

Aplikasi dan Algoritma Penyelesaian Optimal dari Persoalan Tukang Pos Cina

Adhiguna Surya / 13509077

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganeca no. 10, Bandung
e-mail: 13509077@std.stei.itb.ac.id

PERSOALAN TUKANG POS CINA BERARAH

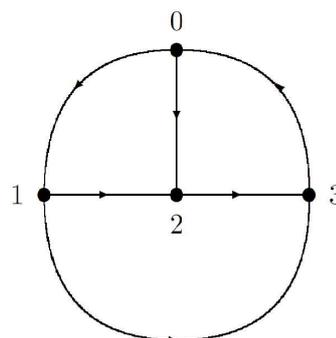
Persoalan tukang pos cina merupakan sebuah persoalan menarik yang muncul dalam teori graf. Persoalan ini pertama dipelajari oleh Mei-Ku Kuan dari Cina pada tahun 1962. Analogi dari persoalan tukang pos cina adalah tukang pos yang ingin mengatarkan surat ke alamat-alamat sepanjang jalan dalam suatu daerah. Dalam graf, sebuah alamat direpresentasikan oleh simpul dan jalan yang dapat dilalui direpresentasikan oleh sisi. Beberapa hal yang membuat persoalan ini menarik adalah fakta bahwa persoalan ini dapat didefinisikan secara singkat, memiliki banyak aplikasi di berbagai bidang, akan tetapi memiliki algoritma penyelesaian yang cukup rumit. Pada makalah ini, penulis membatasi persoalan ini kepada graf berarah, karena dirasa memiliki aplikasi yang jauh lebih luas dan tantangan yang lebih besar. Dalam analogi tukang pos, jalan yang dapat dilalui oleh tukang pos tersebut dapat merupakan jalan satu arah. Pada makalah ini juga akan dibahas secara lengkap algoritma penyelesaian persoalan ini dan optimisasi yang mungkin untuk memangkas kompleksitas. Beberapa aplikasi dari persoalan ini adalah eksplorasi robot dan analisis sistem interaktif.

Kata kunci: Persoalan tukang pos cina, graf berarah, algoritma penyelesaian, optimisasi.

1. PENDAHULUAN

Persoalan tentang seorang pedagang keliling yang ingin mencari rute terdekat merupakan sebuah persoalan yang sangat dikenal dalam dunia ilmu komputer, yang dikenal sebagai *traveling-salesman problem (TSP)*. Persoalan tukang pos cina adalah persoalan yang kurang dikenal dibandingkan dengan TSP, akan tetapi memiliki aplikasi yang beragam dan tidak kalah pentingnya. Ada beberapa variasi dari persoalan ini, yang utama adalah variasi mengenai jalan yang dapat dilalui (satu arah atau dua

arah), dan apakah tukang pos tersebut harus kembali ke titik awal. Persoalan dimana tukang pos harus kembali ke titik awal dapat disebut sebagai persoalan sirkuit tertutup, dan persoalan dimana tukang pos tidak harus kembali ke titik awal dapat disebut sebagai persoalan sirkuit terbuka. Makalah ini difokuskan kepada graf berarah, yang berarti bahwa jalan yang dilalui oleh tukang pos dapat merupakan jalan satu arah ataupun jalan dua arah. Selain itu, makalah ini membahas solusi untuk kedua variasi sirkuit terbuka dan tertutup. Perbedaan antara persoalan sirkuit terbuka dan sirkuit tertutup akan diilustrasikan pada uraian berikut.



Gambar 1.1. Graf Berarah Sederhana

Gambar 1.1 diatas menunjukkan sebuah graf berarah sederhana. Diasumsikan bahwa setiap sisi memiliki panjang yang sama. Penyelesaian optimum dari varian persoalan tukang pos cina sirkuit tertutup menghasilkan lintasan yang melalui 0, 1, 3, 0, 1, 2, 3, 0, 2, 3, 0, yang melewati sepuluh sisi. Sedangkan penyelesaian optimum dari varian sirkuit tertutup memberikan lintasan yang melalui 1, 3, 0, 1, 2, 3, 0, 2, hanya melalui tujuh sisi. Ilustrasi tersebut menekankan perbedaan yang signifikan diantara kedua varian persoalan ini. Pada bagian selanjutnya, akan dibahas aplikasi dan algoritma penyelesaian dari permasalahan tukang pos cina berarah.

