

Penerapan *Seagulls Optimization Algorithm* untuk Menyelesaikan *Open Vehicle Routing Problem*

Laula Ika Setya Rahman¹, Asri Bekti Pratiwi^{1,*} & Herry Suprajitno¹

¹Departemen Matematika, Fakultas Sains dan Teknologi, Universitas Airlangga

*Corresponding author: asri.bekti@fst.unair.ac.id

Abstract. This paper aims to solve the problem of Open Vehicle Routing Problem using Seagulls Optimization Algorithm. Open Vehicle Routing Problem (OVRP) is a variation of Vehicle Routing Problem (VRP) which will not return to the depot after visiting the last customer, is different from VRP which requires the vehicle to return to the depot because the company have insufficient number of vehicles for the distribution of products to customers so they must to rent vehicles and this OVRP aims to minimize the total cost of distributing products with the shortest optimal distance to meet the demands of each customer with private vehicles and rental vehicles. Seagulls Optimization Algorithm (SOA) is the algorithm inspired by the behaviour of seagulls in migrating and ways of attacking the pray of seagulls in nature. In general, the process begins with generating the initial position, evaluating the objective function, the migration process, the attacking process to get a new position, compare the objective function for the new position and the old position, update the position and save the best seagulls in each iteration until the maximum iteration is met. The program used to complete OVRP with Seagulls Optimization Algorithm is Borland C++ and implemented using 3 case examples, small data with 18 customers, medium data 50 customers and large data 100 customers. Based on the implementation results, it can be concluded that the higher number of seagulls, iterations and the smaller the control variable value tend to effect minimum cost gained.

Keywords: *Open Vehicle, Routing Problem, Seagulls Optimization Algorithm.*

1 Pendahuluan

Dalam proses pendistribusian, kapasitas kendaraan pengangkut juga harus memperhatikan suatu permintaan dari konsumen. Penentuan rute dengan mempertimbangkan kapasitas kendaraan pengangkut dan jarak tempuh maksimal dari kendaraan disebut *Vehicle Routing Problem* (VRP). VRP merupakan permasalahan distribusi yang mencari serangkaian rute untuk sejumlah kendaraan dengan kapasitas tertentu dari satu atau lebih depot untuk melayani konsumen mengemukakan tujuan yang ingin dicapai dalam VRP yakni meminimalkan ongkos perjalanan secara keseluruhan, meminimalkan jumlah kendaraan yang digunakan, menyeimbangkan rute, dan meminimalkan keluhan pelanggan [1].

Dalam hal distribusi, tidak semua perusahaan memiliki kendaraan pribadi untuk mengirimkan produknya kepada konsumen. Hal ini menyebabkan perusahaan harus

menyewa beberapa kendaraan untuk mengirimkan produknya kepada konsumen tanpa peduli apakah kendaraan kembali ke depot atau tidak dan tanpa membayar biaya perjalanan antara pelanggan terakhir dan depot. Untuk memecahkan permasalahan tersebut maka digunakan *Open Vehicle Routing Problem* (OVRP) [2]. Dalam menyelesaikan OVRP terdapat beberapa algoritma yang pernah digunakan, salah satunya yaitu metode heuristik [3], *Tabu Search Algorithm* [4], *Particle Swarm Optimization* [5], *Hybrid Genetic Algorithm with Tabu Search Algorithm* [6], *A Multi-start Algorithm* [7], *Simulated Annealing Heuristic* [8], *Ant Colony Optimization* [9].

Seagulls Optimization Algorithm (SOA) merupakan algoritma yang diusulkan oleh Dhiman dan Kumar. Kesederhanaan SOA dengan parameter yang sedikit dan juga mudah diimplementasikan membuatnya dapat diterapkan dalam berbagai bidang. *Seagulls Optimization Algorithm* (SOA) terinspirasi dari perilaku migrasi dan cara menyerang mangsa dari *seagulls* di alam. Ada dua langkah kunci dalam SOA, yaitu tahapan eksplorasi dan tahapan eksploitasi [10].

Menurut [10], SOA telah diuji pada empat puluh empat fungsi uji Benchmark dengan memvalidasi efisiensinya. Hasilnya mengungkapkan bahwa SOA memberikan banyak hasil yang kompetitif dibandingkan dengan *Genetic Algorithm* (GA), *Particle Swarm Optimization* (PSO), dan *Gravitational Search Algorithm* (GSA). Hal ini membuat SOA dapat digunakan dalam kehidupan sehari-hari yaitu untuk aplikasi industri kehidupan nyata seperti desain penyangga optik, desain bejana tekan dan desain peredam kecepatan.

