

## Development of Particle Swarm Optimization and Simulated Annealing Algorithms to Solve Vehicle Routing Problems With Drones

### Pengembangan Algoritma *Particle Swarm Optimization* dan *Simulated Annealing* untuk Menyelesaikan *Vehicle Routing Problem with Drone*

Hasan Aji Prawira<sup>1</sup>, Budi Santosa<sup>2</sup>.

Departemen Teknik Industri, Institut Teknologi Sepuluh Nopember

E-mail: [hasanajip97@gmail.com](mailto:hasanajip97@gmail.com)<sup>1</sup>, [budi\\_s@ie.its.ac.id](mailto:budi_s@ie.its.ac.id)<sup>2</sup>

#### ABSTRACT

*Vehicle Routing Problem with Drone (VRPD)* is a problem of determining the number of routes for delivery of goods from the depot to a number of customers using trucks and drones. Drones are an alternative delivery tool besides trucks, each truck can be equipped with a support drone. Drones can be used to make a delivery while the truck is making others. By combining a truck and a drone, the truck can act as a tool for drone launch and landing so that the drones can reach long distances from the depot. The purpose of this problem is to minimize the cost of sending goods by trucks and drones. In this study, the *Particle Swarm Optimization (PSO)* and the *Simulated Annealing (SA)* are proposed to solve these problems. The *Route Drone* algorithm are used to help change the structure of the *PSO* and *SA* solutions into a *VRPD* solution. The proposed algorithm has been applied to 24 different scenarios ranging from 6 customers to 100 customers. The *PSO* and *SA* algorithms are able to find solutions that are close to optimal. The *SA* is able to find a better solution than the *PSO*.

**Keywords:** *Vehicle Routing Problem, Drone, Particle Swarm Optimization, Simulated Annealing, Metaheuristic, Vehicle Routing Problem with Drone.*

#### ABSTRAK

*Vehicle Routing Problem with Drone (VRPD)* merupakan permasalahan penentuan sejumlah rute pengiriman barang dari depo ke sejumlah kostumer dengan menggunakan truk dan *drone*. *Drone* menjadi alternatif alat pengiriman selain truk, setiap truk dapat dilengkapi dengan *drone* pendukung. *Drone* dapat digunakan untuk melakukan beberapa pengiriman saat truk melakukan pengiriman yang lain. Dengan mengkombinasikan truk dan *drone*, truk dapat berperan sebagai alat untuk peluncuran dan pendaratan *drone* sehingga *drone* dapat mencapai jarak yang jauh dari depot. Tujuan permasalahan ini adalah meminimasi biaya pengiriman barang oleh truk dan *drone*. Dalam penelitian ini, diusulkan algoritma *Particle Swarm Optimization (PSO)* dan algoritma *Simulated Annealing (SA)* untuk menyelesaikan permasalahan tersebut. Algoritma *Route Drone* digunakan untuk membantu mengubah struktur solusi algoritma *PSO* dan *SA* menjadi solusi *VRPD*. Algoritma yang diusulkan telah diaplikasikan untuk 24 macam skenario mulai dari 6 pelanggan hingga 100 pelanggan. Algoritma *PSO* dan *SA* mampu menemukan solusi yang mendekati optimal. Algoritma *SA* mampu menemukan solusi yang lebih baik dibandingkan algoritma *PSO*.

**Kata kunci:** *Vehicle Routing Problem, Drone, Particle Swarm Optimization, Simulated Annealing, Metaheuristik, Vehicle Routing Problem with Drone.*

#### PENDAHULUAN

Pengiriman barang ke *end customer* atau biasa disebut *last mile delivery* merupakan salah satu proses penting dalam sistem logistik. Adanya persaingan yang tinggi di pasar mengakibatkan pelanggan cenderung membeli layanan dan produk dari perusahaan dengan pengiriman yang cepat, fleksibel, dan murah. Oleh karena itu, perusahaan-perusahaan yang bergerak di bidang logistik selalu mencari cara untuk mempercepat dan mengefisienkan biaya pada *last mile delivery*. Salah satu cara yang menjadi perhatian belum lama ini adalah

Pengembangan Algoritma *Particle Swarm Optimization* dan *Simulated Annealing* untuk Menyelesaikan *Vehicle Routing Problem with Drone* / Hasan Aji Prawira, Budi Santosa

Peer reviewed under responsibility of Universitas Muhammadiyah Sidoarjo.

© 2021 Universitas Muhammadiyah Sidoarjo. All Right reserved. This is an open access article under the CC BY licence (<http://creativecommons.org/licenses/by/4.0/>)

penggunaan *Unmanned Aerial Vehicles* (UAVs) atau sering disebut *drone* untuk pengiriman barang ke lokasi pelanggan [1]. *Drone* memiliki beberapa keunggulan dibandingkan dengan truk dalam melakukan pengiriman barang yaitu dapat menghindari kemacetan sehingga lebih cepat dalam pengiriman, biaya per kilometer yang lebih murah [2] dan pengoperasian tanpa biaya pengemudi. Pengiriman barang menggunakan *drone* menjadi pusat perhatian, saat Jeff Bezos, CEO Amazon, mengumumkan bahwa perusahaannya sedang mengembangkan gagasan penggunaan *drone* untuk pengiriman komoditas kecil [3]. Amazon tidak sendirian dalam pengembangan pengiriman menggunakan *drone*. Grup pos dan logistik Jerman Deutsche Post DHL baru-baru ini mengumumkan bahwa *Parcelcopter* mereka telah diberi wewenang untuk mengirimkan pasokan medis ke pulau bebas mobil di lepas pantai Jerman [4]. Perusahaan JD juga telah melakukan percobaan pengiriman menggunakan *drone* di Indonesia [5].

Bedasarkan paparan diatas, *drone* telah dianggap sebagai pilihan yang menjanjikan untuk aplikasi industri logistik. Tetapi, *drone* tidak bisa dianggap sebagai alternatif pengganti kendaraan pengiriman konvensional seperti truk melainkan sebagai alat pengiriman yang saling melengkapi. Hal ini disebabkan *drone* memiliki keterbatasan pada kapasitas muatan dan jarak terbang dibandingkan truk. Jika dibandingkan dengan *drone*, truk dapat membawa barang yang lebih berat dan menempuh jarak yang jauh. Tanpa adanya truk, *drone* tidak bisa memenuhi semua permintaan pengiriman. Pengiriman dengan mengkombinasikan truk dan *drone* menjadi cara yang efisien karena dapat memanfaatkan masing-masing keunggulannya. Penggunaan truk dan *drone* memungkinkan pengiriman lebih cepat karena truk dan *drone* dapat mengirimkan barang secara simultan tanpa adanya tambahan jarak [6]. Berdasarkan literatur, permasalahan pengiriman barang-barang kecil dengan banyak titik pelanggan sering diformulasikan sebagai *Vehicle Routing Problem* (VRP) [7]. Setelah adanya *drone* sebagai alternatif alat pengiriman, setiap truk dapat dilengkapi dengan *drone* pendukung. *Drone* dapat digunakan untuk melakukan beberapa pengiriman saat truk melakukan pengiriman yang lain. Dengan mengkombinasikan truk dan *drone*, truk dapat berperan sebagai alat untuk peluncuran dan pendaratan *drone* sehingga *drone* dapat mencapai jarak yang jauh dari depot. Pemodelan permasalahan ini sering disebut *Vehicle Routing Problem with Drone* (VRP-D) [8].

