

Implementasi Algoritma Neural Network Algorithm Pada Permasalahan Optimasi Berkendala Dalam Bidang Engineering

Author 1, Author 2

Jl. Jamin Ginting 578, Medan 20155

Jl. Ganesha 10, Bandung 40132

email 1, email 2

Abstract. PSO, GA, SDOA merupakan metode-metode metaheuristik yang dapat diandalkan untuk menyelesaikan permasalahan-permasalahan optimasi, selain implementasi yang sederhana metode ini juga memiliki laju kekonvergenan yang tidak dapat diragukan. Namun, metode tersebut kurang dapat diandalkan pada kasus dimana permasalahan semakin kompleks dan berdimensi besar, selain performa komputasi yang semakin meningkat, metode-metode tersebut juga memerlukan pemilihan parameter yang tepat untuk melakukan pemecahan setiap permasalahan optimasi. Sebagai alternatif dari permasalahan tersebut dalam penelitian ini diusulkan metode Neural Network Algorithm, yaitu sebuah metode metaheuristik yang terinspirasi dari konsep ANNs dan yang memiliki konsep free tuning/ tidak diperlukannya parameter-parameter pendukung dalam melakukan optimisasi. Neural Network Algorithm telah diuji pada permasalahan optimasi berdimensi besar dan dibandingkan hasilnya dengan hasil yang didapatkan menggunakan metode SDOA, EJAYA, WOA.

Keywords: ANNs, Neural Network Algorithm, fungsi penalti, engineering, optimasi berkendala

1 Pendahuluan

Optimisasi adalah suatu metode untuk menemukan suatu solusi optimal/ terbaik dari semua solusi yang tersedia (Amir Shabani dkk.2020). Permasalahan optimasi sering sekali ditemukan pada bidang engineering dan saat ini ada banyak algoritma optimasi yang telah dikembangkan untuk menyelesaikan permasalahan optimisasi khususnya pada bidang engineering. Terdapat dua kategori permasalahan optimasi, yaitu permasalahan optimasi dengan kendala dan tanpa kendala. Dalam bidang engineering kendala yang dimaksudkan dapat berupa batasan, baik berupa sumber daya yang ada yang akan dipakai untuk mendesaian atau merancang suatu sistem. Pada bidang spesifik seperti engineering, tujuan akhir dari suatu proses optimisasi adalah memaksimalkan suatu performa dari suatu sistem dengan meminimumkan sumber daya yang ada dalam perancangan sistem tersebut (Pradeep Jangir dkk. 2018).

Dalam penerapannya, algoritma yang diusulkan untuk menyelesaikan suatu permasalahan optimasi memiliki prinsip yang berbeda-beda, seperti algoritma Newton, Quasi Newton dan Conjugate Gradient (Jitsupa dkk. 2021), dalam proses optimisasi algoritma ini memanfaatkan informasi berupa turunan (Huiling Chen dkk. 2019) dari suatu fungsi yang akan di optimisasi. Algoritma dalam kategori ini tidak mampu menemukan titik-titik minimum atau maksimum global dari suatu permasalahan

optimasi, dan jika diimplementasikan pada permasalahan optimasi dalam bidang engineering yang pada umumnya terdiri dari fungsi-fungsi yang bersifat nonlinear dan berdimensi besar, algoritma dalam kategori diatas tidak efisien untuk menyelesaikan permasalahan yang ada.

