Limits: Journal of Mathematics and Its Applications
E-ISSN: 2579-8936
P-ISSN: 1829-605X
Vol. 21, No. 3, Nopember 2024, 429-446
DOI: http://dx.doi.org/10.12962/limits.v21i3.21862

Modified Snow Avalanches Algorithm untuk *Vehicle Routing Problem*

Ayomi Sasmito [1] *, Jovanka Cathrynn [1], Michelle Tanaka [1], Maria Zefanya Sampe [1]
[1] Universitas Prasetiya Mulya, Edutown Kavling Edu 1 No.1, Kab. Tangerang, 15339
[1,2,3]Matematika, Sekolah STEM Universsitas Prasetiya Mulya Indonesia
e-mail:ayomi.sasmito@pmbs.ac.id

Diajukan: 15 Oktober 2024, Diperbaiki: 17 Oktober 2024, Diterima: 1 Nopember 2024

Abstrak

Algoritma metaheuristik sering digunakan untuk mengatasi berbagai masalah optimasi. Dalam beberapa tahun terakhir, banyak algoritma metaheuristik baru telah dikembangkan, seperti *snow avalanches algorithm* (SAA), yang terinspirasi oleh longsoran salju alami. SAA terdiri dari empat fase longsoran: longsoran karena lereng gunung yang curam, faktor manusia, kondisi cuaca lokal, dan hanya memiliki satu parameter kontrol. Seperti kebanyakan algoritma metaheuristik, SAA berpotensi terjebak dalam optimal lokal dikarenakan hanya memiliki satu parameter kendali sehingga dalam penelitian ini disajikan modifikasi SAA, yang disebut *modified* SAA (mSAA), yang mengintegrasikan metode *opposition-based learning* (OBL) dengan SAA untuk meningkatkan proses optimasi. Untuk memvalidasi kinerja mSAA, pengujian dilakukan pada berbagai teknik OBL untuk menentukan kombinasi terbaik dalam menyelesaikan masalah kompleks dan nonlinear yaitu *vehicle routing problem* (VRP) pada tiga jenis dataset VRP (set data D01, D02, dan D03). Kemudian hasilnya dibandingkan dengan *snow avalanches algorithm* (SAA), *hiking optimization algorithm* (HOA), *teaching learning-based optimization* (TLBO), dan *grey wolf optimizer* (GWO). Berdasarkan nilai rata-rata, standar deviasi, dan nilai terbaik metode mSAA bekerja dengan baik dan efektif dalam menyelesaikan VRP menggunakan kombinasi Quasi OBL dan $S_i = 0.6 + 0.4$ rand.

Kata Kunci: Metaheuristik, Optimisasi, *Opposition-based learning (OBL), Vehicle Routing Problem (VRP),Modified Snow Avalanches Algorithm (mSAA).*

Abstract

Metaheuristic algorithms are often used to tackle various optimization problems. In recent years, many new metaheuristic algorithms have been developed, such as the snow avalanches algorithm (SAA), which is inspired by natural snow avalanches. SAA consists of four avalanche phases: avalanches due to steep mountain slopes, human factors, local weather conditions, and it only has one control parameter. Like most metaheuristic algorithms, SAA has the potential to get trapped in local optima due to having only one control parameter. Therefore, this study presents a modification of SAA, called modified SAA (mSAA), which integrates the opposition-based learning (OBL) method with SAA to enhance the optimization process. To validate the performance of mSAA, tests were conducted on various OBL techniques to determine the best combination for solving complex and nonlinear problems, specifically the vehicle routing problem (VRP) on three types of VRP datasets (D01, D02, and D03 datasets). The results were then compared with the snow avalanches algorithm (SAA), hiking optimization algorithm (HOA), teaching learning-based optimization (TLBO), and grey wolf optimizer (GWO). Based on the average value, standard deviation, and best value, the mSAA method performed well and effectively in solving VRP using a combination of Quasi OBL and S_i=0.6+0.4 rand.

***Keywords**: Metaheuristic, Optimization, Opposition-based learning (OBL), Vehicle Routing Problem (VRP),Modified Snow Avalanches Algorithm (mSAA).*

1 Pendahuluan

Saat ini, terdapat hampir 2 miliar kendaraan, dan pada tahun 2040 diproyeksikan akan berlipat ganda menjadi 4 miliar. Pada saat itu 75% populasi dunia akan tinggal di daerah perkotaan, dengan lebih dari 50 kota memiliki populasi melebihi 10 juta. Kemacetan global ini berdampak buruk pada kualitas hidup perkotaan. Mengatasi masalah ini melibatkan beberapa strategi utama seperti manajemen layanan, rantai pasokan, perencanaan transportasi perkotaan, dan transportasi tenaga kerja. Manajemen kendaraan yang efektif, didukung oleh peralatan dan layanan yang tepat, sangatlah penting bagi perusahaan distribusi dan transportasi umum [1].

Pengangkutan barang, komoditas, dan manusia tetap menjadi masalah penting, seperti pada tahun 1959 ketika Dantzig dan Ramser pertama kali mengatasi perumuman dari *travelling salesman* problem [2]. Istilah "*Vehicle routing*" pertama kali digunakan pada tahun 1972 oleh Golden, dkk [3]. Masalah *vehicle routing problem* (VRP) bertujuan untuk mengoptimalkan pembuatan rute distribusi jaringan untuk mengirimkan barang ke pelanggan. Hal ini dilakukan dengan menggunakan armada kendaraan secara efisien untuk meminimalkan biaya transportasi. Pelanggan berada di lokasi yang berbeda.

Seiring berjalannya waktu, untuk mengatasi masalah NP-hard ini, berbagai pendekatan telah digunakan, yang secara umum dapat diklasifikasikan ke dalam tiga kategori; pendekatan eksak, heuristik, dan metaheuristik. Metode eksak mencakup algoritma seperti *branch and bound* [4], *branch and cut* [5], *branch and price* [6], dan *branch and cut and price* [7]. Pendekatan ini menggunakan pendekatan ilmiah dengan memecah masalah menjadi submasalah yang lebih kecil, yang memungkinkan perolehan solusi optimal melalui prosedur yang ketat dan logis. Kelemahan utama dari pendekatan yang akurat adalah kompleksitas waktu yang signifikan. Heuristik untuk VRP dibagi menjadi tiga kelas: dua fase, perbaikan iteratif [8], dan konstruksi. Heuristik dua fase mencakup pemisahan masalah menjadi dua tahap yang berbeda: menugaskan pelanggan ke rute dan menentukan urutan kunjungan pelanggan. Dua klasifikasi strategi seperti *Cluster first-Route second* dan *Route first-Cluster Second* telah dikembangkan [9]. Heuristik konstruksi dimulai dengan solusi kosong dan kemudian mengisinya dengan memeriksa alternatif yang layak. Jenis heuristik ini mencakup metode *savings* [10]. Akibatnya, algoritma dapat terperangkap dalam lokal optima.