Pada makalah ini juga akan dibahas *outline* algoritma penentuan rute tukang pos optimal. *Outline* yang akan dibahas menunjukkan langkah-langkah penentuan rute

tukang pos optimal, yaitu rute yang memiliki bobot paling sedikit.

2. METODE

2.1 Beberapa Aplikasi Persoalan Tukang Pos Cina

Pada kenyataannya, tukang pos tentunya ingin memilih rute yang paling pendek dengan sesedikit mungkin jumlah jalan yang dilalui berulang. Bobot dari sebuah rute didefinisikan sebagai jumlah bobot dari seluruh sisi yang dilalui. Sebuah rute yang optimal didefinisikan sebagai rute yang memiliki bobot minimum.

Aplikasi konvensional dari persoalan tukang pos cina berkaitan dengan penentuan rute yang lebih luas dari sekedar rute tukang pos, misalnya penentuan rute mobil pembersih salju, dan perencanaan pemeliharaan jalan raya. Aplikasi di beberapa bidang keilmuan lain diantaranya adalah di bidang biologi yaitu analisis DNA dan di bidang elektro yaitu *routing* robot. Uraian berikutnya akan menjelaskan secara mendetil beberapa aplikasi yang konkret dari persoalan tukang pos cina.

2.2 Aplikasi Persoalan Tukang Pos Cina dalam Telepon Genggam

Bayangkan sebuah telepon genggam. Menekan sebuah tombol dalam telepon genggam akan membawa alat komunikasi tersebut ke sebuah *state* yang baru. Setelah beberapa waktu mencoba memahami cara kerja dari telepon genggam, mungkin kita dapat memiliki sebuah gambaran, dalam bentuk graf dengan simpul menyatakan *state* dan sisi menyatakan transisi yang mungkin antara dua *state*, mengenai cara kerja dari telepon genggam. Sayangnya, gambaran yang didapat mungkin sangat kompleks dan sulit untuk dites secara sistematis. Dengan graf tersebut, sebuah rute tukang pos akan memberikan sebuah sekuens tes sistematis yang akan menguji setiap transisi antar *state*. Sebuah rute tukang pos yang optimal akan memberikan sekuens test yang memiliki bobot paling kecil, kemudian transisi-transisi yang masih belum sesuai dengan harapan dapat dicatat dan segera diperbaiki.

Sebuah contoh konkrit dari aplikasi ini dapat dilihat pada telepon genggam Nokia 2210.



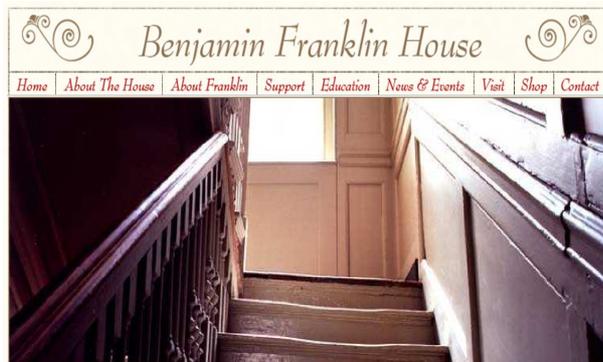
Gambar 2.1 Ponsel Nokia seri 2210

Menurut buku panduan Nokia seri 2210, telepon genggam tersebut memiliki sebuah subsistem menu yang terdiri dari 88 menu dan 273 aksi. Sebuah rute tukang pos optimal memerlukan 515 kali penekanan tombol. Sebagai perbandingan, rute terpendek yang mengunjungi setiap simpul minimal satu kali (solusi untuk persoalan pedagang keliling), hanya memerlukan 98 kali penekanan tombol. Dari contoh ini, dapat dilihat perbedaan yang cukup signifikan dari rute terpendek untuk kedua masalah.

2.3 Aplikasi Persoalan Tukang Pos Cina Dalam Pengecekan Website

Aplikasi persoalan tukang pos cina di bidang lain adalah dalam pengecekan website, terutama dalam pengecekan *broken link*, dimana *link* yang diberikan dalam website tersebut tidak dapat dibuka. Kemungkinan yang bahkan lebih buruk adalah *link* yang diberikan membawa ke halaman yang salah. Karena setiap *link* umumnya bersifat deskriptif, pembuat *website* perlu melakukan pengecekan secara manual terhadap setiap link yang ada di situs tersebut.