Dalam hal efisiensi dan efektivitas pada saat melakukan optimisasi, beberapa algoritma-algoritma baru telah diusulkan oleh para peneliti dimana algoritma-algoritma yang diusulkan memanfaatkan peroses stokastik, diantaranya adalah Particle Swarm Optimization (Keneddy dkk. 1995), Genetic Algorithm (John,1992), Spiral Dynamic Optimization (Kenechi dkk. 2011), Bat Algorithm (Yang, 2012), Moth-Flame Optimizer (Mirjalim 2015), Ant Colony Optimization (Dorigo dkk. 2006), Cuckoo Search (Gandomi dkk. 2013), Henry Gas Solubility Optimization (Fatma dkk. 2019), Whale Optimization Algorithm (Mirjalili dkk. 2016), Differential Evolution (Rainer Storn dkk. 1997), Thermal exchange Optimization (Kaveh dkk.2017), Heat transfer search (Patel dkk. 2015) Water Evaporation Optimization (Kaveh dkk. 2016) Dynamic Virtual Bats Algorithm (Ali dkk. 2016), Cyclical Parthenogenesis Algorithm (Kaveh dkk. 2017) and Mine Blast Algorithm (Sadollah dkk. 2013). Algoritma dalam kategori ini memiliki konsep yang sederhana dan tidak membutuhkan informasi berupa turunan dari suatu fungsi yang akan di optimisasi dan dapat menemukan solusi optimum dari permasalahan optimasi (Nirmal Kumar dkk.2021).

Beberapa algoritma yang telah disebutkan diatas telah diimplementasikan untuk melakukan optimisasi pada permasalahan di bidang engineering diantaranya, (Jiang dkk. 2021) memanfaatkan algoritma Triple Tree-Seed Algorithm, (Chen dkk. 2019) memanfaatkan konsep multi-strategi pada algoritma sine cosine algorithm, (Zahara dkk. 2008) melakukan peroses hybrid Nelder–Mead simplex search dan particle swarm optimization, (Talatahari dkk. 2020) memanfaatkan chaos game optimization, (Kumar dkk. 2021) melakukan peroses hibrid algoritma cuckoo search dan particle swarm optimization dan (Pradeep dkk. 2018) memanfaatkan algoritma Non-Dominated Sorting Grey Wolf Optimizer untuk kasus multi-objektif. Beberapa penelitian dan algoritma yang disebutkan diatas, memiliki isu terkait pemilihan parameter-parameter yang tepat untuk melakukan peroses optimisasi, laju konvergensi dan akurasi dari hasil yang diberikan. Untuk menghindari pemilihan parameter yang tepat dalam melakukan peroses optimisasi dan untuk menghasilkan keakuratan hasil dari suatu permasalahan optimasi, pada penelitian ini Neural Network Algorithm (Sadollah dkk. 2018) diusulkan untuk menyelesaikan permasalahan optimasi berkendala dan penerapannya pada permasalahan optimasi berkendala dalam bidang engineering. Neural Network Algorithm merupakan algoritma optimasi yang berdasarkan pada konsep artificial neural network algorithm (ANNs) dan struktur biologi pada sistem otak manusia dan algoritma ini juga bersifat *free tuning*. Metode yang diusulkan akan diuji pada permasalahan optimasi berkendala dan fungsi benchmark dan 10 masalah optimasi berkendala pada bidang engineering..

2. NNA

Pada umumnya metode metaheuristik yang digunakan dalam menyelesaikan permasalahan optimasi dimulai dengan membangkitkan populasi awal. Neural Network Algorithm (NNA) yang diusulkan dalam penelitian ini juga melakukan hal yang sama dalam menyelesaikan permasalahan optimasi baik berkendala dan tanpa kendala, populasi awal dalam Neural Network Algorithm (NNA) disebut populasi dari pola solusi. Pada permasalahan dengan dimensi D, pola solusi dapat dituliskan kedalam persamaan berikut.

$$\text{pola solusi} = [x_1, x_2, x_3, \dots, \dots, \dots, x_n] \quad (1)$$

Untuk memulai melakukan optimasi, sebuah matrix dengan ukuran $N_{pop} \times D$ dibangkitkan, yang mana nilai dalam matriks tersebut diambil secara random atau acak dengan kendala berupa range yang diberikan pada setiap permasalahan, dalam penelitian ini range yang dimaksud dituliskan secara singkat menjadi LB dan UB yang masing-masing menyatakan batas bawah dan batas atas dari nilai yang diijinkan. Berikut merupakan ilustrasi pembangkitan populasi dari pola solusi, dimana N_{pop} merupakan jumlah populasi yang akan dibangkitkan dan D merupakan dimensi dari permasalahan yang ada.