Penelitian pertama yang mempelajari kerja sama antara truk dan *drone* dilakukan oleh [9] dengan menggunakan *Traveling Salesman Problem* (TSP) sebagai model dasar. [9] Muray menyebut permasalahan tersebut sebagai *Flying Sidekick Traveling Salesman Problem* (FSTSP). Permasalahan tersebut adalah varian dari TSP yang hanya menggunakan satu truk dengan satu *drone* untuk mengirimkan barang ke pelanggan. *Drone* diterbangkan dari suatu lokasi untuk mengirimkan barang ke pelanggan dan mendarat di lokasi pertemuan dengan truk. Saat *drone* terbang, truk dapat mengunjungi pelanggan lain, namun truk harus sampai di lokasi pertemuan sebelum baterai *drone* habis. Dalam permasalahan ini, fungsi tujuannya adalah meminimalkan waktu penyelesaian rute. Studi terbaru, seperti [9], [10], [11], [12], [13], dan [8] telah menyelidiki keuntungan penggunaan kedua kendaraan ini dibandingkan dengan mendistribusikan barang hanya menggunakan truk.

Penelitian ini berfokus pada VRP-D dengan setiap truk dilengkapi satu *drone* pendukung. Permasalahan VRP-D merupakan permasalahan berklasifikasi *NP-hard* sehingga metode *exact* memerlukan waktu yang sangat lama untuk menemukan solusi yang optimal. VRP-D bisa diselesaikan dengan menggunakan metode *exact* jika skala permasalahan kecil atau sekitar 10-20 titik pelanggan [12]. Sedangkan, VRP-D perlu penyelesaian yang cepat karena frekuensi pengiriman yang besar. Oleh karena itu, pengembangan metode heuristik untuk menyelesaikan VRP-D dengan skala permasalahan besar lebih diutamakan dibandingkan metode *exact*. Penelitian ini menggunakan metode metaheuristik untuk menyelesaikan VRP-D. Pendekatan metaheuristik dipilih karena metaheuristik merupakan kelanjutan dari heuristik muncul sebagai alternatif pemecahan masalah-masalah yang sulit diselesaikan menggunakan metode *exact* [14]. Pada penelitian ini algoritma metaheuristik *Particle Swarm Optimization* (PSO) dan *Simulated Annealing* (SA) dipilih untuk menyelesaikan VRP-D. Kedua algoritma ini dipilih karena berdasarkan referensi terkait VRP-D, kedua algoritma ini belum pernah digunakan untuk menyelesaikan VRP-D. Selain itu, kedua algoritma ini merupakan algoritma yang sangat sering digunakan dan telah berhasil menyelesaikan permasalahan kontinyu dengan *constraint*[15], *integer programming*, serta

---

Pengembangan Algoritma *Particle Swarm Optimization* dan *Simulated Annealing* untuk Menyelesaikan *Vehicle Routing Problem with Drone* / Hasan Aji Prawira, Budi Santosa

Peer reviewed under responsibility of Universitas Muhammadiyah Sidoarjo.

© 2021 Universitas Muhammadiyah Sidoarjo. All Right reserved. This is an open access article under the CC BY licence (<http://creativecommons.org/licenses/by/4.0/>)

permasalahan *combinatorial*[16]. Pada penelitian ini kedua algoritma akan dibandingkan untuk mendapatkan algoritma yang menghasilkan solusi yang lebih baik.

## METODE

### A. Model Matematis

*Vehicle Routing Problem with Drone* (VRPD) merupakan salah satu jenis VRP yang tujuannya adalah untuk menentukan serangkaian rute optimal yang dilakukan oleh truk dan *drone* untuk melayani sejumlah pelanggan dengan beberapa kondisi yang harus terpenuhi [17]. Berdasarkan penelitian [8], formulasi matematika untuk VRPD dapat didefinisikan sebagai berikut. Himpunan-himpunan yang akan digunakan untuk memformulasikan permasalahan adalah sebagai berikut.  $C$  merupakan himpunan titik pelanggan.  $C'$  merupakan himpunan bagian dari pelanggan yang dapat dilayani oleh *drone*.  $D$  merupakan titik depot yang menunjukkan awal dan akhir rute.  $N$  merupakan himpunan semua titik.  $N_0$  merupakan himpunan titik yang dapat digunakan untuk keberangkatan kendaraan.  $N_+$  merupakan himpunan titik yang dapat digunakan untuk kedatangan kendaraan.  $\Delta^+(i)$  merupakan himpunan titik yang dapat dicapai dari titik  $i$  yang merupakan elemen himpunan  $N_0$ .  $\Delta^-(i)$  merupakan himpunan titik yang dapat digunakan untuk mencapai titik  $i$  yang merupakan elemen himpunan  $N_+$ .  $A$  merupakan himpunan *feasible arc*.  $V$  merupakan himpunan truk homogen dengan  $m$  adalah jumlah truk yang tersedia.

Parameter-parameter yang digunakan untuk formulasi matematika adalah sebagai berikut.  $\tau_{ij}^T$  adalah waktu perjalanan truk dari  $i \in N_0$  ke  $j \in N_+$ .  $\tau_{ij}^D$  adalah waktu perjalanan *drone* dari  $i \in N_0$  ke  $j \in N_+$ .  $c_{ij}^T$  adalah biaya perjalanan truk dari  $i \in N_0$  ke  $j \in N_+$ .  $c_{ij}^D$  adalah biaya perjalanan *drone* dari  $i \in N_0$  ke  $j \in N_+$ .  $Q$  adalah kapasitas truk.  $q_i$  adalah *demand* pelanggan  $i \in C$ .  $e$  adalah batas waktu terbang *drone*.  $Se_i^T$  adalah waktu pelayanan truk di pelanggan  $i \in C$ .  $Se_i^D$  adalah waktu pelayanan *drone* di pelanggan  $i \in C'$ .  $SL$  adalah waktu yang diperlukan untuk menerbangkan *drone*.  $SR$  adalah waktu yang diperlukan untuk mendaratkan *drone*.  $M$  adalah bilangan yang cukup besar.  $T_{max}$  adalah durasi maksimum setiap rute.

Selanjutnya, notasi tambahan diperlukan untuk mengidentifikasi kemungkinan *sorties* tiga titik dari *drone*. Didefinisikan  $P$  adalah himpunan *sorties* yang mungkin, direpresentasikan dengan *tuple*  $\langle i, j, k \rangle$ . Titik pertama mewakili lokasi peluncuran *drone*, titik kedua mewakili pelanggan yang dikunjungi oleh *drone*, dan titik ketiga mewakili lokasi pendaratan *drone*. Oleh karena itu, elemen  $\langle i, j, k \rangle$  milik himpunan  $P$  berlaku kondisi berikut. Pertama, titik peluncuran  $i \in N_0$  dari *tuple* adalah lokasi yang dapat digunakan untuk peluncuran *drone* sesuai dengan lokasi yang dapat digunakan untuk pemberangkatan truk. Kedua, titik pengiriman  $j \in \{C' : j \neq i\}$  dari *tuple* adalah himpunan pelanggan yang dapat dilayani oleh *drone* dan berbeda dengan posisi peluncuran  $i$ . Ketiga, titik pertemuan atau posisi pemulihan  $k \in \{N_+ : k \neq i, k \neq j, SL + SR + \tau_{ij}^D + \tau_{jk}^D + Se_j^D \leq e\}$  dari *tuple* adalah lokasi pertemuan *drone* dengan truk untuk pemulihan baterai *drone*.

