

Penerapan algoritma paling optimal dalam menyelesaikan Persoalan Chinese Postman dengan Mixed graph.

Marcello Faria 13519086
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13519086@std.stei.itb.ac.id

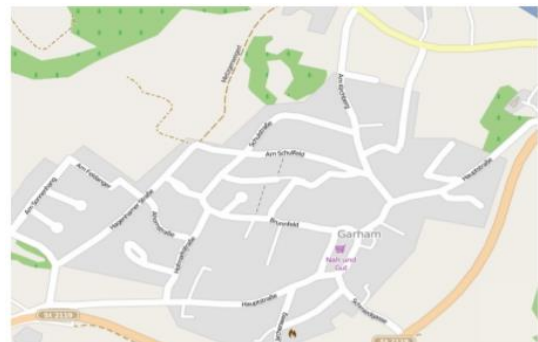
Abstract— Routing Problems atau Routing Inspection dapat dibagi menjadi dua kategori, yaitu node routing dan arc routing. Chinese Postman Problem (CPP) merupakan kasus khusus dalam arc routing problem, dan memiliki potensi dan varians yang banyak untuk diaplikasikan. Salah satu varians dari CPP ini adalah Mixed Chinese Postman Problem (MCP). Penelitian ini bertujuan untuk menerapkan salah satu penyelesaian dari MCP (yang bertipe NP-hard dari CPP dengan mixed graph) dengan menggunakan metode Heuristik. Penyelesaian persoalan MCP memiliki banyak pendekatan, beberapa aproksimasi diantaranya adalah *Mixed 1-2* oleh Frederickson (1979), *Modified Mixed 1-2* oleh Pearn and Liu (1995), *Improved Mixed1-2* oleh Pearn and Chou (1999) dan *Shortest Additional Path Heuristic*, Yaoyuanyong, Charnsethikul, and Chankong (2002). Pada makalah kali ini, cakupan bahasannya akan terbatas dalam pembahasan algoritma yang paling optimal dari semua pendekatan oleh berbagai peneliti.

Keywords— Mixed Chinese Postman Problem, Sirkuit Euler, Metode Heuristik, Mixed 1-2.

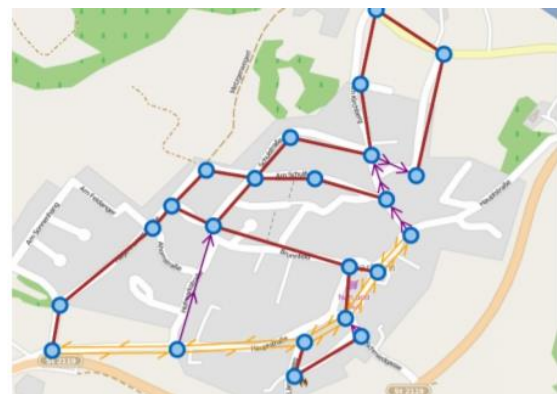
I. INTRODUCTION

Dalam mencari jarak terpendek pada algoritma travelling salesman dengan menelusuri semua kota merupakan hal yang cukup umum. Namun terdapat algoritma yang lebih jarang diketahui dan cukup kompleks bagi beberapa orang, yaitu pada algoritma *Chinese Postman* dalam mengunjungi setiap jalan untuk mengantarkan surat. *Chinese Postman Problem* atau CPP ini pertama kali dikemukakan oleh seorang pakar matematika yang berasal dari Cina di Universitas Shangtung pada awal tahun 1960-an, bernama Meigu Guan atau Kwan Meiko, dalam kesehariannya ia menyisakan waktunya untuk bekerja di suatu kantor pos pada saat revolusi kebudayaan China. Dalam praktiknya, *Chinese Postman Problem* merupakan sebuah masalah yang cukup unik dan simpel, namun belum ada algoritma yang cukup simple dalam implementasinya. Beberapa keunikan dari *Chinese Postman Problem* ini adalah banyaknya variasi dan aplikasinya. Secara umum, CPP memiliki beberapa variasi yaitu undirected CPP atau CPP yang tidak berarah, directed CPP atau CPP yang berarah, dan mixed CPP atau CPP dengan gabungan antara edge yang memiliki arah maupun yang tidak memiliki arah hingga edge yang hanya dapat dikunjungi sekali saja. Pada makalah ini, pembahasan akan hanya dibatasi pada mixed CPP saja. Dalam aplikasinya terhadap masyarakat,

mixed CPP ini tidak hanya terbatas pada pengiriman pos saja, namun juga berguna dalam rute pembersihan jalan, pengambilan tempat pembuangan sampah sementara, penentuan rute tercepat untuk bis sekolah, penjadwalan bis sekolah dan masih banyak lagi.