$$populasi\ dari\ pola\ solusi = \begin{bmatrix} x_1^1 & \dots & x_D^1 \\ \vdots & \dots & \vdots \\ x_1^{n_{pop}} & \dots & x_D^{n_{pop}} \end{bmatrix} \quad (2)$$

Nilai objektif dari fungsi yang diberikan pada permasalahan didapatkan dengan memetakan fungsi tersebut terhadap pola solusi diatas, pemetaan tersebut diilustrasikan sebagai berikut.

$$C_i = f(x_1^1, x_2^2, x_3^3, \dots, x_D^i), i = 1, 2, 3, \dots, n_{pop} \quad (3)$$

Setelah melakukan perhitungan terhadap semua nilai objektif pada masing-masing pola solusi, kemudian Neural Network Algorithm melakukan pemilihan atau pencarian nilai minimum dari nilai objektif yang ada, hasil yang didapatkan disebut target solusi. Pada umumnya dalam konteks jaringan syaraf tiruan diperlukan sebuah bobot dalam fase training, demikian halnya dalam melakukan optimisasi menggunakan Neural Network Algorithm (NNA), pembangkitan bobot tersebut diilustrasikan sebagai berikut.

$$W(t) = [W_1, W_2, W_3, \dots, W_{n_{pop}}] = \begin{bmatrix} W_{1,1} & \dots & x_{n_{pop},1} \\ \dots & \dots & \dots \\ x_{1,n_{pop}} & \dots & x_{n_{pop},n_{pop}} \end{bmatrix} \quad (4)$$

Dimana bobot W merupakan matriks persegi yang berukuran $N_{POP} \times N_{POP}$, bobot diatas dibangkitkan dengan nilai acak pada range (0,1) dan kemudian bobot tersebut dilakukan pembaharuan secara bertahap seiring dengan iterasi yang ada. Dalam melakukan pembangkitan bobot, diberikan kendala ataupun batasan nilai yang diijinkan untuk digunakan sebagai nilai suatu bobot, hal ini bertujuan untuk mengontrol perilaku NNA dan bias yang ada. Secara matematis kendala yang diberikan dalam membangkitkan bobot tersebut adalah sebagai berikut.

$$\sum_{j=1}^{N_{pop}} w_{i,j}(t) = 1, i = 1, 2, 3, \dots, N_{pop} \quad (5)$$

Persamaan diatas memiliki makna bahwa jumlah nilai dari setiap bobot pada baris ke i harus bernilai satu. Setelah melakukan pembangkitan bobot yang memenuhi kendala pada persamaan 5, pola solusi baru kemudian dihitung menggunakan persamaan berikut, persamaan tersebut terinspirasi oleh kensep yang biasa ditemukan pada artificial neural networks (ANNs).

$$X_j^{new}(t+1) = \sum_{i=1}^{N_{pop}} w_{i,j}(t) X_i(t), j = 1, 2, 3, \dots, N_{pop} \quad (6)$$

$$X_i(t+1) = X_i(t) + X_i^{new}(t+1), i = 1, 2, 3, \dots, N_{pop} \quad (7)$$

Dimana t merupakan index dari iterasi. Setelah melakukan perubahan terhadap pola solusi yang didapatkan dari pola solusi sebelumnya, kemudian bobot matriks dilakukan pembaharuan nilai, pembaharuan nilai tersebut dapat diformulasikan kedalam persamaan

Bangkitkan sejumlah bilangan diantara nol dan satu secara acak yang akan digunakan sebagai bobot matrix.

Fase 5 :

Tetapkan target dari solusi yaitu nilai minimum yang didapatkan pada saat melakukan optimasi beserta bobot dari target tersebut.

