

PEMANFAATAN METODE MONTE CARLO DALAM PENCARIAN PATH TERPENDEK PADA GRAF

Said Iskandar Al Idrus

Jurusan Matematika, FMIPA, Universitas Negeri Medan

Email : Said.iskandar.alidrus@gmail.com

ABSTRAK

Pada saat ini ada beberapa cara yang dilakukan untuk mencari path terpendek pada graf. Untuk graf dengan jumlah vertek yang besar, secara konvensional pencarian ini akan menghabiskan waktu yang lama dan keakuratan yang kecil. Dengan bantuan komputer kita dapat mengembangkan banyak algoritma memudahkan kita mencari optimasi dari sebuah graf. Dengan komputasi menggunakan metode monte carlo kita dapat mendistribusikan nilai random untuk dapat memunculkan semua kemungkinan yang terjadi dari path ini. Nilai path akan diseleksi dari generasi ke generasi berdasarkan nilai terkecil dari jumlah edge, waktu yang diperlukan bergantung dari jumlah vertek dan epoch dari sebuah program.

Kata kunci: *komplit graf, path terpendek, tree, monte carlo*

ABSTRAK

At this time there are several ways in which to look for the shortest path in a graph. For a graph with a large number of vertex, conventionally this search will be spending a long time and the accuracy is small. With the help of computer, we can develop a lot of algorithms allows us to search for the optimization of a graph. By computation using monte carlo method we can distribute a random value to be able to bring all of the possibilities arising from this path. Value path will be selected from generation to generation by the smallest value of the number of edges, the time needed depends on the number of vertex and the epoch of a program.

Keywords: complete graph, shortest path, tree, monte carlo

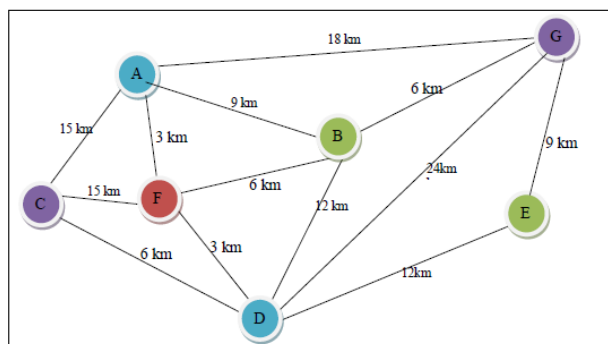
PENDAHULUAN

Dalam kehidupan, sering dilakukan perjalanan dari satu tempat atau kota ke tempat yang lain dengan mempertimbangkan efisiensi, waktu dan biaya sehingga diperlukan ketepatan dalam menentukan jalur terpendek antar suatu kota. Secara matematis kondisi seperti ini dapat direpresentasikan sebagai sebuah graf. Kota-

kota yang akan dikunjungi diasumsikan sebagai sebuah titik (vertek) dan titik ini dihubungkan oleh sebuah jalan atau disebut sebuah edge. Maka satu buah edge akan menghubungkan sebuah dua buah kota atau titik. Semakin banyak titik yang tercipta maka akan semakin penghubung dan edge yang dibutuhkan. Jarak antar satu kota dengan kota yang lain mungkin saja sama

tapi juga tidak tertutup kemungkinan jarak antar satu kota dengan kota yang lainnya berbeda. Jika kita akan melalui semua kota dalam satu kali jalan maka secara optimal kita akan mencari lintasan terpendek dari panjang lintasan tersebut. Masalah ini pada umumnya sering disebut dengan TSP

(Traveling Salesman Problem) namun kasus yang hampir sama terjadi pada beberapa aplikasi lainnya seperti, jaringan komputer, pipa air, pipa minyak, rangkaian komponen pada board elektrik, jaringan telepon, dan lain-lain.



Gambar 1. Kota-kota yang terhubung dengan edge.