Selanjutnya, *tuple*  $\langle 0, i, c + 1 \rangle$  dihilangkan dari  $P$  untuk semua  $i \in C'$ . *tuple* tersebut menunjukkan pengiriman dengan *drone* dengan titik peluncuran dan pertemuan di depot. Biaya *sortie*  $c_s^D$  dari *sortie*  $s = \langle i, j, k \rangle$  dihitung dari  $c_s^D = c_{ij}^D + c_{jk}^D$ . Semua *sortie* dari  $P$  dengan titik peluncuran  $i \in N_0$  didefinisikan dengan notasi  $P_i^+$ . Semua *sortie* dari  $P$  dengan titik pemulihan  $k \in N_+$  didefinisikan dengan notasi  $P_k^-$  dan  $P_j$  adalah semua *sortie* dari  $P$  yang mengirim ke pelanggan  $j \in C'$ . Variabel keputusan yang digunakan untuk formulasi matematika adalah sebagai berikut.  $x_{ij}^v$  adalah variabel biner yang bernilai 1 jika truk  $v \in V$  melalui titik  $i \in N_0$  dan  $j \in N_+$ .  $u_i^v$  adalah variabel kontinu yang menunjukkan posisi kunjungan  $i \in N$  pada rute truk  $v \in V$ .  $t_i^v$  adalah variabel kontinu yang menunjukkan waktu dari rute truk  $v \in V$  saat tiba di lokasi  $i \in N$ .  $t_i^{vD}$  adalah variabel kontinu yang menunjukkan waktu *drone* dari truk  $v \in V$  saat tiba di lokasi  $i \in N$ .  $p_{ij}^v$  adalah variabel biner yang bernilai 1

---

Pengembangan Algoritma *Particle Swarm Optimization* dan *Simulated Annealing* untuk Menyelesaikan *Vehicle Routing Problem with Drone* / Hasan Aji Prawira, Budi Santosa

Peer reviewed under responsibility of Universitas Muhammadiyah Sidoarjo.

© 2021 Universitas Muhammadiyah Sidoarjo. All Right reserved. This is an open access article under the CC BY licence (<http://creativecommons.org/licenses/by/4.0/>)

jika pelanggan  $j \in C$  dikunjungi setelah lokasi  $j \in N_0$  pada rute truk  $v \in V$ .  $y_s^v$  adalah variabel biner yang bernilai 1 jika *sortie*  $s \in P$  digunakan pada rute truk  $v \in V$ .

Bedasarkan himpunan, parameter dan variabel keputusan diatas, permasalahan VRPD dapat diformulasikan sebagai berikut.

Minimize

$$\sum_{v \in V} \left( \sum_{(i,j) \in A} c_{ij}^T x_{ij}^v + \sum_{s \in P} c_s^D y_s^v \right) \quad (1)$$

Subject to:

$$\sum_{v \in V} \sum_{i \in \Delta^-(j)} x_{ij}^v + \sum_{v \in V} \sum_{s \in P_j} y_s^v = 1 \quad \forall j \in C \quad (2)$$

$$\sum_{j \in N_+} x_{0,j}^v \leq 1 \quad \forall v \in V \quad (3)$$

$$\sum_{i \in N_0} x_{i,c+1}^v \leq 1 \quad \forall v \in V \quad (4)$$

$$x_{0,c+1}^v = 0 \quad \forall v \in V \quad (5)$$

$$\sum_{i \in \Delta^-(j)} x_{ij}^v = \sum_{k \in \Delta^+(j)} x_{jk}^v \quad \forall v \in V, j \in C \quad (6)$$

$$u_i^v + 1 \leq u_j^v + M(1 - x_{ij}^v) \quad \forall v \in V, (i, j) \in A \quad (7)$$

$$u_j^v \leq M \sum_{i \in \Delta^-(j)} x_{ij}^v \quad \forall v \in V, j \in N_+ \quad (8)$$

$$\sum_{j \in C} \left( \sum_{k \in \Delta^+(j)} q_j x_{jk}^v + \sum_{s \in P_j} q_j y_s^v \right) \leq Q \quad \forall v \in V \quad (9)$$

$$\sum_{s \in P_i^+} y_s^v \leq 1 \quad \forall v \in V, i \in N_0 \quad (10)$$

$$\sum_{s \in P_k^-} y_s^v \leq 1 \quad \forall v \in V, k \in N_+ \quad (11)$$

$$2y_s^v \leq \sum_{h \in \Delta^+(i)} x_{ih}^v + \sum_{l \in \Delta^-(k)} x_{lk}^v \quad \forall v \in V, s = \langle i, j, k \rangle \in P \quad (12)$$

$$t_0^v = 0 \quad \forall v \in V \quad (13)$$

$$t_0^{v'} = 0 \quad \forall v \in V \quad (14)$$

$$t_{c+1}^v \leq T_{\max} \sum_{i \in N_0} x_{i,c+1}^v \quad \forall v \in V \quad (15)$$

$$t_{c+1}^{v'} \leq T_{\max} \sum_{s \in P_{c+1}^-} y_s^v \quad \forall v \in V \quad (16)$$

$$t_h^v + \tau_{hk}^T + Se_h^T + SL \sum_{s \in P_h^+} y_s^v + SR \sum_{s \in P_k^-} y_s^v \leq t_k^v + T_{\max} (1 - x_{hk}^v) \quad \forall v \in V, (h, k) \in A \quad (17)$$

$$t_i^v + \tau_{hk}^D + SL - T_{\max} \left( 1 - \sum_{s \in P_i^+ \cap P_j} y_s^v \right) \leq t_j^{v'} \quad \forall v \in V, (i, j) \in A \quad (18)$$

Pengembangan Algoritma *Particle Swarm Optimization* dan *Simulated Annealing* untuk Menyelesaikan *Vehicle Routing Problem with Drone* / Hasan Aji Prawira, Budi Santosa

Peer reviewed under responsibility of Universitas Muhammadiyah Sidoarjo.

