

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Penelitian Terdahulu

Archetti *et al.* (2009) menggunakan sebuah metode eksak yaitu *branch-and-price scheme* dan dua metode *metaheuristics* yaitu algoritma *Variable Neighborhood Search* dan *Tabu Search algorithm: Tabu Feasible*, dan *Tabu Admissible*. Penelitian ini menyimpulkan bahwa metode VNS secara keseluruhan memberikan hasil terbaik untuk pada contoh kasus (*instances*) yang diujikan.

Tarantilis *et al.* (2013) menggunakan *Bi-level Filter-and-Fan method* (BiF&F). Metode ini kemudian dikembangkan dengan melakukan optimisasi parameter. Optimisasi parameter menghasilkan *Bi-level Filter-and-Fan fast* (BiF&F-f) dan *Bi-level Filter-and-Fan slow* (BiF&F-s). Penelitian ini menyimpulkan bahwa BiF&F-s memberikan hasil yang lebih baik dibandingkan BiF&F-f. Penelitian ini juga menunjukkan bahwa metode yang digunakan oleh Tarantilis *et al.* (2013) menghasilkan solusi yang lebih baik dibandingkan penelitian Archetti *et al.* (2009).

2.1.1. Branch-and-Price

Metode ini diperkenalkan oleh Barnhart *et al.* (1998) kemudian diimplementasikan oleh Bouscier *et al.* (2006) pada kasus TOP (*Team Orienteering Problem*) dan Feillet *et al.* (2007) pada VRP. Merupakan pengembangan dari metode *branch-and-cut* (Rinaldi, 1991). Filosofi yang dimiliki oleh *branch-and-price* dan *branch-and-cut* adalah sama, perbedaan terletak pada *branch-and-price* yang prosedurnya berfokus pada *column generation* (pembangkitan kolom).

2.1.2. Tabu Search Algorithm

Metode ini digunakan oleh Gendreau *et al.* (1994) untuk memecahkan masalah VRP. Kemudian digunakan oleh Archetti *et al.* (2006) pada VRP dengan pengiriman terpisah (*Split Delivery*). Archetti *et al.* (2009) mengadaptasikan metode ini untuk memecahkan CTOP yaitu *Tabu Feasible* dan *Tabu Admissible*.

2.1.3. Variable Neighborhood Search (VNS)

Menurut Mladenovic & Hansen (1997), *Variable Neighborhood Search* didasarkan pada perubahan struktur daerah sekitar didalam pencarian heuristik

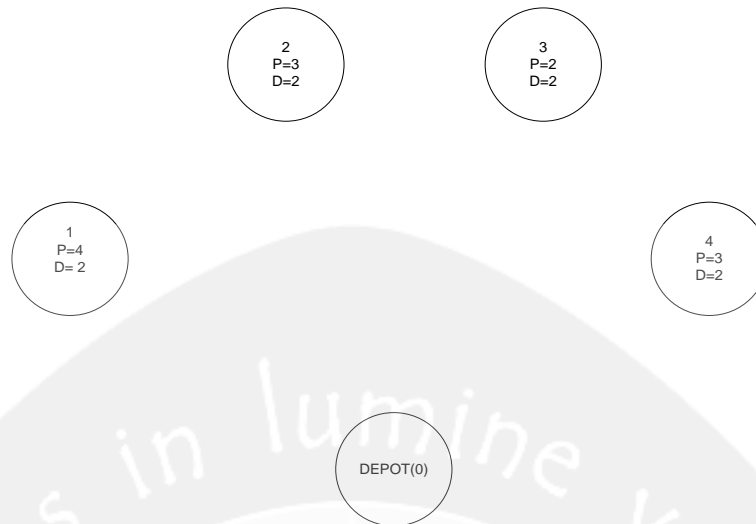
lokal (Gendreau & Potvin, 2005). Metode ini mencari struktur daerah sekitar (*neighborhood structure*) yang terbaik dalam setiap iterasi yang dilakukan.