Metaheuristik telah banyak digunakan untuk memecahkan berbagai versi VRP secara efisien. Metaheuristik utama yang digunakan untuk memecahkan VRP meliputi algoritma *differential evolution* (DE), *ant colony optimization* (ACO), *artificial bee colony* (ABC), *tabu search* (TS), *particle swarm optimization* (PSO), *simulated annealing* (SA), dan *genetic algorithm* (GA). Dalam dekade terakhir, beberapa algoritma metaheuristik baru telah muncul, sering kali

disembunyikan oleh ide-ide inovatif dan formulasi matematika. Terlepas dari ide-ide dan formulasi baru yang diklaim, kebenarannya tetap tidak berubah [11]. Efektivitas setiap algoritma metaheuristik bergantung pada dua komponen mendasar: intensifikasi, yang melibatkan pencarian lokal, dan diversifikasi, yang melibatkan pencarian global. Intensifikasi melibatkan pemeriksaan lokasi terdekat yang menunjukkan potensi untuk menemukan solusi yang lebih efektif. Sebaliknya, diversifikasi menjamin bahwa setiap bagian dari ruang pencarian telah dieksplorasi, yang memungkinkan algoritma untuk keluar dari setiap maksimum lokal [12].

Untuk meningkatkan kinerja pencarian, beberapa peneliti telah menggunakan teknik *opposition-based learning* (OBL) [13], [14], [15], [16]. Keuntungan utama OBL terletak pada kemampuannya untuk membuat solusi kandidat alternatif dari berbagai sisi, sehingga menyediakan cakupan bidang pencarian yang komprehensif. Dalam literatur terkini, banyak pendekatan OBL telah diperkenalkan, seperti Standard-OBL, General-OBL, Quasi Reflection-OBL, Centre-OBL, dan Optimal-OBL [13]. Beberapa algoritma metaheuristik telah ditemukan untuk menggabungkan OBL guna meningkatkan kinerja pencariannya. Ini termasuk *genetic algorithm* (GA), *differential evolution* (DE), *particle swarm optimization* (PSO), *biogeography-based optimization* (BBO), *harmony search* (HS), *ant colony system* (ACS), *gravitational search optimization* (GSO), dan lain-lain [17].

Sebagian besar penelitian yang melibatkan algoritma metaheuristik yang terintegrasi dengan OBL selama ini hanya digunakan untuk menghasilkan solusi alternatif bagi populasi awal sehingga ada kemungkinan terjebak pada optima lokal saat berada dalam ruang pencarian algoritma metaheuristik. Penelitian ini menyajikan pendekatan baru yang mengintegrasikan teknik OBL ke dalam *snow avalanches algorithm* (SAA) sebagai solusi alternatif pada populasi awal dan juga sebagai solusi alternatif setelah memperbarui populasi. *Snow avalanches algorithm* (SAA) dipilih karena mudah diimplementasikan dan hanya memiliki satu parameter. Selain itu, hasil yang cukup beragam menunjukkan bahwa kemampuan strategi SAA yang ada masih terbatas karena belum ada strategi yang lebih unggul. Upaya untuk mengatasi kekurangan tersebut dibenarkan melalui pencarian strategi baru yang memperhitungkan generasi baru algoritma metaheuristik yang dikembangkan.

Berdasarkan penjelasan sebelumnya, di dalam penelitian ini mSAA akan diterapkan untuk menyelesaikan masalah optimasi yang kompleks dan nonlinear yaitu VRP. Metode mSAA menggabungkan algoritma SAA dengan beberapa jenis *opposition-based learning* (OBL) sebagai bentuk solusi alternatif pada populasi awal dan setelah proses update. Pada penelitian ini disajikan beberapa jenis OBL seperti *standard* OBL, *general* OBL, *quasi* OBL, *centroid* OBL, dan *optimal* OBL yang dikombinasikan dengan SAA. Kemudian ditentukan kombinasi OBL terbaik

berdasarkan nilai *fitness* (fungsi objektif VRP) pada tiga ukuran dataset yaitu *small, medium* dan *large.* Kemudian hasilnya dibandingkan dengan empat metode algoritma metaheuristik seperti *snow avalanches algorithm* (SAA) [18], *hiking optimization algorithm* (HOA) [19], *teaching learning-based optimization* (TLBO) [20], dan *grey wolf optimizer* (GWO) [21]. Kemampuan eksplorasi dan eksploitasi dari mSAA dapat ditentukan berdasarkan nilai rata-rata, standar deviasi, dan nilai terbaik. Ini adalah format umum untuk penelitian ini. Pada bagian 2, kami menjelaskan mengenai metode penelitian. Setelah itu, hasil dan pembahasan disajikan pada bagian 3. Di akhir penelitian ini kesimpulan dan saran untuk penelitian lebih lanjut disertakan pada bagian 4.

2 Metode Penelitian

Pada bagian ini dijelaskan teori yang dibutuhkan untuk menyelesaikan VRP dengan mSAA, seperti *vehicle routing problem* (VRP), *snow avalanches algorithm* (SAA), dan *modified snow avalanches algorithm* (mSAA), beserta langkah-langkah SAA yang dimodifikasi dalam memecahkan masalah VRP.