(a) Peta dari sebuah kota



(b) Peta dari sebuah kota dalam bentuk graf
Gambar 1, Abstraksi dari sebuah Map ke Graf

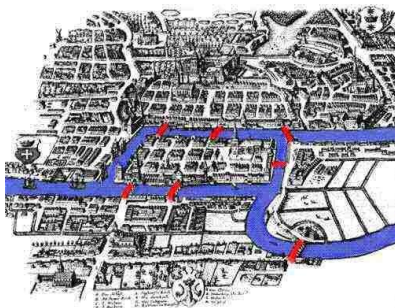
Untuk menyederhanakan ilustrasi dari persoalan Chinese Postman, anda dapat membayangkan jika anda sedang bekerja sebagai tukang pos pembawa surat pada suatu kota dan anda diberi sebuah map (Gambar 1.a) dari kota anda, dan anda sebagai tukang pos harus mampu mendesain sebuah perjalanan atau lintasan yang mengunjungi semua kota setidaknya sekali untuk mengantarkan surat. Dengan waktu yang terbatas, maka

anda harus mencari jalur tercepat dalam mengunjungi setiap kota dalam map tersebut. Dengan catatan bahwa kota tersebut memiliki jalur yang sedikit berbeda. Yaitu terdapat beberapa jalur yang cukup luas sehingga dapat diakses dalam dua arah, jalur yang sempit yang hanya dapat dilalui dalam satu arah, dan adapun jalur yang hanya dapat diakses sebanyak satu kali.

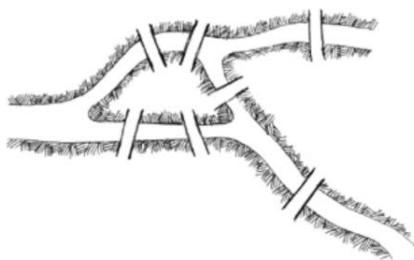
II. TEORI DASAR

A. Tur Euler

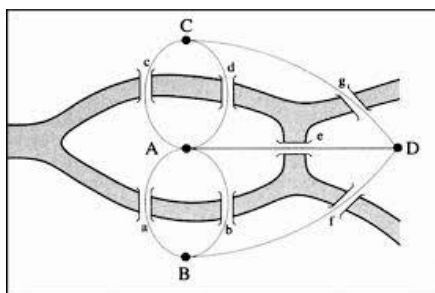
Dalam persoalan Arc routing problems (ARPs), kita diharuskan untuk menentukan biaya minimal untuk melintasi suatu subset tertentu dari suatu graf. Salah satu studi kasus pertama yang dilakukan pada ARPs ini adalah Königsberg Bridge Problem, dan persoalan tersebut telah diselesaikan oleh salah satu matematikawan dari Swiss yang bernama Leonhard Euler pada tahun 1736. Dia berkata jika suatu graf yang terhubung terdapat jalan tertutup yang mengunjungi setiap node dengan minimal satu kali dan setiap busur tepat satu kali, maka graf tersebut dinamakan dengan graf Eulerian.



(a) Gambar kota Königsberg.



(b) Gambar ilustrasi kota Königsberg.



(c) Gambar ilustrasi jembatan Königsberg dalam graf.

Gambar 2, The seven bridges of Königsberg

Gambar 2.a tersebut merupakan gambar kota Königsberg, yang merupakan ibu kota dari German East Prussia. Pada tahun 1254, King Ottoker menemukan sebuah karya yang menggambarkan kota tersebut dalam bentuk Gambar 2.b. Kota tersebut dibagi menjadi 4 bagian daratan oleh sebuah sungai yang diberikan nama Pregel. Dan terdapat jembatan dengan total 7 dibangun untuk menyeberangi sungai dan untuk melintasi satu daratan dengan daratan lainnya.

Dibutuhkan waktu hampir 150 tahun sebelum para matematikawan dapat mengilustrasikan Königsberg Bridge Problem sebagai graf yang terdiri dari nodes sebagai suatu daratan dan edges sebagai suatu jembatan seperti yang dapat dilihat pada gambar 2.c. Derajat dari vertex sebuah graf menentukan angka dari suatu edges yang bersisian terhadapnya. Dalam teori graf yang modern, Lintasan Eulerian mengunjungi setiap edge dari sebuah graf hanya sekali. Sehingga, pernyataan Euler bahwa sebuah graf yang mempunyai lintasan dengan jumlah maksimal dua nodes dengan derajat ganjil merupakan teorema pertama dalam teori graf.

B. Mixed Chinese Postman Problem

Definisi B.1

Mixed Graph $G = (V, E, A)$ merupakan suatu triple yang merupakan suatu set yang terdiri dari simpul V , suatu set yang terdiri dari edge yang tak berarah E dan suatu set busur yang berarah A . Dalam mixed graph, terdapat sebuah penghubung yang dapat kita nyatakan ke semua element sebagai $l \in E \cup A$. Dengan setiap l ditetapkan sebuah harga $c_l \in \mathbb{R}$. Busur disini memiliki arti yang berbeda dengan edge/lempang, busur memiliki definisi lempang yang berarah dan sebaliknya edge/lempang didefinisikan sebagai lempang yang tak berarah secara umum.

Definisi B.2

Jika diberikan sebuah Mixed graph $G = (V, E, A)$, sebuah lintasan berarah dari v ke u merupakan sebuah urutan dari penghubung $v = v_1 \xrightarrow{l_1} v_2 \xrightarrow{l_2} \dots \xrightarrow{l_{n-1}} v_n = u$ dengan $l_i = (v_i, v_{i+1}) \in A$ atau $l_i = (v_i, v_{i+1}) \in E$. Dapat kita nyatakan bahwa G merupakan graf yang terhubung dengan kuat jika terdapat sebuah lintasan yang berarah dari simpul apapun terhadap simpul lainnya. Tour dapat dinyatakan sebagai sebuah lintasan π akan mulai dan berakhir pada vertex yang sama dengan jumlah biaya ke semua penghubung dinyatakan sebagai: $C_\pi = \sum_{l \in \pi} c_l$.