2.1.4. Bi-level Filter-and-Fan method

Metode ini diperkenalkan oleh Tarantilis *et al.* (2013). Sesuai namanya, metode ini dibagi kedalam dua tingkat (*level*). Tingkat pertama (*upper level*), optimisasi berorientasi pada maksimasi keuntungan yang didapatkan. Pada tingkat ini, metode optimisasi yang digunakan adalah *Filter-and-Fan method*. Tingkat kedua (*lower level*), optimasi berorientasi pada minimasi jarak tempuh kendaraan, metode optimasi yang digunakan pada tingkat ini adalah *Variable Neighborhood Descent* (VND) (Gendreau & Potvin, 2005)

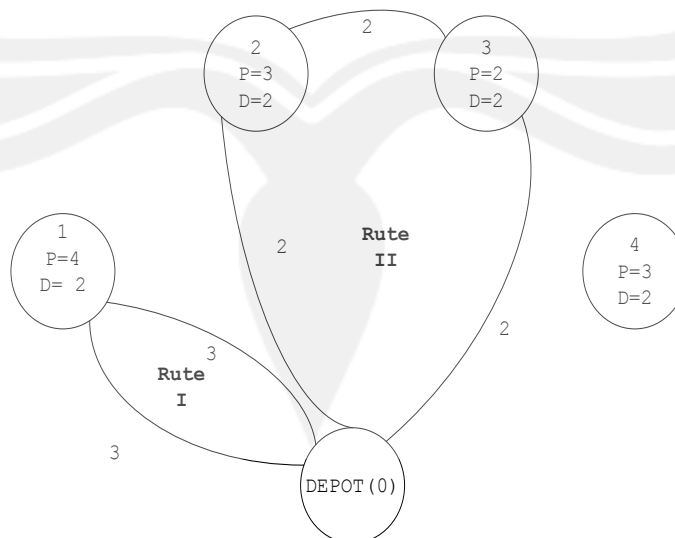
2.2. Capacitated Team Orienteering Problem

Capacitated Team Orienteering Problem (CTOP) merupakan pengembangan dari TOP. CTOP didefinisikan oleh Archetti *et al.* (2009) sebagai masalah transportasi dimana sejumlah konsumen potensial tersedia, setiap konsumen memiliki *demand* dan skor (*profit*). Konsumen tertentu harus dipilih guna memaksimalkan skor, tanpa melanggar batas durasi waktu tiap rute dan daya angkut (kapasitas) tiap kendaraan. Pada CTOP setiap konsumen hanya akan dipilih (dikunjungi) sekali dan setiap rute berawal dari depot dan berakhir di depot. Ada dua perbedaan antara CTOP dan TOP. Perbedaan pertama adalah keberadaan *demand* pada setiap *vertex* konsumen. Perbedaan kedua adalah adanya kendala kapasitas kendaraan pada setiap kendaraan. Sehingga untuk setiap kendaraan yang mengunjungi konsumen, tidak hanya mempertimbangkan keterbatasan jarak/waktu tempuh, tetapi juga keterbatasan kemampuan/daya angkut kendaraan. Ilustrasi sederhana dari kasus CTOP dapat dilihat pada Gambar 2.1.



Gambar 2.1. Contoh Kasus CTOP

Pada Gambar 2.1 terdapat sebuah depot (*vertex* 0) dan 4 buah konsumen (*vertex* 1, 2, 3, dan 4). Masing-masing konsumen memiliki *profit* (P) dan *demand* (D). Waktu tempuh dari satu *vertex* ke *vertex* lainnya (C_{ij}) adalah 2 satuan waktu, kecuali untuk C_{01} dan C_{13} yaitu 3 satuan waktu. Jika jumlah kendaraan (m) = 2 buah; Kapasitas kendaraan (Q_{max}) = 5 satuan; Waktu tempuh maksimum (T_{max}) = 6 satuan waktu. Maka solusi untuk contoh kasus CTOP pada Gambar 2.1 dapat digambarkan pada Gambar 2.2.



Gambar 2.2. Solusi contoh kasus CTOP

Pada Gambar 2.2 terbentuk 2 buah rute sesuai dengan jumlah kendaraan (m). Pada rute I, total waktu tempuh adalah 6 satuan waktu (tidak melebihi T_{max}) sedangkan total *demand* yang diangkut adalah 2 satuan (tidak melebihi Q_{max}). Pada rute II, total waktu tempuh adalah 6 satuan waktu (tidak melebihi T_{max}) sedangkan total *demand* yang diangkut adalah 5 satuan (tidak melebihi Q_{max}). Sehingga total *profit* yang diperoleh adalah $P_1 + P_2 + P_3 = 9$.