Oleh karena itu, sangat menarik untuk menyelesaikan *Open Vehicle Routing Problem* (OVRP) digunakanlah *Seagulls Optimization Algorithm* (SOA). Diharapkan penggunaan algoritma tersebut memberikan hasil yang terbaik.

2 *Open Vehicle Routing Problem*

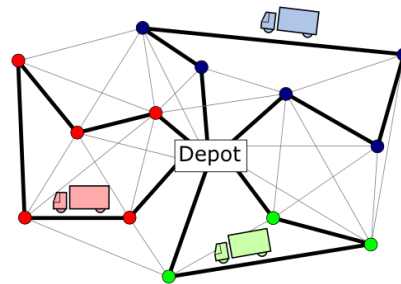
Vehicle Routing Problem merupakan istilah umum untuk suatu permasalahan lebih dari satu kendaraan pengangkut dengan kapasitas tertentu untuk mengunjungi sejumlah pelanggan dengan permintaannya masing-masing yang berkaitan dengan penentuan pengoptimalan rute. Dalam penentuan tersebut kendaraan diharuskan dimulai dan diakhiri di depot. Setiap pelanggan dikunjungi tepat satu kali dengan total permintaan semua pelanggan dalam satu rute tidak boleh melebihi kapasitas kendaraan yang melayani rute tersebut [11].

Tujuan dari penyelesaian VRP antara lain [1] :

1. Meminimumkan keseluruhan biaya pengiriman, berdasarkan pada total jarak yang harus ditempuh dengan menggunakan beberapa kendaraan yang diikuti dengan berbagai macam *constraint* dan setiap pelanggan dilayani tepat satu kali oleh satu kendaraan saja.

2. Meminimumkan jumlah kendaraan pengangkut yang dibutuhkan untuk melayani semua pelanggan.
3. Menyeimbangkan rute dengan waktu tempuh dan muatan.

Berikut pada Gambar 1 memperlihatkan ilustrasi VRP dengan melayani 13 pelanggan oleh satu depot dan menggunakan 3 kendaraan, sehingga terbentuk 3 rute pengiriman.



Gambar 1 Ilustrasi VRP dengan 13 pelanggan dan 3 kendaraan

OVRP merupakan masalah yang populer di bidang distribusi dan dapat digunakan untuk memodelkan banyak masalah kehidupan nyata. Misalnya jika perusahaan tidak memiliki kendaraan pribadi dan harus menyewa beberapa kendaraan untuk mengirimkan produknya kepada konsumen, tidak peduli apakah kendaraan kembali ke depot atau tidak, serta tidak membayar biaya perjalanan antara pelanggan terakhir dan depot. Tujuannya adalah untuk menemukan rute terpendek yang optimal untuk sebuah kendaraan (kendaraan) agar setiap permintaan pelanggan terpenuhi [2].

Menurut [2] untuk meminimumkan jarak tempuh dan biaya perjalanan maka tujuan permasalahan OVRP dapat dimodelkan sebagai berikut :

$$\text{Meminimumkan } f = c_o \sum_{i=0}^{N_0} \sum_{\substack{j \neq 0 \\ j \neq i}}^{N_0} d_{ij} x_{ij} + c_r \sum_{i=0}^{N_0} \sum_{\substack{j=0 \\ j \neq i}}^{N_0} d_{ij} y_{ij} + c_f \sum_{j=0}^{N_0} y_{0j} \quad (1)$$

Variabel keputusan yang akan digunakan adalah sebagai berikut :

$$x_{ij} = \begin{cases} 1, & \text{jika kendaraan pribadi melayani pelanggan } j \text{ setelah mengunjungi pelanggan } i. \\ 0, & \text{jika tidak demikian.} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{jika kendaraan sewa melayani pelanggan } j \text{ setelah mengunjungi pelanggan } i. \\ 0, & \text{jika tidak demikian.} \end{cases}$$

dengan kendala sebagai berikut :

1. Setiap pelanggan dan depot yang dikunjungi tepat satu kali oleh satu kendaraan pengangkut, baik kendaraan pribadi atau kendaraan sewa.

$$\sum_{i \in N_0} x_{ij} + \sum_{i \in N_0} y_{ij} = 1, \quad j \in N, i \neq j \quad (2)$$

2. Kendaraan tidak perlu berangkat dari setiap pelanggan, karena rute kendaraan sewa berakhir setelah pelanggan yang terakhir.