© 2021 Universitas Muhammadiyah Sidoarjo. All Right reserved. This is an open access article under the CC BY licence (<http://creativecommons.org/licenses/by/4.0/>)

$$t_j^{iv} + \tau_{jk}^D + Se_j^D + SR - T_{\max} \left( 1 - \sum_{s \in P_j \cap P_k^-} y_s^v \right) \leq t_k^{iv} \quad \forall v \in V, j \in C', k \in \Delta^+(j) \quad (19)$$

$$t_i^v - T_{\max} \left( 1 - \sum_{s \in P_i^+} y_s^v \right) \leq t_i^{iv} \quad \forall v \in V, i \in N_0 \quad (20)$$

$$t_i^v + T_{\max} \left( 1 - \sum_{s \in P_i^+} y_s^v \right) \geq t_i^{iv} \quad \forall v \in V, i \in N_0 \quad (21)$$

$$t_k^v - T_{\max} \left( 1 - \sum_{s \in P_k^-} y_s^v \right) \leq t_k^{iv} \quad \forall v \in V, k \in C \quad (22)$$

$$t_k^v + T_{\max} \left( 1 - \sum_{s \in P_k^-} y_s^v \right) \geq t_k^{iv} \quad \forall v \in V, k \in C \quad (23)$$

$$e + T_{\max} \left( 1 - \sum_{s \in P_i^+ \cap P_k^-} y_s^v \right) \geq t_k^{iv} - t_i^{iv} \quad \forall v \in V, i \in N_0, k \in N_+ \quad (24)$$

$$(u_j^v - u_i^v) \leq M (p_{ij}^v) \quad \forall v \in V, i \in N_0, j \in C \setminus \{i\} \quad (25)$$

$$(u_j^v - u_i^v) \geq M (p_{ij}^v - 1) + 1 \quad \forall v \in V, i \in N_0, j \in C \setminus \{i\} \quad (26)$$

$$t_k^{iv} - T_{\max} \left( 3 - \sum_{s \in P_i^+ \cap P_k^-} y_s^v - \sum_{s \in P_i^+} y_s^v - p_{ib}^v \right) \leq t_b^{iv} \quad \forall v \in V, i \in N_0, k \in N_+, b \in C \setminus \{b\} \quad (27)$$

$$x_{ij}^v \in \{0, 1\} \quad \forall v \in V, (i, j) \in A \quad (28)$$

$$y_s^v \in \{0, 1\} \quad \forall v \in V, s \in P \quad (29)$$

$$u_i^v, t_i^v, t_i^{iv} \geq 0 \quad \forall v \in V, i \in N \quad (30)$$

$$p_{ij}^v \in \{0, 1\} \quad \forall v \in V, i \in N_0, j \in C \setminus \{i\} \quad (31)$$

Persamaan 1 menunjukkan fungsi objektif dari permasalahan yaitu meminimumkan biaya operasional ketika mengunjungi pelanggan. Persamaan 2 memastikan setiap pelanggan hanya dikunjungi satu kali menggunakan truk ataupun *drone*. Persamaan 3 memastikan semua truk harus berangkat dari depot maksimal satu kali. Persamaan 4 memastikan semua truk harus kembali ke depot maksimal satu kali. Persamaan 5 memastikan tidak ada perjalanan antar depot. Persamaan 6 memastikan aliran rute. Persamaan 7 dan 8 mengeliminasi *subtour* pada rute truk. Persamaan 9 memastikan barang yang dibawa truk tidak melebihi kapasitas truk. Persamaan 10 dan 11 memastikan setiap titik hanya bisa satu kali menjadi titik peluncuran dan pendaratan *drone*. Persamaan 12 memastikan jika *drone* diluncurkan di titik  $i \in N_0$  dan mendarat di titik  $k \in N_+$  maka truk harus mengunjungi titik tersebut. Persamaan 13 dan 14 memberi nilai waktu awal untuk truk dan *drone*. Persamaan 15 dan 16 memastikan waktu truk dan *drone* kembali ke depot tidak melebihi  $T_{\max}$ . Persamaan waktu 17 untuk truk mendefinisikan waktu truk tiba di lokasi ketika terdapat perjalanan *drone* diantaranya. Persamaan waktu 18 dan 19 untuk *drone* mendefinisikan waktu *drone* tiba di lokasi pelanggan berdasarkan lokasi truk. Persamaan 18 memastikan jika *drone* diluncurkan dari lokasi  $i \in N_0$  ke pelanggan  $j \in C'$ , maka waktu kedatangan *drone* di pelanggan  $j$  harus lebih besar dari waktu kedatangan truk di lokasi  $i$  ditambah waktu perjalanan *drone* dan waktu untuk peluncuran *drone*. Persamaan 19 memastikan jika *drone* selesai melayani pelanggan  $j \in C'$  dan terbang kembali ke truk di lokasi  $k \in N_+$ , maka waktu kedatangan *drone* di lokasi  $k$  harus lebih besar dari

Pengembangan Algoritma *Particle Swarm Optimization* dan *Simulated Annealing* untuk Menyelesaikan *Vehicle Routing Problem with Drone* / Hasan Aji Prawira, Budi Santosa

Peer reviewed under responsibility of Universitas Muhammadiyah Sidoarjo.

© 2021 Universitas Muhammadiyah Sidoarjo. All Right reserved. This is an open access article under the CC BY licence (<http://creativecommons.org/licenses/by/4.0/>)

waktu kedatangan *drone* di lokasi  $j$  ditambah waktu perjalanan *drone*, waktu melayani pelanggan, dan waktu pemulihan *drone*. Persamaan 20 sampai 23 memastikan sinkronisasi waktu truk dan *drone*. Persamaan 24 memastikan waktu terbang *drone* tidak melebihi batas waktu terbang *drone*. Pada persamaan 25 dan 26 variabel biner  $p_{ij}^v$  didefinisikan untuk menentukan urutan rute truk, menunjukkan apakah suatu pelanggan dikunjungi sebelum pelanggan lainnya. Persamaan 27 memastikan tidak ada peluncuran *drone* baru ketika *drone* sedang terbang. Persamaan 28 sampai 31 menunjukkan bentuk variabel.

## B. Pengembangan Algoritma *Particle Swarm Optimization*

*Particle Swarm Optimization* (PSO) merupakan salah satu metode metaheuristik atau teknik optimasi global. PSO pertama kali diperkenalkan oleh [18]. Algoritma PSO meniru perilaku dari sekumpulan organisme contohnya segerombolan ikan ataupun burung. PSO merupakan salah satu teknik metaheuristik berdasarkan populasi. PSO telah diimplementasikan untuk menyelesaikan permasalahan-permasalahan sulit dan dapat menghasilkan solusi yang kompetitif [15]. Algoritma PSO dikembangkan untuk menyelesaikan VRPD dengan menggabungkannya dengan algoritma *Route Drone*. Algoritma *Route Drone* membantu untuk merubah vektor posisi PSO yang berbentuk vektor kontinu menjadi rute kendaraan VRPD. Berikut ini merupakan langkah-langkah algoritma PSO yang telah dikembangkan untuk menyelesaikan VRPD. Pertama, penentuan data permasalahan. Data permasalahan yang digunakan pada algoritma ini adalah *node* depot dan pelanggan, *demand* setiap pelanggan, kapasitas kendaraan, jumlah kendaraan, dan rute truk dari solusi VRP. Selanjutnya, penentuan parameter yaitu  $N$  merupakan jumlah individu setiap iterasi.  $It_{max}$  merupakan maksimum iterasi yang dilakukan.  $c_p$  merupakan parameter bobot perubahan kecepatan berdasarkan *pbest*.  $c_g$  merupakan parameter bobot perubahan kecepatan berdasarkan *gbest*. LB merupakan batas bawah koordinat. UB merupakan batas atas koordinat. Selanjutnya perhitungan matriks jarak. Matriks jarak berisi nilai jarak antar setiap pelanggan dan depot. Perhitungan jarak setiap pelanggan dan depot berdasarkan titik koordinat yang telah kumpulkan sebelumnya. Perhitungan dilakukan dengan menggunakan rumus jarak *euclidean*. Selanjutnya, inisialisasi vektor posisi, kecepatan, dan *pbest* awal. Inisialisasi vektor posisi awal dilakukan untuk setiap partikel  $x_i$  dengan membangkit nilai *random* sejumlah  $n$  pelanggan yang bisa dilayani oleh *drone*. Inisialisasi kecepatan awal dilakukan dengan memberi angka nol dengan panjang vektor yang sama dengan panjang vektor posisi awal untuk semua partikel  $v_i^1 = 0$ , dan *pbest* ditetapkan sama dengan vektor posisi awal  $p_i^1 = x_i^1$ . Selanjutnya, pembentukan rute *drone*. Pembentukan rute *drone* dilakukan berdasarkan pembangkitan partikel  $x_i$  yang telah dilakukan sebelumnya. Pembentukan rute ini dilakukan dengan menggunakan algoritma *Route Drone*. Selanjutnya, perhitungan nilai *fitness*. Untuk setiap partikel  $i = 1 \dots I$ , evaluasi nilai *fitness*  $\varphi(x_i^t)$  berdasarkan nilai biaya transportasi yang ditempuh dari rute *drone* yang terbentuk. Selanjutnya, perbarui vektor *pbest*. Pembaruan vektor *pbest* untuk setiap partikel  $i = 1 \dots I$  berdasarkan persamaan berikut.

