JURNAL UPI

Jurnal EurekaMatika

Journal homepage: https://ejournal.upi.edu/index.php/JEM

Penyelesaian *Multi Depot Vehicle Routing Problem with Time Windows* Menggunakan *Particle Swarm Optimization Algorithm*

Khairunnisa Aulia Azzahra, Khusnul Novianingsih, dan Dewi Rachmatin*

Program Studi Matematika, Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia

*Correspondence: E-mail: k_novianingsih@upi.edu

A B S T R A K

Penelitian ini membahas *Multi Depot Vehicle Routing Problem with Time Windows* (MDVRPTW), masalah penentuan rute kendaraan dari sejumlah depot ke beberapa pelanggan dengan mempertimbangkan batasan *time windows* dalam setiap rutenya. Tujuan penyelesaian MDVRPTW adalah mendapatkan rute optimal dengan total *travel time* terkecil dan tidak melebihi *time windows*-nya. Algoritma Particle Swarm Optimization (PSO) digunakan untuk menyelesaikan MDVRPTW. Cara kerja PSO diadaptasi dari perilaku sosial dari sekawanan burung dalam mencari makan. Algoritma ini bekerja dengan cara melakukan inisialisasi, mengevaluasi, mengonstruksi rute, dan memperbaharui rute hingga optimal. Penelitian diuji pada studi kasus pengambilan bahan baku suatu perusahaan dengan 2 depot penyimpanan dan 169 agen. Implementasi PSO berhasil membentuk rata-rata *travel time* setiap rute adalah 7,83 jam yang artinya time windows tidak dilanggar dan kapasitas kendaraan terpenuhi.

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima 26 Februari 2024
Direvisi 8 April 2024
Disetujui 25 April 2024
Tersedia online 28 April 2024
Dipublikasikan 2 Mei 2024

Kata Kunci:
Multi Depot Vehicle Routing Problem,
Particle Swarm Optimization,
Rute,
Time Windows

A B S T R A C T

This research addresses the Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW), the problem of determining vehicle routes from several depots to multiple customers while considering time window constraints for each route. The goal of solving MDVRPTW is to obtain optimal routes with the shortest total travel time without exceeding their respective time windows. The Particle Swarm Optimization (PSO) algorithm is used to solve MDVRPTW, adapted from the social behavior of a flock of birds in search of food. The algorithm operates through initialization, evaluation, route construction, and route updates to achieve optimality. The research was tested on a case study involving raw material pickup for a company with 2 storage depots and 169 agents. The implementation of PSO successfully generated an average travel time of 7.83 hours for each route, indicating adherence to time windows and fulfillment of vehicle capacity.

Keywords:
Multi Depot Vehicle Routing Problem,
Particle Swarm Optimization,
Route,
Time Windows

1. PENDAHULUAN

Vehicle Routing Problem (VRP) merupakan salah satu permasalahan distribusi untuk menentukan rute optimal dari sejumlah kendaraan yang mengunjungi sejumlah tempat untuk mengantar atau menjemput barang. Titik awal dari perjalanan tersebut disebut dengan depot. Rute optimal harus berawal dan berakhir di depot yang sama. Setiap kendaraan memiliki kapasitas angkut, dan setiap pelanggan memiliki permintaan yang harus dipenuhi, dengan jumlah kunjungan tepat satu kali dan total permintaan tidak boleh melebihi kapasitas angkut dari kendaraan tersebut (Yeun *et al.,* 2008).

Dalam kehidupan sehari-hari, masalah VRP memiliki banyak perluasan kendala. Salah satunya adalah MDVRPTW di mana memiliki kendala tambahan berupa depot yang lebih dari satu dan terikat dengan *time windows*. Tujuan penyelesaian dari MDVRPTW ini adalah untuk menentukan himpunan rute perjalanan beberapa kendaraan dari beberapa depot untuk melayani ke beberapa pelanggan dengan mempertimbangkan total *travel time* dan waktu pelayanan pelanggan yang terbatas sekaligus tidak boleh melebihi kapasitas kendaraan yang digunakan (Bae & Moon, 2016). Sejauh ini, terdapat beberapa penelitian terkait penyelesaian MDVRPTW seperti menggunakan Algoritma *Simulated Annealing* oleh Sinaga (2015), Algoritma *Ant Colony System* oleh Nugroho (2015), dan Algoritma *Adaptive Genetic* oleh Fazarudin *et al.* (2015). Jauh sebelum Bae & Moon memasukkan *time windows* pada tahun 2016, Crevier *et al.* (2007) meneliti *The multi-depot vehicle routing problem* namun dengan pendekatan route *inter* depot. Bai, Q. (2010) memperbaiki algoritma PSO dengan memperhatikan bobot inersia, menambah faktor konvergen, pemilihan dan membaurkan algoritma PSO dengan algoritma-algoritma intelijen lainnya.