$$\sum_{j \in N_0} x_{ij} + \sum_{j \in N_0} y_{ij} \leq 1, \quad i \in N, i \neq j \quad (3)$$

3. Menghindar adanya *sub-tour* (Untuk kendaraan sewa dan pribadi)

$$u_i + q_j - g(1 - x_{ij}) \leq u_j \quad i \in N_0, j \in N, j \in N, i \neq j \quad (4)$$

$$u_i + q_j - g(1 - y_{ij}) \leq u_j \quad i \in N_0, j \in N, j \in N, i \neq j \quad (5)$$

4. Setiap kendaraan harus meninggalkan pelanggan yang telah dikunjungi (kendaraan pribadi).

$$\sum_{i \in N_0} x_{ij} - \sum_{i \in N_0} x_{ji} = 0, \quad j \in N, i \neq j \quad (6)$$

5. Setiap pelanggan dalam rute yang dilalui setiap kendaraan pengangkut tidak boleh melebihi kapasitas kendaraan.

$$q_i \leq u_i \leq g \quad i \in N \text{ dengan } N = \{1, 2, \dots, n\} \quad (7)$$

6. Nilai perbaikan u_i diatur oleh

$$u_0 = 0 \quad (8)$$

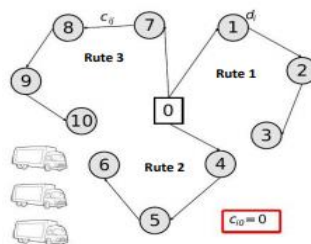
7. Mendefinisikan variabel x untuk setiap kendaraan sewa.

$$x_{ij} \in \{0, 1\}, \quad i, j \in N_0 \text{ dengan } N_0 = N \cup \{0\} \quad (9)$$

8. Mendefinisikan variabel y untuk setiap kendaraan pribadi.

$$y_{ij} \in \{0, 1\}, \quad i, j \in N_0 \text{ dengan } N_0 = N \cup \{0\} \quad (10)$$

Berikut pada Gambar 2 memperlihatkan salah satu ilustrasi OVRP dengan 3 rute pengiriman.



Gambar 2 Ilustrasi OVRP dengan 3 rute pengiriman

3 Seagulls Optimization Algorithm

Seagulls Optimization Algorithm (SOA) terinspirasi dari perilaku migrasi dan cara menyerang mangsa dari *Seagulls* (Burung Camar) di alam. *Seagulls* merupakan burung yang cerdas, mereka juga menggunakan kecerdasan mereka untuk mencari dan menyerang mangsanya. Hal yang penting dari *seagulls* yaitu mengenai cara mereka migrasi dan cara mereka menyerang mangsanya [10].

Berikut ini langkah-langkah dalam mengimplementasikan *Seagulls Optimization Algorithm* (SOA) :

1. Inisialisasi masalah dan input parameter. Menentukan permasalahan optimasi, variabel keputusan dan kendala. Kemudian, menginput parameter dari SOA (iterasi maksimal ($max_iterasi$), populasi *seagulls* (N), variabel kendali dari A (f_c) dan bentuk spiralnya (α, β).
2. Membangkitkan posisi awal *seagulls*. Membangkitkan bilangan real secara acak pada interval $[0,1]$ sebanyak pelanggan yang mewakili posisi dari setiap *seagulls*.
3. Mengevaluasi fungsi tujuan awal *seagulls*. Menghitung nilai fungsi tujuan awal $F_i, i = 0,1, \dots, N$
4. Menentukan fungsi tujuan terbaik sementara (P_{best}^t).
5. Melakukan tahap migrasi.
6. Melakukan proses menghindari tabrakan antar *seagulls*.
 - Menghitung nilai a .

$$a = f_c - \left(t \times \left(\frac{f_c}{max_iterasi} \right) \right) \quad (11)$$

- Menghitung nilai $P_{N_i}^t$.

$$P_{N_i}^t = a \times P_{C_i}^t, \quad t = 0,1,2, \dots, Maks_iterasi \quad (12)$$

7. Melakukan gerakan menuju *seagulls* terdekat yang terbaik.
 - Membangkitkan bilangan real acak (r_d) pada interval $[0,1]$.
 - Menghitung nilai konstanta b .

$$b = 2 \times a^2 \times r_d \quad (13)$$

- Menghitung $d_{e_i}^t$.