Fase 6 :

Bangkitkan pola solusi baru dan lakukan perubahan pada pola solusi tersebut.

Fase 7 :

Lakukan pembaharuan pada nilai setiap bobot matrix yang diberikan pada iterasi sebelumnya.

Fase 8 :

Lakukan pengujian terhadap bias, dimana jika nilai acak menggunakan fungsi rand() lebih kecil dari nilai beta, perbaharui bias untuk menghasilkan pola solusi dan bobot baru.

Fase 9 :

Namun jika nilai random yang dibangkitkan lebih besar dari nilai beta, implementasikan fungsi operator (TF) untuk mendapatkan posisi baru dari pola solusi.

Fase 10 :

Hitung nilai objektif dari fungsi yang ada untuk semua pola solusi yang telah diperbaharui.

Fase 11 :

Lakukan perubahan terhadap target solusi dan bobot dari target solusi.

Fase 12 :

Lakukan perubahan terhadap nilai beta menggunakan formula reduksi, seperti yang disebutkan pada salah satu persamaan reduksi diatas.

Fase 13 :

Lakukan pengujian terhadap nilai maximum iterasi yang diharapkan, jika belum mencapai nilai maximum, kembali kepada fase 6 untuk mendapatkan target solusi beserta bobot baru yang lebih baik.

3. Percobaan Numerik

Untuk melakukan verifikasi terhadap performa dari metode yang diusulkan pada penelitian ini, kami melakukan sejumlah percobaan numerik yang diambil dari berbagai referensi terkait optimisasi dan membandingkan performa yang dihasilkan oleh metode yang diusulkan pada penelitian ini dengan berbagai metode yang telah diusulkan oleh para peneliti seperti Whale Optimization Algorithm (Mirjalili dkk.2016), Enhanced Jaya Algorithm (Zhang dkk.2021), Spiral Dynamic Optimization Algorithm (Tamura dkk. 2011), Social Network Search (Bayzidi dkk.2021) dan metode-metode lainnya. Pengujian dilakukan terhadap fungsi-fungsi yang terdiri dari $n=3, 30, 100$ dimensi dengan kondisi maximum iterasi pada fungsi berdimensi 100 adalah $K_{\max} = 1500$ dan maximum N_{POP} pada fungsi berdimensi 100 adalah 120. Pengujian dilakukan pada komputer dekstop dengan sistem operasi debian buster GNU/ Linux, RAM 4 Gb dengan bahasa pemrograman c++ dan compiler g++.

3.1 Percobaan Pada Permasalahan Engineering

Problem 1: Permasalahan Three-bars Truss

Pada permasalahan three-bars truss, tujuan utama dari permasalahan yang diberikan adalah meminimumkan bobot dari sebuah three-bars truss terhadap tiga kendala yang ada. Permasalahan tersebut dimodelkan secara matematis pada (Gandomi dkk. 2013). Hasil optimal yang didapatkan dalam penelitian ini adalah $f^*(x) = 263.8960$. Permasalahan ini telah diuji dengan menggunakan metode-metode lain seperti yang pernah dilakukan pada (Gandomi dkk. 2013) dengan menggunakan metode CS,

(Bayzidi dkk. 2021) menggunakan metode SNS dan (Melo dkk. 2013). Tabel 2 menunjukkan hasil performa yang didapatkan menggunakan beberapa metode komputasi yang berbeda dalam menyelesaikan permasalahan three-bars truss. Dari tabel tersebut terlihat bahwa hasil yang didapatkan pada penelitian ini mampu memberikan performa yang begitu baik dalam menyelesaikan permasalahan three-bars truss.