Metode lain yang akan digunakan untuk menyelesaikan permasalahan ini adalah Algoritma *Particle Swarm Optimization* (PSO). Algoritma ini terinspirasi terhadap perilaku sosial burung dalam mencari makanan. Perilaku sosial dari tindakan tiap individu dalam sekawanannya yang memberikan pengaruh terhadap kawanannya dikenalkan pertama kali oleh Kennedy & Eberhart (1995) melalui presentasinya pada *the IEEE international conference on neural networks* dengan judul pada Particle swarm optimization?. Algoritma PSO berbeda dengan teknik komputasi lainnya, di mana setiap partikel di dalam PSO berhubungan dengan suatu *velocity*. Partikel-partikel tersebut akan bergerak melalui penelusuran ruang dengan *velocity* yang dinamis dan disesuaikan menurut perilaku historisnya. Oleh sebab itu partikel-partikel tersebut memiliki kecenderungan untuk bergerak dan mengubah pola sesuai dengan pengalaman belajarnya sendiri dan dari anggola lain (Wang *et al.*, 2018). Venter & Sobieszczanski-Sobieski (2003) meningkatkan optimasi PSO dengan memodifikasi kriteria kekonvergenan, optimisasi konstrain, perhatian khusus pada partikel-partikel dengan konstrain *violated*, serta melakukan observasi pada variabel disain diskret/bilangan bulat. Pada penelitian yang telah dilakukan Nugroho (2015), Algoritma PSO menghasilkan travel time terpendek dibandingkan dengan Algoritma *Ant Colony System* dan Algoritma *Simulated Annealing*. Pada artikel ini akan dibahas cara kerja algoritma PSO dalam menyelesaikan MDVRPTW, serta implementasi algoritma PSO pada kasus pengambilan bahan baku oleh salah satu perusahaan di Bandung.

2. METODE

Pada bagian ini dibahas mengenai model MDVRPTW dan teknik penyelesaian MDVRPTW dengan menggunakan Algoritma PSO.

2.1 Model MDVRPTW

Asumsi-asumsi yang akan digunakan dalam penelitian ini adalah sebagai berikut :

- Setiap rute memiliki *time windows* yang berlaku yaitu 10 jam, dimulai dari jam 7.00 WIB hingga 17.00 WIB
- Setiap pelanggan memiliki batasan waktu pelayanan yaitu 30 menit.
- Setiap jarak jalan yang terbentuk dapat berlaku 2 arah.

Untuk menurunkan model MDVRPTW, terlebih dahulu akan didefinisikan himpunan-himpunan yang digunakan oleh model sebagai berikut:

- $\boldsymbol{V}$ adalah himpunan gabungan dari pelanggan dan depot
- $\boldsymbol{N}$ adalah himpunan pelanggan
- $\boldsymbol{M}$ adalah himpunan depot
- $\boldsymbol{K}$ adalah himpunan kendaraan pengangkut

Adapun variabel keputusan pada model didefinisikan sebagai berikut :

$$x_{ijk} = \begin{cases} 1, \text{jika kendaraan } k \text{ mengunjungi pelanggan } j \text{ setelah pelanggan } i \\ 0, \text{yang lainnya.} \end{cases}$$

Pada permasalahan MDVRPTW akan dicari rute dengan total *travel time* terkecil dari beberapa kendaraan yang berangkat dari suatu depot menuju beberapa pelanggan dan kembali ke depot yang sama dengan mempertimbangkan *time windows* dan kapasitas kendaraan yang terbatas. Oleh karena itu fungsi tujuan dari MDVRPTW dapat direpresentasikan sebagai berikut (Bae *et al.*, 2016):

$$\sum_{k\in K}\sum_{i\in V}\sum_{j\in V} x_{ijk} \,.\, t_{ij} + s_{jk}$$

Adapun kendala dari model adalah sebagai berikut (Bae *et al.*, 2016) :