$$d_{e_i}^t = b \times \left(P_{best}^t - P_{C_i}^t \right) \quad (14)$$

8. Melakukan gerakan menuju *seagull* terbaik $D_{e_i}^t$.

$$D_{e_i}^t = \left| P_{N_i}^t + d_{e_i}^t \right| \quad (15)$$

9. Melakukan tahap *Attacking*
 - Membangkitkan bilangan t secara acak pada interval $[0,2\pi]$.
 - Menghitung nilai jari-jari spiral r .

$$r = \alpha \times e^{\beta t} \quad (16)$$

- Menghitung nilai dari \hat{x} , \hat{y} dan \hat{z} .

$$\hat{x} = r \times \cos(t) \quad (17)$$

$$\hat{y} = r \times \sin(t) \quad (18)$$

$$\hat{z} = r \times t \quad (19)$$

- Memperbarui nilai P_c dengan P_c^t .

$$P_c^t = (D_e^t \times \hat{x} \times \hat{y} \times \hat{z}) + P_{best_i}^t \quad (20)$$

10. Periksa Kelayakan posisi baru *seagulls*. Jika posisi baru *seagulls* layak, *seagulls* akan memperbarui posisinya. Jika tidak, *seagulls* tetap pada posisi yang lama (saat itu) dan tidak pindah ke posisi yang baru.
11. Mengevaluasi fungsi tujuan dari posisi baru. Menghitung fungsi tujuan untuk posisi baru dari setiap *seagulls*.
12. Cek kriteria keputusan. Ulangi langkah 4-7 sampai *max_iterasi* tercapai.
13. Mencetak hasil solusi terbaik (P_c^t)

Berikut ini dijelaskan langkah-langkah dalam prosedur *Open Vehicle Routing Problem* menggunakan *Seagulls Optimization Algorithm*, ditunjukkan pada Gambar 3.

Prosedur SOA untuk menyelesaikan OVRP

begin

Input data OVRP();

Inisialisasi Parameter SOA dan OVRP();

Bangkitkan posisi awal();

Evaluasi fungsi tujuan posisi awal *seagulls*();

do

update posisi *seagulls*();

evaluasi fungsi tujuan posisi baru dan *update* memori *seagulls*();

if (*iterasi* < *max_iterasi*)

Menentukan *seagulls* terbaik sementara();

do

Tahapan Migrasi ();

Tahapan *Attacking*();

while (*a* < 0)

Menyimpan posisi terbaik();

end if

while(*iterasi* = *max_iterasi*)

menentukan *seagulls* terbaik dari memori();

end

Gambar 3 Prosedur SOA untuk menyelesaikan OVRP

4 Hasil dan Pembahasan

Data yang digunakan dalam skripsi ini diambil dari <https://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm.old>. Data terdiri dari data berukuran kecil (18 pelanggan), data berukuran sedang (50 pelanggan) dan data berukuran besar (100 pelanggan). Parameter yang akan digunakan adalah $\alpha = 1, \beta = 1$ berdasarkan [10], dengan $f_c = 10; 100; 1000$, $jum_sea = 10; 50; 100$, $MaxIterasi = 10; 100; 1000$. Hasil *output runing program* dengan data kecil dapat dilihat pada Tabel 1.

Tabel 1 Hasil Output Data 18 Pelanggan

f_c	Iterasi	Jumlah Seagulls		
		10 Seagulls	50 Seagulls	100 Seagulls
10	10	238.713	207.074	221.779
	100	234.617	217.683	213.299
	1000	231.129	222.17	212.929
100	10	241.365	234.786	225.058
	100	238.022	228.458	226.438
	1000	235.617	222.346	228.805
1000	10	249.861	235.105	225.598
	100	238.287	231.248	213.699
	1000	237.521	227.665	230.166

Dari Tabel 1, dapat dilihat biaya total minimum yang didapatkan dari data kecil dengan 18 pelanggan adalah 207.074 € dengan 50 seagulls, 10 iterasi *Seagulls Optimization Algorithm* dan f_c bernilai 10. Hal ini menunjukkan bahwa semakin banyak jumlah seagulls, semakin banyak jumlah iterasi dan semakin kecil nilai f_c cenderung mempengaruhi biaya minimum yang yang didapatkan. Selanjutnya, Hasil *output runing program* dengan data sedang dapat dilihat pada Tabel 2.