Definisi B.3

Diberikan sebuah mixed graph yang terhubung dengan kuat, $G = (V, E, A)$. MCPP merupakan sebuah persoalan dalam mencari biaya minimum dalam melintasi seluruh penghubung $l \in E \cup A$ setidaknya sebanyak satu kali.

Untuk sebuah mixed graph $G = (V, E, A)$ dan dua buah set yang terputus dari sebuah simpul $S_1, S_2 \subset V$,

- $(S_1:S_2) = \{(i, j) \in E \cup A : (i \in S_1 \wedge j \in S_2) \text{ or } (i \in S_2 \wedge j \in S_1)\}$

$\wedge j \in S1\}$ menyatakan semua penghubung diantara sets $S1$ dan $S2$.

- $\delta(S) = (S:V \setminus S)$ dinamakan link cut-set dari G yang didefinisikan oleh S ,
- $A^+(S) = \{(i, j) \in A: i \in S \wedge j \notin S\}$, dengan semua busur keluar dari S ,
- $A^-(S) = \{(i, j) \in A: i \notin S \wedge j \in S\}$, dengan semua busur masuk ke S ,
- $E(S) = \{(i, j) \in E: i \in S \wedge j \in S\}$, untuk semua edges keluar dari S .

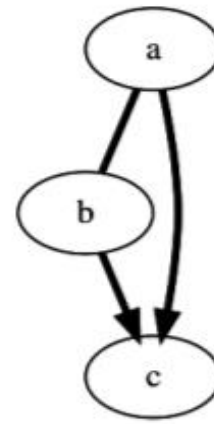
Dengan syarat bahwa masuknya terhadap MCPP terdiri dari mixed graph G yang terhubung secara kuat dan tidak mengandung multiedges apapun. Namun jika tur pada G yang mengunjungi penghubung apapun dengan jumlah setidaknya sekali mungkin akan membutuhkan untuk melewati sebuah penghubung lebih dari sekali. Untuk merepresentasikannya dengan lebih jelas kita misalkan sebagai sebuah objek matematis baru: sebuah augmented mixed graph $G^F = (V, E^F \cup A^F)$ yang didapat dengan menganggap setiap salinan dari sebuah tur sebagai elemen yang berbeda. Hal tersebut akan membuat kita mudah dalam mendefinisikan dalam genap, balans ataupun simetrik; dan hal tersebut merupakan karakteristik dari augmented mixed graph yang sesuai dengan tur MCPP.

Untuk membuat hubungan antar augmented mixed graph G^F menjadi dasar dari mixed graph G , kita harus menunjukkan x_1 dengan banyaknya salinan penghubung $l \in E \cup A$ ada dalam G^F . Dengan mengkombinasikan semua x_1 pada incidence vector $x \in R^{|E \cup A|}$ dan diberi $T \subset E \cup A$, maka dapat ditulis bahwa $x(T)$ untuk $\sum_{e \in T} X_e$. Untuk membuktikan bahwa augmented mixed graphs G^F merupakan tur yang valid dalam G , maka kita akan mendefinisikan untuk augmented mixed graph $(V, E^F \cup A^F)$, sebuah vertex $v \in V$ dapat dinamakan

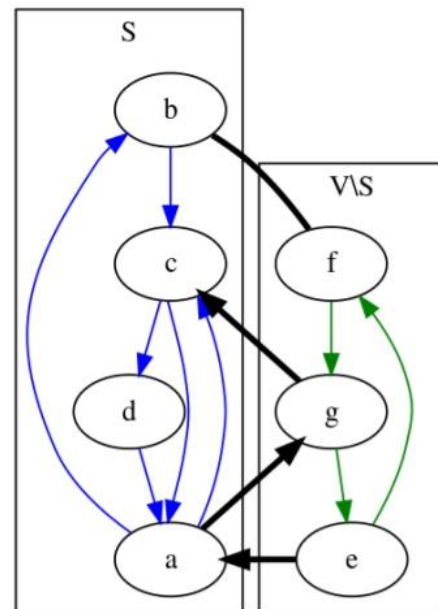
- genap bila v bersisian dengan angka genap dari salinan penghubungnya.
- Selain itu maka dinamakan ganjil.
- dan G akan dinamakan
- genap bila setiap vertex memiliki jumlah yang genap
- balans bila setiap $S \subset V$, perbedaan antara jumlah salinan busur dalam $A^+(S)$ dan jumlah salinan busur dalam $A^-(S)$ lebih kecil atau sama dengan jumlah Salinan dari edges dalam $E(S)$. Kondisi tersebut biasa disebut dengan balanced-set condition untuk set S . Dengan simbol:

$$x(A^+(S)) - x(A^-(S)) \leq x(E(S)), \forall S \subset V$$

- simetrik apabila setiap vertex dengan setiap salinan busur yang memasukkinya memiliki jumlah yang sama dengan busur yang keluar darinya.