1) Setiap pelanggan dikunjungi sekali oleh kendaraan pengangkut, dan direpresentasikan oleh persamaan:

$$\sum_{k\in K}\sum_{i\in V} x_{ijk} = 1, \forall j \in N$$

$$\sum_{k\in K}\sum_{j\in V} x_{ijk} = 1, \forall i \in N$$

2) Setiap kendaraan berangkat dari depot dan kembali ke depot yang sama sebanyak $\boldsymbol{K}$, dan direpresentasikan oleh persamaan:

$$\sum_{j\in N} x_{ijk} \leq 1, \forall k \in K, \forall i \in M$$

$$\sum_{i\in N} x_{ijk} \leq 1, \forall k \in K, \forall j \in M$$

3) Total bahan yang dibawa oleh kendaraan pengangkut tidak melebihi kapasitas $\boldsymbol{Q}$, dan direpresentasikan oleh persamaan:

$$\sum_{i \in N} \sum_{j \in V} d_i x_{ijk} \leq Q, \forall k \in K$$

4) Setiap kendaraan harus patuh pada kendala time windows yang telah ditetapkan pada setiap rute, dan direpresentasikan oleh persamaan :

$$a_j \geq a_i + s_{ik} + t_{ij} + \left(\sum_{k \in K} x_{ijk} - 1 \right), \forall i \in V, j \in N$$

$$t_k \geq a_j \geq a_i + s_{ik} + t_{ij} + \left(x_{ijk} - 1 \right), \forall i \in N, j \in M, k \in K$$

5) Setiap kendaraan akan memulai pelayanan pada time window di setiap lokasi pelanggan, dan direpresentasikan oleh persamaan :

$$e_i \leq a_i + s_{ik} \leq l_i, \forall i \in N$$

2.2 Teknik Penyelesaian

Terdapat dua tahapan untuk menyelesaikan MDVRPTW, yaitu *Classification* kemudian dilanjut dengan *Optimization*. Tahap pertama adalah dengan melakukan klasifikasi untuk mengelompokkan sejumlah agen menjadi *single depot* VRPTW. Pertama-tama dibentuk matriks jarak menggunakan rumus *Haversine*:

$$d = 2r.arcsin\left\{ \sqrt{sin^2\left(\frac{Lat_1 - Lat_2}{2}\right) + \cos(Lat_2) \cdot \cos(Lat_2) \cdot \sin^2\left(\frac{Long_1 - Long_2}{2}\right)} \right\}$$

dengan *r* merupakan jari-jari bumi yaitu 6.371 km. Tahap selanjutnya adalah mengelompokkan agen ke depot terdekat sehingga depot tersebut dapat memaksimalkan operasionalnya. Dengan menggunakan aturan berikut (Fitriana *et al*., 2019):

- Jika $d_{nd_i} < d_{nd_2}$, maka pelanggan n dikelompokkan dengan depot 1.
- Jika $d_{nd_i} > d_{nd_2}$, maka pelanggan n dikelompokkan dengan depot 2.
- Jika $d_{nd_i} = d_{nd_2}$, maka pelanggan n bebas dikelompokkan dengan depot 1 maupun depot 2.

Setelah gugus agen dengan depot terbentuk akan dilakukan *Optimization* untuk setiap gugus depot. Optimisasi ini dilakukan untuk mengetahui urutan kunjungan kendaraan menuju agen pertama, kedua, hingga agen terakhir dalam setiap gugusnya. Metode yang digunakan adalah Algoritma *Particle Swarm Optimization* dengan langkah-langkah berikut (Shami *et al*., 2022):

1) Inisialisasi Partikel Awal

Penentuan posisi awal partikel adalah nilai *travel* time yang ditentukan secara acak, sehingga akan dibentuk posisi awla dengan cara memanggil secara random antara [0,1] dengan banyak baris sebanyak node dalam rute, dan baris sebanyak *swarm*. Dengan terbentunya posisi awal tersebut, selanjutnya akan diurutkan pada setiap *swarm* partikel dari yang paling kecil ke yang paling besar. Sehingga akan terbentuk rute dari posisi bilangan random awal sebelum diurutkan (Gamayanti *et al.*, 2015).