Untuk melakukan pengecekan dengan paling efisien, dapat dicari rute tukang pos terpendek, dengan setiap halaman situs direpresentasikan dengan simpul dan transisi antara dua halaman direpresentasikan dengan graf berarah.



Gambar 2.2 Situs Benjamin Franklin House, memiliki 66 halaman dan 1191 link yang kesemuanya harus divalidasi dalam proses pembuatannya

Untuk situs-situs yang telah didesain dengan baik, pembuat situs tidak perlu mengecek setiap *link* secara eksplisit. Gambar 2.2 menunjukkan sebuah contoh situs yang memiliki 66 halaman dan 1191 *link* dalam proses pembuatannya. Dengan menggunakan rute tukang pos optimal, diperlukan 2248 langkah untuk melakukan validasi terhadap seluruh *link* yang ada pada situs tersebut. Tentunya bagi manusia pengecekan 2248 *link* secara manual merupakan proses yang sangat melelahkan dan sulit untuk direalisasikan.

Idealnya, sebuah sistem pembuat *website* menyediakan mekanisme pengecekan *link* yang memanfaatkan rute tukang pos optimal. Untungnya, kebanyakan *compiler website* sudah menyediakan fasilitas yang dapat digunakan untuk mengecek *link* secara otomatis, sehingga jumlah *link* yang harus diperiksa secara langsung oleh pembuat situs menjadi jauh berkurang. Aplikasi berikutnya yang akan dibahas adalah persoalan eksplorasi robot.

2.4 Persoalan Eksplorasi Robot dengan Menggunakan Prinsip Rute Terpendek Tukang Pos

Sebuah persoalan lainnya yang dapat diselesaikan dengan prinsip rute terpendek tukang pos adalah persoalan eksplorasi robot, dimana sebuah robot harus menelusuri sebuah jaringan tanpa mengetahui jenis dari jaringan tersebut. Misalkan jaringan direpresentasikan oleh sebuah graf, maka robot tersebut harus menelusuri setiap sisi dan mengunjungi setiap simpul dari graf tersebut dengan jarak tempuh minimum.

Untuk sebuah jaringan dengan m buah sisi, telah ditemukan sebuah algoritma penyelesaian yang membutuhkan paling banyak $m\varphi^{O(\log \varphi)}$ langkah, dimana φ adalah defisiensi dari sebuah graf. Defisiensi adalah sebuah parameter yang menandakan kemudahan sebuah sistem untuk digunakan. Defisiensi mengaitkan seberapa sulitnya sebuah sistem untuk dimengerti tanpa menggunakan graf yang menggambarkan transisi antar *state* dari sistem tersebut, atau seberapa sulitnya sebuah jaringan, misalnya bangunan atau jaringan, untuk dipelajari. Perlu diketahui bahwa algoritma penentuan rute tukang pos optimal juga akan mencari defisiensi dari graf tersebut.

2.5 Pengaruh Variasi Persoalan Tukang Pos Cina Berarah Terhadap Aplikasi

Seperti yang telah disinggung pada bab satu, ada beberapa variasi persoalan tukang pos cina berarah, misalnya apakah tukang pos harus kembali ke titik awal. Di kehidupan nyata, variasi sirkuit terbuka jarang digunakan karena tukang pos pasti harus pulang ke titik awal.

Akan tetapi, pada aplikasinya, misalnya pada pengecekan *link* di *website*, pembuat situs tidak perlu kembali ke titik awal. Setelah selesai mengecek semua *link* yang direpresentasikan oleh sisi dalam graf, pembuat situs tidak perlu kembali ke halaman awal. Contoh lainnya adalah aplikasi pengecekan *state* telepon genggam. Dalam aplikasi telepon genggam, proses pengecekan dapat segera diakhiri setelah transisi antar *state* telah teruji. Dengan demikian dapat disimpulkan bahwa persoalan pengecekan

link pada situs merupakan persoalan tukang pos cina berarah sirkuit terbuka. Seperti yang telah diilustrasikan di bagian pendahuluan pada gambar 1.1, kedua variasi ini kemungkinan akan menghasilkan perbedaan rute terpendek yang cukup tajam.

Oleh karena itu, salah satu hal yang harus dilakukan pertama kali adalah identifikasi jenis variasi sirkuit terbuka atau sirkuit tertutup untuk menghindari ketidakefisienan dari algoritma penyelesaian.