Tabel 2 Hasil Output Data 50 Pelanggan

f_c	Iterasi	Jumlah Seagulls		
		10 Seagulls	50 Seagulls	100 Seagulls
10	10	783.012	758.507	754.784
	100	781.368	739.247	732.31
	1000	757.405	728.753	727.324
100	10	784.657	767.406	724.569
	100	789.081	762.947	707.161
	1000	791.07	761.163	728.863
1000	10	795.296	783.711	779.77
	100	789.175	772.696	752.824
	1000	788.162	777.216	772.101

Dari Tabel 2, dapat dilihat biaya total minimum yang didapatkan dari data kecil dengan 50 pelanggan adalah 707.161 € dengan 100 *seagulls*, 100 iterasi *Seagulls Optimization Algorithm* dan f_c bernilai 100. Hal ini menunjukkan bahwa semakin banyak jumlah *seagulls*, semakin banyak jumlah iterasi dan semakin kecil nilai f_c cenderung mempengaruhi biaya minimum yang yang didapatkan. Selanjutnya, Hasil *output running program* dengan data besar dapat dilihat pada Tabel 3.

Tabel 3 Hasil Output Data 100 Pelanggan

f_c	Iterasi	Jumlah Seagulls		
		10 <i>Seagulls</i>	50 <i>Seagulls</i>	100 <i>Seagulls</i>
10	10	1647.93	1614.9	1596.64
	100	1624.2	1589.83	1625.93
	1000	1612.73	1604.63	1604.27
100	10	1674.06	1666.48	1660.95
	100	1671.24	1661.31	1629.06
	1000	1625.9	1638.97	1622.23
1000	10	1701.85	1713.22	1635.95
	100	1671.35	1653.05	1647.04
	1000	1658.75	1647.84	1608.8

Dari Tabel 3, dapat dilihat biaya total minimum yang didapatkan dari data kecil dengan 100 pelanggan adalah 1589.83 € dengan 50 *seagulls*, 100 iterasi *Seagulls Optimization Algorithm* dan f_c bernilai 10. Hal ini menunjukkan bahwa semakin banyak jumlah *seagulls*, semakin banyak jumlah iterasi dan semakin kecil nilai f_c cenderung mempengaruhi biaya minimum yang yang didapatkan.

5 Kesimpulan

Seagulls Optimization Algorithm (SOA) dapat diterapkan untuk menyelesaikan *Open Vehicle Routing Problem* (OVRP). Dari hasil implementasi ketiga contoh kasus dapat disimpulkan bahwa semakin banyak jumlah *seagulls*, semakin banyak jumlah iterasi, dan semakin kecil nilai f_c maka cenderung semakin minimum total biaya yang didapatkan.

6 Daftar Pustaka

- [1] Toth, P., dan Vigo, D., 2002, *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia, USA.
- [2] Čičková Z, dkk., 2015, *Cost Analysis of Open VRP*, 2nd Logistics Internasional Conference, pp. 8-13.

- [3] Sariklis, D., dan Powell, S., 2000. *A Heuristic Method for Open Vehicle Routing Problem*. The Journal of the Operational Research Society, Vol. 51, No. 5 : 564 - 573.
- [4] Brandao, J., 2004, A Tabu Search Algorithm for Open Vehicle Routing Problem, *European Journal of Operational Research* 157 : 552-564.
- [5] Mirhassani, S.A., dan Abolghasemi, N., 2011, *A Particle Swarm Optimization Algorithm for Open Vehicle Routing Problem*, *Expert Systems with Applications*, pp. 11547-11551.
- [6] Yu, Sh., Ding, Ch., dan Zhu, K., 2011, A Hybrid GA-TS Algorithm for Open Vehicle Routing Optimization of Coal Mines Material, *Journal of Expert Systems with Applications* 38 : 10568-10573.
- [7] Sánchez, A.D.L., Diaz, A.G.H., Vigo, D., Cabellero, R., dan Molina, J., 2014, A Multi-Start Algorithm for A Balanced Real-World Open Vehicle Routing Problem, *European Journal of Operational Research*, Vol.238, pp. 104-113.
- [8] Yu, V.F., dan Lin S.Y., A, 2014, Simulated Annealing for Open Location Routing Problem, *Computers & Operations Research*.
- [9] Gurpreetsingh, Er dan Dhir, V., 2014. *Open Vehicle Routing Problem by Ant Colony Optimization*. International Journal of Advanced Computer Science and Application (IJACSA), Vol. 5, No. 3.
- [10] Dhiman, G., dan Vinjay, K., 2019, *Seagull Optimization Algorithm : Theory and its Applications for Large – Scale Industrial Engineering Problems*. Journal Knowledge – Based Systems, Vol. 165 : 169 – 196.
- [11] Belfiore, P., Tsugunobu, H., dan Yoshizaki, Y., 2008, *Vehicle Routing Problem*, Intech, Croatia.