2) *Fitness* Partikel

Nilai *fitness* yang digunakan dalam hal ini adalah *travel time* dari masing-masing rute yang terbentuk. Dengan menggunakan rumus :

$$T_1 = t_1 + a_1$$

di mana T_1 adalah total *travel time swarm* pertama, t_1 adalah total waktu tempuh dari *swarm* pertama dengan membagi total jarak dengan kecepatan rata-rata kendaraan

pengankut, dan a_1 adalah total waktu pelayanan dari setiap agen pada *swarm* pertama (Gamayanti *et al.*, 2015).

3) Evaluasi nilai pBest dan gBest

Nilai pBest adalah nilai posisi partikel local. Penentuan pBest pada iterasi pertama adalah dengan menggunakan nilai *fitness swarm* pertama, lalu pada iterasi berikutnya dengan menggunakan nilai *fitness* terbaik dari iterasinya. Sedangkan gBest adalah nilai partikel *global*, artinya adalah nilai *fitness* terbaik dari semua iterasi yang dilakukan, sehingga nilai gBest dapat berubah jika nilai pada iterasi terkini lebih optimal daripada iterasi sebelumnya (Gamayanti *et al.*, 2015).

4) *Update* Kecepatan Partikel (V)

Semua partikel bergerak menuju titik optimal dengan suatu kecepatan. Kecepatan awal dari semua partikel diasumsikan dengan suatu nilai pada iterasi pertama. Selanjutnya pada iterasi selanjutnya dihitung dengan rumus :

$$V_1(C_1) = \omega V_0 + c_1 r_1(p_{best} - C_1) + c_2 r_2(g_{1best} - C_1)$$

dengan $j = 1,2,3, \dots N$. c_1 dan c_2 adalah *learning rates* untuk kempuan individu dan pengaruh social. r_1 dan r_2 adalah bilangan acak yang terdistribusi seragam dalam interval 0-1. Dan ω adalah *constant inertia weight* (Gamayanti *et al*., 2015).

5) *Update* Posisi Partikel

Posisi pada partikel akan dihitung dengan :

$$C_1' = C_1 + V_1$$

di mana C_1 adalah nilai random awal yang belum diurutkan. Selanjutnya hasil C_1' akan dibagi dengan jumlah total dari C_1' pada setiap partikelnya. Perhitungan ini dilakukan pada setiap *swarm*. Selanjutnya adalah melakukan pengecekan apakah solusi terkini sudah konvergen, yaitu ketika semua nilai partikel menuju ke satu nilai yang sama. Jika belum konvergen maka dilakukan *updating* kecepatan dengan memperbaharui iterasi $i = i + 1$, dengan menghitung nilai baru dari pBest dan gBest. Proses ini akan terus berlanjut hingga semua partikel menuju titik solusi yang sama (Gamayanti et al., 2015).

3. HASIL DAN PEMBAHASAN

Pada bagian ini algoritma *Particle Swarm Optimization* akan diimplementasikan pada masalah pengambilan bahan baku dari suatu perusahaan di Bandung. Perusahaan tersebut memiliki 2 gudang penyimpanan dengan 169 agen yang harus dikunjungi untuk mengambil bahan baku. Kendaraan pengangkut dengan kapasitas angkut 10.000 liter sebanyak 15 buah berkendara dengan rata-rata kecepatan kendaraan 25 km/jam. Dengan batas waktu operasional dimulai dari pukul 7.00 WIB sampai 17.00 WIB atau 10 jam dan batas waktu pelayanan adalah 30 menit pada setiap agen.

Langkah pertama yang dilakukan adalah dengan membagi 169 agen menjadi 62 agen untuk Depot Pertama dan 107 agen untuk Depot Kedua dengan menggunakan parameter $r_1 = 0{,}2$, $r_2 = 0{,}4$, c_1 & $c_2 = 2$, dan $\omega = 0{,}9$, dan menggunakan kecepatan awal partikel 0,1. Setelah dilakukan perhitungan menggunakan *software python* sebanyak 62 iterasi untuk Depot Pertama dan sebanyak 107 iterasi untuk Depot Kedua diperoleh hasil rute pada Tabel 3.1 dan Tabel 3.2.

