 56 & 7 \\ 28 & 46 & 88 & 33 & 0 & 25 \\ 3 & 88 & 18 & 46 & 92 & 0 \end{pmatrix}$$

Keputusan

i. Dengan menggunakan Kaedah Pemasukkan Terjauh, laluan = (1 4 6 3 2 5 1)

Jumlah kos = 104

ii. Dengan menggunakan Kaedah Batas dan Ranting,

laluan = (1 4 6 3 5 2 1)

Jumlah kos = 104

iii. Kaedah Dua-optimum dan kaedah Tiga-Optimum tidak boleh digunakan.

Contoh 5

Selesaikan Masalah Perjalanan Jurujual dengan matriks pemerat W seperti di bawah,

$$(W)_{10x10} = \begin{bmatrix} 0 & 10 & 55 & 78 & 93 & 79 & 48 & 49 & 76 & 11 \\ 10 & 0 & 77 & 46 & 64 & 86 & 37 & 8 & 29 & 35 \\ 55 & 77 & 0 & 79 & 9 & 82 & 13 & 54 & 80 & 84 \\ 78 & 46 & 79 & 0 & 34 & 29 & 9 & 35 & 98 & 91 \\ 93 & 64 & 9 & 34 & 0 & 74 & 83 & 44 & 67 & 68 \\ 79 & 86 & 82 & 29 & 74 & 0 & 42 & 98 & 85 & 8 \\ 48 & 37 & 13 & 9 & 83 & 42 & 0 & 84 & 5 & 2 \\ 49 & 8 & 54 & 35 & 44 & 98 & 84 & 0 & 98 & 28 \\ 76 & 29 & 80 & 98 & 67 & 85 & 5 & 98 & 0 & 48 \\ 11 & 35 & 84 & 91 & 68 & 8 & 2 & 28 & 48 & 0 \end{bmatrix}$$

Keputusan

i Kaedah Pemasukkan Terjauh

laluan = (1 10 8 4 6 7 3 5 9 2 1)

Jumlah kos = 273

ii. Kaedah Batas dan Ranting

laluan = (1 2 9 7 3 5 8 4 6 10 1)

Jumlah kos = 193

iii. Kaedah Dua-optimum

laluan = (1 3 5 4 8 2 9 7 6 10 1)

Jumlah kos = 236

iv. Kaedah Tiga-optimum

laluan = (1 10 6 4 8 5 3 7 9 2)

Jumlah kos = 193

Contoh 6

Matriks pemberat W diberikan sebagai

$$(W)_{nxn} = \begin{pmatrix} 0 & 42 & 31 & 17 & 7 & 63 \\ 42 & 0 & 24 & 51 & 10 & 15 \\ 31 & 24 & 0 & 5 & 6 & 11 \\ 17 & 51 & 5 & 0 & 8 & 22 \\ 7 & 10 & 6 & 8 & 0 & 3 \\ 63 & 15 & 11 & 22 & 3 & 0 \end{pmatrix}$$

Keputusan

i. Dengan menggunakan Kaedah Pemasukkan Terjauh,

laluan = (1 4 3 6 2 5 1)

Jumlah kos = 65

ii. Dengan menggunakan Kaedah Batas dan Ranting,

laluan = (1 5 2 6 3 4 1)

Jumlah kos = 65

iii. Dengan menggunakan Kaedah Dua-optimum

laluan = (1 4 3 6 2 5 1)

Jumlah kos = 65

iv. Dengan menggnauakan Kaedah Tiga-optimum

laluan = (1 4 3 6 2 5 1)

Jumlah kos = 65

Contoh 7

Matriks pemberat W diberikan sebagai,

$$(W)_{nxn} = \begin{pmatrix} 0 & 52 & 35 & \infty & 43 & \infty \\ 19 & 0 & \infty & 85 & \infty & 13 \\ 18 & 43 & 0 & 11 & \infty & \infty \\ \infty & \infty & 18 & 0 & 21 & 10 \\ \infty & 16 & \infty & 77 & 0 & 32 \\ 28 & 13 & \infty & \infty & 62 & 0 \end{pmatrix}$$

Keputusan

i. Menggunakan Kaedah Pemasukkan Terjauh,

laluan = (1 5 6 2 4 3 1)

Jumlah kos = 200

ii. Menggunakan Kaedah Batas dan Ranting, laluan = (1 3 4 5 6 2)

Jumlah kos = 122

iii. Kaedah Dua-optimum dan Kaedah Tiga-Optimum tidak boleh digunakan.

9. KESIMPULAN

Daripada contoh 1, Kaedah Batas dan Ranting memberikan penyelesaian yang baik dengan jumlah kos = 63. Dalam contoh 2 kaedah Batas dan ranting memberikan jumlah kos = 535. Dalam contoh 4, Kaedah Batas dan Ranting dan Pemasukkan Terjauh memberikan jumlah kos yang sama iaitu 104 tetapi laluan yang berlainan. Dalam contoh 5, Kaedah batas dan ranting dan kaedah Tiga-optimum memberikan jumlah kos yang sama iaitu 193 tetapi laluan yang berlainan. Dalam contoh 6, kesemuanya memberikan jawapan yang sama iaitu dengan jumlah kos = 65.

Sebagai kesimpulan didapati bahawa Kaedah Batas dan Ranting adalah yang terbaik jika dibandingkan dengan 3 kaedah yang lain, walupun pada keadaan-keadaan tertentu kaedah yang lain mampu memberikan jawapan yang sama.

10. PENGHARGAAN

Ucapan terima kasih kepada Universiti Pertanian Malaysia di atas biaya penyelidikan.

RUJUKAN

Bellmore, M. and Nemhauser, G.L. (1968) : "The Traveling Salesman Problems: A Survey", *Operations Research*, Vol. 16, m.s. 538-338.

Croes, A. (1958) : A Method for Solving Traveling-Salesman Problems, Oper. res., 5, 791 - 812.

Lin, S. and Kernigham, B.W. (1973) : "An Effective Hueristic Algorithm for the Traveling Salesman problem", *Operations Research*, Vol. 21, m.s. 598-516.

Micheal Held and Richard M. Karp (1970) : *"The Traveling Salesman Problems and Minimum Spanning Trees"*, *Operations Research*, Vol. 18, m.s. 1138-1162.

Mohd, I. B. (1989) : Pengenalan Kaedah Berangka Dalam Pengoptimuman, Dewan Bahasa dan Pustaka, Kuala Lumpur.

Robert R. Korfhage, *Discrete Computational Structures*, New York: Southern Methodist University.

Swamy, M.N.S., K. Thulsiraman (1981) : *Graphs, Networks and Algorithms*, John Wiley, New York.