JAVA Journal of Electrical and Electronics Engineering
Volume 13, Number 1, April 2015

Optimisasi *Multi Depot Vehicle Routing Problem (MDVRP)* dengan Variabel *Travel Time* Menggunakan Algoritma *Particle Swarm Optimization*

Nurlita Gamayanti[1], Abdullah Alkaff[2], Randi Mangatas[3]
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya, Indonesia
lita@ee.its.ac.id[1], abdullah.alkaff@gmail.com[2], randimangatas10@gmail.com[3]

***Abstract*—Multi Depot Vehicle Routing Problem (MDVRP) merupakan permasalahan optimasi yang memiliki peranan penting dalam manajemen sistem distribusi dengan tujuan meminimalkan waktu yang diperlukan, dimana penentuan waktu berkaitan dengan jarak dari rute yang ditempuh oleh armada distribusi. Penelitian ini memberikan sebuah formulasi dari kasus *Multi Depot Vehicle Routing Problem* (*MDVRP*) yang diselesaikan dengan metode algoritma *Particle Swarm Optimization* (*PSO*). Terdapat dua tahapan untuk menyelesaikan MDVRP yaitu *clustering* dan *assignment*. Dalam clustering digunakan metode *simplified parallel assignment*, sedangkan untuk *assignment* digunakan metode *particle swarm optimization*. PSO merupakan salah satu teknik komputasi. Algoritma *Particle Swarm Optimization* dapat menghasilkan waktu tempuh yang minimum.**

***Keywords*—Rute; clustering; assignment; Multi Depot Vehicle Routing Problem; Particle Swarm Optimization.**

I. PENDAHULUAN

Permasalahan distribusi merupakan permasalahan yang hampir tidak bias lepas dalam dunia industri terutama yang bergerak dalam bidang produksi. Permasalahan distribusi sering ditemukan dalam kehidupan sehari-hari seperti pengangkutan sampah, pengiriman galon atau pada pengantaran dan penjemputan yang dilakukan bus sekolah.

Dalam dunia industri permasalahan distribusi merupakan salah satu faktor yang penting dalam mempengaruhi pendapatan. Berdasarkan penelitian beberapa ahli yang menyatakan bahwa biaya distribusi rata-rata sebesar 16% dari harga jual barang yang dihasilkan, oleh karena itu perlu adanya suatu metode yang digunakan untuk mengurangi biaya yang digunakan untuk mendistribusikan barang.[11]

Vehicle Routing Problem (VRP) merupakan salah satu jenis permasalahan distribusi untuk menentukan suatu himpunan rute-rute kendaraan, dengan setiap kendaraan berangkat dari depo yang sama, untuk melayani beberapa pelanggan, dan kendaraan tersebut kembali ke depo yang sama. Salah satu jenis VRP adalah *Multi Depot Vehicle Routing Problem with Time Window* (MDVRPTW) dimana dalam jenis ini, selain ada *time window* yang merupakan interval waktu untuk dilakukannya pelayanan, tujuan utama dari jenis ini adalah untuk menemukan rute kendaraan agar dapat melayani setiap pelanggan dengan waktu seminimal mungkin. Untuk mendapatkan penjadwalan pengiriman yang handal digunakan metode *Particle Swarm Optimization* (PSO).

Partical Swarm Optimization adalah sebuah algoritma untuk mencari lintasan optimal berdasarkan perilaku sosial dari binatang, seperti sekumpulan burung dalam suatu swarm. PSO merupakan salah satu dari teknik komputasi evolusioner yang mana populasi pada PSO didasarkan pada penelusuran algoritma dan diawali dengan suatu populasi yang random yang disebut *Particle.*[5]

Sisa dari makalah ini dirancang seperti ini: kami memberikan penjelasan tentang deskripsi masalah dan model matematika pada Bagian II. Pada bagian III kami menjelaskan mengenai penyelesaian MDVRPTW. Hasil Simulas dan analisa dijelaskan pada Bagian IV dan Kesimpulan pada Bagian V.