Gambar 1 menunjukkan salah satu contoh graf. Kota-kota di representasi menjadi titik-titik dan jalan penghubung antar dua kota di berikan sebuah garis. Dalam graf, walk adalah barisan titik dan garis yang saling berselang seling. Jika pada walk tidak terjadi perulangan titik maka walk tersebut disebut dengan path. Path terpendek dapat ditentukan secara konvensional dengan menghitung secara berurutan dan mencoba dengan berbagai edge yang ada di gambar. Namun semakin banyak verteks yang terjadi semakin rumit penghitungannya dan semakin tidak akurat karena bisa saja bukan path terpendek yang kita hasilkan dan kemungkinan untuk satu edge terhitung dua kali akan semakin besar. Untuk itu dibutuhkan sebuah algoritma dan metode yang untuk dapat mencari secara akurat path terpendek ini, salah satu metode yang dapat digunakan adalah dengan menggunakan monte carlo yang selanjutnya akan diimplementasikan dalam program komputer. Dengan algoritma dan kecepatan kompembangkit acak yang ada di monte

carlo kita dapat menampilkan hampir keseluruhan path yang ada dalam sebuah graf.

Secara konvensional perhitungan path terpendek akan menghabiskan waktu yang lama dan memiliki keakuratan yang rendah baik dari hasil path terpendek maupun dari perulangan edge yang dihitung, kemungkinan edge akan terhitung dua kali sangat besar. Metode monte carlo diharapkan nantinya untuk memperoleh hasil yang akurat dengan membangkitkan nilai acak untuk edge yang akan memungkinkan semua path akan muncul untuk memperoleh path terpendek.

Penulisan akan dibatasi pada jenis metode monte carlo dan graf yang akan di jelaskan adalah komplit graf, didalam jenis graf ini satu verteks berhubungan dengan semua verteks yang lain, tidak memiliki loop dan paralel edge. Nilai random acak akan dihubungkan dengan histori yang digambarkan dengan tree dan matrik ketetanggaan untuk melihat nilai bobot. Penelitian ini bertujuan menyelesaikan

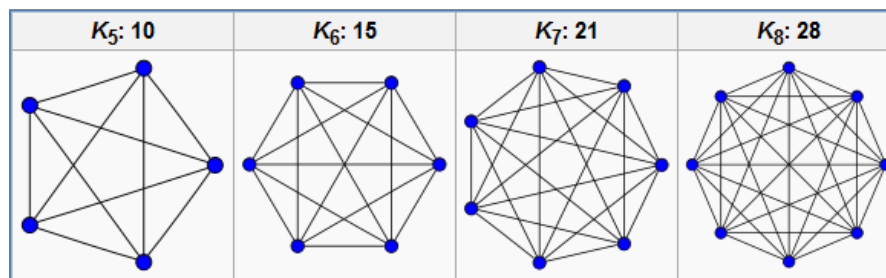
masalah optimasi menggunakan metode monte carlo pada komplit graf dan mencoba mensimulasikannya dalam sebuah program komputer. Manfaat yang dapat diambil dari penelitian adalah: Menawarkan penyelesaian yang lebih mudah dalam perhitungan path terpendek, dapat dikembangkan dalam bentuk software untuk banyak aplikasi.

METODE PENELITIAN

Jenis penelitian yang dilakukan adalah studi literatur dan komputasi numerik. Penulis melakukan kajian kepustakaan tentang hal yang akan dibahas. Selanjutnya penulis membuat algoritma komputasi numerik metode monte carlo untuk mencari dan menghitung path terpendek dari graf komplit.

HASIL DAN PEMBAHASAN

Komplit graf



Gambar 2. Komplit graf.

Jumlah edge yang ada mengikuti dari jumlah verteks, dari Gambar 2 dapat dilihat bahwa jika $v=5$ maka $e=10$, $v=6$ maka $e=15$ dan dinotasikan dengan $G(5,10)$, $G(6,15)$ dan seterusnya. Jumlah edge pada komplit graf dapat di formulasi dengan

$$e = n \frac{(n - 1)}{2}$$

Dimana e =banyaknya edge dan n =banyaknya verteks.

Jumlah edge ini akan menjadi jumlah indikasi pembangkit nilai random yang akan dibangkitkan dalam metode monte carlo

Graf adalah kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi/busur (*edges*). Suatu Graf G terdiri dari dua himpunan yaitu himpunan V dan himpunan E .

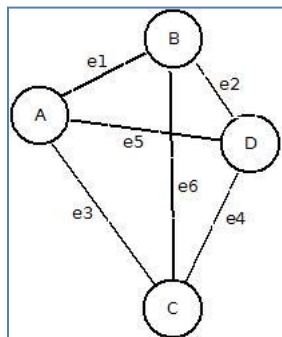
- Verteks (simpul) : V = himpunan simpul yang terbatas dan tidak kosong
- Edge (sisi/busur): E = himpunan busur yang menghubungkan sepasang simpul.