(a) Graf genap yang tidak simetrik dan juga tidak balans sehingga jelas tidak memiliki tur euler.



(b) Upagraf dari S merupakan set yang balans. Gambar 3 Contoh dari graf

Ke-balans-an dari graf tersebut bisa saja tidak jelas atau tidak dapat langsung ditebak, jika jumlah dari busur yang masuk dan keluar tidaklah sama, maka akan terdapat jumlah dari edges yang cukup dalam menyeimbangkannya. Sebagai contoh, dapat diambil dari Gambar 3.b bahwa subset atau upagraf dari S balans, namun grafnya sendiri tidak balans. Gambar 3.a menunjukkan contoh dari augmented mixed graph yang genap.

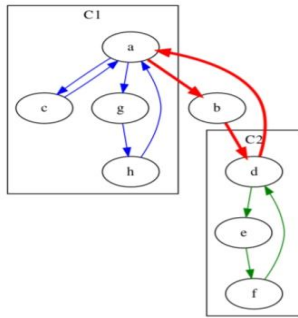
Teorema B.4

Sebuah mixed graph $G = (V, E, A)$ memiliki tur euler apabila 3 syarat berikut terpenuhi.

- G tersebut terhubung antara semua simpulnya,
- G genap,
- G juga balans.

Untuk membuktikannya, dapat kita pastikan dengan membangun sebuah sirkulasi dalam G , pada kasus ini terdapat kasus edge berarah maupun tidak berarah. Lalu dengan mengarahkan beberapa busur original tak berarah berdasarkan

sirkulasi tersebut dengan tujuan membentuk tur euler.



Gambar 4, Visualisasi dari Langkah induksi, saat menghapus tur berwarna merah, tersisa dua komponen dimana dengan induksi hipotesis dapat memiliki tur euler secara sendirinya.

Dengan properti sirkulaton tersebut, sekarang kita telah memiliki bukti yang cukup untuk memperlihatkan adanya tur euler di G_2 . Hal tersebut dapat dilakukan dengan induksi pada jumlah busur di graf tersebut. Jika G_2 tidak memiliki busur dan terhubung, maka dia akan terdiri dari vertex yang tunggal dengan panjang tur euler 0. Dengan mengasumsi bahwa graf dengan jumlah busur $m > 0$ memenuhi properti (a), (b) and (c) memiliki tur euler. Lalu dengan mempertimbangkan bahwa graf G_2 tersebut memiliki tepat m busur. Dengan (b) dan (c) memiliki busur yang kurang dari m , maka dia memiliki tur euler dengan asumsi induktif. Sehingga dapat disimpulkan bahwa G_2 memiliki tur euler.

Kita dapat menyatakan bahwa F merupakan sebuah tur untuk MCPP saat kondisi berikut dipenuhi :

- F mengandung semua penghubung yang diperlukan
- Graf $(V, E^F \cup A^F)$ terhubung
- Graf $(V, E^F \cup A^F)$ genap
- Graf $(V, E^F \cup A^F)$ seimbang

C. Algoritma Dijkstra

Algoritma Dijkstra umumnya digunakan dalam mencari path atau lintasan terpendek pada graf, digraf, maupun graf campuran dengan ketentuan yang memiliki bobot maupun tidak memiliki bobot. Jika diberikan sebuah graf berbobot $G = (V, E)$ dengan sebuah himpunan $V = \{1, 2, 3, 4 \dots n\}$ namun dengan syarat bahwa bobot setiap lengan tidak berupa negatif. Setiap Langkah yang dilakukan, dapat didefinisikan terlebih dahulu sebuah $PARENT(v)$ yang menyatakan sebuah vertex yang mendahului atau terlebih dahulu digunakan sebelum vertex v pada lintasan terpendek u_0-v yang telah didapat. Variabel tersebut akan terus diupdate (diganti) saat terdapat lintasan u_0-v yang lebih kecil atau pendek darinya. Jika dimisalkan sebuah S berupa suatu himpunan semua vertex dari G , maka berikut langkah-langkah Dalam menemukan rute / path terpendek:

a) input suatu vertex awal u_0 dengan jarak $d(u_0)$ sebagai 0, dengan $(i \leftarrow 0)$, $S \leftarrow \{u_0\}$ dan $\hat{S} \leftarrow V(G) - \{u_0\}$ dan generate semua jarak lain $d(u_0)$ dengan tak hingga untuk semua $v \in V(G) - \{u_0\}$. Jika p bernilai 0 maka langkah yang dilakukan akan dihentikan, selain dari itu, proses tetap dilanjutkan.

b) Untuk setiap $v \in \hat{S}$ sehingga $u_i v \in E(G)$, akan kita periksa bila terdapat $l(v) \leq d(u_i) + w(u_i, v)$, proses akan tetap dilanjutkan; selain dari itu maka $l(v)$ akan diinput dengan $d(u_i) + w(u_i, v)$ dan $PARENT(v)$ akan diinput u_i .

c) Dengan menentukan $m = \min \{d(v) | v \in \hat{S}\}$. Apabila v_j merupakan anggota dari \hat{S} , dengan $d(v_j) = m$, m akan menjadi jarak antara u_0 , dengan v_j dan $u_{i+1} \leftarrow v_j$.

d) S merupakan irisan dari S dan $\{u_{i+1}\}$,

e) $i \leftarrow i+1$, namun saat $i = p-1$, proses tersebut dihentikan dan kembali ke (b).