II. DESKRIPSI MASALAH DAN MODEL MATEMATIKA

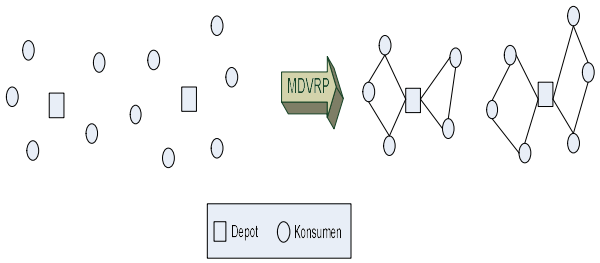
Permasalahan MDVRPTW merupakan permasalahan VRP dengan kondisi dimana depot yang digunakan sebagai pusat distribusi barang bisa lebih dari satu dan distribusi dilakukan dalam *Time Windows* tertentu. *Time Windows* disini maksudnya adalah pelanggan atau konsumen dilayani dalam *range* waktu tertentu sesuai permintaan dari konsumen. Terdapat pula waktu pelayanan yang diperlukan untuk melayani tiap konsumen. Tujuan dari permasalahan MDVRPTW ini adalah mencari sejumlah rute dan *travel time* minimum yang diperlukan untuk mengunjungi suatu himpunan konsumen dimana kendaraan berangkat dan kembali lagi ke depot dan konsumen dilayani tepat satu kali oleh tepat satu kendaraan dengan tidak melanggar kendala kapasitas yang ada.

Berikut ini merupakan formulasi matematis dari MDVRPTW:

Fungsi Tujuan :

Min $\sum_{i=1}^{N+M}\sum_{j=1}^{N+M}\sum_{m=1}^{M}\sum_{k=1}^{Km} t_{i,j}X_{i,j}^{mk} + w_j^{mk}$

Fungsi Kendala :



Gambar 1 MDVRP dengan dua depot dan empat rute

- *Flow Constraint*

a. Setiap *node* hanya dikunjungi sekali saja oleh satu armada pengangkut

$$\sum_{i=0}^{N+M} \sum_{j=1}^{N+M} \sum_{m=1}^{M} \sum_{k=1}^{Km} X_{j,i}^{mk} = 1 \qquad (2)$$

$$\sum_{j=0}^{N+M} \sum_{i=1}^{N+M} \sum_{m=1}^{M} \sum_{k=1}^{Km} X_{i,j}^{mk} = 1 \qquad (3)$$

b. Setiap kendaraan berangkat dari depot dan kembali ke depot.

$$\sum_{k=1}^{Km} \sum_{j=1}^{N} X_{0j}^{mk} = 1 \qquad (4)$$

$$\sum_{k=1}^{Km} \sum_{i=1}^{N} X_{i0}^{mk} = 1 \qquad (5)$$

- *Capacity Constraint*

Total permintaan pada armada pengangkut tidak boleh melebihi kapasitas maksimum

$$\sum_{i=1}^{Km} d_i \sum_{k=1}^{Km} \sum_{j=1}^{N+M} X_{i,j}^{mk} \leq Q \qquad (6)$$

- *Time Window Constraint*

Setiap *node* mempunyai *time window* termasuk depot, setiap kendaraan harus datang tidak melebihi *time window* tiap *node*.

$$x_{ij}(b_i+t_{ij}) \leq l_j \qquad (7)$$

$$b_i = a_i + s_i \qquad (8)$$

$$a_j = max\{e_j, b_i + t_{i,j}\} \qquad (9)$$

$$e_j > b_i + t_{ij} \rightarrow w_j = e_j - (b_i + t_{ij}) \qquad (10)$$

$$e_j > b_i + t_{ij} \rightarrow w_j = 0 \qquad (11)$$

a. Setiap kendaraan mulai service pada time window tiap node

$$e_i \leq a_i \leq l_i \qquad (12)$$

Keterangan:

$N+M$= Himpunan Node Konsumen dan Depot

K_m = Himpunan armada pengangkut
Q = Kapasitas maksimum tiap armada pengangkut
d_i = Jumlah permintaan node konsumen ke i
t_{ij} = Waktu tempuh dari node i ke node j
w_j^{mk} = Waktu tunggu (*waiting time*) armada pengangkut k di node j
$X_{i,j}^{mk}$= 1 : jika armada pengangkut k mengunjungi node j segera setelah node i, $i \neq j$
= 0 : jika lainnya.
b_j = Waktu selesai service di node j
a_i = Waktu mulai service di node i
s_i = Waktu service di node i
e_j = Waktu awal (buka) time window di node j
l_i = Waktu akhir (tutup) time window di node i