Problem 2: Permasalahan Piston Lever

Pada permasalahan piston lever, tujuan utama dari permasalahan yang diberikan adalah menempatkan komponen-komponen dari piston yaitu H, B, D dan X dengan meminimumkan volume dari minyak ketika lever dari piston tersebut diangkat dari sudut 0^0 sampai 45^0 . Permasalahan yang diberikan juga terdiri dari empat kendala, dimana tiap-tiap kendala bersifat nonlinear. Permasalahan tersebut dimodelkan secara matematis pada (Bayzidi dkk. 2021). Hasil optimal yang didapatkan dalam penelitian ini adalah $f^*(x) = 8.41285$. Permasalahan ini telah diuji dengan menggunakan metode-metode lain seperti yang pernah dilakukan pada (Hamzehkolaei dkk. 2021) dengan memanfaatkan metode MCEO, (Gandomi dkk. 2013) dengan menggunakan metode CS serta (Hamzehkolaei dkk. 2021) dengan memanfaatkan metode HPSO serta beberapa metode metaheuristik lainnya. Performa terbaik, rata-rata berserta standart deviasi dari permasalahan piston lever diberikan pada tabel 6. Performa dari metode yang diusulkan dalam penelitian ini mampu mengungguli keseluruhan metode yang pernah diusulkan dalam menyelesaikan permasalahan yang sama, kecuali dalam penelitian yang dilakukan pada (Hamzehkolaei dkk. 2021) metode yang diusulkan mampu memberikan hasil optimal yang lebih baik.

Problem 3: Permasalahan Cantilever beam

Struktur dari suatu permasalahan cantilever beam digambarkan pada (Bayzidi dkk. 2021). Pada gambar tersebut cantilever beam terdiri dari lima blok berongga persegi, permasalahan ini terdiri dari lima variable yang digunakan untuk mendesain suatu cantilever yang optimal dalam hal ini bobot optimal dari suatu cantilever beam. Permasalahan menentukan bobot optimal dari sebuah cantilever ini dibatasi dengan sebuah kendala yang bersifat nonlinear. Permasalahan menentukan bobot optimal dari sebuah cantilever beam diekspresikan secara matematis pada (Bayzidi dkk. 2021).

Problem 4: Permasalahan Speed Reducer

Permasalahan speed reducer pertama kali diperkenalkan oleh (Golinski 1973), dimana tujuan utama dari permasalahan tersebut adalah meminimumkan bobot dari sebuah speed reducer. Permasalahan meminimumkan bobot dari suatu speed reducer terdiri dari tujuh variable desain dan sebelas kendala yang keseluruhannya bersifat nonlinear. Pada kasus ini, permasalahan untuk menentukan bobot optimal dari speed reducer merupakan contoh dari sebuah permasalahan mix-integer programming dimana salah satu variable desain merupakan bilangan *integer* atau bilangan bulat. Permasalahan tersebut dimodelkan secara matematis pada (Hadaideh dkk. 2021). Hasil optimal yang didapatkan dalam penelitian ini adalah $f^*(x) = 2996.35$. Permasalahan ini telah diuji dengan menggunakan metode-metode lain seperti yang pernah dilakukan pada (Hadaideh dkk. 2021) dengan menggunakan metode RL-PSO, RL-DE, RL-NES, (Shabani dkk. 2020) dengan memanfaatkan metode MDE serta berbagai metode komputasi lainnya yang telah dirangkum pada tabel 4. Tabel tersebut menunjukkan performa dari metode yang diusulkan dalam penelitian ini mampu memberikan hasil optimal yang sangat baik dibandingkan dengan beberapa metode sebagaimana telah disebutkan diatas.

Problem 5: Permasalahan Pressure Vessel

Permasalahan pressure vessel pertama kali diusulkan oleh Kannan dan Krammer), target atau tujuan utaman dari permasalahan pressure vessel adalah melakukan optimisasi