2.1 *Vehicle Routing Problem* (VRP)

Vehicle routing problem (VRP) merupakan masalah kombinatorial yang digambarkan oleh graf $G(V, A)$. Diberikan notasi $V = \{v_0, v_1, \dots, v_n\}$ merepresentasikan himpunan titik dan $A = \{(v_i, v_j) |\ v_i, v_j \in V, i \neq j\}$ merepresentasikan himpunan busur yang menghubungkan titik v_i dan v_j dengan jarak (biaya) d_{ij}. Fungsi objektif dari model VRP adalah meminimumkan biaya transportasi, yang disajikan oleh persamaan berikut [2] :

$$z = min \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{N} d_{ij} x_{ijk} \tag{1}$$

dengan variabel keputusan

$$x_{ijk} = \begin{cases} 1, & \text{jika kendaraan } k \text{ mengunjungi pelanggan } j \text{ setelah } i \\ 0, & \text{jika kendaraan } k \text{ tidak mengunjungi pelanggan } j \text{ setelah } i \end{cases} \tag{2}$$

dan batasan kendala

a. Setiap pelanggan dikunjungi sekali oleh satu kendaraan

$$\sum_{i=0}^{N} \sum_{k=1}^{K} x_{ijk} = 1, j = 1,2 \dots, N, i \neq j, \tag{3}$$

b. Setelah mengunjungi pelanggan, setiap kendaraan akan meninggalkan pelanggan

$$\sum_{i=0}^{N} x_{ijk} - \sum_{i=0}^{N} x_{jik} = 0, k = 1,2, \dots, K, j = 0,1, \dots, N, \tag{4}$$

c. Kendaraan K dimulai dari depot

$$\sum_{j=1}^{N} x_{0jk} = 1, k = 1,2,\dots,K, \quad (5)$$

d. Kendaraan K akan kembali ke depot setelah melayani pelanggan terakhir

$$\sum_{i=1}^{N} x_{i0k} = 1, k = 1,2,\dots,K, \quad (6)$$

e. Total permintaan dalam satu rute tidak melebihi kapasitas kendaraan

$$\sum_{i=0}^{N}\sum_{j=1}^{N} q_j x_{ijk} \leq W, k = 1,2,\dots,K. \quad (7)$$

Sebagai tambahan berikut adalah definisi dari setiap variabel yang didefinisikan :

x_{ijk} : variabel keputusan kendaraan k melayani pelanggan j setelah pelanggan i,

d_{ij} : jarak pelanggan i dan pelanggan j,

q_i : permintaan pelanggan i,

W : kapasitas maksimum kendaraan,

N : jumlah pelanggan,

K : jumlah kendaraan yang tersedia pada depot,

i, j : indeks pelanggan,

k : indeks kendaraan.

2.2 *Snow avalanches algorithm* (SAA)

Snow avalanches algorithm (SAA) merupakan algoritma metaheuristik yang terinspirasi oleh fenomena longsoran salju. Pergerakan longsoran salju dikonseptualisasikan dan diibaratkan sebagai pergerakan populasi. Kebaruan dari SAA terletak pada keunikannya sebagai algoritma metaheuristik yang hanya melibatkan satu parameter kontrol, tidak seperti pada kebanyakan algoritma serupa yang melibatkan beberapa parameter. Hal ini membuat SAA relatif lebih mudah dan lebih fleksibel untuk diimplementasikan. SAA terdiri dari ukuran populasi N (tumpukan salju), yang dihasilkan secara acak dalam rentang $X_{\min}$ dan X_{maks}. Pendekatan ini mencerminkan adaptasi populasi terhadap variasi yang diperlukan dalam pencarian solusi optimal, menjadikan SAA sebagai metode yang meminimalkan kompleksitas implementasi.

Analisis menggunakan SAA melibatkan 4 tahap utama yang disebabkan oleh berbagai faktor penyebab longsoran salju. Keempat tahap tersebut melibatkan fase longsoran salju akibat lereng gunung, faktor manusia, kondisi cuaca setempat, dan kondisi normal. Pada awal implementasi SAA, populasi awal pada ruang pencarian diinisialisasi secara acak menggunakan Persamaan (8):

$$X_i: x_{i,j} = lb + r \cdot (ub - lb),\ i = 1,2,\dots,N,\ j = 1,2,\dots,D \quad (8)$$

dengan X_i merupakan populasi awal dari longsoran i pada ruang pencarian, $x_{i,j}$ merupakan dimensi ke j, N adalah ukuran populasi, D adalah ukuran dimensi permasalahan, r merupakan

bilangan real acak pada interval [0,1], dan ub dan lb masing-masing merupakan batas atas dan batas bawah.

Dalam SAA, ukuran kualitas solusi kandidat adalah nilai *fitness* sehingga anggota populasi mengarah ke nilai terbaik untuk nilai *fitness* yang dikenal sebagai populasi terbaik (X_{best}). Ketika i mencapai batas tertentu iterasi, langkah selanjutnya adalah menetapkan $X_{\text{best}}, X_{r_1}, X_{r_2}$, dan X_{r_3} dengan menerapkan randomisasi sesuai dengan jumlah indeks populasi dimana X_{best} adalah nilai terkecil yang didapatkan. Setelah itu, variabel S_i dan R_i akan dibangkitkan secara acak dari interval [0,1]. Jika $R_i < S_i$, maka hasil yang diberikan adalah *true*. Sementara, jika $R_i > S_i$ maka akan dihasilkan *false.* SAA terbagi menjadi empat fase yaitu :

Fase 1 : longsoran yang terjadi akibat lereng gunung

Fase pertama adalah longsoran yang disebabkan oleh kemiringan gunung. Pada tahap ini, diasumsikan X_{best} akan bertindak sebagai lereng gunung yang menyebabkan longsoran salju sehingga pergerakan massa salju akan menuju arah dan posisi yang optimal. Oleh karena itu, posisi terbaru X_i^{new} dapat dihasilkan dengan rumus berikut :

$$X_i^{new} = X_{\text{best}} + rand(1, D) \times (X_{r_1} - X_{r_2}) \quad (9)$$

dengan $rand(1, D)$ merupakan bilangan real acak pada interval [0,1] dengan ukuran $1 \times D$. Sementara X_{r_1} dan X_{r_2} merupakan populasi yang dipilih secara acak.