Formula (1) berfungsi untuk meminimalkan total waktu tempuh kendaraan. Formula (2) menyatakan jumlah kendaraan yang akan keluar dari depot m K_m. Formula (3) menyatakan bahwa setiap kendaraan berangkat dari depo dan kembali ke depo. Formula (4) dan (5) memastikan bahwa pelanggan dilayani tepat satu kali oleh satu kendaraan. Formula (6) menyatakan kapasitas kendaraan. Formula (7) menyatakan bahwa kendaraan tidak boleh berangkat dari depo menuju depo lainnya. Formula (8) dan (9) untuk memastikan *time windows*nya.

III. PENYELESAIAN MDVRPTW

Pada penyelesaian permasalahan ini secara garis besar akan dibagi keempat bagian seperti berikut:

- Clustering : merupakan tahapan untuk mengumpulkan sejumlah konsumen dengan sebuah depot agar menjadi single Depot VRPTW.
- Assignment : tahapan untuk menentukan urutan kunjungan mulai dari konsumen pertama, kedua, ketiga, hingga konsumen terakhir dalam sebuah gugus
- Routing : dalam sebuah rute pengiriman, terdapat banyak alternative jalan yang dapat ditempuh dari sebuah depot konsumen pertama, atau dari sebuah kosumen ke konsumen selanjutnya, tahapan ini akan menentukan melalui jalan mana pengiriman akan dilakukan agar mencapai rute optimal yakni rute dengan jarak terpendek
- Mapping : agar data hasil optimasi dapat ditampilkan secara menarik dan mudah dipahami oleh pengguna maka dibuatlah rute yang akan menghubungkan antar node(depot-konsumen/konsumen-konsumen) sesuai hasil perhitungan dalam algoritma djikstra pada sebuah peta geografis.

A. Simplified Parallel Assingment [6]

Prosedur Dalam metode ini dinamakan parallel karena urgensi dari masing-masing konsumen diperhitungkan sembari memperhitungkan faktor kedekatan dengan depot. Metode ini membandingkan biaya (*travel time*) antara konsumen dengan depot terdekat dan konsumen dengan depot lainnya. Konsumen yang memiliki prioritas pertama ialah konsumen dengan nilai μ maksimum[8]. Konsumen dengan nilai μ paling tinggi dikelompokkan dengan depot yang terdekat (satu gugus dengan depot terdekat). Agar metode ini dapat bekerja dengan baik

dengan kendala time window, urgensi μ dari masing-masing pelanggan ditentukan melalui persamaan berikut :

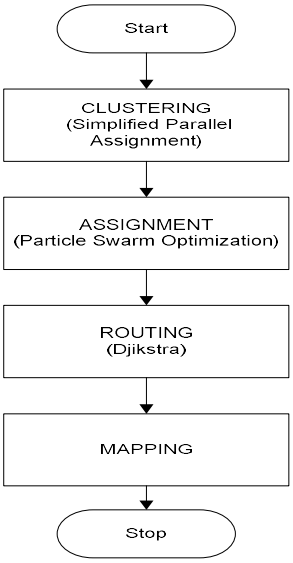
$$\mu_c = Closeness(c, dc'(c)) - Closeness\big(c, dc''(c)\big) \quad (10)$$

Dimana :

μ_c = urgensi masing-masing konsumen

d(c,dep'(c)) = jarak antara konsumen *c* dengan depot terdekat.

d(c,dep''(c)) = jarak antara konsumen *c* dengan depot terdekat nomor dua.



Gambar 2 *Flowchart* pengerjaan MDVRPTW

Closeness dipengaruhi oleh nilai afinitas dan jarak, persamaan *closeness* dicari melalui persamaan :

$$Closeness(i,j) = \frac{d(i,j)}{Afinitas(i,j)} \qquad j \in D, i \in C \quad (11)$$

Sedangkan afinitas(i,j) diperoleh dari persamaan :

$$Afinitas(i,d) = \left\{ \frac{\sum_{j \in C(d) \cup \{d\}} e^{-(DTW(i,j)+TV_{ij})}}{|C|} \right\} d \in D | \, i,j \in C \quad (12)$$

Dimana :

D = himpunan depot dalam MDVRPTW

C = himpunan konsumen dalam MDVRPTW

C(d)=himpunan dari konsumen yang telah dikelompokkan dengan depot d.

