KINERJA BEBERAPA ALGORITME METAHEURISTIK PADA PENYELESAIAN *TRAVELLING SALESMAN PROBLEM*

Prapto Tri Supriyo[1)*], Bib Paruhum Silalahi[2)], Farida Hanum[3)]
Ahmad Syauqi[4)], Emde Jati[5)], Farahdila Sahara[6)], Nurul Fathiah[7)]
Institut Pertanian Bogor
Jalan Raya Dramaga, Babakan, Dramaga, Babakan, Dramaga, Bogor, Jawa Barat
email: [1)] praptosu@apps.ipb.ac.id, [2)]bipsi@apps.ipb.ac.id, [3)]fhanum@apps.ipb.ac.id,
[4)]ahmad_syauqi5@apps.ipb.ac.id, [5)]emde_jati@apps.ipb.ac.id, [6)]farahdila_shr@apps.ipb.ac.id,
[7)]nurulfathiah393@gmail.com

Abstrak

Travelling Salesman Problem (TSP) yang memilki kompleksitas NP complete seringkali digunakan sebagai standar untuk menilai kinerja algoritme-algoritme optimisasi metaheuristik. Paper ini bertujuan membandingkan kinerja Firefly Algorithm (FA), Ant Colony Optimization (ACO), Bee Colony Optimization (BCO), dan Simulated Annealing (SA) untuk menyelesaikan TSP. Keempat algoritme metaheuristik tersebut selanjutnya dibandingkan dengan algoritme eksak Integer Linear Programming (ILP) sebagai acuan yang menghasilkan solusi optimum. Hasil yang diperoleh terhadap tiga kasus yang dibangkitkan memperlihatkan bahwa solusi yang dihasilkan FA lebih baik dibandingkan dengan ACO, BCO, dan SA.

Kata Kuci: *Integer linear programming, Metaheuristik, Travelling salesman problem*

PENDAHULUAN

Travelling Salesman Problem (TSP) merupakan salah satu permasalahan dalam sistem distribusi yang sering kali kita temui. Penyelesaian terhadap TSP ini adalah upaya untuk memperoleh rute terpendek dalam suatu proses distribusi. TSP dikenal sebagai salah satu permasalahan optimisasi klasik yang berat untuk dipecahkan secara konvensional. Penyelesaian eksak terhadap persoalan ini akan melibatkan algoritme yang mengharuskan untuk mencari semua kemungkinan solusi yang ada. Sebagai akibatnya, kompleksitas waktu dari eksekusi algoritme ini akan menjadi eksponensial terhadap ukuran dari masukan yang diberikan. Berbagai penelitian telah dilakukan untuk mendapatkan algoritme yang dapat menghasilkan solusi TSP semakin baik dan semakin efisien. Karenanya, TSP merupakan topik yang sangat menarik di bidang optimisasi dan masih *up to date* untuk didiskusikan.

Penyelesaian eksak terhadap masalah TSP mengharuskan untuk melakukan perhitungan terhadap semua kemungkinan rute yang dapat diperoleh, kemudian dipilih rute yang terpendek. Untuk itu jika terdapat k kota yang harus dikunjungi, maka diperlukan proses pencarian sebanyak $\frac{(k-1)!}{2}$ rute. Dengan cara ini waktu komputasi yang diperlukan akan jauh meningkat seiring dengan bertambahnya jumlah tempat yang harus dikunjungi. Sebagai ilustrasi, untuk 16 tempat saja, diperlukan proses pencarian rute sebanyak 653.8 milyar rute. Penjelasan ini menunjukkan bahwa solusi eksak dengan penerapan algoritme deterministik terhadap masalah TSP pada jumlah tempat yang besar membutuhkan waktu eksekusi yang relatif lama.

Beberapa kasus nyata seringkali dapat diformulasikan sebagai TSP yang perlu segera diselesaikan, seperti misalnya dalam pendistribusian logistik di daerah bencana. Sehingga muncullah berbagai algoritme metaheuristik untuk menyelesaiakan kasus-kasus semacam ini. Oleh karenanya, mendapatkan atau memilih algoritme yang memerlukan waktu eksekusi yang relatif cepat serta

mendapatkan solusi terbaik merupakan suatu keniscayaan.