Fase 2 : longsoran yang terjadi akibat faktor manusia

Fase kedua adalah longsoran yang diakibatkan oleh faktor manusia. Pada tahapan ini, diasumsikan X_{r_3} merupakan tindakan manusia yang menyebabkan terjadinya longsoran salju sehingga pergerakan massa salju akan menuju arah dan posisi yang optimal. Oleh karena itu, posisi terbaru X_i^{new} dapat dihasilkan dengan rumus berikut :

$$X_i^{new} = X_{r_3} + rand(1, D) \times (X_{r_1} - X_{r_2}) \quad (10)$$

Fase 3 : longsoran yang terjadi akibat cuaca di wilayah setempat

Fase ketiga adalah fase longsoran salju yang terjadi akibat kondisi cuaca setempat. Longsoran salju biasa terjadi saat ada hujan dan angin. Pada tahap ini, X_i berperan signifikan sebagai faktor kondisi cuaca yang menyebabkan longsoran salju sehingga pergerakan massa salju terjadi ke arah optimal. Oleh karena itu posisi terbaru X_i^{new} dapat dihasilkan dengan rumus berikut :

$$X_i^{new} = X_i + rand(1, D) \times (X_{r_1} - X_{r_2}) \quad (11)$$

Fase 4 : kondisi normal

Fase keempat dan fase terakhir adalah fase tidak terjadinya longsoran salju dan massa salju disebabkan oleh faktor lain seperti angin atau hujan sehingga pergerakan massa salju terjadi ke arah optimal. Oleh karena itu posisi terbaru X_i^{new} dapat dihasilkan dengan rumus berikut :

$$X_i^{new} = X_i + rand(1, D) \times (X_{\text{maks}} - X_{\text{min}}) \quad (12)$$

Namun apabila hasil $R_i > S_i$ atau *false,* maka fase yang akan diimplementasikan adalah keempat atau terakhir. Jika $f(X_i^{new}) < f(X_i)$ maka $X_i = X_i^{new}$ sehingge $f(X_i) = f(X_i^{new})$. Sedangkan jika $f(X_i) < f(X_{\text{best}})$ maka $X_{\text{best}} = X_i$ sehingge $f(X_{\text{best}}) = f(X_i)$. Untuk prosedur lengkap dari SAA, dapat dilihat pada **Algoritma 1.**

Begin
Set parameter SAA: banyaknya iterasi maksimum $(Iter_{max})$, jumlah populasi (N), dan s_i
Bangkitkan secara acak populasi awal X_i $(i = 1,2, \dots, N)$
Hitung nilai *fitness* X_i, $f(X_i)$
While $t \leq Iter_{max}$ do
  For $i = 1$ to N
    Pilih $X_{\text{best}}, X_{r_1}, X_{r_2}, X_{r_3}$
    If $rand < s_i$
      Hitung X_i^{new} menggunakan Persamaan (9)
    else if $rand < s_i$
      Hitung X_i^{new} menggunakan Persamaan (10)
    else if $rand < s_i$
      Hitung X_i^{new} menggunakan Persamaan (11)
    else
      Hitung X_i^{new} menggunakan Persamaan (12)
    end if
    if $f(X_i^{new}) < f(X_i)$
      $X_i = X_i^{new}$
      $f(X_i) = f(X_i^{new})$
    end if
    if $f(X_i) < f(X_{best})$
      $X_{\text{best}} = X_i$
      $f(X_{\text{best}}) = f(X_i)$
    end if
  end for
end while
return the solusi terbaik (X_{best})

Algorithm 1. Pseudocode dari *snow avalanches algorithm* (SAA)

2.3 *Opposition-based learning* (OBL)

Pada subbab ini, teori mengenai *opposition-based learning* (OBL) dan variasinya yang digunakan dalam penelitian ini disajikan. OBL merupakan alat yang efektif dalam masalah optimisasi yang dikembangkan oleh Tizhoosh pada tahun 2005 [16]. Ide dasar dari metode OBL adalah untuk mengevaluasi kesesuaian estimasi dan estimasi balikan secara bersamaan, untuk menemukan solusi kandidat yang lebih unggul.

2.3.1 *Standard OBL*

Opposition-based learning (OBL) adalah sebuah konsep yang menyatakan bahwa kebalikan dari solusi saat ini mungkin lebih unggul daripada solusi saat ini itu sendiri. Tujuannya adalah

untuk meningkatkan kemungkinan menemukan solusi optimal $\underline{X}$ dengan memperbaiki solusi yang ada x dengan cara berikut:

$$\underline{X} = lb + ub - X \tag{13}$$

dengan lb dan ub masing-masing merupakan batas bawah dan batas atas X.

2.3.2 *General OBL*

General opposition-based learning (G-OBL) [19] menggunakan prinsip dasar OBL dengan mutasi Cauchy (yaitu bobot acak) untuk memfasilitasi keluarnya solusi dari minimum lokal.

$$\underline{X} = r * (lb + ub - X) \tag{14}$$

dengan $\boldsymbol{r}$ adalah bilangan real acak pada interval $[\mathbf{0}, \mathbf{1}]$.

2.3.3 *Quasi OBL (Q-OBL)*

Quasi opposition-based learning (Q-OBL) menghasilkan titik yang dipilih secara acak yang berada di antara kedua solusi balikan, yaitu titik pusat dan titik OBL dari $\boldsymbol{X}$**.** Q-OBL dicirikan sebagai berikut :

$$\underline{X} = \begin{cases} rand(C, lb + ub - X), & X < C \\ rand(lb + ub - X, C), & X > C \end{cases} \tag{15}$$

$$C = \frac{ub + lb}{2} \tag{16}$$

2.3.4 *Centroid OBL (C-OBL)*

Centroid opposition-based learning (C-OBL) menghasilkan titik-titik yang dipilih secara acak berdasarkan oposisi sentroid. C-OBL dicirikan sebagai berikut :

$$\underline{X} = 2.C - X \tag{17}$$

$$C = \frac{\sum_{i=1}^{n} X_i}{N} \tag{18}$$

dengan N merupakan ukuran populasi.

2.3.5 *Optimal OBL (O-OBL)*

Centroid opposition-based learning (C-OBL) menghasilkan titik-titik berdasarkan populasi terbaik saat ini. O-OBL dicirikan sebagai berikut :

$$\underline{X} = 2.X_{\text{best}} - X \tag{19}$$

dengan X_{best} populasi terbaik.

