Penyelesaian Masalah *Mixed Integer Nonlinear Programming* Menggunakan Modifikasi *Salp Swarm Algorithm*

Anton Sudirman[1], Ahmad Belva Savero Ergaputra[2]
Institut Teknologi Bandung[1]
Institut Teknologi Bandung[2]
antonsudirman66@students.itb.ac.id

Abstrak — Optimisasi adalah proses menentukan solusi terbaik suatu permasalahan dalam berbagai bidang, seperti *engineering*, bisnis, olahraga dan industri. Beberapa contoh masalah optimisasi adalah meminimumkan konsumsi energi dan biaya, memaksimumkan keuntungan, dan memaksimumkan efisiensi kerja. Permasalahan dalam bidang *engineering* dan masalah praktis biasanya dapat dimodelkan sebagai masalah *Mixed Integer Nonlinear Programming* (MINLP). Makalah ini mengusulkan modifikasi *Salp Swarm Algorithm* (SSA) untuk menyelesaikan beberapa masalah optimisasi, mencakup bidang *engineering* dan olahraga. Contoh masalah yang dapat diselesaikan menggunakan modifikasi SSA adalah masalah desain pereduksi kecepatan. Perbedaan SSA dengan modifikasi SSA untuk masalah MINLP terletak pada penggunaan fungsi *round* dalam perhitungan. Eksperimen secara numerik dari modifikasi metode ini dapat digunakan untuk memperoleh solusi optimal dari masalah-masalah optimisasi seperti desain pereduksi kecepatan, dan optimisasi pemilihan pelari dalam lari estafet. Hal ini akan banyak berperan dalam menyelesaikan masalah-masalah optimisasi diberbagai bidang, termasuk dalam menghadapi tantangan Revolusi Industri 4.0.

Kata kunci: Optimisasi, Salp Swarm Algorithm, Mixed Integer Nonlinear Programming

I. PENDAHULUAN

Permasalahan *Mixed Integer Nonlinear Programming* (MINLP) sering muncul dalam berbagai bidang. Fokus dari permasalahan MINLP adalah menentukan solusi optimal dari suatu fungsi objektif yang dibatasi oleh satu atau lebih kendala. Beberapa masalah MINLP bersifat *non-linear* dan dapat menghasilkan fungsi objektif multi-modal sehingga metode optimisasi lokal seperti metode *steepest descent* solusinya tidak dapat digunakan. Oleh karena itu, metode optimisasi global harus digunakan untuk mendapatkan solusi dari permasalahan MINLP [1].

Beberapa metode optimisasi heuristik dan meta-heuristik untuk menyelesaikan masalah MINLP telah dikembangkan dalam beberapa artikel. Sebagai contoh Cagnina dkk yang menggunakan metode *Particle Swarm Optimization* (PSO) untuk menyelesaikan masalah optimisasi dalam bidang *engineering* [2], Garg menggunakan metode *Artificial Bee Colony Algorithm* (ABC) untuk menyelesaikan masalah optimisasi desain *engineering* struktural dengan kendala *non-linear* [3], dan Yan dkk mengembangkan metode *Line-up Competition Algorithm* (LCA) untuk menyelesaikan masalah *Mixed Integer Nonlinear Programming* [4]. Tujuan penelitian ini adalah menyelesaikan masalah *Mixed Integer Nonlinear Programming*. Metode yang digunakan adalah modifikasi *Salp Swarm Algorithm*.

II. METODE PENELITIAN

A. *Mixed-Integer Non-Linear Programming*

Bentuk umum masalah *Mixed-Integer Non-Linear Programming* (MINLP) dapat dituliskan sebagai

$$\begin{aligned} &\text{Minimumkan } f(\boldsymbol{x}) \qquad (1)\\ &\text{terhadap } g_i(\boldsymbol{x}) = 0,\ i = 1, \dots, M\\ &\qquad h_j(\boldsymbol{x}) \le 0,\ j = 1, \dots, N\\ &\qquad \boldsymbol{x} = \left(x_1, x_2, \dots, x_q, x_{q+1}, \dots, x_n\right)^T \end{aligned}$$

dengan $x_1, x_2, \dots, x_q$ merupakan bilangan bulat untuk suatu konstanta q.