pada sebuah pressure vessel dengan meminimumkan total biaya desain dari pressure vessel tersebut seperti biaya pembentukan, biaya material. Permasalahan desain pressure vessel terdiri dari 4 variabel desain dan terdiri dari empat kendala yang dijadikan batasan dalam melakukan desain dari sebuah pressure vessel yang masing-masing merupakan fungsi yang bersifat linear dan kendala yang bersifat nonlinear. Permasalahan tersebut dimodelkan secara matematis pada (Saddollah dkk. 2013). Hasil optimal yang didapatkan dalam penelitian ini adalah $f^*(x) = 5915.61$. Permasalahan ini telah diuji dengan menggunakan kurang lebih sekitar dua belas metode-metode komputasi yang berbeda seperti yang pernah dilakukan pada (Ferreira dkk. 2018) dengan menggunakan metode C-ITGO, (Ferreira dkk. 2018) dengan memanfaatkan metode IAPSO serta berbagai metode komputasi lainnya yang telah dirangkum pada tabel 3. Berdasarkan tabel tersebut dapat disimpulkan bahwa performa terbaik hanya didapatkan menggunakan metode yang diusulkan dalam penelitian ini.

Problem 6: Permasalahan Weldead Beam

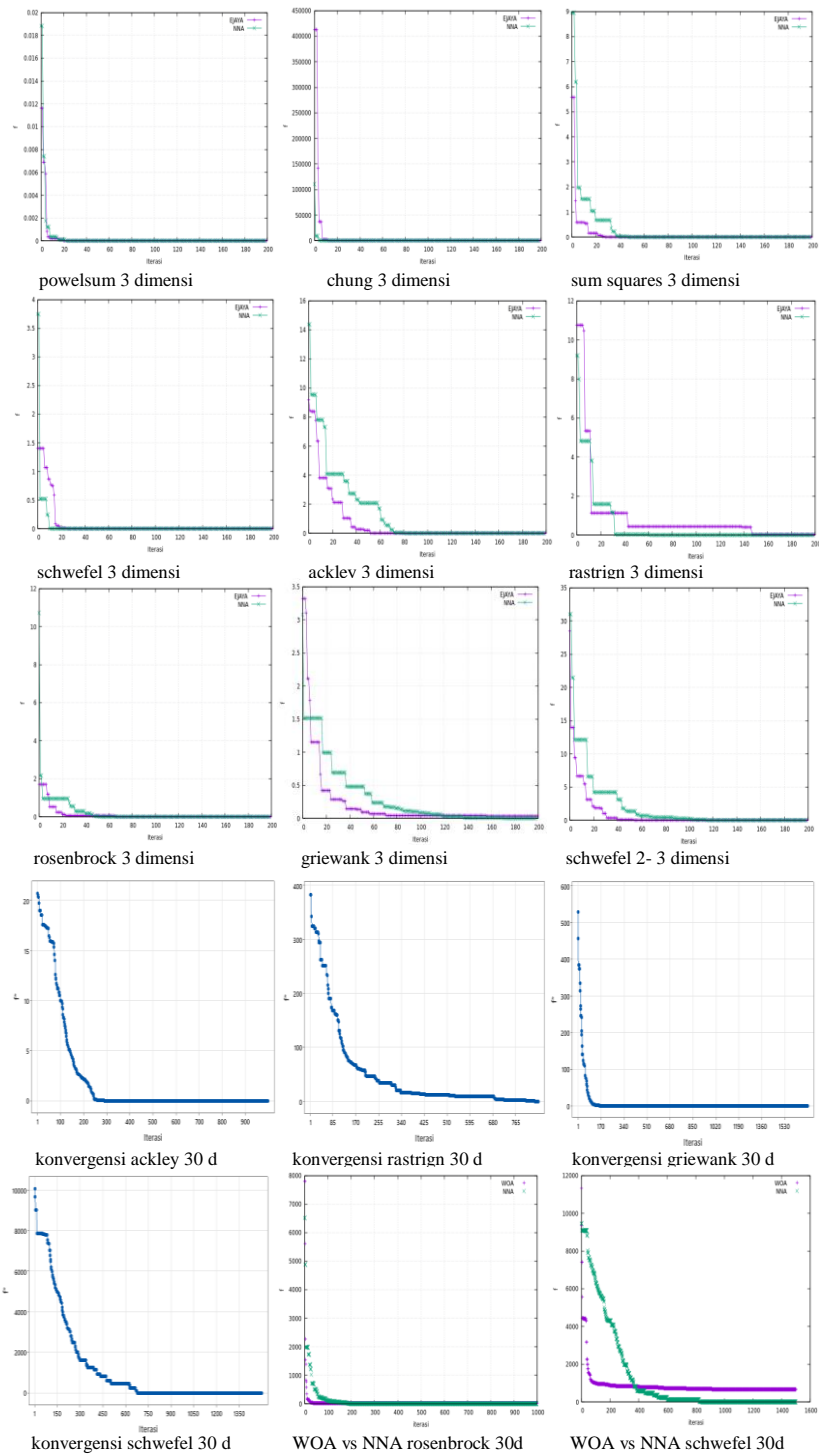
Permasalahan weldean beam pertama kali diperkenalkan oleh Debb, dimana tujuan utama dari permasalahan tersebut adalah meminimumkan biaya produksi dari suatu beam dengan mengoptimalkan pemilihan parameter desain dari beam tersebut. Permasalahan meminimumkan biaya produksi dari beam ini terdiri dari empat variable desain dengan tujuh kendala yang bersifat linear dan nonlinear. Struktur dari beam secara detail dijelaskan pada (Hadaideh dkk. 2021). Permasalahan tersebut dimodelkan secara matematis pada (Ferreira dkk. 2018). Hasil optimal yang didapatkan dalam penelitian ini adalah $f^*(x) = 1.72486$. Permasalahan ini telah diuji dengan menggunakan metode-metode lain seperti yang pernah dilakukan pada (Hadaideh dkk. 2021) dengan memanfaatkan metode RL-PSO, RL-DE, (Ferreira dkk. 2018) dengan memanfaatkan metode C-ITGO beserta metode-metode metaheuristik lainnya. Tabel 7 merupakan hasil perbandingan dari berbagai metode yang pernah dilakukan untuk menyelesaikan permasalahan weldead beam. Berdasarkan tabel tersebut dari plot konvergensi permasalahan weldead beam, terlihat bahwa metode yang diusulkan dalam penelitian ini mampu memberikan hasil yang cukup memuaskan dan performa yang dapat bersaing dengan metode-metode komputasi yang disebutkan dalam tabel 7.