2.4 Modified SAA untuk menyelesaikan VRP

Implementasi utama algoritma metaheuristik yang digunakan dalam teknik yang disarankan untuk menyelesaikan *vehicle routing problem* (VRP) adalah *snow avalanches algorithm* (SAA). Untuk menghasilkan solusi yang berlawanan, SAA yang baru dirilis mengintegrasikan OBL. OBL yang dipilih untuk iterasi berikutnya ditentukan dengan mengevaluasi solusi kandidat saat ini dan solusi yang berlawanan secara bersamaan. Oleh karena itu, untuk meningkatkan kinerja SAA

secara efektif, metode untuk memilih kombinasi OBL dan SAA menjadi penting. Pseudocode SAA yang dimodifikasi dapat dilihat pada **Algoritma 2.**

Begin
Set SAA parameters: iterasi maksimum $(Iter_{max})$, jumlah populasi (N), and s_i
Bangkitkan secara acak populasi awal X_i $(i = 1,2, \dots, N)$
Bangkitkan populasi oposisi $\underline{X_i}$ $(i = 1,2, \dots, N)$
Hitung nilai *fitness* $f(X_i)$ dan $f(\underline{X_i})$
if $f(X_i) > f(\underline{X_i})$
  $X_i = \underline{X_i}$
  $f(X_i) = f(\underline{X_i})$
end if
While $t \leq Iter_{max}$ do
  For $i = 1$ to N
    Pilih $X_{best}, X_{r_1}, X_{r_2}, X_{r_3}$
    if $rand < s_i$
      Hitung X_i^{new} menggunakan Persamaan (9)
    else if $rand < s_i$
      Hitung X_i^{new} menggunakan Persamaan (10)
    else if $rand < s_i$
      Hitung X_i^{new} menggunakan Persamaan (11)
    else
      Hitung X_i^{new} menggunakan Persamaan (12)
    end if
    if $f(X_i^{new}) < f(X_i)$
      $X_i = X_i^{new}$
      $f(X_i) = f(X_i^{new})$
    end if
    Bangkitkan populasi oposisi $\underline{X_i}$ $(i = 1,2, \dots, N)$
    Hitung nilai fitnes setiap individu $f(\underline{X_i})$
    if $f(X_i) > f(\underline{X_i})$
      $X_i = \underline{X_i}$
      $f(X_i) = f(\underline{X_i})$
    end if
    if $f(X_i) < f(X_{best})$
      $X_{best} = X_i$
      $f(X_{best}) = f(X_i)$
    end if
  end for
end while
return solusi terbaik (X_{best})

Algorithm 2. Pseudocode dari *modified snow avalanches algorithm* (SAA)

3 Hasil dan Pembahasan

Studi dan temuan penyelesaian VRP yang memanfaatkan SAA yang dimodifikasi (mSAA) dengan *opposition-based learning* (OBL) ditunjukkan pada bagian ini. OBL diimplementasikan selama iterasi SAA awal dan digunakan mengikuti proses pembaruan posisi bola salju berdasarkan empat fase. Penggunaan teknik *opposition-based learning* (OBL) setelah proses pembaruan

bertujuan untuk memperoleh solusi pembaruan yang lebih baik dengan membandingkan posisi yang diperbarui dengan lawannya. Iterasi awal teknik OBL dimaksudkan untuk menghasilkan solusi kandidat awal yang lebih baik berdasarkan angka lawan yang diperoleh. PC yang dilengkapi dengan Intel Core i7 13700H @2.4GHz / RAM 16GB di bawah Windows 11. digunakan untuk mengodekan SAA yang ditingkatkan dalam C++ dan menjalankannya untuk menilai efektivitasnya dalam menyelesaikan VRP.

Tiga dataset dataset P VRP dipilih dari https://vrp.galgos.inf.puc-rio.br/index.php/en/ yang terdiri dari permintaan setiap pelanggan dan titik koordinat setiap pelanggan. Data yang digunakan adalah P-n20-k2, P-n76-k4, dan P-n101-k4. Untuk mempermudah penulisan setiap dataset ini dapat direpresentasikan dengan simbol D01, D02, dan D03.

SAA yang dimodifikasi digabungkan dengan lima bentuk OBL yang berbeda: *standard* OBL, *general* OBL, *quasi* OBL, *centroid* OBL, dan *optimal* OBL dengan tujuan menemukan pendekatan OBL terbaik dalam algoritma SAA. Dalam penelitian ini, ukuran populasi untuk algoritma ditetapkan sebesar 30, jumlah iterasi maksimum sebesar 500, dan setiap dataset dievaluasi dengan 30 kali uji coba secara independen. Selain itu, karena algoritma SAA memiliki parameter acak yang dibangkitkan dari bilangan riil acak dalam interval [0,1], maka percobaan dilakukan untuk lima nilai S_i yang berbeda, yaitu $0.1 + 0.9rand; 0.2 + 0.8\,rand; 0.4 + 0.6\,rand$, dan $0.8 + 0.2\,.rand$ dengan rand sebagai bilangan riil acak dalam interval [0,1]. Kombinasi optimal dari pengujian tersebut ditemukan dengan menghitung nilai rata-rata (avg), standar deviasi (st.dev), dan nilai terbaik (best) untuk setiap percobaan.

3.1 Hasil *running* pada data D01

Hasil untuk data D01 disajikan dalam bentuk nilai rata-rata (avg), standar deviasi (st.dev), dan nilai terbaik (best), dengan sistem penilaian untuk mengidentifikasi kombinasi optimal dari eksperimen.

Tabel 1. Hasil standar deviasi pada data D01

S_i	OBL				
	Standard	**General**	**Quasi**	**Centroid**	**Optimal**
0.1 +0.9 rand()	16.44	8.86	18.69	8.42	14.64
Rank	4	2	5	1	3
0.2 +0.8 rand()	9.26	9.33	13.12	14.01	8.46
Rank	2	3	4	5	1
0.4 +0.6 rand()	17.71	10.80	11.12	11.87	5.52
Rank	5	2	3	4	1
0.6 +0.4 rand()	15.17	14.06	13.57	14.24	9.27
Rank	5	3	2	4	1
0.8 +0.2 rand()	10.37	6.18	10.42	7.78	15.23
Rank	3	1	4	2	5

Tabel 2. Hasil rata-rata pada data D01

S_i	OBL				
	Standard	**General**	**Quasi**	**Centroid**	**Optimal**
0.1 +0.9 rand()	358.47	330.18	320.72	359.4	362.39
Rank	3	2	1	4	5
0.2 +0.8 rand()	354.28	325.34	319.90	359.02	367.95
Rank	3	2	1	4	5
0.4 +0.6 rand()	355.68	319.92	320.10	364.49	359.11
Rank	3	1	2	5	4
0.6 +0.4 rand()	364.37	315.03	317.36	362.77	364.19
Rank	5	1	2	3	4
0.8 +0.2 rand()	359.81	324.74	323.50	360.62	348.00
Rank	4	2	1	5	3