2.6 Definisi dan Outline dari Algoritma Penentuan Rute Tukang Pos Terpendek Variasi Sirkuit Tertutup

Dalam makalah ini, graf akan didefinisikan sebagai kumpulan dari berbagai sisi dengan label i, j , dan c , dimana sisi tersebut menghubungkan simpul i dan simpul j dan c adalah bobot dari sisi tersebut. Dalam aplikasi pengecekan *link* pada situs, misalnya, sebuah halaman dari situs tersebut adalah sebuah simpul dengan *link* sebagai sebuah sisi dan waktu transisi antara dua halaman dimisalkan sebagai bobot dari sisi. Tujuan dari algoritma penyelesaian persoalan tukang pos cina adalah mendapatkan daftar label dari sisi-sisi yang membuat rute tukang pos berbobot minimum.

Karena ruang lingkup dari makalah ini dibatasi pada persoalan tukang pos cina berarah, maka setiap simpul memiliki derajat masuk dan derajat keluar. Misal jumlah sisi yang terarah ‘masuk’ kepada sebuah simpul v adalah $d^-(v)$ dan jumlah sisi yang terarah ‘keluar’ dari sebuah simpul v adalah $d^+(v)$. Misalkan δ adalah selisih antara derajat masuk dan derajat keluar, didefinisikan $\delta(v) = d^+(v) - d^-(v)$. Apabila $\delta(v) = 0$, dapat dikatakan bahwa simpul tersebut seimbang.

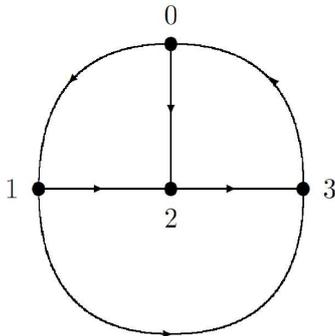
Sirkuit Euler adalah lintasan yang melalui setiap sisi tepat satu kali dan kembali ke titik awal. Sebuah teorema yang perlu diketahui adalah sebuah graf berarah memiliki sirkuit euler jika dan hanya jika setiap simpul dalam graf tersebut adalah simpul seimbang. Terdapat beberapa algoritma standar untuk mencari sirkuit euler dalam graf, dan beberapa kelas dari graf Euler hanya memiliki solusi trivial, akan tetapi hal tersebut berada diluar cakupan makalah ini. Sirkuit Euler dalam graf pasti merupakan rute tukang pos optimal, karena setiap sisi hanya dilalui maksimal satu kali.

Permasalahan baru muncul ketika graf tidak memiliki sirkuit Euler, yang berarti harus ada cara lain untuk mencari rute tukang pos optimal dari graf tersebut. Misal D^+ adalah daftar dari graf-graf yang memiliki jumlah sisi keluar lebih banyak dan D^- adalah daftar dari graf-graf yang memiliki jumlah sisi masuk lebih banyak. Hal tersebut dapat dituliskan secara formal menjadi :

$$D^+ = \{v \mid \delta(v) > 0\} /$$

$$D^- = \{v \mid \delta(v) < 0\}$$

Untuk lebih jelasnya, dapat dilihat kembali contoh graf pada gambar 1.1 yang ditaruh ulang dibawah ini.



Gambar 2.3 Contoh Graf Berarah Sederhana

Karena ada sisi yang tidak seimbang, jelaslah bahwa graf pada gambar 2.2 tidak mengandung sirkuit Euler. Kita mempunyai $D^+ = \{0,1\}$ dan $D^- = \{2,3\}$, dan δ untuk setiap simpul bernilai +1 atau -1. Dengan demikian, rute tukang pos optimal tentu harus melalui rute tambahan untuk ‘menyeimbangkan’ nilai δ yang masih belum seimbang. Untuk sebuah rute tukang pos yang optimal, pilihan jalur tambahan yang harus dilalui tentunya harus bernilai minimum.