2.3. Particle Swarm Optimization

Particle Swarm Optimization (PSO) adalah metode optimisasi yang didasarkan pada populasi (Kennedy & Eberhart, 1995). Metode ini meniru kebiasaan dari organisme berkelompok seperti kerumunan lebah, sekelompok ikan, dan kawanan burung. PSO meniru gerakan fisik dari sebuah individu didalam kawanan tersebut sebagai metode pencarian.

Partikel pada PSO dapat dianalogikan pada ilustrasi berikut. Sebuah sarang lebah terletak ditengah sebuah taman. Pada sarang lebah terdapat lima ekor lebah, masing-masing lebah A, B, C, D, dan E. Kelima lebah memiliki satu tujuan, yaitu membawa madu kualitas terbaik ke sarang mereka sebelum pukul 5 sore tanpa melewati batas area pencarian yaitu taman. Pada awal pencarian, masing-masing lebah terbang ke posisi acak (menurut kebiasaan masing-masing) dari sarang mereka dengan kecepatan awal tertentu. Saat tiba pada suatu bunga, masing-masing lebah melaporkan ke lebah lainnya mengenai data yang mereka peroleh. Data tersebut berisi posisi bunga (posisi lebah terkini) dan kualitas madu bunga tersebut. Para lebah kemudian membandingkan laporan tersebut dengan masing-masing data yang mereka miliki. Setelah dievaluasi ternyata (misal) lebah A memiliki kualitas madu yang lebih baik dibandingkan keempat lebah yang lain, sehingga keempat lebah yang lain menjadikan madu yang ditemukan lebah A sebagai madu terbaik keseluruhan saat itu. Dari hasil evaluasi, kelima lebah kemudian terbang mencari bunga selanjutnya. Kecepatan lebah mencari posisi bunga selanjutnya yang akan dihindangi oleh lebah didasarkan pada tiga kriteria berikut ini:

1. Posisi lebah terkini
2. Posisi bunga dengan madu terbaik (lebah A)
3. Inersia akibat kecepatan awal lebah

Lebah yang hinggap ke bunga selanjutnya kemudian membandingkan kualitas madu terkini dan sebelumnya. Hal ini terus berulang hingga tenggat waktu

pencarian madu yaitu pukul 5 sore. Lebah dengan kualitas madu terbaik pada pukul 5 sore yang akan membawa madu kesarang.

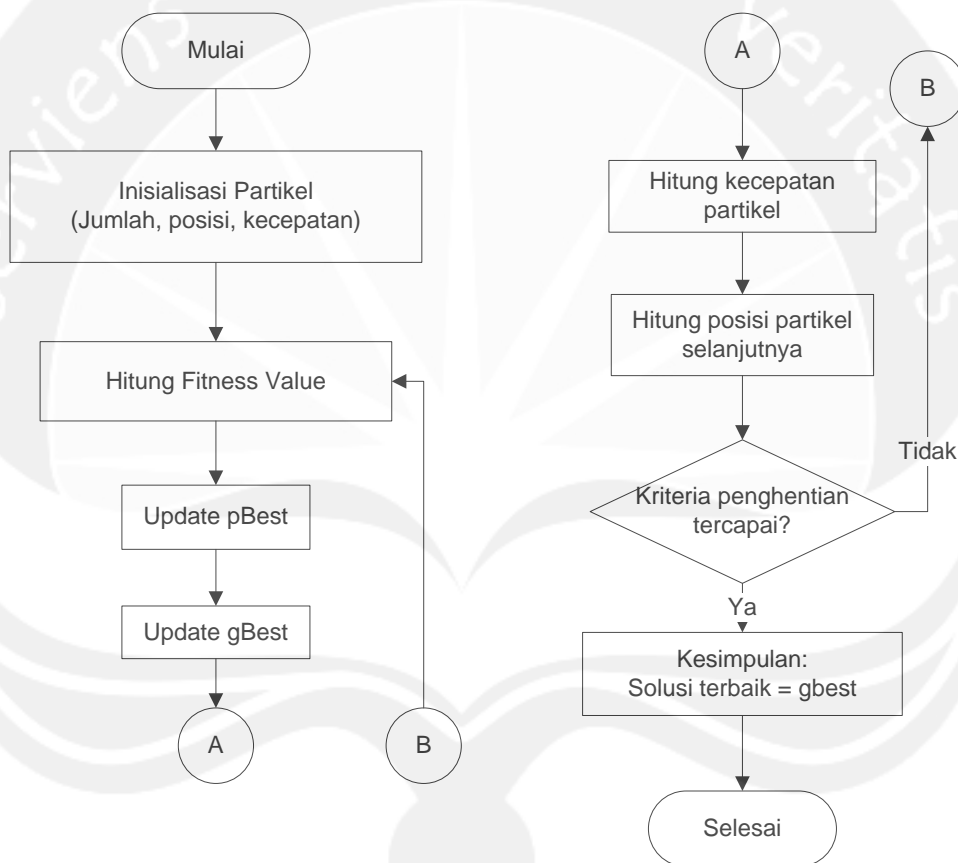
Ilustrasi di atas memiliki poin-poin penting yang erat kaitannya dengan PSO, yaitu:

1. Lebah merepresentasikan partikel pada PSO
2. Tujuan yang dimiliki lebah merepresentasikan fungsi tujuan pada PSO, baik itu minimasi atau maksimasi.
3. Batas area pencarian lebah yaitu taman, merepresentasikan batas atas dan batas bawah area pencarian solusi dari sebuah masalah optimasi.
4. Batas pencarian pukul 5 sore merepresentasikan batas pencarian dari sebuah masalah optimasi (pada kasus-kasus, secara umum, batas ini berupa jumlah iterasi).
5. Awal pencarian merupakan inialisasi partikel atau iterasi ke-0 dengan posisi awal partikel (θ_τ).
6. Data yang dimiliki oleh para lebah merepresentasikan *pBest* (*personal best*). Dalam bahasa optimisasi, data ini berisi sebuah titik (posisi) dalam area pencarian dan nilai dari fungsi yang ingin dioptimasi.
7. Saat membandingkan data masing-masing lebah, diketahui bahwa lebah A merupakan lebah dengan madu terbaik. Proses membandingkan merepresentasikan proses membandingkan *fitness value* terhadap *pBest*. Data yang dimiliki lebah A merepresentasikan *gBest* (*global best*).
8. Lebah yang mencari bunga selanjutnya adalah partikel yang mencari posisi selanjutnya (kemungkinan lebih baik atau lebih buruk dari sebelumnya), sekali lagi dilakukan perbandingan antara *fitness value* terhadap *pBest*.
9. Kecepatan lebah saat mencari bunga selanjutnya merepresentasikan kecepatan ($\omega_{\tau+1}$) partikel mencari posisi selanjutnya yang dipengaruhi oleh *pBest*, *gBest*, dan inersia akibat kecepatan awal ($W\omega_\tau$).
10. Posisi bunga selanjutnya yang akan dihindangi merepresentasikan posisi partikel selanjutnya ($\theta_{\tau+1}$), ditentukan oleh posisi partikel saat ini (θ_τ) dan ($\omega_{\tau+1}$).
11. Persamaan-persamaan yang digunakan pada PSO (Ai, 2008) adalah:

$$\theta_{\tau+1} = \theta_\tau + \omega_{\tau+1} \quad (2.1)$$

$$\omega_{\tau+1} = W\omega_\tau + C_p u(\psi_{p,\tau} - \theta_\tau) + C_g u(\psi_{g,\tau} - \theta_\tau) \quad (2.2)$$