Simpul-simpul pada graf dapat merupakan obyek sembarang seperti kota, atom-atom suatu zat, nama anak, jenis buah, komponen alat elektronik dan sebagainya. Busur dapat menunjukkan hubungan (relasi) sembarang seperti rute penerbangan, jalan raya, sambungan telepon, ikatan kimia, dan lain-lain. Notasi graf: $G(V,E)$ artinya graf G memiliki V simpul dan E busur. Pada komplit graf semua verteks terhubung dengan verteks yang lain.

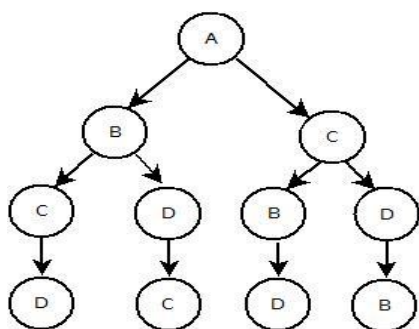
dalam satu generasi random.

Tree

Tree pada graf sangat penting untuk merekam histori dari tiap titik yang telah dikunjungi. Setiap titik yang telah terekam akan dihilangkan dalam random selanjutnya dalam satu generasi penjumlahan edge. Misalkan kita ambil contoh dengan komplit graf empat verteks



Gambar 3. Komplit graf 4 verteks.



Gambar 4. Tree pada graf empat verteks.

Tree yang terbentuk adalah diagram alir perjalanan satu titik ke titik yang lain. Dengan bantuan tree perhitungan sebuah edge tidak akan pernah berulang, ini akan membuat perhitungan menjadi akurat.

Bobot

Dari graf $G = (V(G), E(G))$ di atas diperoleh, jika $A=v1, B=v2, C=v3, D=v4$ maka

$$V(G) = \{v1, v2, v3, v4\},$$

$$E(G) = \{e1, e2, e3, e4, e5, e6\}$$

$$= \{\{v1,v2\}, \{v2,v4\}, \{v1,v3\}, \{v4,v3\}, \{v1,v4\}, \{v2,v3\}\}.$$

Graf G dikatakan graf berbobot (*weighted graph*) jika setiap sisinya mempunyai sebuah nilai real $w(e)$ yang

disebut sebagai bobot.

Monte Carlo.

Dasar dari simulasi Monte Carlo adalah percobaan elemen kemungkinan dengan menggunakan sampel random (acak). Metode ini terbagi dalam 5 tahapan:

1. Membuat distribusi kemungkinan untuk variabel penting.
2. Membangun distribusi kemungkinan kumulatif untuk tiap-tiap variabel di tahap pertama.
3. Menentukan interval angka random untuk tiap variabel.
4. Membuat angka random.
5. Membuat simulasi dari rangkaian percobaan.

PEMBAHASAN

Dalam simulasi ini ada beberapa langkah yang dilakukan baik dari inisialisasi graf dan penentuan variabel random dalam monte carlo.

Langkah 1: Inisialisasi jumlah verteks (n).

Langkah 2: Dari jumlah verteks kita akan dapat menghitung jumlah edge(e).

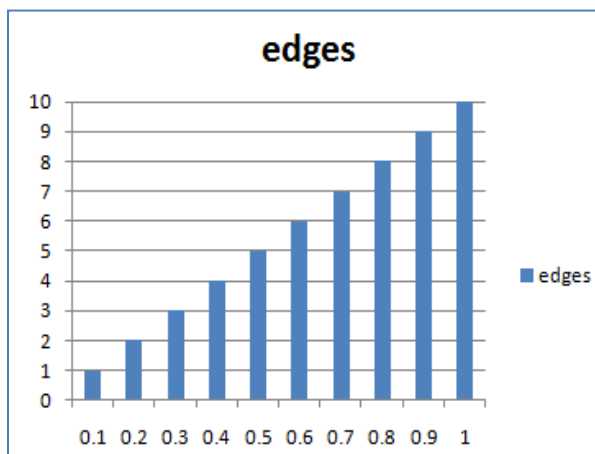
$$e = n \frac{(n-1)}{2}$$

Langkah 3: Inisialisasi bobot dari semua edge(w). Jika kita menggunakan $v=5$, maka jumlah $e(w)=(e1,e2,...,e10)$, untuk setiap edge mempunyai $e1=$ nilai bobot, $e2=$ nilai bobot, ..., $e10=$ nilai bobot, nilai bobot diinisialisasi sesuai dengan ketentuan nilai bobot.