Algoritme metaheuristik yang terinspirasi dari alam atau yang sering disebut *Nature-Inspired Algorithms* (NIA) merupakan algoritme yang sangat kuat dalam menyelesaikan masalah optimisasi. (Yang, 2008). Beberapa NIA yang sangat dikenal antara lain adalah *Firefly Algorithm* (FA), *Ant Colony Optimization* (ACO), *Bee Colony Optimization* (BCO), *Simulated Annealing* (SA), *Genetic Algorithms* (GA), *Harmony Search* (HS), dan *Particle Swarm Optimization* (PSO)

Penelitian ini bertujuan membandingkan kinerja empat metode NIA yakni FA, ACO, BCO, dan SA untuk menyelesaikan TSP. Selanjutnya, hasil eksekusi keempat algoritme metaheuristik tersebut dibandingkan dengan metode eksak *Integer Linear Programming* (ILP).

KAJIAN LITERATUR

NIA adalah algoritme-algoritme pengoptimuman yang terinspirasi dari fenomena alam terhadap situasi yang menantang. Algoritme ini telah berhasil dimanfaatkan untuk mengatasi berbagai masalah pengoptimuman (Yang, 2008).

Algoritme ACO direpresentasikan dari perilaku semut dalam dunia nyata untuk membangun jalur terpendek antara sumber makanan dan sarangnya. Setiap semut memulai turnya melalui sebuah titik yang dipilih secara acak. Secara berulang kali, satu persatu titik yang ada dikunjungi oleh semut dengan tujuan untuk menghasilkan sebuah tur. Pemilihan titik-titik yang akan dilaluinya didasarkan pada suatu fungsi probabilitas. Semut lebih suka untuk bergerak menuju ruas yang memiliki tingkat feromon yang tinggi (Dorigo dan Gambardella, 1997). Semakin pendek sebuah tur yang dihasilkan oleh setiap semut, jumlah feromon yang ditinggalkan pada edge-edge yang dilaluinya pun semakin besar. Hal ini menyebabkan edge-edge yang diberi feromon lebih banyak akan lebih diminati pada tur-tur selanjutnya, dan sebaliknya edge-edge yang memiliki sedikit feromon menjadi kurang diminati. Rute terpendek yang ditemukan oleh semut disimpan. Proses di atas kemudian diulangi sampai tur-tur yang dilakukan mencapai jumlah maksimum (Liu *et al*., 2009).

Algoritme SA terinspirasi dari proses fisika, yaitu pendinginan bahan logam yang disebut annealing. Proses annealing dapat diartikan sebagai penurunan suhu secara perlahan pada suatu benda yang sebelumnya sudah dipanaskan sampai keadaan dimana benda tersebut mencapai freezing point atau dengan kata lain benda mencapai titik beku. Analogi antara proses annealing pada proses pendinginan logam dan annealing pada permasalahan optimasi antara lain keadaan sistem pada proses pendinginan logam adalah solusi awal pada optimasi, energi pada proses pendinginan logam adalah jarak pada optimasi, perubahan keadaan pada proses pendinginan logam adalah solusi sementara pada optimasi, temperatur pada proses pendinginan logam adalah parameter kontrol pada optimasi, keadaan beku pada proses pendinginan logam adalah solusi optimal pada optimasi (Alpianty dan Lesmono, 2018).

Algoritme BCO terinspirasi dari simulasi perilaku koloni lebah madu dalam mencari makanan di alam. Dalam algoritme BCO, jumlah lebah yang dilepas sama dengan jumlah lokasi sumber makanan. Setiap lebah mulai mengunjungi lokasi sumber makanan yang dipilih secara acak. Proses ini dilakukan berulangkali untuk membangun jalur yang layak. Lokasi sumber makanan yang dipilih lebah ditentukan dengan aturan fungsi probabilitas. Nilai probabilitas tertinggi akan dipilih lebah dalam menentukan lokasi sumber makanan. Setelah masing-masing lebah membangun jalur lengkap, lebah pekerja akan kembali ke sarang dan melakukan suatu tarian kepada lebah pengamat. Lebah pekerja akan menari dalam durasi tertentu untuk memberikan informasi rute

terpendek yang didapat kepada lebah pengamat (Wong *et al*., 2009).