Pada bagian ini akan ditentukan beberapa notasi yang akan membantu pendefinisian algoritma penyelesaian yang memiliki bobot paling kecil. Misal $i \rightarrow j$ mendefinisikan lintasan terpendek dari simpul i ke simpul j , dan $c(i, j)$ adalah bobot dari simpul tersebut. Apabila terdapat beberapa sisi yang menghubungkan simpul i dan simpul j , lintasan yang terpendek tentunya melalui sisi yang memiliki bobot terkecil. Untuk graf pada contoh, terdapat dua cara untuk memilih rute tambahan yang harus dilalui. Apabila dipilih $2 \rightarrow 0$ sebagai salah satunya, maka rute yang satu lagi tentunya adalah $3 \rightarrow 1$. Jalur alternatif lain yang dapat dipilih adalah $2 \rightarrow 1$ dan $3 \rightarrow 0$. Pada contoh ini, kebetulan kedua alternatif tersebut memiliki bobot yang sama, $c(2,1) + c(3, 0) = c(2,0) + c(3,1) = 4$, yang berarti bahwa salah satu dari kedua alternatif tersebut dapat dipilih untuk menghasilkan rute tukang pos yang optimal. Andaikan graf pada contoh ditambah dua lintasan baru sebagai sisi sehingga setiap simpul memiliki derajat masuk sama dengan derajat keluar, dapat dipastikan graf tersebut mengandung sirkuit Euler. Dengan demikian, rute tukang pos yang berbobot minimum dapat dicari dengan menggunakan algoritma pencari sirkuit Euler.

Dalam graf sederhana pada gambar 2.2, hanya diperlukan dua lintasan tambahan untuk ‘menyeimbangkan’ derajat masuk dan keluar dari graf tersebut. Secara umum, diperlukan k buah lintasan tambahan, dengan rumus k diberikan pada gambar 2.3.

$$k = \sum_{v \in D^+} \delta(v) \quad (1)$$

Sisi pertama dari lintasan pada (1) dapat melalui sisi manapun, sehingga ada k kemungkinan, kemudian sisi kedua dari lintasan tersebut memiliki $k-1$ kemungkinan, sisi ketiga memiliki $k-2$ kemungkinan, dan seterusnya. Dalam kasus terburuk, terdapat $k!$ alternatif, sehingga algoritma penyelesaiannya memiliki kompleksitas faktorial. Untuk itu, diperlukan algoritma yang lebih mangkus.

Secara umum, sebuah rute tukang pos berbobot minimum dapat melalui lintasan $D^- \rightarrow D^+$ lebih dari sekali. Misalkan $f(i,j)$ adalah kekerapan sisi $i \rightarrow j$ harus dilalui, atau lebih spesifiknya berapa kali sisi $i \rightarrow j$ harus ditambahkan untuk membuat graf tersebut memiliki sirkuit Euler. Sekarang, kita dapat menyatakan inti dari persoalan rute tukang pos optimal, yaitu mencari lintasan untuk membuat rute tambahan yang harus diambil menjadi sesedikit mungkin. Adapun bobot dari rute tambahan yang harus diambil dapat dinyatakan oleh (2).

$$\phi = \sum c_{ij} f_{ij} \quad (2)$$

Pada kasus ini, (2) menyatakan bobot tambahan yang muncul dari menelusuri jalur tambahan antara simpul-simpul yang memiliki derajat tidak seimbang.

Pada akhirnya, bobot total dari rute tukang pos yang optimal adalah (2) ditambahkan dengan bobot asli dari setiap sisi, karena setiap sisi harus ditelusuri minimal sebanyak satu kali dalam kasus apapun. Langkah-langkah penentuan algoritma rute tukang pos optimal akan dijelaskan lebih lanjut setelah ini.

determine from the graph : δ, D^-, D^+, c

find f to minimise : $\sum c_{ij} f_{ij}$

$$\text{subject to : } \begin{cases} f_{ij} & \text{integer} \\ f_{ij} & \geq 0 \\ \sum_{j \in D^+} f_{ij} & = -\delta(i) \\ \sum_{i \in D^-} f_{ij} & = \delta(j) \end{cases}$$

construct : Eulerian circuit, by using least cost paths $i \rightsquigarrow j$, each path repeated $f_{ij} > 0$ times

Gambar 2.4 Langkah-langkah Penentuan Rute Tukang Pos Optimal Variasi Sirkuit Tertutup

Gambar 2.3 diatas menjelaskan langkah-langkah penentuan rute tukang pos optimal. Hal pertama yang harus dilakukan adalah mencari δ , D^- , D^+ , dan c (bobot sisi). Langkah berikutnya adalah menentukan f sehingga menghasilkan persamaan (2) yang minimum. Langkah terakhir adalah membentuk sirkuit Euler dengan memanfaatkan rute yang memiliki bobot minimum dari simpul $i \rightarrow j$, dengan setiap lintasan diulangi sebanyak $f(i,j)$ kali. Setelah ini akan diuraikan contoh penggunaan algoritma penentuan rute tukang pos optimal dengan graf berarah sederhana pada gambar 2.2 sebagai contoh.