III. PEMBAHASAN

Pada bagian teori dasar sebelumnya, telah dibahas penyelesaian Mixed Chinese Postman Problem (MCP) secara umum, namun pada bab ini kita akan membahas pendekatan secara heuristik, dengan cara yang sedikit berbeda dari teori awal, oleh Yaoyuenyong, Charnsethikul, and Chankong (2002) dengan metode Shortest Additional Path Heuristic (SAPH).

3.1 Konsep dari SAPH

Diberikan sebuah $G = (N, E, A)$ sebagai Mixed Graph yang orisinal, dengan asumsi tersebut, maka G haruslah terhubung dengan kuat dan untuk semua nodes tetangga, terdapat satu edge (lengan) atau busur diantaranya. Lalu dimisalkan Kembali $G^* = (N, E^*, A^*)$ sebagai suatu solusi yang mungkin untuk MCP pada G dengan algoritma apapun, Kita dapat menggunakan Algoritma Dijkstra untuk mendeteksi jalur terpendek yang akan ditambahkan pada G^* sehingga akan menghapus beberapa tambahan edge atau busur yang berlebihan/tidak digunakan dan akan memberikan solusi akhir (G^* baru). Sehingga pada metode SAPH ini, G^* akan dioptimalkan dengan mengurangi jumlah total biaya tambahan pada G^* . Berikut beberapa notasi yang akan sering digunakan:

- $(i-j)$ edge(lengan) orisinal dan yang tidak berorientasi pada E dengan penghubung pada node i dan node j . Misalkan C_{ij} sebagai biaya/panjang dari $(i-j)$ sehingga $C_{ij} = C_{ji}$
- $\langle i \rightarrow j \rangle$ busur orisinal dari A dari node i ke j , dan $C_{ij} \neq C_{ji}$, karena C_{ij} bernilai tak hingga
- $\langle i \rightarrow j \rangle'$ edge orisinal yang telah diorientasi dari node i menuju j
- $(i \rightarrow j)$ edge tambahan yang telah diorientasi dari i menuju j (duplikat dari $(i-j)$)
- $(i \rightarrow j)'$ busur tambahan (duplikat dari $\langle i \rightarrow j \rangle$)
- SP_{ij} lintasan minimum dari node i menuju j dengan biaya $C(SP_{ij})$.

3.1.1 Tipe busur/lengan pada G*

- Type (a) $(i \rightarrow j)$, lengan yang tidak orisinil dan terorientasi
- Type (b) $(i \rightarrow j)$ atau $(j \rightarrow i)$, terorientasi namun tidak duplikat
- Type (c) $(i \rightarrow j) + (i \rightarrow j)'$ or $(j \rightarrow i) + (j \rightarrow i)'$ + ... + $(j \rightarrow i)'$, terorientasi dan terduplikasi, dengan semua lengan tambahan telah diorientasi pada arah yang sama dengan yang orisinil
- Type (d) $(i \rightarrow j) + \langle i \rightarrow j \rangle'$ terorientasi dan terduplikasi, dan hanya satu busur tambahan yang telah diorientasi dengan arah yang berlawanan dengan orisinil.
- Type (e) $\langle i \rightarrow j \rangle$, yang masih belum diduplikat (busur orisinil).
- Type (f) $\langle i \rightarrow j \rangle + \langle i \rightarrow j \rangle'$ atau $\langle i \rightarrow j \rangle + \langle i \rightarrow j \rangle' + \dots + \langle i \rightarrow j \rangle'$ terorientasi dan terduplikasi,

3.1.2 Konsep SAPH 1

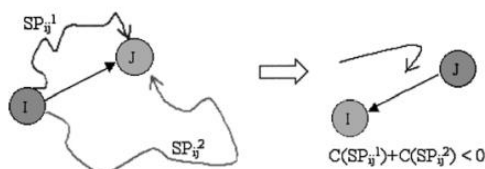
Untuk setiap lengan orisinil tambahan, jika terdapat SP_{ij} sehingga $C(SP_{ij}) < C_{ij}$, lalu kita perbaharui $e(i \rightarrow j)'$ dengan (SP_{ij}) , dengan $(i \rightarrow j)'$ membuat biaya C_{ij} dan SP_{ij} membuat biaya (SP_{ij}) . Sehingga G^* menjadi lebih efisien dengan $C_{ij} - C(SP_{ij})$. Ide dasar dari SAPH adalah dengan mengganti tambahan lengan orisinil $(i \rightarrow j)'$ pada G^* dengan lintasan berarah terpendek yang ada dengan biaya yang lebih kecil dari C_{ij} . Ide tersebut tidak hanya berlaku pada lengan tambahan, tapi juga dapat digunakan pada busur tambahan $\langle i \rightarrow j \rangle'$.