Algoritme FA terinspirasi dari perilaku berkedipnya kunang-kunang dan fenomena komunikasi kunang-kunang melalui cahaya. Semua kunang-kunang adalah *unisex* atau berkelamin tunggal, sehingga setiap kunang-kunang akan tertarik satu sama lain. Tingkat daya tarik kunang-kunang proposional dengan tingkat kecerahan atau intensitas cahaya yang dimilikinya. Kunang-kunang yang memiliki cahaya yang kurang terang akan tertarik dan bergerak ke kunang-kunang yang memiliki cahaya lebih terang. Jika tidak ada kunang-kunang yang memiliki cahaya lebih terang dari dirinya, maka kunang-kunang akan bergerak secara acak. Tingkat kecerahan kunang-kunang ditentukan oleh fungsi objektif (Yang, 2008). Algoritme FA ini pertama kali diperkenalkan oleh Xin-she Yang di tahun 2007.

METODE PENELITIAN

Penelitian diawali dengan mendiskripsikan TSP secara informal dan formal, kemudian membangun model-model optimisasi beserta analisis matematiknya. Model selanjutnya divalidasi dan diimplementasikan menggunakan bantuan *software*. Terakhir, dilakukan penilaian dan perbandingan kinerja terhadap masing-masing algoritme.

Kinerja masing-masing algoritme dinilai berdasarkan total jarak yang dihasilkan oleh algoritme tersebut. Perbandingan kinerja dilakukan terhadap tiga kasus yang dibangkitkan.

Model formal TSP dikutip dari Winston (2004). Sedangkan model FA diadaptasi dari Jati dan Suyanto (2011), model ACO diadaptasi dari Dorigo & Gambardella (1997), model BCO diadaptasi dari Wong *et al* (2008), dan model SA diadaptasi dari Bayram & Sahin (2013).

Algoritme eksak dibangun berdasarkan ILP dan diselesaikan menggunakan bantuan *software* LINGO. Sementara itu, algoritme FA dan ACO dibangun secara berturut-turut menggunakan bantuan *software* Julia dan MATLAB, sedangkan algoritme BCO dan SA dibangun menggunakan bantuan *software* Python.

HASIL DAN PEMBAHASAN

Formulasi TSP

Misalkan k adalah banyaknya node/kota yang harus dikunjungi, c_{ij} adalah jarak node i ke node j, dan u_i variabel tambahan saat mengunjungi node i. Didefinisikan variabel keputusan x_{ij} bernilai 1 jika ada perjalanan dari node i ke node j, dan bernilai 0 jika selainnya. Fungsi objektif TSP adalah meminimumkan total jarak tempuh kendaraan, yaitu

$$\min Z = \sum_{i=1}^{k} \sum_{j=1}^{k} c_{ij} x_{ij}$$

dengan kendala-kendala sebagai berikut:

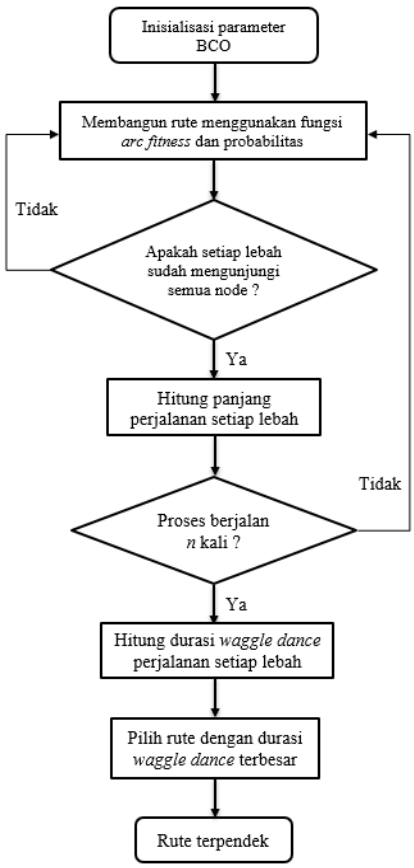
1. Setiap node harus dikunjungi tepat satu kali
   $\sum_{i=1}^{k} x_{ij} = 1, \qquad \forall j = 1,2,\dots,k$
   $\sum_{j=1}^{k} x_{ij} = 1, \qquad \forall i = 1,2,\dots,k$
2. Tidak ada subtur yang terbentuk pada rute perjalanan, sehingga hanya ada satu tur rute perjalanan berupa *cycle*
   $$u_i - u_j + k x_{ij} \leq k - 1, i \neq j, \forall i = 2,3,\dots,k; \forall j = 2,3,\dots,k; u_j \geq 0.$$
3. Kendala biner
   $$x_{ij} \in \{0,1\}, \forall i = 1,2,\dots,k; \forall j = 1,2,\dots,k$$