TV_{ij} = adalah *travel time* antara i dengan j.

DTW mengukur jarak dalam *time window* dari konsumen dengan konsumen yang lain atau dengan depot :

$$DTW(i,j) = \begin{cases} e_j - l_i \; si \; l_i < e_j \\ e_i - l_j \; si \; l_j < e_i \\ 0, untuk \; yang \; lain \end{cases} \quad (13)$$

Dimana l adalah waktu awal *time window* dan e adalah waktu berakhirnya *time window*.

Apabila perhitungan hanya berdasarkan pada *time window*, idealnya konsumen dikelompokkan dengan depot yang memiliki konsumen dengan kendala *time window* yang berdekatan, dengan cara demikian suatu depot dapat memaksimalkan afinitasnya. Sebaliknya, apabila perhitungan hanya didasarkan pada jarak, seorang konsumen akan dikluster dengan depot terdekat.

B. Particle Swarm Optimization

Partical Swarm Optimization (PSO) adalah salah satu dari teknik komputasi evolusioner, yang mana populasi pada PSO didasarkan pada penelusuran algoritma dan diawali dengan suatu populasi yang random yang disebut *particle*.

Berbeda dengan teknik komputasi evolusioner lainnya, setiap *particle* di dalam PSO juga berhubungan dengan suatu *velocity*. *Particle-particle* tersebut bergerak melalui penelusuran ruang dengan *velocity* yang dinamis yang disesuaikan menurut perilaku historisnya.oleh karena itu, *particle-particle* mempunyai kecenderungan untuk bergerak ke area penelusuran yang lebih baik setelah melewati proses penelusuran.

Pada algoritma PSO vektor *velocity* diupdate untuk masing-masing *particle* kemudian menjumlahkan vektor *velocity* tersebut ke posisi *particle*. Update *velocity* dipengaruhi oleh kedua solusi yaitu *global best* yang berhubungan dengan biaya yang paling rendah yang pernah diperoleh dari suatu *particle* dan solusi *local best* yang berhubungan dengan biaya yang paling rendah pada populasi awal. Jika solusi *local best* mempunyai suatu biaya yang kurang dari biaya solusi global maka solusi *local best* menggantikan solusi *global best.*

Prosedur standar untuk menerapkan algoritma PSO adalah sebagai berikut :

1) Inisialisasi Partikel

Parameter yang dioptimisasi dalam sistem ini adalah travel time yang ter-interkoneksi pada sistem tersebut dengan memperhitungkan *time window.* Sehingga penentuan posisi awal partikel adalah nilai *travel time* yang ditentukan secara acak.

Nilai *travel time* tersebut, pada iterasi ke-0, akan dibagikan ke setiap partikel. Proses pembagian nilai pada setiap partikel ada sama dengan metode inti dari PSO, dengan menggunakan pBest dan gBest yang didapat setelah perhitungan *travel time* setiap partikel. Sehingga pada iterasi ke-0, nilai posisi partikel dalam proses inisialisasi ini bernilai total *travel time* setiap partikel.

2) Fitness Partikel

Pada fase ini sudah memasuki tahap iterasi ke-1 hingga iterasi maksimum. Pada tiap iterasi, nilai fitness partikel sama dengan nilai dari total *travel time*. Namun fungsi evaluasi objektif atau *fitness* ini ditujukan untuk perbandingan nilai di setiap iterasi. Nilai *fitness partikel* disetiap iterasi berbeda-beda. Nilainya akan berubah dengan nilai total *travel time* pada iterasi ke-i, apabila total *travel time* lebih optimal dari nilai iterasi sebelumnya maka akan dipakai nilai total *travel time* yang baru.

Dan akan bernilai tetap apabila total *travel time* tidak seoptimal nilai total *travel time* sebelumnya.