Misalkan graf berarah sederhana pada gambar 2.2 adalah graf G .

Daftar sisi graf $G : (0,1), (0,2), (1,2), (1,3), (2, 3), (3,0)$

δ setiap simpul: $\delta(0)=1, \delta(1)=1, \delta(2)= -1, \delta(3)= -1$

$D^- = \{2,3\}$

$D^+ = \{0,1\}$

Variabel : $f(2,0), f(2,1), f(3,0), f(3,1)$

Konstrains : $f(2,0) + f(3,0) = 1$

$f(2,0) + f(2,1) = 1$

$f(2,1) + f(3,1) = 1$

$f(3,0) + f(3,1) = 1$

Dengan menyelesaikan persamaan diatas, dapat diperoleh $f(2,0) = 0, f(2,1) = 1, f(3,0) = 1, \text{ dan } f(3,1) = 0$

Jalur tambahan : $f(i,j) \times i \rightarrow j \Rightarrow \{(2,1) \rightarrow (3,0)\}$

Sirkuit Euler : 2, 1, 3, 0, 2, 3, 0, 1, 2

Lintasan minimum untuk sisi-sisi yang tidak ada di G adalah $2 \rightarrow 1 : 2, 3, 0, 1$

Rute tukang pos optimal : 2, 3, 0, 1, 3, 0, 2, 3, 0, 1, 2

Uraian diatas menjelaskan langkah-langkah yang diperlukan untuk mendapat rute tukang pos optimal, yaitu rute tukang pos dengan bobot yang minimum. Perlu diingat bahwa variasi yang digunakan disini adalah variasi sirkuit tertutup. Hal tersebut ditunjukkan oleh titik keberangkatan awal yang sama dengan titik keberangkatan akhir, yaitu pada simpul 2. Hal lain yang perlu diperhatikan adalah pada contoh diatas, diasumsikan setiap sisi memiliki bobot yang sama. Pada kenyataannya, dalam kebanyakan kasus setiap sisi memiliki bobot yang berbeda-beda.

2.7 Algoritma Penentuan Rute Tukang Pos Berbobot Minimum Variasi Sirkuit Terbuka dan Optimisasinya

Pada beberapa aplikasi persoalan tukang pos cina berarah, terutama aplikasi yang berhubungan dengan *testing*, alat yang diuji tidak perlu kembali ke titik awal. Misalkan solusi optimal untuk persoalan tukang pos cina berarah variasi sirkuit terbuka berawal di simpul u dan berakhir di simpul v . Solusi persoalan variasi sirkuit terbuka dapat ditentukan dengan memisalkan penambahan sebuah sisi baru yaitu sisi (v,u) dengan prekondisi bahwa bobot sisi ini sangat besar sehingga hanya dapat digunakan tepat satu kali dalam solusi persoalan variasi sirkuit tertutup. Dengan asumsi bahwa u diketahui, solusi persoalan variasi terbuka diperoleh dengan menyelesaikan persoalan variasi tertutup untuk setiap v yang mungkin dan memilih nilai v yang memberikan bobot paling kecil.

Akan tetapi, ada algoritma penyelesaian yang lebih baik daripada mencoba setiap nilai v . Apabila graf tersebut mengandung sirkuit Euler, maka solusi persoalan variasi terbuka yang optimal tetap adalah sirkuit Euler dalam graf tersebut, biarpun sirkuit Euler bersifat tertutup (selalu kembali ke titik awal).

Dengan demikian, untuk persoalan variasi sirkuit terbuka yang mengandung sirkuit Euler, kita tidak perlu 'menambahkan' sisi baru yang menghubungkan antara u

dan v , karena penyelesaian optimalnya pasti merupakan sirkuit Euler yang terkandung dalam graf tersebut.