Implementasi FA, ACO, BCO, SA dan ILP pada TSP

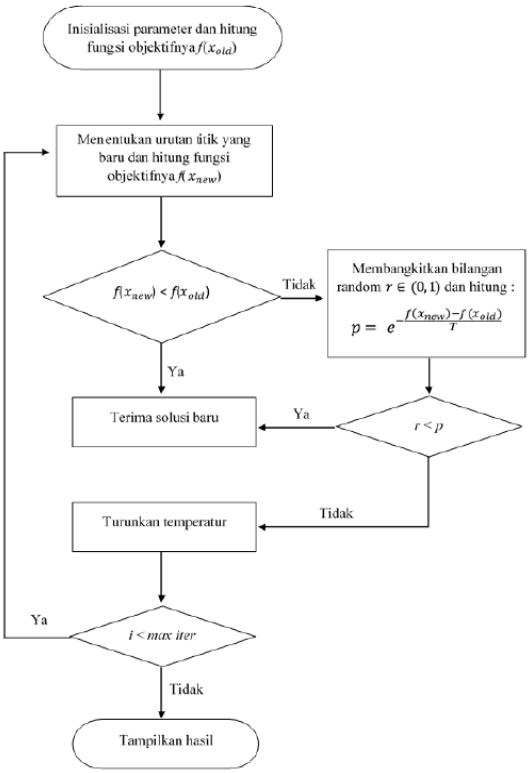
Flowchart langkah-langkah penyelesaian BCO, SA, FA dan ACO pada TSP secara berturut-turut diberikan pada Gambar 1, 2, 3 dan 4. Implementasi secara teknis FA, ACO, BCO dan SA secara berturut-turut dapat dilihat di Jati dan Suyanto (2011), Dorigo & Gambardella (1997), Wong *et al* (2008) dan Bayram & Sahin (2013).

Implementasi FA, ACO, BCO, SA dan ILP dilakukan pada tiga kasus TSP yang dibangkitkan secara acak, yakni kasus I sebanyak 30 node, kasus II sebanyak 35 node, dan kasus III sebanyak 38 node. Implementasi FA diselesaikan mengguna-kan bantuan *software* Julia. Implementasi ACO diselesaikan menggunakan bantuan *software* MATLAB. Implementasi BCO dan SA diselesaikan menggunakan bantuan *software* Python. Sementara implementasi ILP diselesaikan mengguna-kan bantuan *software* LINGO.

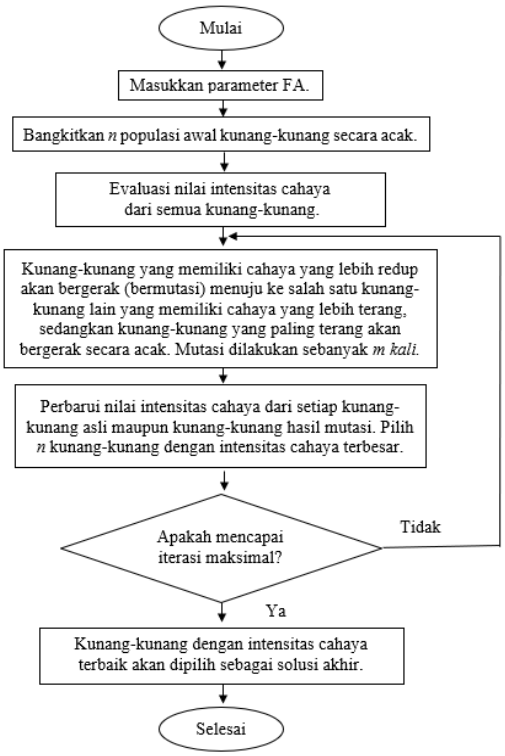
Implementasi FA pada kasus I, total jarak terkecil 390.2161 diperoleh pada saat nilai parameter koefisien dasar serap cahaya ditetapkan 0.01, parameter tingkat daya tarik dua kunang-kunang berjarak nol ditetapkan 1, jumlah populasi ditetapkan 60, jumlah iterasi ditetapkan 50, dan jumlah mutasi setiap kunang-kunang ditetapkan sebanyak satu. Implementasi FA pada kasus II, total jarak terkecil 431.4759 diperoleh pada saat nilai parameter koefisien dasar serap cahaya ditetapkan 0.01, parameter tingkat daya tarik dua kunang-kunang berjarak nol ditetapkan 1, jumlah populasi ditetapkan 50, jumlah iterasi ditetapkan 50, dan jumlah mutasi setiap kunang-kunang ditetapkan sebanyak dua. Implementasi FA pada kasus III, total jarak terkecil 442.2757 diperoleh pada saat nilai parameter koefisien dasar serap cahaya ditetapkan 0.01, parameter tingkat daya tarik dua kunang-kunang berjarak nol ditetapkan 2, jumlah populasi ditetapkan 60, jumlah iterasi ditetapkan 60, dan jumlah mutasi setiap kunang-kunang ditetapkan sebanyak satu.