$$p_i^t = \begin{cases} x_i^t & \text{jika } \varphi(x_i^t) < \varphi(p_i^t) \\ p_i^t & \text{sebaliknya} \end{cases} \quad (32)$$

Vektor *pbest*  $p_i^t = x_i^t$  jika nilai *fitness* posisi partikel kurang dari nilai *fitness* *pbest*. Selanjutnya, perbarui vektor *gbest*. Pembaruan vektor *gbest*  $g^t$  berdasarkan vektor *pbest*  $p_i^t$  dengan nilai *fitness* paling kecil. Selanjutnya, perbarui kecepatan dan vektor posisi. Pembaruan kecepatan dan posisi setiap partikel  $i$  berdasarkan persamaan berikut.

$$v_i^{t+1} = v_i^t + c_p u(p_i^t - x_i^t) + c_g u(g^t - x_i^t) \quad (33)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (34)$$

dengan  $u \sim U(0,1)$ .  $c_p$  dan  $c_g$  adalah parameter bobot perubahan kecepatan yang telah ditentukan sebelumnya.  $u$  adalah bilangan random dengan distribusi *uniform* 0 sampai 1. Selanjutnya, pengecekan kriteria pemberhentian.

---

Pengembangan Algoritma *Particle Swarm Optimization* dan *Simulated Annealing* untuk Menyelesaikan *Vehicle Routing Problem with Drone* / Hasan Aji Prawira, Budi Santosa

Peer reviewed under responsibility of Universitas Muhammadiyah Sidoarjo.

© 2021 Universitas Muhammadiyah Sidoarjo. All Right reserved. This is an open access article under the CC BY licence (<http://creativecommons.org/licenses/by/4.0/>)

Pemberhentian iterasi dilakukan jika kriteria pemberhentian terpenuhi, jika belum maka kembali ke langkah pembentukan rute. Pada penelitian ini menggunakan dua kriteria pemberhentian yang digunakan yaitu standar deviasi dari nilai *fitness* individu telah kurang dari 1 dan jumlah iterasi. Jika kriteria pemberhentian terpenuhi dilakukan penentuan rute terbaik. Rute terbaik dibentuk dengan menggunakan algoritma *Route Drone* berdasarkan vektor *gbest*. Hasil keluaran algoritma berupa rute truk baru dan rute *drone*. Algoritma dijalankan untuk setiap rute truk dari solusi VRP. Solusi algoritma dari masing-masing rute truk dikumpulkan menjadi kesatuan solusi VRPD.

### C. Pengembangan Algoritma *Simulated Annealing*

*Simulated Annealing* (SA) merupakan salah satu metode metaheuristik iteratif atau teknik optimasi global yang dikembangkan berdasarkan algoritma Monte Carlo. SA pertama kali diperkenalkan oleh [19] untuk menyelesaikan permasalahan *Traveling Salesman Problem* (TSP). Algoritma SA meniru cara kerja proses *annealing* yang biasa terjadi dalam bidang metalurgi. Algoritma SA dikembangkan untuk menyelesaikan VRPD dengan menggabungkannya dengan algoritma *Route Drone*. Algoritma *Route Drone* membantu untuk merubah vektor posisi SA yang berbentuk vektor diskrit menjadi rute kendaraan VRPD. Berikut ini merupakan langkah-langkah algoritma SA yang telah dikembangkan untuk menyelesaikan VRPD. Pertama, penentuan data permasalahan. Data permasalahan yang digunakan pada algoritma ini adalah *node* depot dan pelanggan, *demand* setiap pelanggan, kapasitas kendaraan, jumlah kendaraan, dan rute truk dari solusi VRP. Selanjutnya, dilakukan penentuan parameter yang akan digunakan yaitu *It\_max* merupakan maksimum iterasi yang dilakukan,  $T_0$  merupakan parameter temperatur awal,  $\alpha$  merupakan parameter penurunan suhu, LB merupakan batas bawah koordinat, dan UB merupakan batas atas koordinat. Kemudian, dilakukan perhitungan matriks jarak. Matriks jarak berisi nilai jarak antar setiap pelanggan dan depot. Perhitungan jarak setiap pelanggan dan depot berdasarkan titik koordinat yang telah kumpulan sebelumnya. Perhitungan dilakukan dengan menggunakan rumus jarak *euclidean*. Kemudian, dilakukan inisialisasi vektor solusi sementara. Inisialisasi vektor posisi awal *s* dilakukan dengan membangkit urutan acak sejumlah pelanggan yang dapat dilayani *drone*. Vektor ini digunakan untuk menentukan prioritas pelanggan dalam penentuan rute *drone*. Kemudian, dilakukan pembentukan vektor baru. Pembentukan vektor baru dilakukan dengan cara memodifikasi solusi sementara. Pembentukan vektor baru ini dilakukan dengan tiga cara yaitu membalik (*flip*), menukar (*swap*), dan menggeser (*slide*). Tiga cara ini dilakukan berdasarkan teori yang ada di buku Santosa & Ai (2017). Selanjutnya, dilakukan pembentukan rute *drone*. Pembentukan rute *drone* dilakukan berdasarkan solusi sementara *s* dan solusi baru  $s_{new}$ . Pembentukan rute ini dilakukan dengan menggunakan algoritma *Route Drone*. Evaluasi nilai *fitness*  $\varphi(s)$  dari solusi sementara dan  $\varphi(s_{new})$  dari solusi baru berdasarkan nilai biaya transportasi dari keseluruhan rute yang terbentuk. Selanjutnya, perbarui vektor Solusi berdasarkan temperatur. Pembaruan vektor solusi sementara berdasarkan persamaan berikut.

$$s = \begin{cases} s_{new} & \text{jika } u < e^{\frac{\varphi(s_{new})-\varphi(s)}{T}} \\ s & \text{sebaliknya} \end{cases} \quad (35)$$

dengan  $u \sim U(0,1)$ . Solusi sementara  $s = s_{new}$  jika nilai acak *u* lebih kecil dari probabilitas penerimaan  $e^{\frac{\varphi(s_{new})-\varphi(s)}{T}}$ . Selanjutnya, perbarui vektor solusi berdasarkan *fitness*. Pembaruan vektor solusi sementara berdasarkan persamaan berikut.

$$s = \begin{cases} s_{new} & \text{jika } \varphi(s_{new}) < \varphi(s) \\ s & \text{sebaliknya} \end{cases} \quad (36)$$

Solusi sementara  $s = s_{new}$  jika nilai *fitness* solusi baru lebih baik dari solusi sementara. Selanjutnya, perbarui nilai parameter temperatur. Pembaruan nilai parameter temperatur berdasarkan penurunan suhu  $\alpha$ .

$$T = \alpha \times T \quad (37)$$

Pengembangan Algoritma *Particle Swarm Optimization* dan *Simulated Annealing* untuk Menyelesaikan *Vehicle Routing Problem with Drone* / Hasan Aji Prawira, Budi Santosa

Peer reviewed under responsibility of Universitas Muhammadiyah Sidoarjo.