Tabel 3. Hasil nilai terbaik pada data D01

S_i	OBL				
	Standard	**General**	**Quasi**	**Centroid**	**Optimal**
0.1 +0.9 rand()	328.29	315.41	299.62	330.61	349.32
Rank	3	2	1	4	5
0.2 +0.8 rand()	335.77	313.29	306.32	332.52	358.26
Rank	4	2	1	3	5
0.4 +0.6 rand()	331.50	299.14	298.89	342.46	339.63
Rank	3	2	1	4	5
0.6 +0.4 rand()	351.54	293.97	291.47	346.49	349.97
Rank	5	2	1	3	4
0.8 +0.2 rand()	342.51	309.95	316.38	346.32	323.84
Rank	4	1	2	5	3

Berdasarkan Tabel 1, hasil standar deviasi dari 30 kali uji coba menunjukkan bahwa *Optimal* OBL memiliki simpangan baku terkecil dibandingkan dengan jenis OBL lainnya. Selanjutnya, untuk nilai S_i yang ditetapkan sebesar $0.4 + 0.6\,rand$, diperoleh simpangan baku terkecil dibandingkan dengan nilai S_i lainnya. Oleh karena itu, dapat disimpulkan bahwa untuk D01, standar deviasi terkecil, yang menunjukkan sebaran data terkecil di sekitar nilai rata-rata, dicapai dengan menggabungkan SAA yang dimodifikasi menggunakan Optimal OBL dan menetapkan nilai S_i sebesar $0.4 + 0.6\,rand$. *Quasi* OBL memiliki hasil uji coba rata-rata terkecil, menurut hasil rata-rata Tabel 2 dari 30 kali uji coba. Selanjutnya, jika dibandingkan dengan berbagai kombinasi S_i, nilai rata-rata terkecil ditemukan dengan nilai S_i yang ditetapkan sebesar $0.6 + 0.4\,rand$. Tabel 3 menunjukkan bahwa dari 30 kali uji coba, *Quasi* OBL menghasilkan hasil terbaik dengan nilai terbaik terkecil. Selain itu, nilai optimal terkecil diperoleh saat dibandingkan dengan nilai S_i yang berbeda dengan nilai S_i ditetapkan pada $0.6 + 0.4\,rand$.

3.2 Hasil *running* pada data D02

Hasil untuk data D02 disajikan dalam bentuk nilai rata-rata (avg), standar deviasi (st.dev), dan nilai terbaik (best), dengan sistem penilaian untuk mengidentifikasi kombinasi optimal dari eksperimen.

Tabel 4. Hasil standar deviasi pada data D02

S_i	OBL				
	Standard	**General**	**Quasi**	**Centroid**	**Optimal**
0.1 +0.9 rand()	29.69	27.30	48.66	26.39	32.10
Rank	3	2	5	1	4
0.2 +0.8 rand()	31.95	21.56	33.19	28.26	39.07
Rank	3	1	4	2	5
0.4 +0.6 rand()	37.78	19.29	48.24	33.31	29.23
Rank	4	1	5	3	2
0.6 +0.4 rand()	20.97	31.24	46.80	29.06	67.44
Rank	1	3	4	2	5
0.8 +0.2 rand()	66.34	30.60	33.85	37.76	33.39
Rank	5	1	3	4	2

Tabel 5. Hasil rata-rata pada data D02

S_i	OBL				
	Standard	**General**	**Quasi**	**Centroid**	**Optimal**
0.1 +0.9 rand()	2232.11	2109.45	2101.51	2228.79	2236.49
Rank	4	2	1	3	5
0.2 +0.8 rand()	2225.43	2121.92	2094.48	2212.38	2213.36
Rank	5	2	1	3	4
0.4 +0.6 rand()	2220.67	2132.36	2095.02	2218.85	2222.68
Rank	3	2	1	4	5
0.6 +0.4 rand()	2240.85	2112.86	2082.09	2231.29	2353.87
Rank	4	2	1	3	5
0.8 +0.2 rand()	2229.56	2116.88	2113.26	2219.61	2205.28
Rank	5	2	1	3	4

Tabel 6. Hasil nilai terbaik pada D02

S_i	OBL				
	Standard	**General**	**Quasi**	**Centroid**	**Optimal**
0.1 +0.9 rand()	2195.88	2066.22	2042.43	2189.33	2159.03
Rank	5	2	1	4	3
0.2 +0.8 rand()	2190.94	2090.69	2051.35	2162.79	2153.09
Rank	5	2	1	4	3
0.4 +0.6 rand()	2164.48	2109.69	2018.51	2166.99	2168.29
Rank	3	2	1	4	5
0.6 +0.4 rand()	2203.95	2056.89	1993.60	2186.05	2177.40
Rank	5	2	1	4	3
0.8 +0.2 rand()	2112.50	2052.66	2064.50	2133.46	2152.52
Rank	3	1	2	4	5

Berdasarkan Tabel 4, hasil standar deviasi dari 30 kali uji coba menunjukkan bahwa *general* OBL memiliki standar deviasi terkecil dibandingkan dengan jenis OBL lainnya. Lebih lanjut, untuk nilai S_i yang ditetapkan pada $0.4 + 0.6\,rand$, diperoleh standar deviasi terkecil dibandingkan dengan nilai S_i lainnya. Oleh karena itu, dapat disimpulkan bahwa untuk D02, standar deviasi terkecil, yang menunjukkan sebaran data terkecil di sekitar nilai rata-rata, dicapai dengan menggabungkan SAA yang dimodifikasi menggunakan *general* OBL dan menetapkan nilai S_i pada $0.4 + 0.6\,rand$. *Quasi* OBL memiliki hasil uji coba rata-rata paling sedikit, menurut hasil rata-rata Tabel 5 dari 30 kali uji coba. Lebih lanjut, jika dibandingkan dengan berbagai kombinasi S_i, nilai rata-rata terkecil ditemukan dengan nilai S_i yang ditetapkan pada $0.6 + 0.4\,rand$. Tabel 6 menunjukkan bahwa dari 30 kali uji coba, *Quasi* OBL menghasilkan hasil terbaik dengan nilai terbaik terkecil. Selain itu, nilai optimal terkecil diperoleh saat dibandingkan dengan nilai S_i yang berbeda dengan nilai S_i ditetapkan pada $0.6 + 0.4\,rand$.