3.1.3 Konsep SAPH 2

Untuk lengan orisinal apapun, $(i \rightarrow j)$,

1. Cari lintasan terpendek SP^1_{ij} , pada G^* saat C_{ij} bernilai tak hingga,
 2. Misalkan sebuah $G_{baru} = G^* + SP^1_{ij}$.
 3. Cari lintasan terpendek kedua SP^2_{ij} , pada G_{baru} saat C_{ij} bernilai tak hingga,
- Jika $C(SP^1_{ij}) + C(SP^2_{ij}) < 0$ maka misalkan $G^* = G^* + C(SP^1_{ij}) + C(SP^2_{ij})$ dan balikkan orientasi dari lengan dari $(i \rightarrow j)$ to $(j \rightarrow i)$. Selain itu, G^* tidak akan berubah. Karena $(i \rightarrow j)$ menghasilkan C_{ij} unit, namun $(j \rightarrow i) + (SP^1_{ij}) + (SP^2_{ij})$ menghasilkan $C_{ij} + C(SP^1_{ij}) + C(SP^2_{ij})$ unit, sehingga total biaya pada G^* berkurang sebanyak $C(SP^1_{ij}) + C(SP^2_{ij})$.

3.2 Implementasi dari SAPH



Gambar 5 lengan yang diputar jika $C(SP^1_{ij}) + C(SP^2_{ij}) < 0$

Dalam SAPH, dilakukan langkah untuk menyeimbangkan semua simpul dengan *inoutdegree*, lalu dengan algoritma Dijkstra, SAPH melakukan eliminasi terhadap semua lengan/busur yang terorientasi dan mengorientasi beberapa

lengan orisinil.

Metode heuristik tersebut diimplementasikan dengan cara berikut :

1. Melakukan penyelesaian *inoutdegree* pada mixed graph G .
2. Dengan menginisiasi G^* sebagai solusi awal dari SAPH, kita dapat memisalkan U bukanlah himpunan kosong, lalu G^* menjadi solusi yang mungkin digunakan pada MCPP, jika tidak G^* bukanlah solusinya. SAPH mampu mengimprovisasi G^* saat U bukanlah himpunan kosong.
3. Menyelesaikan SAPH pada G^* dengan prosedur berikut.

3.2.1 Prosedur SAPH

Input: $G = (N, E, A)$, mixed graph yang orisinil, $G^* = (N, M, U)$ dan M^* , berupa solusi yang dihasilkan dari *inoutdegree* pada G
 $G_M = (N, E + E_M, A + A_M)$, berupa solusi yang dihasilkan oleh *evendegree* pada G . Dengan memisalkan E_M dan A_M sebagai set yang terdiri atas busur dan lengan yang bersesuaian dengan solusi.

Output: Graf eulerian yang berarah.

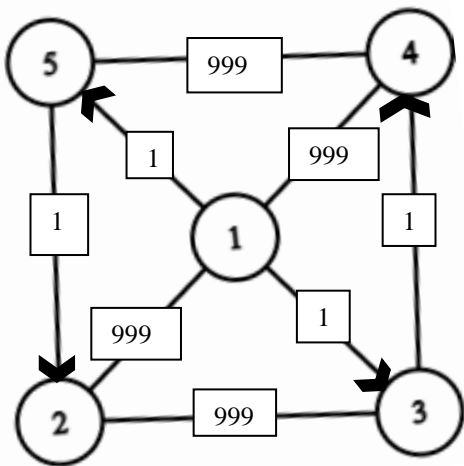
1. Dengan memilih secara acak lengan/ pada type (a), (c), (d), atau (f) pada G^* .
2. Pilih suatu lengan $(i \rightarrow j)$, lalu misalkan $G^{1_{ij}} = G$, dan $G^{2_{ij}} = G$
3. Lakukan algoritma Dijkstra untuk menyusuri semua busur/lengan yang bersesuaian dengan solusi dari hasil *evendegree*. Dengan memisalkan C^*_{fg} sebagai biaya non – negatif dari busur / lengan (f, g) pada graf orisinil G . Dengan memisalkan K sebagai suatu biaya ≤ 0 . Lalu ganti biaya busur/lengan pada G_{ij} . Pada G_{ij} , jika terdapat suatu lengan $e(f \rightarrow g) \in E_M$, input C_{fg} dengan $C_{gf} = K$. Dan jika terjadi suatu busur $\langle f \rightarrow g \rangle \in A_M$. Jika bertipe f , maka pada G_{ij} input $C_{fg} = C_{gf}$. Namun jika bertipe e , maka input C_{fg} dengan 0 dan C_{fg} dengan tak hingga.
4. Lakukan algoritma Dijkstra untuk menyusuri semua busur/lengan pada $G^{1_{ij}}$ dan $G^{2_{ij}}$ yang dapat ditingkatkan secara langsung saat selesai dalam memilih busur/lengan dengan tipe (a), (c), (d), (f). Dengan memisalkan C^*_{fg} sebagai biaya non – negatif dari busur / lengan (f, g) pada graf orisinil G . Lalu ganti biaya busur/lengan pada G_{ij} . Pada G^* jika terdapat lengan bertipe (a) atau (d) maka pada G_{ij} input $C_{fg} = -C_{fg}$ dan $C_{gf} = -C_{gf}$. Namun jika terdapat lengan dengan tipe c atau f , input $C_{gf} = -C_{fg}$.
- . Terakhir, ganti biaya pada busur/lengan yang dipilih pada kedua arah menjadi tak hingga.
5. Implementasi konsep SAPH 1.
6. Kembali pada langkah 1 hingga tidak ada tipe (a), (c), (d), (f) yang dapat ditingkatkan.
7. Pilih secara acak lengan bertipe (b).
8. Implementasi konsep SAPH 2.
9. Kembali pada langkah (7) hingga tidak ada busur beripe(b) yang dapat ditingkatkan.
10. Jika masih ada peningkatan pada langkah (8),

kembali ke langkah (1).