© 2021 Universitas Muhammadiyah Sidoarjo. All Right reserved. This is an open access article under the CC BY licence (<http://creativecommons.org/licenses/by/4.0/>)

Selanjutnya, cek kriteria pemberhentian. Pemberhentian iterasi dilakukan jika kriteria pemberhentian terpenuhi, jika belum maka kembali ke langkah pembentukan vektor baru. Pada penelitian ini menggunakan kriteria pemberhentian yang digunakan yaitu jumlah iterasi. Jika kriteria pemberhentian terpenuhi maka dilakukan penentuan rute terbaik. Rute terbaik dibentuk dengan menggunakan algoritma *Route Drone* berdasarkan vektor solusi sementara. Hasil keluaran algoritma berupa rute truk baru dan rute *drone*. Algoritma dijalankan untuk setiap rute truk dari solusi VRP. Solusi algoritma dari masing-masing rute truk dikumpulkan menjadi kesatuan solusi VRPD.

#### D. Pengembangan Algoritma *Route Drone*

Algoritma *Route Drone* dikembangkan untuk membantu algoritma SA dalam menyelesaikan VRPD. Algoritma ini digunakan untuk mengubah vektor algoritma SA menjadi rute *drone*. Berikut ini merupakan garis besar tahapan algoritma *Route Drone* yang dikembangkan. Pertama, penentuan data masukan. Data masukan yang digunakan pada algoritma ini adalah urutan prioritas *drone*, rute truk, dan biaya rute. Selanjutnya, penentuan jumlah pelanggan yang dilayani *drone*. Jumlah pelanggan yang akan dilayani *drone* ditentukan berdasarkan jumlah pelanggan yang memiliki *demand* kurang dari kapasitas *drone*. Hal ini dilakukan karena tidak semua pelanggan bisa dilayani oleh *drone*. Penentuan jumlah pelanggan yang akan dilayani *drone*  $\beta$  dilakukan berdasarkan persamaan berikut.

$$\beta = \min(\max(c_{low}, \delta \cdot |C|), c_{lim}) \tag{38}$$

dengan  $\delta$  merupakan rasio jumlah pelanggan yang akan dilayani oleh *drone*,  $c_{low}$  dan  $c_{lim}$  adalah batas bawah dan batas atas jumlah pelanggan yang harus dilayani oleh *drone*, dan  $|C|$  adalah jumlah pelanggan yang memiliki *demand* kurang dari kapasitas *drone*. Selanjutnya, penghapusan pelanggan *drone* dari rute truk. Pelanggan yang memiliki urutan prioritas pertama sampai  $\beta$  dihapus dari rute truk. Pelanggan-pelanggan ini dihapus dari rute truk karena mereka akan dimasukkan ke rute *drone*. Selanjutnya, perhitungan biaya rute truk. Biaya rute truk dihitung kembali untuk penurunan biaya pada rute truk. Penurunan biaya ini akan menjadi *constraint* pada pemasukan pelanggan ke rute *drone*. Selanjutnya, pemasukan pelanggan ke rute *drone*. Pemasukan pelanggan ke rute *drone* dilakukan satu persatu sampai jumlah pelanggan yang telah ditentukan sebelumnya. Urutan pemasukan pelanggan dilakukan berdasarkan urutan prioritas pelanggan. Pemasukan pelanggan ini menggunakan algoritma *cheapest insertion*. Setiap pemasukan pelanggan ke rute *drone* dilakukan pengecekan terhadap *constraint*. Selanjutnya, pengecekan *constraint*. *Constraint* yang digunakan pada algoritma ini adalah total biaya rute *drone* harus lebih kecil dari penurunan biaya rute truk, waktu rute *drone* untuk setiap pelanggan harus lebih kecil dari pada waktu ketahanan *drone*, *drone* hanya bisa melayani satu pelanggan setiap rute. Jika *constraint* terpenuhi maka dilakukan perhitungan total biaya rute. Total biaya dihitung berdasarkan total biaya rute truk dan total biaya rute *drone*.

### HASIL DAN PEMBAHASAN

#### A. Desain Eksperimen

Data yang digunakan berdasarkan skenario yang ada di dalam penelitian [8]. Dalam penelitian [8] data kasus yang terdiri dari data jumlah pelanggan, jumlah barang, koordinat pelanggan, biaya operasional truk, biaya operasional *drone*, kecepatan truk, kecepatan *drone*, waktu peluncuran, dan waktu pendaratan *drone* yang dijelaskan pada **Tabel 1**.

**Tabel 1.** Parameter Data Uji

Parameter	Nilai	Satuan
Waktu peluncuran	1	menit

Pengembangan Algoritma *Particle Swarm Optimization* dan *Simulated Annealing* untuk Menyelesaikan *Vehicle Routing Problem with Drone* / Hasan Aji Prawira, Budi Santosa

Peer reviewed under responsibility of Universitas Muhammadiyah Sidoarjo.

© 2021 Universitas Muhammadiyah Sidoarjo. All Right reserved. This is an open access article under the CC BY licence (<http://creativecommons.org/licenses/by/4.0/>)

Waktu pendaratan	1	menit
Kecepatan truk	35	mil/jam
Kecepatan drone	50	mil/jam
Batas waktu terbang drone	30	menit
Kapasitas truk	1300	kg
Kapasitas drone	5	kg
Biaya operasional truk	0.1274	euro/km
Biaya operasional drone	0.0127	euro/km

Penelitian ini mengevaluasi kinerja kedua metode PSO dan SA dengan 32 set data masalah acak. Data skenario mempunyai skala yang berbeda, dari skala kecil hingga skala besar, dengan 6 sampai 100 pelanggan. Setiap skenario data memiliki tingkat kerumitan yang berbeda-beda. Perbedaan tingkat kerumitan untuk setiap data skenario digunakan untuk menganalisis sensitivitas dari algoritma yang digunakan. Pengujian algoritma dengan skenario data diatas dilakukan dengan parameter yang telah dijelaskan sebelumnya. *Stopping criteria* yang digunakan adalah standar deviasi kurang dari  $10^{-5}$  dan jumlah maksimum iterasi. Parameter yang digunakan untuk algoritma PSO adalah  $N = 50$ ,  $It\_max = 100$ ,  $c_p = 0.8$ , dan  $c_g = 0.8$ . Parameter yang digunakan pada algoritma SA adalah  $It\_max = 5000$ ,  $k = 1$ , dan  $\alpha = 0.9$ . Hasil eksperimen kedua algoritma dibandingkan dengan solusi *best known* dari penelitian Sacramento dkk. (2019).