Apabila graf tersebut tidak mengandung sirkuit Euler, maka cara untuk mengoptimalkan algoritma penyelesaiannya adalah memilih simpul akhir, v , hanya dari himpunan D^- . Hal ini dapat dibuktikan dengan cukup mudah. Misalkan dipilih v bukan anggota himpunan D^- , yaitu simpul yang memiliki derajat masuk dan derajat keluar seimbang atau simpul yang memiliki derajat keluar lebih banyak daripada derajat masuk. Apabila simpul tersebut ditambahkan sisi baru yang menghubungkan simpul u dan simpul v , maka simpul tersebut akan memiliki derajat keluar yang lebih banyak daripada derajat masuk, sehingga menambah jumlah jalur yang harus dilalui untuk mendapatkan rute tukang pos yang optimal.

Teknik optimisasi diatas sangat bermanfaat, karena algoritma penyelesaian tidak perlu mencoba semua kemungkinan v , melainkan cukup mencoba kemungkinan v yang anggota D^- . Umumnya, jumlah simpul yang tergabung dalam himpunan D^- berjumlah kurang lebih setengah dari seluruh simpul yang ada, sehingga efisiensi algoritma penyelesaian dapat naik sampai dua kali lipatnya.

Untuk variasi persoalan dimana simpul awal u tidak diketahui, kita dapat membuktikan dengan cara yang sama bahwa u yang membuat rute tukang pos optimum pasti merupakan anggota dari himpunan D^+ . Dengan demikian, efisiensi dari algoritma penyelesaian akan meningkat karena tidak harus mencoba semua kemungkinan simpul u .

2.8 Optimisasi Algoritma Penentuan Rute Tukang Pos Terpendek

Algoritma penyelesaian yang dipaparkan pada beberapa subbab diatas masih memiliki ruang untuk optimisasi, salah satunya karena algoritma tersebut masih memanfaatkan skema pencarian linier. Algoritma penentuan rute terpendek menemukan seluruh rute yang berbobot minimum, akan tetapi hanya lintasan terpendek antara D^+ dan D^- yang digunakan.

Dengan mencari seluruh rute terpendek, secara otomatis kita juga akan menemukan pohon merentang minimum, yang salah satu dari pohon merentang minimum dari graf tersebut akan digunakan untuk mencari sirkuit Euler secara efisien.

Tentunya, penentuan rute tukang pos minimum memerlukan lintasan terpendek dan solusi untuk persoalan pemrograman bilangan linier. Akan tetapi, bentuk persoalan pemrograman bilangan linier dapat direduksi secara cukup signifikan dengan menjadikan sebuah graf dimana rute tukang pos optimal merepresentasikan solusinya.

3. Kesimpulan

Persoalan tukang pos cina berarah memiliki banyak aplikasi yang menarik dan berguna di berbagai bidang. Persoalan ini dapat diselesaikan dengan menggunakan beberapa kombinasi menarik dari algoritma standar. Makalah ini membahas beberapa contoh aplikasi dari persoalan tukang pos cina berarah dan *outline* algoritma penyelesaian untuk menentukan rute terpendek yang dapat ditempuh oleh tukang pos, baik untuk variasi sirkuit terbuka maupun variasi sirkuit tertutup.

Dari sebuah sisi, sangat disayangkan bahwa persoalan tukang pos cina berarah tidak memiliki algoritma penyelesaian yang singkat. Dari sisi yang lain, persoalan tukang pos cina berarah tetap menjadi persoalan yang menarik karena tidak adanya algoritma penyelesaian yang singkat, dan persoalan ini mengasah kemampuan studi kasus dalam mengintegrasikan beberapa algoritma.

4. Ucapan Terima Kasih

Puji syukur penulis panjatkan kepada Tuhan YME karena atas izin-Nya makalah ini dapat selesai. Penulis mengucapkan terima kasih kepada bapak Rinaldi Munir selaku dosen mata kuliah struktur diskrit, dan juga kepada para ilmuwan bidang informatika yang telah menghasilkan karya-karya yang berguna untuk menyelesaikan makalah ini.

REFERENSI

- [1] Thimbleby, Harold, "The Directed Chinese Postman Problem", UCLIC, 2006
- [2] Cormen, et.al, "Algorithms", The MIT Press, 1994
- [3] R. K. Ahuja, T. L. Magnanti & J. B. Orlin, *Network Flows : Theory Algorithms and Applications*, Prentice-Hall (Simon and Schuster), 1993
- [4] Bollobas, B., "Graph Theory", Springer-Verlag, 1970
- [5] Knuth, D.E., "The Stanford Graph Base", Addison-Wesley, 1993
- [6] Aldous, J.M. & R.J. Wilson, "Graphs and Applications", The Open University, Springer-Verlag, 2000