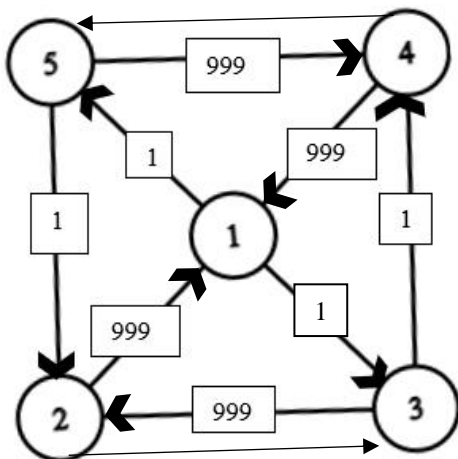
11. Jika masih terdapat lengan yang bertipe (a),(i-j) pada G^* , kita dapat menggantinya dengan $(i \rightarrow j) + (j \rightarrow i)'$.

12. Berhenti dan hasil dari G^* merupakan solusi dari SAPH.

3.3 Contoh testing pada SAPH



Gambar 6 (a) $G = (N,E,A)$ orisinal



Gambar 6(b) G^* yang dimodifikasi pada algoritma mixed2

Graf orisinal G terdiri dari 4 lengan yang tidak terorientasi dan lengan yang orisinal. $E = \{(1-2), (1-4), (2-3), (5-4)\}$ and 4 busur; $A = \{(\langle 1 \rightarrow 2 \rangle, \langle 1 \rightarrow 4 \rangle), \langle 2 \rightarrow 3 \rangle, \langle 5 \rightarrow 4 \rangle\}$. Jumlah seluruh biaya adalah 999 unit. Graf tersebut merupakan graf

Pearn and Chou (1999), yaitu salah satu graf dengan kompleksitas worse case. G^* merupakan solusi modifikasi Mixed2. Sebelum menggunakan prosedur SAPH, langkah yang harus dilakukan adalah penyelesaian pada *inoutdegree* dan solusinya akan sama dengan G^* .

Berikut langkah SAPH saat input G, G^* dan G_M .

(1) Misalkan sebuah G^* dengan memilih lengan $(5 \rightarrow 4) + (4 \rightarrow 5)'$ bertipe (d).

(2) Misalkan $G_{45}^1 = G, G_{45}^2 = G$. Dengan mengkategorikan lengan dan busur pada G^* :

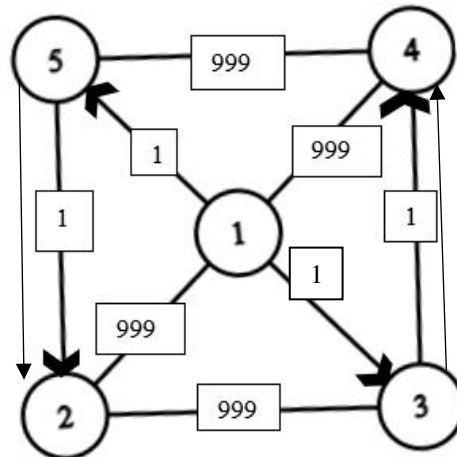
Tipe (b): $(2 \rightarrow 1), (4 \rightarrow 1)$

Tipe (d): $(5 \rightarrow 4) + (4 \rightarrow 5)', (2 \rightarrow 3) + (3 \rightarrow 2)'$

Type (e): $\{(\langle 1 \rightarrow 2 \rangle, \langle 1 \rightarrow 4 \rangle), \langle 2 \rightarrow 3 \rangle, \langle 5 \rightarrow 4 \rangle\}$

Type (a), (c), and (f): -

(3) Lakukan step 3 pada prosedur SAPH pada G_{45} dan misalkan K dengan 0. Lalu selesaikan CPP tak berarah pada G dengan mengabaikan semua arah dari usur. Misalkan E_M dan A_M sebagai suatu set busur/lengan dengan solusi yang bersesuaian dengan yang didapat pada *evendegree*. Hasil yang didapatkan adalah E_M berupa himpunan kosong dan $A_M = \{ \langle 5 \rightarrow 2 \rangle, \langle 3 \rightarrow 4 \rangle \}$.



(4) Lakukan step 4 pada prosedur SAPH dengan G_{45}^1 dan G_{45}^2 .