## B. Analisis Hasil Komputasi

Algoritma PSO dan SA dapat digunakan untuk menyelesaikan *Vehicle Routing Problem with Drone* dengan cara mengkombinasikannya dengan algoritma *Route Drone*. Algoritma *Route Drone* digunakan untuk mengubah struktur solusi PSO dan SA menjadi solusi *Vehicle Routing Problem with Drone*. Dari hasil eksperimen pada 6 skenario jumlah pelanggan, algoritma PSO dan algoritma SA dapat menghasilkan solusi yang mendekati solusi optimal. Hal tersebut dapat dilihat dari nilai gap dan nilai *Relative Percentage Deviation* antara hasil dari eksperimen dan hasil dari referensi pada **Tabel 2**.

**Tabel 2.** Nilai Gap dan RPD Semua Skenario

Skenario	PSO-Referensi		SA-Referensi		Skenario	PSO-Referensi		SA-Referensi	
	RPD	Gap	RPD	Gap		RPD	Gap	RPD	Gap
6.5.1	0.00%	0.00	0.00%	0.00	20.5.1	-1.02%	-0.02	-1.99%	-0.04
6.5.2	0.00%	0.00	0.00%	0.00	20.5.2	-4.22%	-0.08	-4.22%	-0.08
6.5.3	-11.78%	-0.14	-11.78%	-0.14	20.5.3	0.06%	0.00	0.00%	0.00
6.5.4	0.11%	0.00	0.00%	0.00	20.5.4	0.00%	0.00	0.00%	0.00
10.5.1	0.00%	0.00	0.00%	0.00	50.10.1	-4.15%	-0.24	-6.61%	-0.39
10.5.2	0.00%	0.00	0.00%	0.00	50.10.2	3.28%	0.18	0.00%	0.00
10.5.3	-2.24%	-0.03	-2.24%	-0.03	50.10.3	4.16%	0.23	-0.01%	0.00
10.5.4	0.00%	0.00	0.00%	0.00	50.10.4	10.64%	0.55	-0.66%	-0.03
12.5.1	0.00%	0.00	0.00%	0.00	100.10.1	14.14%	0.97	1.33%	0.09
12.5.2	0.00%	0.00	0.00%	0.00	100.10.2	8.72%	0.66	0.86%	0.07
12.5.3	0.00%	0.00	0.00%	0.00	100.10.3	12.27%	0.88	0.69%	0.05
12.5.4	0.00%	0.00	0.00%	0.00	100.10.4	5.80%	0.43	-0.25%	-0.02
<b>Rata-rata</b>						<b>-1.16%</b>	<b>-0.01</b>	<b>-1.17%</b>	<b>-0.01</b>

Pada **Tabel 2** dapat dilihat bahwa rata-rata RPD untuk algoritma PSO dan SA dibandingkan nilai referensi memiliki nilai dibawah 5% yaitu sebesar 1,49% dan -1.04%. Hal ini menunjukkan bahwa algoritma PSO dan SA dapat menghasilkan solusi yang mendekati solusi dari referensi. Pada skenario 6 pelanggan algoritma PSO dan SA mampu menghasilkan solusi yang sama dan lebih baik dari referensi. Hal ini dapat dilihat pada skenario 6.5.1, 6.5.2, dan 6.5.4 kedua algoritma memiliki RPD yang bernilai 0,00% serta pada skenario 6.5.3 kedua algoritma memiliki nilai RPD negatif yaitu -11,78%. Nilai RPD negatif menunjukkan bahwa algoritma yang dikembangkan mampu menghasilkan solusi yang lebih baik dari referensi. Berdasarkan hasil dapat disimpulkan bahwa algoritma PSO dan SA yang dikembangkan dapat menemukan solusi optimal apabila jumlah pelanggan sedikit. Sama halnya pada skenario 10 dan 12 pelanggan, algoritma PSO dan SA yang dikembangkan dapat

Pengembangan Algoritma *Particle Swarm Optimization* dan *Simulated Annealing* untuk Menyelesaikan *Vehicle Routing Problem with Drone* / Hasan Aji Prawira, Budi Santosa

Peer reviewed under responsibility of Universitas Muhammadiyah Sidoarjo.

© 2021 Universitas Muhammadiyah Sidoarjo. All Right reserved. This is an open access article under the CC BY licence (<http://creativecommons.org/licenses/by/4.0/>)

menghasilkan solusi yang sama bahkan dapat menghasilkan solusi yang lebih baik dari pada referensi pada beberapa skenario. Tetapi pada skenario 20 pelanggan yaitu pada skenario 20.5.3, algoritma PSO yang dikembangkan hanya dapat menghasilkan solusi yang mendekati solusi referensi. Sedangkan untuk algoritma SA masih mampu menghasilkan solusi yang sama dan lebih baik dari referensi. Pada skenario 50 pelanggan algoritma PSO hanya dapat menghasilkan solusi yang mendekati solusi referensi kecuali pada skenario 50.10.1 algoritma PSO mampu menghasilkan solusi yang lebih baik dari referensi. Sedangkan untuk algoritma SA masih mampu menghasilkan solusi yang sama dan lebih baik dari referensi. Pada skenario 100 pelanggan, kedua algoritma hanya dapat menghasilkan solusi yang mendekati solusi referensi kecuali pada skenario 100.10.4 algoritma SA mampu menghasilkan solusi yang lebih baik dari referensi. Algoritma PSO dan SA tidak dapat menghasilkan solusi yang sama dengan referensi dikarenakan dalam pencarian solusi ada kemungkinan terjebak pada solusi lokal optimal.

Algoritma PSO dan SA yang dikembangkan menggunakan metode heuristik yang dikembangkan sendiri oleh peneliti yaitu algoritma *Route Drone*. Metode ini juga merupakan metode *local search* sehingga solusi yang dihasilkan merupakan solusi lokal optimal. Pada algoritma *Route Drone* ini tidak mengakomodasi pembuatan rute baru oleh truk sehingga rute *drone* hanya berpaku pada rute yang dihasilkan dari penyelesaian VRP. Hal tersebut menyebabkan terdapat potensi rute *drone* yang tidak bisa dilakukan. Sehingga pembentukan rute *drone* sangat bergantung dari rute kendaraan awal yang dihasilkan dari penyelesaian VRP. Oleh karena itu penggunaan metode heuristik pada pembentukan rute *drone* berpengaruh pada kemungkinan solusi yang dihasilkan oleh algoritma PSO dan SA yang dikembangkan terjebak pada solusi lokal optimal.