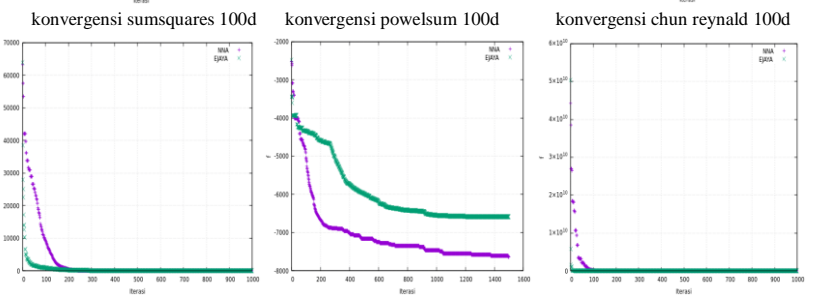
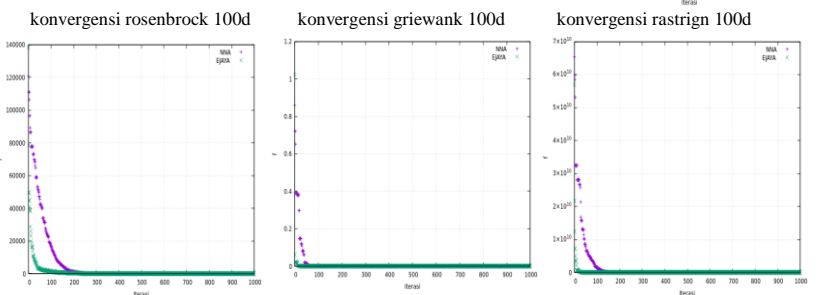
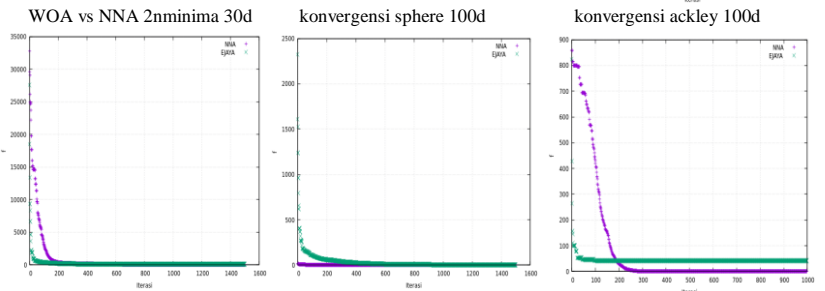
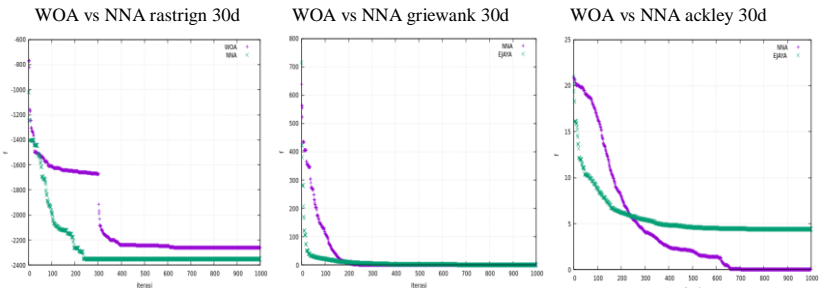
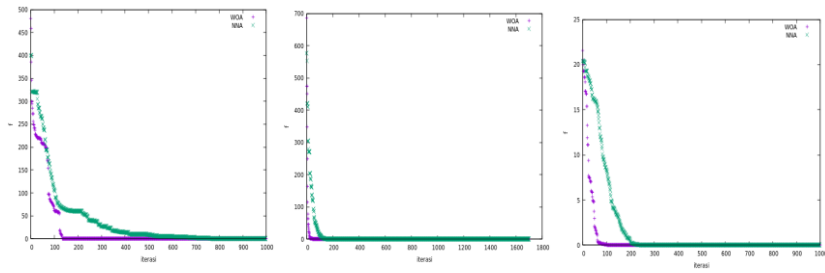
Problem 7: Permasalahan Tension/ Compression Design