Berdasarkan definisi fungsi penalti, masalah optimisasi dengan kendala (1) dapat ditransformasikan menjadi masalah optimisasi tanpa kendala. Definisikan

$$F(\boldsymbol{x}, \alpha_i, \beta_j) = f(\boldsymbol{x}) + \sum_{i=1}^{M} \alpha_i g_i^2(\boldsymbol{x}) + \sum_{j=1}^{N} \beta_j (maks\{h_i(\boldsymbol{x}), 0\})^2 \tag{2}$$

dengan α_i dan β_j adalah konstanta penalti. Konstanta tersebut dapat ditentukan sebesar 10^{10} sampai 10^{15} [5]. Dalam makalah ini akan digunakan $\alpha_i = \beta_j = 10^{15}$ untuk setiap i dan j.

B. Salp Swarm Algorithm

Metode *Salp Swarm Algorithm* (SSA) yang diperkenalkan oleh Mirjalili dkk pada tahun 2017 merupakan metode optimisasi meta-heuristik yang terinspirasi oleh alam yang digunakan untuk menyelesaikan berbagai masalah optimisasi baik yang memiliki fungsi uni-modal maupun multi-modal [6]. Metode optimisasi ini terinspirasi oleh perilaku kawanan salp di laut. Salp merupakan anggota dari keluarga *Salpidae* dan sangat mirip dengan ubur-ubur. Secara umum, salp hidup berkelompok dan membentuk kawanan yang disebut rantai salp. Rantai ini terbagi menjadi dua grup: *Leader* dan *Followers*. *Leader* menempati urutan pertama dari rantai, sedangkan sisanya adalah *Followers*.

Didefinisikan $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})^T \in \mathbb{R}^n, i = 1,2,\dots,m$, sebagai posisi salp dengan n adalah jumlah variabel dan m adalah jumlah salp. Posisi salp $\boldsymbol{x}_i \in \mathbb{R}^n, i = 1,2,\dots,m$, dibagi menjadi dua yaitu posisi *Leader* salp: $\boldsymbol{x}_i \in \mathbb{R}^n, i = 1,2,\dots,\frac{m}{2}$, dan posisi *Followers* salp: $\boldsymbol{x}_i \in \mathbb{R}^n, i = \left(\frac{m}{2}+1\right), \left(\frac{m}{2}+2\right) \dots, m$. Diasumsikan ada sumber makanan sebagai target yang akan dituju oleh kawanan salp. Sumber makanan salp didefinisikan sebagai $\boldsymbol{s} = (s_1, s_2, \dots, s_n)^T \in \mathbb{R}^n$ dan memiliki nilai fungsi objektif terbaik dari semua salp. Posisi *Leader* salp diperbarui menggunakan persamaan (3) [7]:

$$\boldsymbol{x}_i = \begin{cases} \boldsymbol{s} + c_1\big((\boldsymbol{ub} - \boldsymbol{lb})c_2 + \boldsymbol{lb}\big), & c_3 < 0.5 \\ \boldsymbol{s} - c_1\big((\boldsymbol{ub} - \boldsymbol{lb})c_2 + \boldsymbol{lb}\big), & c_3 \geq 0.5 \end{cases} \tag{3}$$

dengan $\boldsymbol{x}_i \in \mathbb{R}^n, i = 1,2,\dots,\frac{m}{2}$, merupakan posisi *Leader* salp, m adalah jumlah salp, $\boldsymbol{s}$ merupakan posisi sumber makanan, $\boldsymbol{lb}$ dan $\boldsymbol{ub}$ merupakan batas bawah dan batas atas dari $\boldsymbol{x}_i$, c_2 dan c_3 adalah bilangan acak pada selang [0,1] dan c_1 merupakan parameter yang penting untuk menyeimbangkan eksplorasi dan eksploitasi dan dituliskan sebagai:

$$c_1 = 2e^{\left(\frac{4t}{T}\right)^2} \tag{4}$$

dengan t merupakan iterasi dan T merupakan maksimum dari iterasi. Posisi *Followers* dari salp diperbarui menggunakan persamaan (5):

$$\boldsymbol{x}_i = \frac{1}{2}(\boldsymbol{x}_i + \boldsymbol{x}_{i-1}) \tag{5}$$

dengan $\boldsymbol{x}_i \in \mathbb{R}^n, i = \left(\frac{m}{2}+1\right), \left(\frac{m}{2}+2\right), \dots, m$, merupakan posisi *Followers* salp, dan m adalah jumlah salp. Pseudo-code dari metode *Salp Swarm Algorithm* dapat dilihat pada Gambar 1 [7].

```
Inisialisasi populasi salp: x_i, i = 1,2,...,m dengan mempertimbangkan ub
dan lb.
While (kondisi akhir belum terpenuhi)
Hitung nilai fungsi objektif dari masing-masing posisi salp
s=posisi salp terbaik
Perbarui c_1 menggunakan (4)
    For masing-masing salp (x_i)
            If (i = 1,2,...,m/2)
            Perbarui posisi Leader salp menggunakan (3)
            Else (i = (m/2 + 1), (m/2 + 2), ..., m)
            Perbarui posisi Followers salp menggunakan (5)
            End
    End
Posisi salp dirubah berdasarkan batas atas dan batas bawah dari x_i
End
Return s
```

GAMBAR 1. PSEUDO-CODE *SALP SWARM ALGORITHM*.

C. Modifikasi Salp Swarm Algorithm

Modifikasi yang dilakukan pada metode SSA untuk menyelesaikan masalah MINLP yaitu berupa penambahan fungsi *round* pada nilai-nilai dari populasi salp jika disyaratkan bilangan bulat sebelum dievaluasi ke fungsi objektif. Pseudo-code dari modifikasi *Salp Swarm Algorithm* dapat dilihat pada Gambar 2.

Inisialisasi populasi salp $\boldsymbol{x}_i, i = 1,2, \dots, m$ dengan mempertimbangkan $\boldsymbol{ub}$, dan $\boldsymbol{lb}$.
While (kondisi akhir belum terpenuhi)
    If $\boldsymbol{x}_i$ disyaratkan bilangan bulat
    round($\boldsymbol{x}_i$)
    end
Hitung nilai fungsi objektif dari masing-masing posisi salp
$\boldsymbol{s}$=posisi salp terbaik
Perbarui c_1 menggunakan (4)
    For masing-masing salp ($\boldsymbol{x}_i$)
        If ($i = 1,2, \dots, \frac{m}{2}$)
        Perbarui posisi *Leader* salp menggunakan (3)
        Else ($i = \left(\frac{m}{2}+1\right), \left(\frac{m}{2}+2\right), \dots, m$)
        Perbarui posisi *Followers* salp menggunakan (5)
        End
    End
Posisi Salp dirubah berdasarkan batas atas dan batas bawah dari $\boldsymbol{x}_i$.
End
Return $\boldsymbol{s}$