Pada **Tabel 2** dapat dilihat bahwa nilai RPD SA-Referensi memiliki nilai yang lebih kecil dari pada nilai RPD PSO-Referensi. Hal tersebut menunjukkan bahwa solusi yang dihasilkan oleh algoritma SA lebih baik dari pada solusi yang dihasilkan oleh algoritma PSO. Hal tersebut dapat disebabkan kurang tepatnya pembuatan struktur solusi algoritma karena ketidaksesuaian algoritma dengan permasalahan. Permasalahan VRPD merupakan permasalahan kombinatorial yang berupa urutan pelanggan. Algoritma PSO merupakan algoritma yang pertama kali dikembangkan untuk menyelesaikan permasalahan kontinyu. Tetapi algoritma PSO telah banyak digunakan untuk menyelesaikan permasalahan kombinatorial dengan mengubah struktur nilai kontinyu yang dihasilkan algoritma PSO menjadi prioritas urutan. Perubahan struktur ini tidak lebih bagus dalam menyelesaikan permasalahan kombinatorial jika dibandingkan algoritma SA yang pada dasarnya dikembangkan untuk menyelesaikan permasalahan kombinatorial ataupun kontinyu. Selain pengaruh ketidaksesuaian algoritma, perbedaan hasil algoritma PSO dan SA juga disebabkan oleh adanya langkah-langkah khusus untuk keluar dari lokal optimal pada algoritma SA. Pada algoritma SA terdapat suatu langkah penerimaan individu berdasarkan peluang penerimaan meskipun solusi individu tersebut lebih buruk dari individu sebelumnya. Peluang penerimaan ini dipengaruhi oleh parameter temperatur pada algoritma SA. Semakin tinggi temperatur iterasi maka semakin besar peluang penerimaan. Langkah tersebut dilakukan dengan harapan penerimaan individu tersebut dapat menuju individu yang lebih baik pada iterasi selanjutnya. Langkah tersebut membuat arah pergerakan individu pada algoritma SA dapat keluar dari lokal optimal dimana hal tersebut tidak dilakukan pada algoritma PSO.

## SIMPULAN

Algoritma PSO dan SA berhasil dikembangkan untuk menyelesaikan *Vehicle Routing Problem with Drone* dengan cara mengkombinasikan algoritma PSO dan SA dengan algoritma *Route Drone* untuk penentuan rute *drone*. Algoritma PSO dapat menghasilkan solusi yang lebih baik atau sama dengan solusi referensi pada skenario berukuran 6 – 12 pelanggan tetapi pada skenario 50 – 100 pelanggan algoritma PSO belum mampu menghasilkan solusi yang lebih baik atau sama dengan solusi referensi karena ada kemungkinan terjebak solusi lokal optimal. Algoritma SA dapat menghasilkan solusi yang lebih baik atau sama dengan solusi referensi pada skenario berukuran 6 – 50 pelanggan tetapi pada skenario 100 pelanggan terdapat beberapa skenario yang algoritma SA belum mampu menghasilkan solusi yang lebih baik atau sama dengan solusi referensi. Berdasarkan

---

Pengembangan Algoritma *Particle Swarm Optimization* dan *Simulated Annealing* untuk Menyelesaikan *Vehicle Routing Problem with Drone* / Hasan Aji Prawira, Budi Santosa

Peer reviewed under responsibility of Universitas Muhammadiyah Sidoarjo.

© 2021 Universitas Muhammadiyah Sidoarjo. All Right reserved. This is an open access article under the CC BY licence (<http://creativecommons.org/licenses/by/4.0/>)

nilai *Relavite Percentage Deviation* (RPD) dari hasil eksperimen pada 6 skenario jumlah pelanggan dapat disimpulkan bahwa algoritma SA menghasilkan solusi yang lebih baik daripada algoritma PSO.

## DAFTAR PUSTAKA

- [1] Rao, B., Gopi, A. G., & Maione, R. (2016). The societal impact of commercial drones. *Technology in Society*, 45, 83–90. <https://doi.org/10.1016/j.techsoc.2016.02.009>
- [2] Wohlsen, M. (2014). The Next Big Thing You Missed: Amazon’s Delivery Drones Could Work—They Just Need Trucks. Retrieved from <https://www.wired.com/2014/06/the-next-big-thing-you-missed-delivery-drones-launched-from-trucks-are-the-future-of-shipping/>
- [3] Rose, C. (2013). Amazon’s Jeff Bezos looks to the future. Retrieved from <https://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/>
- [4] Bryan, V. (2014). Drone delivery: DHL “parcelcopter” flies to German isle. Retrieved from <https://www.reuters.com/article/us-deutsche-post-drones/drone-delivery-dhl-parcelcopter-flies-to-german-isle-idUSKCN0HJ1ED20140924>
- [5] Mulia, K. (2019). JD warms Indonesia up to drone deliveries with first government-endorsed test flight. Retrieved from <https://kr-asia.com/jd-warms-indonesia-up-to-drone-deliveries-with-first-government-endorsed-test-flight>
- [6] Trop, J. (2016). Drone Delivery is About to Disrupt the Trucking Industry. Retrieved from <https://www.trucks.com/2016/06/21/drone-delivery-reshape-trucking/>
- [7] Toth, P., & Vigo, D. (2002). *Vehicle Routing: Problems, Methods, and Applications* (2nd ed.). SIAM - Society for Industrial and Applied Mathematics.
- [8] Sacramento, D., Pisinger, D., & Ropke, S. (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*, 102(March), 289–315. <https://doi.org/10.1016/j.trc.2019.02.018>
- [9] Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54, 86–109. <https://doi.org/10.1016/j.trc.2015.03.005>
- [10] Ponza, A. (2016). *Optimization of Drone-Assisted Parcel Delivery*. 80.
- [11] Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2018). On the min-cost Traveling Salesman Problem with Drone. *Transportation Research Part C: Emerging Technologies*, 86(May 2016), 597–621. <https://doi.org/10.1016/j.trc.2017.11.015>
- [12] Wang, Z., & Sheu, J. B. (2019). Vehicle routing problem with drones. *Transportation Research Part B: Methodological*, 122, 350–364. <https://doi.org/10.1016/j.trb.2019.03.005>
- [13] Ham, A. M. (2018). Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transportation Research Part C: Emerging Technologies*, 91(March), 1–14. <https://doi.org/10.1016/j.trc.2018.03.025>
- [14] Santosa, B., & Ai, T. J. (2017). *Pengantar Metaheuristik - Implementasi dengan Matlab* (1st ed.). Surabaya: ITS Tekno Sains.
- [15] Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization: An overview. *Swarm Intelligence*, 1(1), 33–57. <https://doi.org/10.1007/s11721-007-0002-0>
- [16] Ai, T. J., & Kachitvichyanukul, V. (2009). Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. *Computers and Industrial Engineering*, 56(1), 380–387. <https://doi.org/10.1016/j.cie.2008.06.012>
- [17] Schermer, D., Moeini, M., & Wendt, O. (2018). Algorithms for Solving the Vehicle Routing Problem with Drones. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10751 LNAI, 352–361. [https://doi.org/10.1007/978-3-319-75417-8\\_33](https://doi.org/10.1007/978-3-319-75417-8_33)

Pengembangan Algoritma *Particle Swarm Optimization* dan *Simulated Annealing* untuk Menyelesaikan *Vehicle Routing Problem with Drone* / Hasan Aji Prawira, Budi Santosa

Peer reviewed under responsibility of Universitas Muhammadiyah Sidoarjo.

© 2021 Universitas Muhammadiyah Sidoarjo. All Right reserved. This is an open access article under the CC BY licence (<http://creativecommons.org/licenses/by/4.0/>)

- 
- [18] Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 4, 1942–1948. <https://doi.org/10.1109/TST.2016.7442504>
- [19] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680. <https://doi.org/10.1126/science.220.4598.671>