3.3 Hasil *running* pada data D03

Hasil untuk data D03 disajikan dalam bentuk nilai rata-rata (avg), standar deviasi (st.dev), dan nilai terbaik (best), dengan sistem penilaian untuk mengidentifikasi kombinasi optimal dari eksperimen.

Table 7. Hasil standar deviasi pada data D03

S_i	OBL				
	Standard	**General**	**Quasi**	**Centroid**	**Optimal**
0.1 +0.9 rand()	37.15	40.01	23.59	78.18	103.11
Rank	2	3	1	4	5
0.2 +0.8 rand()	48.18	30.83	39.97	57.16	57.37
Rank	3	1	2	4	5
0.4 +0.6 rand()	54.41	31.29	31.83	47.53	47.61
Rank	5	1	2	3	4
0.6 +0.4 rand()	68.55	50.54	29.95	62.70	54.66
Rank	5	2	1	4	3
0.8 +0.2 rand()	40.47	42.45	53.59	49.92	37.80
Rank	2	3	5	4	1

Berdasarkan Tabel 7, hasil standar deviasi dari 30 kali uji coba menunjukkan bahwa *Quasi* OBL memiliki standar deviasi terendah jika dibandingkan dengan jenis OBL lainnya. Nilai S_i sebesar $0.4 + 0.6\,rand$ menghasilkan standar deviasi terkecil jika dibandingkan dengan nilai lainnya. Untuk D03, menggabungkan SAA yang dimodifikasi dengan *Quasi* OBL dan menetapkan nilai S_i sebesar $0.4 + 0.6\,rand$ menghasilkan penyebaran data terkecil di sekitar rata-rata.

Tabel 8. Hasil nilai rata-rata pada data D03

S_i	OBL				
	Standard	**General**	**Quasi**	**Centroid**	**Optimal**
0.1 +0.9 rand()	3038.76	2930.53	2915.33	3013.20	3012.09
Rank	5	2	1	4	3
0.2 +0.8 rand()	3042.08	2928.85	2891.33	2994.75	3039.60
Rank	5	2	1	3	4
0.4 +0.6 rand()	3061.47	2919.21	2932.41	3019.05	3059.97
Rank	5	1	2	3	4
0.6 +0.4 rand()	3071.33	2934.70	2873.76	3021.93	3053.93
Rank	5	2	1	3	4
0.8 +0.2 rand()	3045.72	2923.79	2906.08	3020.17	3062.35
Rank	4	2	1	3	5

Table 9. Hasil nilai terbaik pada D03

S_i	OBL				
	Standard	**General**	**Quasi**	**Centroid**	**Optimal**
0.1 +0.9 rand()	2973.58	2861.21	2828.57	3013.20	2839.67
Rank	4	3	1	5	2
0.2 +0.8 rand()	2957.63	2891.52	2928.57	2994.75	2930.95
Rank	4	1	2	5	3
0.4 +0.6 rand()	2968.20	2942.33	2873.26	3019.05	2992.92
Rank	3	2	1	5	4
0.6 +0.4 rand()	3023.18	2874.92	2824.36	3021.93	2960.87
Rank	5	2	1	4	3
0.8 +0.2 rand()	2995.78	2865.67	2939.52	3020.17	2984.99
Rank	4	1	2	5	3

Berdasarkan Tabel 8, hasil rata-rata dari 30 kali uji coba menunjukkan bahwa *Quasi* OBL memiliki hasil uji coba rata-rata terkecil. Selain itu, untuk nilai S_i yang ditetapkan sebesar $0.6 + 0.4\,rand$, diperoleh nilai rata-rata terkecil dibandingkan dengan kombinasi S_i lainnya. Berdasarkan Tabel 9, hasil terbaik dari 30 kali percobaan diperoleh dari *Quasi* OBL, yang memiliki nilai terbaik terkecil. Selanjutnya, untuk nilai S_i yang ditetapkan pada $0.6 + 0.4\,rand$, nilai terbaik terkecil dicapai dibandingkan dengan nilai S_i lainnya.

3.4 Perbandingan dengan Algoritma Metaheuristik lainnya

Pada bagian ini menyajikan analisis perbandingan *modified snow avalanches algorithm* (mSAA) dengan algoritma metaheuristik lainnya, termasuk SAA, *grey wolf optimizer* (GWO), *teaching learning-based optimization* (TLBO), dan *hiking optimization algorithm* (HOA).

Algoritma perbandingan ini dipilih karena karakteristik yang sama yaitu memiliki lebih sedikit parameter. Hasil SAA yang dimodifikasi yang digunakan untuk perbandingan ini adalah hasil yang diperoleh dari kombinasi algoritma *Quasi* OBL dan S_i yang ditetapkan ke $0.6 + 0.4\,rand$. Perbandingan dilakukan dengan memeriksa nilai rata-rata, standar deviasi, dan nilai terbaik untuk setiap algoritma dalam menyelesaikan *Vehicle Routing Problem* (VRP) untuk setiap set data. Hasil secara detail disajikan dalam Tabel 10.

Tabel 10. Perbandingan performansi algoritma

Algorithm	Terms	D01	D02	D03
SAA [18]	Average	352.12	2184.87	3009.55
	St.Dev	23.66	37.96	65.23
	Best	301.61	2123.77	2874.19
HOA [19]	Average	328.42	2104.86	2925.75
	St.Dev	7.47	34.01	32.40
	Best	313.06	2042.05	2853.73
GWO [21]	Average	328,92	2.137,17	2.925,26
	St.Dev	7,54	31,02	37,66
	Best	314,32	2.084,36	2.853,15
TLBO [20]	Average	303.33	2123.06	2890.95
	St.Dev	11.53	17.78	30.00
	Best	291.34	2098.19	2839.63
mSAA	Average	317.36	2082.09	2873.76
	St.Dev	13.57	46.8	29.95
	Best	291.47	1993.60	2824.36

Menurut Tabel 10, *modified* SAA secara konsisten mengungguli algoritma lain dalam hal mencapai solusi terbaik untuk setiap set data. Namun, dalam hal standar deviasi, *modified* SAA hanya mendominasi di D03, sementara kinerja rata-ratanya diungguli oleh TLBO di D01. Temuan ini menunjukkan bahwa kombinasi S_i dan teknik OBL yang dirancang dalam modifikasi ini secara efektif menghasilkan solusi kandidat yang lebih baik berdasarkan angka-angka yang berlawanan dan secara efektif menangani potensi optima lokal yang terjadi dalam versi SAA asli.