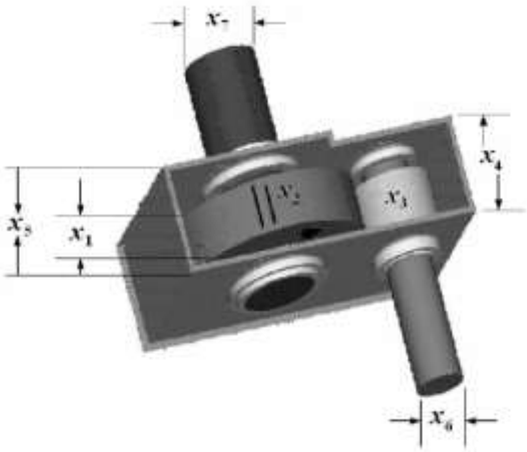
GAMBAR 2. PSEUDO-CODE MODIFIKASI *SALP SWARM ALGORITHM.*

III. HASIL DAN PEMBAHASAN

Eksperimen secara numerik menggunakan SSA dan modifikasi SSA diuji untuk masalah-masalah MINLP. Eksperimen ini dilakukan dengan menggunakan Laptop Pribadi dengan spesifikasi: Intel ® Celeron® CPU 1007U @1.50 GHz RAM 4GB dan aplikasi yang digunakan adalah MATLAB R2015a.

A. Masalah 1

Masalah ini diambil dari masalah *engineering* yaitu masalah optimisasi desain pereduksi kecepatan [2]. Desain pereduksi kecepatan dapat dilihat pada Gambar 3.



GAMBAR 3. DESAIN PEREDUKSI KECEPATAN.

Pada Gambar 3, x_1 adalah *face width*, x_2 adalah modul gigi, x_3 adalah jumlah gigi pada pinion, x_4 adalah panjang poros pertama antar bantalan, x_5 adalah panjang poros kedua antar bantalan, x_6 adalah diameter poros pertama dan x_7 adalah diameter poros kedua (semua variabel kontinu kecuali x_3 yang berupa bilangan bulat). Masalah optimisasi desain pereduksi kecepatan adalah meminimumkan kecepatan terhadap kendala-kendala: tekanan dari gigi, tegangan permukaan, defleksi melintang dari poros dan tekanan diporos. Secara matematis, masalah desain pereduksi kecepatan dapat diformulasikan sebagai:

$$\text{Minimumkan } f(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1..508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \quad (6)$$

terhadap

$$g_1(\boldsymbol{x}) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0$$
$$g_2(\boldsymbol{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0$$
$$g_3(\boldsymbol{x}) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0$$
$$g_4(\boldsymbol{x}) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0$$
$$g_5(\boldsymbol{x}) = \frac{1}{110x_6^3}\sqrt{\left(\frac{745x_4}{x_2x_4}\right)^2 + 16.9 \times 10^6} - 1 \leq 0$$
$$g_6(\boldsymbol{x}) = \frac{1}{85x_7^3}\sqrt{\left(\frac{745x_5}{x_2x_4}\right)^2 + 157.5 \times 10^6} - 1 \leq 0$$
$$g_7(\boldsymbol{x}) = \frac{x_2x_3}{40} - 1 \leq 0$$
$$g_8(\boldsymbol{x}) = \frac{5x_2}{x_1} - 1 \leq 0$$
$$g_9(\boldsymbol{x}) = \frac{x_1}{12x_2} - 1 \leq 0$$
$$g_{10}(\boldsymbol{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0$$
$$g_{11}(\boldsymbol{x}) = \frac{1.1x_7 + 1.9}{x_3} - 1 \leq 0$$

dengan $2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3, 7.8 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9$, dan $5 \leq x_7 \leq 5.5$.

Jika masalah optimisasi (6) diselesaikan dengan menggunakan metode SSA maka solusi yang diperoleh adalah $\boldsymbol{x} = (3.5, 0.7, 17.008, 7.30273, 7.82697, 3.35025, 5.28669)$ dengan waktu eksekusi 7.71 detik dan nilai fungsi objektif sebesar $f(\boldsymbol{x}) = 2998.3598$. Namun, jika dilihat dari solusi yang diperoleh, variabel x_3 bukan berupa bilangan bulat, hal ini tidak sesuai dengan syarat bahwa x_3 harus bilangan bulat. Oleh karena itu, modifikasi SSA akan diterapkan pada masalah ini.