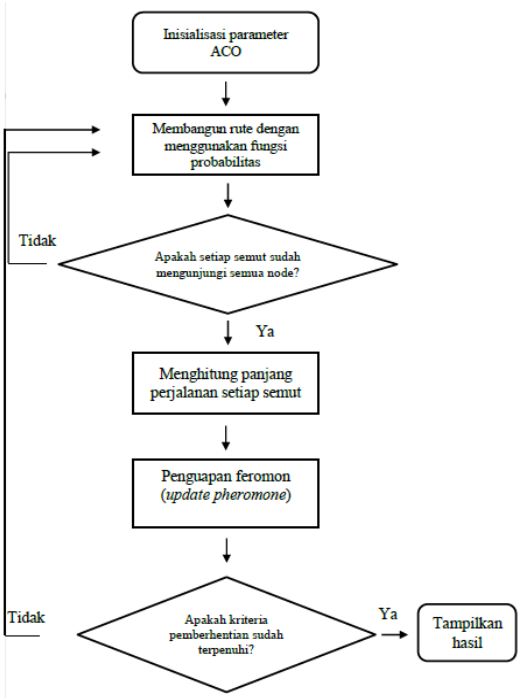
Gambar 1. *Flowchart* BCO pada TSP



Gambar 2. *Flowchart* SA pada TSP



Gambar 3. *Flowchart* FA pada TSP



Gambar 4. *Flowchart* ACO pada TSP

Implementasi ACO pada kasus I, total jarak terkecil 392.8014 diperoleh pada saat nilai parameter pengendali feromon ditetapkan 0.9, parameter pengendali jarak ditetapkan 5, jumlah populasi ditetapkan 50, dan jumlah iterasi ditetapkan 500. Implementasi ACO pada kasus II, total jarak terkecil 463.4509 diperoleh pada saat nilai parameter pengendali feromon ditetapkan 1, parameter pengendali jarak ditetapkan 5, jumlah populasi ditetapkan 50, dan jumlah iterasi ditetapkan 700. Implementasi ACO pada kasus III, total jarak terkecil 464.7083 diperoleh pada saat nilai parameter pengendali feromon ditetapkan 0.9, parameter pengendali jarak ditetapkan 5, jumlah populasi ditetapkan 70, dan jumlah iterasi ditetapkan 800.

Implementasi BCO pada kasus I, total jarak terkecil 390.2264 diperoleh pada saat nilai parameter signifikansi relatif *arc fitness* ditetapkan 1, parameter tingkat signifikansi jarak antar node ditetapkan 2, probabilitas sebuah node dikunjungi seekor lebah ditetapkan 0.2, dan faktor skala *waggle dance* ditetapkan 1. Implementasi BCO pada kasus II, total jarak terkecil 461.8564 diperoleh pada saat semua nilai parameter yang ditetapkan sama dengan yang ditetapkan pada kasus II. Implementasi BCO pada kasus III, total jarak terkecil 453.4944 diperoleh pada saat nilai parameter signifikansi relatif *arc fitness* ditetapkan 1, parameter tingkat signifikansi jarak antar node ditetapkan 4, probabilitas sebuah node dikunjungi seekor lebah ditetapkan 0.2, dan faktor skala *waggle dance* ditetapkan 1. Sedangkan jumlah populasi dan jumlah iterasi sebanyak node di setiap kasusnya.