Tabel 3. 1 Hasil Implementasi Algoritma PSO - Depot Pertama

Vehicle	***Route***	***Demand* (Liter)**	***Travel Time* (Jam)**
1	Depot KPSBU → Ciburial → Gombong → Nyampai → Bale → Caringin → Pentas → Lapang → Ciputri → H. Odih → Depot KPSBU	9999	5,37
2	Depot KPSBU → Gunung Putri → Cibogo → Nyalindung → Cikareumbi →Babakan → Legok → Cireyod 2 → Pagerwangi → Pagermaneuh → PKS → Benteng → Depot KPSBU	9976	6,31
3	Depot KPSBU → Gunung Payung → Pak Katimin → Pencut → Pasir Wangi → Areng → Cibulakan → Bukamanah → Suka Muti 1 → Cilanguk 1 → Suka Mukti 2 → Bapak Ekon → Cikadu → Pabrik Gitar → Ibu Endah → Depot KPSBU	5524	8,64
4	Depot KPSBU → Pak Suhada → Cigalukguk → Cijero Kaso → Areng → Asmara → Pasing Angling Bawah → Cikapundung → Pasing Angling → Sukmana → Gandok → Babakan → Legok Barong → Dago → Sukamaju → Cece → Depot KPSBU	4313	8,96
5	Depot KPSBU → Penampungan Odih → Cireyod 1 → Bunisari → Ibu Yeti → Bukatanah → Batu Lonceng Kulon → Sukatinggal → Batu Lonceng Wetan → Gunung Batu → Patrol → Bukanagara → Cinungku → Depot KPSBU	4146	9,45

Tabel 3. 2 Hasil Implementasi Algoritma PSO - Depot Kedua

Vehicle	***Route***	***Demand* (Liter)**	***Travel Time* (Jam)**
6	Depot Pojok → The Pic → Baru Ahad → Cibadak → P. Caringin → Pak Kurnia → Pak Oom → Pak Sumpena → Pak Udin → Cileweng → KHS 08 → Undang → Citalio → Udin → Wawan → Jompo → Depot Pojok	3532	8,84
7	Depot Pojok → Monoko → Babakan → H. Hamzah → Cipariuk → Kancah 2 → Bongkor → Depot Pojok	2487	4,02

Vehicle	*Route*	*Demand* (Liter)	*Travel Time* (Jam)
8	Depot Pojok → Eutik Cahya → Sukadami → Kancah 3 → Dano → Kancah → Sukawana → Panaruban → Ciater → Cigeureung → Jagar Naek → Cisaat → Palasari → Cibeureum → Genteng → Cisaroni Atas → Depok Pojok	8908	9,24
9	Depok Pojok → Cibodas → Pasir Atas → Banuragri 1 → Kancah 4 → Karyawangi 1 → Kancah 1 → Barunagri 2 → Suka Mekar → Pasir Bawah → Citespong → Depot Pojok	7690	5,81
10	Depot Pojok → Joro → Tugu 2 → Cijanggel → Jajang → B. Lele → Pak Udin → Tatang → Herman → Mekarwangi → Bos Entang → Adang → Ayi S. → Gunung Masagit → Penampungan Bu Ai → Cipariuk → Depot Pojok	4967	8,83
11	Depot Pojok → Barunagri 3 → Adetex → Pancuran 12 → Joro → Panyairan → Bbk Karyawangin → Ganda (Cipeusing) → Pak Ade Carna → Ayi bing Abing → Ai Aning → Kampung Baru → Cu Nagrak → Pojok Tengah → Pasir Handap → Cisaroni Bawah → Depot Pojok	9773	8,65
12	Depot Pojok → Nyampai → Mokla → Ajang S (Cipeusing) → Pak Ukat → Wisto → Ikin → Ence → Dase → Ikin → Centris → Ari → PMK Specta → Cisarua → Tutugan → Kebon Hui → Depot Pojok	3231	8,78
13	Depok Pojok → Karyawangi 2 → Tugu 1 → PMK Jagal → Cahya Diana → Taufik → Dede → Eman → Pak Rodiat → Cipanas → Iis Jaja → Jajang W → Sopian → Edi → Atim → Undang → Depot Pojok	2921	8,95