Masalah optimisasi (6) dapat diselesaikan menggunakan modifikasi SSA sebanyak 100 kali eksekusi dengan populasi salp sebanyak 50 dan iterasi maksimum 1000, kemudian dipilih nilai fungsi objektif yang paling minimum. Solusi yang diperoleh dengan menggunakan modifikasi SSA adalah $\boldsymbol{x} = (3.5, 0.7, 17, 7.31831, 7.80461, 3.35025, 5.28668)$ dengan waktu eksekusi 7.74 detik. Nilai fungsi objektif berdasarkan solusi tersebut adalah $f(\boldsymbol{x}) = 2996.62$. Jika dilihat dari solusi yang diperoleh, variabel x_3 berupa bilangan bulat sehingga solusi yang diperoleh sudah optimal dengan nilai fungsi objektif 2996.62 hampir sama dengan yang diperoleh pada [2] sebesar 2996.35.

B. Masalah 2

Masalah ini diperoleh dari permasalahan olahraga, yaitu masalah pemilihan pelari dalam lari estafet 4×100 m [8]. Dalam masalah ini, kita mempunyai 4 pelari yang masing-masing berada pada setiap bagian dari lintasan 4×100 m. Para pelari tersebut dipilih dari grup yang beranggotakan 6 pelari untuk memperoleh tim yang memiliki waktu tercepat. Masing-masing pelari dalam suatu bagian dari lintasan beserta performanya dapat dilihat pada Tabel 1.

TABEL 1. WAKTU (DETIK) DARI SETIAP PELARI PADA MASING-MASING BAGIAN.

Pelari	Bagian			
	Bagian 1	***Bagian 2***	***Bagian 3***	***Bagian 4***
Pelari 1	12.27 s	11.57 s	11.54 s	12.07 s
Pelari 2	11.34 s	11.45 s	12.45 s	12.34 s
Pelari 3	11.29 s	11.50 s	11.45 s	11.52 s
Pelari 4	12.54 s	12.34 s	12.32 s	11.57 s
Pelari 5	12.20 s	11.22 s	12.07 s	12.03 s
Pelari 6	11.54 s	11.48 s	11.56 s	12.30 s

Pada permasalahan ini akan dipilih 4 dari 6 pelari sehingga total waktu yang diperlukan minimum. Pemilihan hanya bergantung pada persyaratan berikut: setiap pelari hanya ditugaskan pada satu bagian; setiap bagian hanya dapat dikerjakan oleh satu pelari. Masalah ini dapat diformulasikan sebagai berikut:

$$\text{Minimumkan } \sum_{j=1}^{4}\sum_{i=1}^{6}\tau_{ij}x_{ij} \tag{7}$$

$$\text{terhadap } \sum_{i=1}^{6}x_{ij} = 1, \forall j, 1 \leq j \leq 4, \text{ dan } \sum_{i=1}^{4}x_{ij} \leq 1, \forall i, 1 \leq i \leq 6,$$

$$x_{ij} = \begin{cases} 1, & \text{jika pelari ke} - i \text{ pada bagian ke} - j \\ 0, & \text{lainnya} \end{cases}$$

dengan τ_{ij} adalah waktu yang ditempuh pelari ke-*i* pada bagian ke-*j*.

Jika masalah optimisasi (7) diselesaikan dengan metode SSA maka solusi yang diperoleh adalah $\boldsymbol{x}$ =(0.24139, 0.5215, 0.09201, 0.0940, 0.0103, 0.0408, 0.0080, 0.0499, 0.1870, 0.0044, 0.4771, 0.2737, 0.4450, 0.1674, 0.1135, 0.0022, 0.0947, 0.1772, 0.2998, 0.2327, 0.0450, 0.0002, 0.4076, 0.0150) dengan waktu eksekusi 1.02 detik.Total waktu pelari yang diperoleh dari solusi tersebut adalah 51.34 detik. Namun, dapat dilihat bahwa solusi yang diperoleh bukan berupa bilangan bulat, hal ini tidak sesuai dengan syarat bahwa solusi harus bilangan bulat. Oleh karena itu, modifikasi metode SSA akan diterapkan pada masalah ini.