(5) Cari SP_{45} dan SP_{54} pada graf G_{45}^1 . Karena lengan tipe (d) ekuivalen dengan tipe (a), maka dapat diubah menjadi tipe (a)

$SP_{45} = 5 \rightarrow 2 \rightarrow 3 \rightarrow 4$ dengan biaya $0 - 999 + 0 = -999$

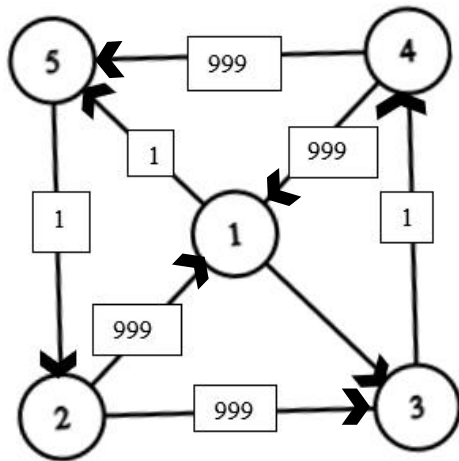
$SP_{54} = 4 \rightarrow 999$ dengan biaya $999 + 1 = 1000$

Dengan menggunakan biaya aktual pada lintasan minimum tersebut yaitu

$C(SP_{54}) = 1 - 999 + 1 = -997$ dan

$C(SP_{45}) = 999 + 1 = 1000$

Karena $C_{45} = 999$ sehingga $C(SP_{45}) < C_{45}$ dan $C(SP_{54}) < C_{54}$, lalu ganti $(5 \rightarrow 4)$ dengan (SP_{54}) ganti lengan $(4-5)$. Jika kita perhatikan, akan ada sirkuit yang berlebihan / tidak digunakan pada simpun 2 dan 3 dan akan dihapus. Sehingga hasil dari graf G^* akan menjadi Eulerian dan tidak terdapat solusi lagi yang ditingkatkan pada G^* , sehingga G^* merupakan graf yang paling optimal. Total dari biaya tersebut menjadi 2.



IV. CONCLUSION

Setelah melakukan berbagai riset dari metode heuristik yang ada, terdapat berbagai formulasi untuk menyelesaikan MCP. Beberapa peneliti diantaranya adalah Kappauf and Koehler [8] dan Christofides et al. [9] yang menyelesaikan dan membuat algoritma eksak untuk menyelesaikan MCP secara optimal, algoritmanya berbasis pada algoritma branch and bound dengan menggunakan relaksasi lagrange, lalu terdapat peneliti lain yang menggunakan teknik cutting-plane dan program linear dengan mengonversikan MCP tersebut menjadi persoalan flow-with-gains. Namun metode tersebut hanya efisien dan optimal pada ukuran atau cakupan yang kecil hingga sedang. Oleh karena itu, digunakan algoritma heuristik Mixed 1-2, yang merupakan algoritma yang telah diimprovisasi dari berbagai penelitian terbaru dalam meningkatkan efisiensi algoritma dan melalui metode heuristik tersebut, dikembangkanlah penyelesaian dengan metode Shortest Additional Path Heuristic (SAPH).

V. ACKNOWLEDGMENT

Dalam makalah ini, penulis ingin berterimakasih kepada Tuhan yang Maha Esa karena atas kelimpahan berkat dan anugrah-Nya lah penulis dapat menyelesaikan makalah ini hingga sampai titik ini dengan tepat waktu. Penulis juga ingin berterimakasih terhadap Ibu Harlili sebagai dosen dari kelas K02 pada mata kuliah Matematika Diskrit ini yang telah membagikan ilmunya kepada kami, para mahasiswa, selama satu semester ini.

Penulis juga tidak lupa untuk berterimakasih terhadap semua orang yang telah berkontribusi secara langsung maupun tidak langsung dalam kelancaran penulisan makalah ini. Dan tak lupa, penulis ingin berterimakasih terhadap semua penulis dan pemilik referensi yang tercantum pada makalah ini yang memberi kelancaran dari penulisan ini. Terakhir, penulis ingin mengucapkan permohonan maaf jika ada kesalahan dalam penulisan ini,

REFERENCES

- [1] Edmonds J, Johnson E. Matching, Euler tours and the Chinese postman. *Mathematical Programming* 1973;5: pp. 88—124.
- [2] HSIAO-FAN WANG, Yu-PIN WEN. Time-Constrained Chinese Postman Problem 2002: pp 1-13.
- [3] L. R. Ford and D. R. Fulkerson. *Flows in networks*. Vol. 1962. Princeton University Press, 1962.
- [4] Maximilian P.L. Haslbeck, *Algorithms for the Mixed Chinese Postman Problem* 2015: pp. 1-15.
- [5] M. K. Gordenko, S. M. Avdoshin, The mixed chinese postman problem, *Proceedings of ISP RAS*, 2017, Volume 29, Issue 4, 107– 122
- [6] Zulfikar, A. Y. (2012) 'Penyelesaian Masalah Chinese Postman Pada Graf Campuran Menggunakan Metode Heuristik Balans-Genap Ali Yudha Zulfikar'.
- [7] Yaoyuenyong, Charnsethikul, and Chankong. 'A Heuristic Algorithm for the Mixed Chinese Postman Problem' 2002 : pp. 1-15.
- [8] Kappauf CH, Koehler GJ. The mixed Chinese postman problem. *Discrete Applied Mathematics* 1979;1:89—103.
- [9] Christofides N, Benavent B, Campos V, Corberan A, Mota E. An optimal method for the mixed postman problem. *System modeling and optimization. Lecture Notes in Control and Information Sciences* 1983;59:641—9.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Desember 2020

Marcello Faria - 13519086