4 Simpulan

Berdasarkan program yang diproses, tiga dataset dikembangkan, yaitu D01, D02, dan D03. Dalam penelitian ini, program mengevaluasi skenario-skenario ini dengan mempertimbangkan beberapa metrik kinerja untuk menentukan pendekatan yang paling efektif. Metrik kinerja meliputi nilai rata-rata (avg), standar deviasi (st dev), dan nilai terbaik (best). Melalui analisis dan

perbandingan yang cermat dari metrik-metrik ini, penelitian ini mengidentifikasi bahwa kombinasi Quasi Oppositional-Based Learning (Quasi OBL) dan $S_i = 0.6 + 0.4\,rand$ menghasilkan hasil yang optimal. Kombinasi spesifik ini menunjukkan kinerja yang unggul, sebagaimana dibuktikan oleh metrik yang dipertimbangkan, dan dengan demikian disimpulkan sebagai strategi yang paling efektif untuk skenario yang diberikan. Akibatnya, penelitian ini menyoroti pentingnya kombinasi ini dalam mencapai hasil terbaik yang mungkin dalam implementasi algoritma SAA.

5 Daftar Pustaka

[1] E. M. Toro O., A. H. Escobar Z., and M. Granada E., "Literature review on the vehicle routing problem in the green transportation context," *Luna Azul*, no. 42, pp. 362–387, Dec. 2015, doi: 10.17151/luaz.2016.42.21.

[2] G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," *Manage Sci*, vol. 6, no. 1, pp. 80–91, 1959, doi: 10.1287/mnsc.6.1.80.

[3] B. L. Golden, T. L. Magnanti, and H. Q. Nguyen, "Implementing vehicle routing algorithms," *Networks*, vol. 7, no. 2, pp. 113–148, Jun. 1977, doi: https://doi.org/10.1002/net.3230070203.

[4] G. Laporte and Y. Nobert, "Exact Algorithms for the Vehicle Routing Problem**The authors are grateful to the Canadian Natural Sciences and Engineering Research Council (grants A4747 and A5486) and to the Quebec Government (FCAC grant 80EQ04228) for their financial support.," in *North-Holland Mathematics Studies*, vol. 132, S. Martello, G. Laporte, M. Minoux, and C. Ribeiro, Eds., North-Holland, 1987, pp. 147–184. doi: https://doi.org/10.1016/S0304-0208(08)73235-3.

[5] J.-F. O. Cordeau, "A Branch-and-Cut Algorithm for the Dial-a-Ride Problem," 2003.

[6] A. Pessoa, M. P. de Aragão, and E. Uchoa, "Robust Branch-Cut-and-Price Algorithms for Vehicle Routing Problems," in *The Vehicle Routing Problem: Latest Advances and New Challenges*, B. Golden, S. Raghavan, and E. Wasil, Eds., Boston, MA: Springer US, 2008, pp. 297–325. doi: 10.1007/978-0-387-77778-8_14.

[7] R. Baldacci, P. Toth, and D. Vigo, "Exact algorithms for routing problems under vehicle capacity constraints," *Ann Oper Res*, vol. 175, no. 1, pp. 213–245, 2010, doi: 10.1007/s10479-009-0650-0.

[8] M. Gendreau, A. Hertz, and G. Laporte, "New Insertion and Postoptimization Procedures for the Traveling Salesman Problem." [Online]. Available: https://www.jstor.org/stable/171722

[9] J. E. Beasley, "Route first—Cluster second methods for vehicle routing," *Omega (Westport)*, vol. 11, no. 4, pp. 403–408, 1983, doi: https://doi.org/10.1016/0305-0483(83)90033-6.

[10] G. Clarke and J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Oper Res*, vol. 12, no. 4, pp. 568–581, Aug. 1964, doi: 10.1287/opre.12.4.568.

[11] X.-S. Yang, "Chapter 1 - Introduction to Algorithms," in *Nature-Inspired Optimization Algorithms (Second Edition)*, X.-S. Yang, Ed., Academic Press, 2021, pp. 1–22. doi: https://doi.org/10.1016/B978-0-12-821986-7.00008-1.

[12] N. Pholdee and S. Bureerat, "Comparative performance of meta-heuristic algorithms for mass minimisation of trusses with dynamic constraints," *Advances in Engineering Software*, vol. 75, pp. 1–13, 2014, doi: https://doi.org/10.1016/j.advengsoft.2014.04.005.

[13] S. Gupta and K. Deep, "A hybrid self-adaptive sine cosine algorithm with opposition based learning," *Expert Syst Appl*, vol. 119, pp. 210–230, 2019, doi: https://doi.org/10.1016/j.eswa.2018.10.050.

[14] M. Ventresca and H. R. Tizhoosh, "Simulated Annealing with Opposite Neighbors," in *2007 IEEE Symposium on Foundations of Computational Intelligence*, 2007, pp. 186–192. doi: 10.1109/FOCI.2007.372167.

[15] M. Ventresca and H. R. Tizhoosh, "Opposite Transfer Functions and Backpropagation Through Time," in *2007 IEEE Symposium on Foundations of Computational Intelligence*, 2007, pp. 570–577. doi: 10.1109/FOCI.2007.371529.

[16] H. R. Tizhoosh, "Opposition-Based Learning: A New Scheme for Machine Intelligence," in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, 2005, pp. 695–701. doi: 10.1109/CIMCA.2005.1631345.

[17] S. Mahdavi, S. Rahnamayan, and K. Deb, "Opposition based learning: A literature review," *Swarm Evol Comput*, vol. 39, pp. 1–23, Apr. 2018, doi: 10.1016/j.swevo.2017.09.010.

[18] K. Golalipour *et al.*, "Snow avalanches algorithm (SAA): A new optimization algorithm for engineering applications," *Alexandria Engineering Journal*, vol. 83, pp. 257–285, Nov. 2023, doi: 10.1016/j.aej.2023.10.029.

[19] S. O. Oladejo, S. O. Ekwe, and S. Mirjalili, "The Hiking Optimization Algorithm: A novel human-based metaheuristic approach," *Knowl Based Syst*, vol. 296, p. 111880, Jul. 2024, doi: 10.1016/j.knosys.2024.111880.

[20] R. V Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011, doi: https://doi.org/10.1016/j.cad.2010.12.015.

[21] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014, doi: https://doi.org/10.1016/j.advengsoft.2013.12.007.