Selanjutnya dilakukan analisis parameter PSO untuk mengetahui pengaruh dari parameter-parameter tersebut terhadap solusi MDVRPTW. Parameter yang dianalisis terdiri dari *learning rates* (c_1, c_2), *inertia weight* (ω), kecepatan awal partikel (V_0), jumlah *swarm*, dan jumlah iterasi. Berdasarkan hasil uji parameter di atas maka nilai-nilai parameter yang direkomendasikan untuk menyelesaikan MDVRPTW pada Depot 1 adalah jumlah *swarm* = 62 buah, *inertia weight* = 0.5, dan *learning rate* = 1,1. Sedangkan nilai-nilai parameter yang direkomendasikan untuk menyelesaikan MDVRPTW pada Depot 2 adalah jumlah *swarm* = 107 buah, *inertia weight* = 0.9, dan *learning rate* = 1,1.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan tentang model *Multi Depot Vehicle Routing Problem with Time Windows* (MDVRPTW) dan implementasi Algoritma *Particle Swarm Optimization* (PSO) dapat diambil kesimpulan bahwa MDVRPTW dapat diselesaikan menggunakan Algoritma PSO dengan fungsi tujuan meminimumkan total waktu tempuh setiap kendaraan. Algoritma PSO bekerja dengan cara mengklasifikasikan pelanggan ke setiap depot, kemudian melakukan optimisasi dengan inisialisasi partikel, evaluasi nilai fitness, melakukan pembaharuan partikel hingga diperoleh solusi rute yang optimal.

Algoritma Particle Swarm Optimization (PSO) berhasil diimplementasikan untuk menyelesaikan masalah penentuan rute pengambilan bahan baku susu di salah satu KPSBU di Lembang. Hasil optimisasi berhasil membentuk 5 rute optimal untuk Depot Pertama dan 8 rute optimal untuk Depot kedua dengan total travel time minimum, time windows yang ditetapkan tidak dilanggar, dan kapasitas kendaraan terpenuhi. Hasil ini menggambarkan efektivitas Algoritma PSO dalam menyelesaikan permasalahan MDVRPTW dan memberikan kontribusi pada perencanaan rute pengiriman yang efisien dan optimal.

5. DAFTAR PUSTAKA

Bae, H., & Moon, I. (2016). Multi-depot vehicle routing problem with time windows considering delivery and installation vehicles. *Applied Mathematical Modelling*, *40*(13-14), 6536-6549.

Bae, H., Moon, I., & Yun, W. (2016). A Time-Varying Lot Sizes Approach for The Economic Lot Scheduling Problem with Return. *International Journal of Production Research*. 54(11). p.3380-3396.

Bai, Q. (2010). Analysis of Particle Swarm Optimization Algorithm. *Computer and Information Science (3:1).* p.180.

Crevier, B., Cordeau, J. F., & Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European journal of operational research*, *176*(2), p.756-773.

Fitriana, R., & Moengin, P., & Kusumaningrum, U. (2019). Improvement Route for Distribution Solutions MDVRP (Multi Depot Vehicle Routing Problem) using Genetic Algorithm. *IOP Conference Series Materials Science and Engineering*. 528(1). p.12–42. IOP Publishing.

Gamayanti, N., Alkaff, A., Mangatas, R. (2015). Optimisasi Multi Depot Vehicle Routing Problem (MDVRP) dengan Variabel Travel Time Menggunakan Algoritma Particle Swarm Optimization. *JAVA Journal of Electrical and Electronics Engineering*. 13(1). Surabaya

Kennedy J, Eberhart RC (1995) Particle swarm optimization?. *Proceedings of the IEEE international conference on neural networks*. p. 1942–1948.

Shami, T.M., El-Saleh, A.A., Alswaitti, M., Al-Tashi, Q., Summakieh, M.A., & Mirjalili, S. (2022). Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access*. 10. p. 10031-10061.

Venter, G., & Sobieszczanski-Sobieski, J. (2003). Particle swarm optimization. *AIAA journal*, *41*(8), 1583-1589.

Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: an overview. *Soft computing*, *22*, p. 387-408.

Yeun, L.C., Ismail, W.R., Omar, K., & Zirour, M. (2008). Vehicle routing problem: models and solutions. *Journal of Quality Measurement and Analysis.*