Masalah optimisasi (7) dapat diselesaikan menggunakan modifikasi SSA sebanyak 100 kali eksekusi dengan populasi salp sebanyak 50 dan iterasi maksimum 1000, kemudian dipilih nilai fungsi objektif yang paling minimum. Solusi yang diperoleh adalah $\boldsymbol{x} = (0,1,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,1,0,0)$ dengan waktu eksekusi 1.56 detik dan nilai fungsi objektif $f(\boldsymbol{x}) = 45.58$. Berdasarkan solusi yang diperoleh sudah berupa bilangan bulat sehingga solusi sudah optimal dan nilai fungsi objektifnya sebesar 45.58 lebih baik jika dibandingkan dengan yang diperoleh pada [8] sebesar 45.62. Pelari-pelari yang terpilih dapat dilihat pada Tabel 2.

TABEL 2. WAKTU MINIMUM YANG DIPEROLEH DARI PELARI-PELARI YANG TERPILIH.

Bagian				Total waktu
Bagian 1	***Bagian 2***	***Bagian 3***	***Bagian 4***	
Pelari 2 (11.34 detik)	Pelari 5 (11.22 detik)	Pelari 3 (11.45 detik)	Pelari 4 (11.57 detik)	45.58 detik

IV. SIMPULAN DAN SARAN

Modifikasi SSA untuk masalah MINLP terletak pada penggunaan fungsi *round* dalam perhitungan. Berdasarkan hasil dan pembahasan, dapat disimpulkan bahwa modifikasi SSA dapat digunakan untuk menyelesaikan masalah MINLP lebih baik dibandingkan dengan SSA tanpa modifikasi. Untuk penelitian selanjutnya, dapat dilakukan perbandingan antara metode modifikasi SSA dengan metode-metode optimisasi yang lain untuk menyelesaikan masalah MINLP.

DAFTAR PUSTAKA

[1] A. Kania, K. A. Sidarto, "Solving Mixed Integer Nonlinear Programming Problems Using Spiral Dynamics Optimization Algorithm," AIP Conference Proceedings, vol. 1716, 2016.

[2] L. C. Cagnina, S. C. Esquival, C. A. Coello Coello, "Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer," Informatica, vol. 32, pp. 319-326, 2008.

[3] H. Garg, "Solving Structural Engineering Design Optimization Problems Using Artificial Bee Colony Algorithm," Journal of Industrial and Management Optimization, vol. 10, pp. 777-794, 2014.

[4] L. Yan, K. Shen, S. Hun, "Solving Mixed Integer Nonlinear Programming Problems with Line-up Competition Algorithm," Computers & Chemical Engineering, vol. 28(12), pp.2647-2657, November 2004.

[5] X. S. Yang, Engineering Optimization, John Wiley & Sons Inc., New Jersey 2010.

[6] H. Faris, M. M. Mafarja, A. A. Heidari, I. Aljarah, A. Z. Alam, S. Mirjalili, H. Fujita, "An efficient binary Salp Swarm algorithm with crossover scheme for feature selection problems," Knowledge Based System, vol. 154, pp. 43-67, 2018.

[7] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili, "Salp Swarm algorithm: a bio-inspired optimizer for engineering design problems," Advanced Engineering Software, vol. 114, pp. 163-191, 2017.

[8] F. Masedu and M. Angelozzi, "Modelling Optimum Fraction Assigment in the 4x100 m Relay Race by Integer Linear Programming," Italian Journal of Sports Sciences, vol. 13, pp. 74-77, 2008.