3) Menentukan Nilai pBest dan gBest

pBest adalah nilai posisi partikel lokal, penentuan pBest pada iterasi ke-1 hingga iterasi maksimum adalah nilai total *travel time* di setiap partikel. Nilai pBest didapat setelah melalui fase *fitness particle.* Nilai pBest bergantung pada nilai *fitness particle,* sehingga berubah nilainya setiap iterasinya. Sedangkan nilai gBest adalah yang terbaik dari setiap nilai pBest. Dari nilai pBest, menentukan nilai yang paling optimal, pada penelitian ini adalah pencarian nilai minimum *travel time.*

4) *Updating* Kecepatan

Pada tahap ini, hitung kecepatan dari semua partikel. Semua partikel bergerak menuju titik optimal dengan suatu kecepatan. Awalnya semua kecepatan dari partikel diasumsikan sama dengan nol. Set iterasi i=1.

$$V_j(i) = V_j(i\text{-}1) + c_1 r_i\left[pBest\text{-}x_j(i\text{-}1)\right] + c_2 r_2\left[gBest\text{-}\ x_j(i\text{-}1)\right], j = 1,2, \dots, N \quad (14)$$

Dimana c1 dan c2 masing-masing adalah *learning rates* untuk kemampuan individu dan pengaruh social, dan r1 dan r2 bilangan random yang terdistribusi uniformal dalam interval 0-1. Jadi parameters c1 dan c2 menunjukkan bobot dari memory (*position)* sebuah partikel terhadap memori dari kelompok (*swarm*). Nilai dari c1 dan c2 biasanya adalah 2 sehingga perkalian c1r1 dan c2r2 memastikan bahwa partikel-partikel akan mendekati target sekitar setengah selisihnya.

5) *Updating* Posisi

Hitung posisi pada partikel dengan cara

$$X_j(i) = X_j(i-1) + V_j(i); j = 1,2, \dots N \quad (15)$$

Cek apakah solusi yang sekarang sudah konvergen. Jika posisi semua partikel menuju ke satu nilai yang sama, maka ini disebut konvergen. Jika belum konvergen maka dilakukan *updating* kecepatan dengan memperbaharui iterasi i=i+1, dengan cara menghitung nail baru dari pBest dang Best. Proses iterasi ini dilanjutkan sampai semua partikel menuju satu titik solusi yang sama.

C. Algoritma Dijkstra

Hasil dari algoritma ini adalah jarak terpendek dari *node* sumber ke *node* tujuan beserta rutenya.

IV. HASIL SIMULASI DAN ANALISA

Untuk simulasi pada program ini menggunakan *software* MATLAB. Sedangkan plant yang digunakan pada penelitian ini adalah perusahaan air minum dalam kemasan "Cleo Pure Water Surabaya" dengan konsumen yang diasumsikan. Selanjutnya simulasi ini akan diterapkan pada data jaringan jalan di Surabaya yang sesuai dengan penelitian sebelumnya [10].

A. Pemilihan Nilai Parameter PSO

Pengujian dilakukan dengan menggunakan 50 node konsumen, dan jumlah iterasi maksimal 20, sedangkan untuk nilai c1, c2, dan swarm bervariasi. Tujuan pengujian yang dilakukan adalah untuk mengetahui pengaruh terhadap rute optimum dengan adanya variasi dari parameter-parameter Ant Colony System. Nilai yang dibandingkan dari pengujian ini adalah total *travel time* dan jumlah kendaraan akhir.

```
algorithm Dijkstra;
begin
    S : = ⦰;
    d(i): = ∞ for each node i∈N;
    d(s): = 0 and pred(s): = 0
    while |S| < n do
    begin
        let i∈  be a node for which d(i) =
min{d(j) : j∈  };
        S: = S    {i};
         : =   – {i};
        for each (i,j) ϵ A(i) do
            if d(j) >d(i) + cij then d(j): = d(i)
+ cij and pred (j): = i;
    end;
end;
```

Pengujian yang dilakukan dengan menentukan nilai c1 = 1, c2 = 1, iterasi maksimal = 20, dengan nilai *swarm* yang divariasikan mulai dari 20, 35, 50 untuk mencari nilai *swarm* yang terbaik. Pengujian selanjutnya dilakukan dengan cara yang sama sesuai dengan parameter uji yang dilakukan dimana parameter uji divariasikan dan parameter lainnya konstan. (bisa dilihat pada lampiran)