Langkah 4: Penentuan distribusi nilai random terhadap edges. Jika kita akan melalui semua titik pada graf maka pada satu titik kita akan berhenti. Untuk itu semua edge yang ada pada graf memiliki peluang untuk terpilih dengan distribusi nilai random. Misal r adalah nilai random dengan

interval $0 \leq r \leq 1$, dengan distribusi

edge	random
e1	≤ 0.1
e2	≤ 0.2
e3	≤ 0.3
e4	≤ 0.4
e5	≤ 0.5
e6	≤ 0.6
e7	≤ 0.7
e8	≤ 0.8
e9	≤ 0.9
e10	≤ 1



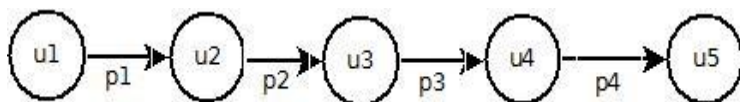
Gambar 5. Distribusi nilai random terhadap edges.

Maka setiap edge akan mempunyai probabilitas yang sama dalam kemunculannya.

Langkah 5: Menentukan banyak edge yang dibutuhkan dalam path terpendek. Setiap edge hanya menghubungkan 2 verteks, maka jumlah edge yang diperlukan adalah

$$P = n - 1$$

P=jumlah edge yang diambil dalam random.
 n=jumlah verteks.



Gambar 6. $U(H)=V(H)$, $P(H) \subseteq E(H)$

$U(H)=\{u1,u2,u3,u4,u5\}$, adalah urutan kemunculan verteks hasil dari distribusi random.

$P(H)=\{p1,p2,p3,p4\}$. Jika dalam distribusi random, histori edge yang muncul akan disimpan di $P(H)$ jika muncul edge yang sama maka random akan diulang.

Langkah 6: Seleksi path. Satu generasi adalah urutan vertek berdasarkan distribusi random dan jumlah dari semua edge yang muncul.

$$f_i = \sum_{i=1}^{n-1} p_i$$

Dimana $i=1,2,3,\dots$

f =hasil path dalam generasi. Generasi terbaik adalah generasi yang memiliki path terpendek. Generasi yang muncul adalah bergantung banyaknya epoch yang dilakukan, makin banyak besar epoch yang dilakukan makin banyak generasi yang bisa diseleksi dan semakin besar peluang untuk mendapatkan path terpendek.

KESIMPULAN DAN SARAN

Kesimpulan

- Pemanfaatan monte carlo pada path terpendek akan memunculkan semua kemungkinan jumlah dari path karena distribusi random tersebar secara merata.
- Secara konvensional memungkinkan edge terhitung dua kali memiliki akurasi yang rendah.
- Semakin banyak vertek maka semakin banyak epoch yang diperlukan dan semakin lama waktu yang dibutuhkan.

Saran

- Monte carlo dapat dikembangkan dalam graf untuk optimalisasi dalam non-

- komplit graf dengan menggunakan random pada matriks ketetanggaan.
- b. Metode ini dapat dibandingkan lebih lanjut dengan metode heuristic yang lain.

DAFTAR PUSTAKA

- [1] Cook, Williem.2012. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton University Press.
- [2] David L. Applegate, Robert E. Bixby, Vasek Chvátal & William J. Cook. 2007. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press.
- [3] Edgar G. Goodoire, Michael M. Pormenlte. 2002.*Discrete Mathematics with Graph Theory*.PRENTICE HALL.
- [4] G. Gutin, A.P. Punnen.2002.*The Traveling Salesman Problem and Its Variations*.Springer Science & Business Media, 31.05.2002 - 830 Seiten.
- [5] Graham, Carl, Talay, Denis.2013. *Stochastic Simulation and Monte Carlo Methods*.Stochastic Modelling and Applied Probability, Vol. 68,2013, XVI, 260 p. 4 illus.
- [6] Mdvín H. Kalos, Paula A. Whitlock. 2004.*Monte Carlo Methods*. WILEY-VCH Verlag GmbH & Co. KgaA.
- [7] Munir, Rinaldi. 2003. *Matematika Diskrit*. Informatika Bandung, 2003.
- [8] Robert, Christian, Casella, George. 2004. *Monte Carlo Statistical Methods*.2nd ed. 2004, XXX, 649 p.