Implementasi SA pada kasus I, total jarak terkecil 393.132 diperoleh pada saat nilai parameter rasio pendinginan ditetapkan 0.99, temperatur awal ditetapkan 10000, dan jumlah iterasi ditetapkan 15000. Implementasi SA pada kasus II, total jarak terkecil 440.9545 diperoleh pada saat nilai parameter rasio pendinginan ditetapkan 0.999, temperatur awal ditetapkan 30000, dan jumlah iterasi ditetapkan 60000. Implementasi SA pada kasus III, total jarak terkecil 448.2024 diperoleh pada saat nilai parameter rasio pendinginan ditetapkan 0.999, temperatur

awal ditetapkan 50000, dan jumlah iterasi ditetapkan 3500000.

Perbandingan total dan selisih jarak hasil eksekusi metode FA, SA, BCO dan ACO terhadap metode eksak ILP diberikan pada Tabel 1 dan Tabel 2.

Tabel 1. Perbandingan total jarak

Kasus	Total Jarak				
	ILP	**FA**	**SA**	**BCO**	**ACO**
I	388.3712	390.2161	393.132	390.2264	392.8014
II	427.5839	431.4759	440.9545	461.8564	463.4509
III	435.4919	442.2757	448.2024	453.4944	464.7083

Tabel 2. Perbandingan selisih total jarak

Kasus	Selisih Total Jarak Terhadap ILP			
	FA	**SA**	**BCO**	**ACO**
I	0.475%	1.226%	0.478%	1.141%
II	0.91%	3.127%	8.015%	8.388%
III	1.558%	2.919%	4.134%	6.709%

Berdasarkan Tabel 2, dapat dilihat bahwa untuk setiap kasus, metode FA memiliki solusi yang lebih baik dibandingkan dengan metode SA, BCO, dan ACO. Selisih jarak yang diperoleh antara metode eksak ILP dengan metode FA relatif kecil, yaitu tidak lebih dari 2%.

KESIMPULAN

FA, ACO, BCO, SA dapat digunakan untuk menyelesaikan TSP. Pada tiga kasus TSP yang dibangkitkan secara acak, metode FA memiliki solusi rute dengan jarak yang lebih pendek dibandingkan dengan menggunakan metode ACO, BCO, dan SA. Perbandingan selisih jarak yang dihasilkan antara metode FA dengan metode eksak ILP pada kasus pertama 30 node, kasus kedua 35 node, dan kasus ketiga 38 node secara berturut-turut sebesar 0.475%, 0.91%, dan 1.588%.

REFERENSI

Alpianty CN & Lesmono JD. (2018). Penerapan Algoritme Simulated Annealing untuk Menyelesaikan Asymmetric Traveling Salesman Problem. *Seminar Nasional Matematika (SEMMAT UNPAR)*; 2018 Sep 22; Bandung, Indonesia. Bandung: UNPAR. hlm 63-69.

Bayram H, Sahin R. (2013). A New Simulated Annealing Approach for Traveling Salesman Problem. *Mathematical and Computational Applications*. 18(3): 313-322.

Dorigo M & Gambardella L. (1997). Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation.* 1(1): 53-66. doi: 10.1109/4235.585892.

Jati, G.K, dan Suyanto. (2011). Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem. *In Proceedings of the 2nd International Conference on Adaptive and Intelligence System (ICAIS)*. 6943:393-403.

Liu W, Li S, Zhao F, Zheng A.(2009). An Ant Colony Optimization Algorithm for The Multiple Traveling Salesmen Problem. *IEEE Conference on Industrial Electronics and Applications*. doi: 10.1109/ICIEA.2009.5138451.

Wong LP, Low MYH, Chong CS. (2008). A Bee Colony Optimization Algorithm for Travelling Salesman Problem. *IEEE International Conference on Industrial Informatics*. 818-823. doi: 10.1109/AMS.2008.27.

Wong LP, Low MYH, Chong CS. (2009). An Efficient Bee Colony Optimization Algorithm for Traveling Salesman Problem using Frequency-based Pruning. *IEEE International Conference on Industrial Informatics*. 775-782. Doi: 10.1109/INDIN.2009.5195901.

Winston WL. (2004). *Operations Research: Applications and Algorithms 4thed*. California (US): Duxbury.

Yang X.S. (2008). *Nature-inspired Metaheuristic Algorithm*. Bristol: Luniver Press.