B. Hasil Simulasi

1. Hasil Clustering

Setelah memasukkan data depot dan data konsumen, proses berikutnya ialah melakukan *clustering*. Dari metode *simplified parallel assignment* yang digunakan dalam program ini akan menghasilkan hasil *clustering* sebagai berikut :

- Depot 1 (22 konsumen):
  189 174 113 46 62 60 141 81 44 39 28 140 101 4 21 254 12 7 235 144 1 104
- Depot 2 (13 konsumen):
  408 95 399 296 135 292 242 205 244 161 181 175 106
- Depot 3 (15 konsumen) :
  210 386 248 289 298 264 221 355 256 315 402 400 333 177 376

2. Hasil Optimasi (depot 2-kendaraan 1)

a) Posisi node = 'Kramat Gantung' 'Jaksa Agung Suprapto' 'Simorejo II' 'Pecindilan' 'Gang Besar' 'Pacuan Kuda' 'Indrapura' 'Semarang' 'Tidar' 'Anjasmoro' 'Kali Butuh' 'Petemon Kali' 'Petemon Barat' 'Simorejo II'
b) *Travel time* = 32 menit 8 detik
c) Jumlah akhir kendaraan = 3

- Rute Kendaraan 1 = 277 95 135 277

Jika rute tersebut diubah sesuai kondisi Surabaya menjadi :

277 → 267 → 260 → 206 → 191 → 161 → 148 → 130 → 120 → 98 → 95 → 98 → 120 → 139 → 136 → 135 → 136 → 139 → 142 → 146 → 148 → 161 → 191→ 206 407 → 224 → 295 → 277

Dan jika dinyatakan dalam nama jalannya adalah sebagai berikut:

(DEPOT 277) 'Baliwerti' → 'Praban' 'Kranggan' → 'Kali Butuh' → 'Kali Butuh' → 'Kali Butuh' → 'Kali Butuh' → 'Asem Raya' → 'Asem Raya' → 'Dupak Rukun' → 'Dupak Rukun' →(95) 'Asem Raya' → 'Asem Raya' → 'Kali Butuh' → 'Tembok Buntaran' 'Tidar' →(135) 'Pacuan Kuda' → 'Pacuan Kuda' → 'Pacuan Kuda' 'Pacuan Kuda' → 'Pacuan Kuda' → 'Pacuan Kuda' → 'Tidar' 'Tembok Buntaran' → 'Kali Butuh' → 'Kali Butuh' → 'Kali Butuh' → 'Semarang' → 'Semarang' → 'Semarang' → 'Semarang' → 'Semarang' → 'Tembaan' → 'Tembaan' →(DEPOT 277) 'Pahlawan'

V. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan dapat diambil kesimpulan sebagai berikut:

1) *Clustering* pada Multi Depot Vehicle Routing Problem dapat menggunakan metode Simplified Parallel Assignment dengan mempertimbangkan jarak konsumen ke depot dan time window masing-masing depot dan konsumen.
2) Penggunaan parameter *particle swarm optimization* yang sesuai akan menghasilkan solusi dengan travel time paling minimum

DAFTAR PUSTAKA

[1] Zhu,Q., Qian, L., Li, Y., Zhu, S., "An Improved Particle Swarm Optimization Algorithm for Vehicle Routing Problem with Time Windows", Beijing, China, 2006.
[2] Wen,L., Meng, F., "An Improved PSO for Multi Depot Vehicle Routing Problem with Time Windows", Hebei, China, 2008.
[3] Ravindra, K.A., Magnati, T.L., and Orlin, J.B. , "Network Flows-Theory,Algorithms, and Application", Prentice Hall, 1993, New Jersey, USA
[4] _____, "Comput. & Ops Res Vol.10. No.2, pp.63-211,1983, Great Britain.
[5] Tansini, L., Urquhart, M., Viera, O.,"Comparing assignment algorithms for the Multi-Depot VRP",2007,Uruguay.
[6] Alam, Akhmad Fajar Nurul , "Algoritma Improved Ant Colony System Untuk Menyelesaikan Dynamic Vehicle Routing Problem With Time Window dengan Variabel Travel Time", 2011,Surabaya.