Persamaan (2.1) menunjukkan bahwa posisi partikel periode berikutnya ($\theta_{\tau+1}$) diperoleh dari penjumlahan posisi saat ini (θ_{τ}) dengan kecepatan periode berikutnya ($\omega_{\tau+1}$). Persamaan (2) menunjukkan bahwa kecepatan periode berikutnya ($\omega_{\tau+1}$) diperoleh dari penjumlahan dari hasil kali bobot inersia suatu partikel (W) dengan kecepatan saat ini (ω_{τ}), konstanta kognitif personal (C_p) dengan posisi terbaik partikel tersebut ($\psi_{p,\tau} - \theta_{\tau}$), dan konstanta kognitif global (C_g) dengan posisi terbaik keseluruhan partikel ($\psi_{g,\tau} - \theta_{\tau}$). Masing konstanta kemampuan kognitif dikali dengan u yaitu suatu bilangan acak [0,1]. Diagram alir PSO dapat dilihat pada Gambar 2.3.



Gambar 2.3. Diagram Alir PSO

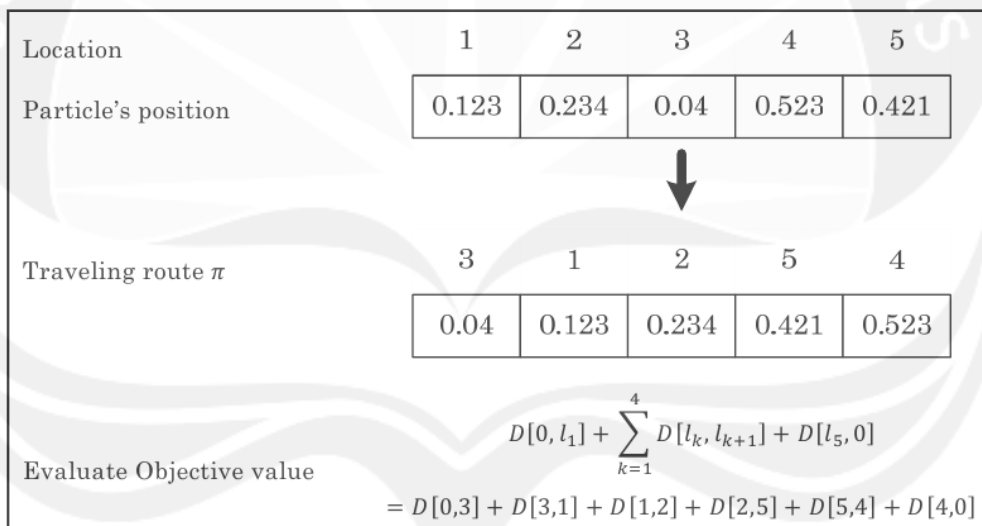
2.4. ET-Lib versi 1.0

ET-Lib adalah *object-oriented library* yang ditulis menggunakan bahasa pemrograman C#. ET-Lib dikembangkan pada tahun 2008 di Asian Institute of Technology (AIT), Thailand. ET-Lib dibuat sebagai sebuah *tool* yang umum dan

efektif bagi para peneliti dan mahasiswa yang mempelajari berbagai macam permasalahan optimisasi. ET-Lib versi 1.0 memiliki fungsi obyektif minimasi.

Versi awal ET-Lib menggunakan algoritma *Particle Swarm Optimization with multiple social learning terms* (GLNPSO). Berbeda dengan PSO dasar, GLNPSO memiliki kemampuan *social learning*, ditandai dengan adanya bobot sosial C_l dan C_n , serta variabel sosial *local best* ($lbest$) dan *near neighbor best* ($nbest$). Keberadaan C_l , C_n , $lbest$, dan $nbest$ akan memengaruhi kecepatan suatu partikel dan secara tidak langsung memengaruhi posisi partikel berikutnya.

ET-Lib telah digunakan untuk menyelesaikan berbagai masalah optimisasi, yaitu *Travelling Salesman Problem* dan *Job Shop Scheduling Problem*. Pada TSP, dimensi partikel sesuai dengan jumlah *node* yang ada, kemudian dimensi dari posisi partikel diurutkan, kemudian urutan dimensi partikel tersebut yang menjadi solusi rute TSP. Gambar 2.4 menunjukkan proses *decoding* posisi partikel menjadi solusi pada contoh kasus TSP di User's Manual ET-Lib.



Gambar 2.4. Decoding TSP (Nguyen et al., 2010)