Permasalahan tension/ compression desain pertama kali dideskripsikan atau diusulkan oleh arora. Permasalahan yang diusulkan dimaksudkan untuk melakukan optimisasi dari suatu bobot tension/ compression terhadap empat kendala. Variabel independent/ bebas yang digunakan dalam melakukan desain bobot suatu tension ini terdiri dari tiga variable dengan keseluruhan kendala yang diberikan dalam melakukan desain adalah fungsi-fungsi yang bersifat nonlinear. Permasalahan tersebut dimodelkan secara matematis pada (Talatari dkk. 2020). Hasil optimal yang didapatkan dalam penelitian ini adalah $f^*(x) = 0.0126947$. Permasalahan ini telah diuji dengan menggunakan metode-metode lain seperti yang pernah dilakukan pada (Talatari dkk. 2020) dengan memanfaatkan metode CGO, (Gandomi dkk. 2013) dengan menggunakan metode cuckoo search algorithm beserta metode komputasi lainnya. Hasil perbandingan dari setiap metode komputasi yang berbeda dirangkum dalam tabel 4. Berdasarkan tabel tersebut dapat dilihat bahwa metode yang diusulkan dalam penelitian ini mampu memberikan hasil yang akurat sampai empat digit pada metode yang diusulkan oleh (Talatari dkk. 2020), (Gandomi dkk. 2013).

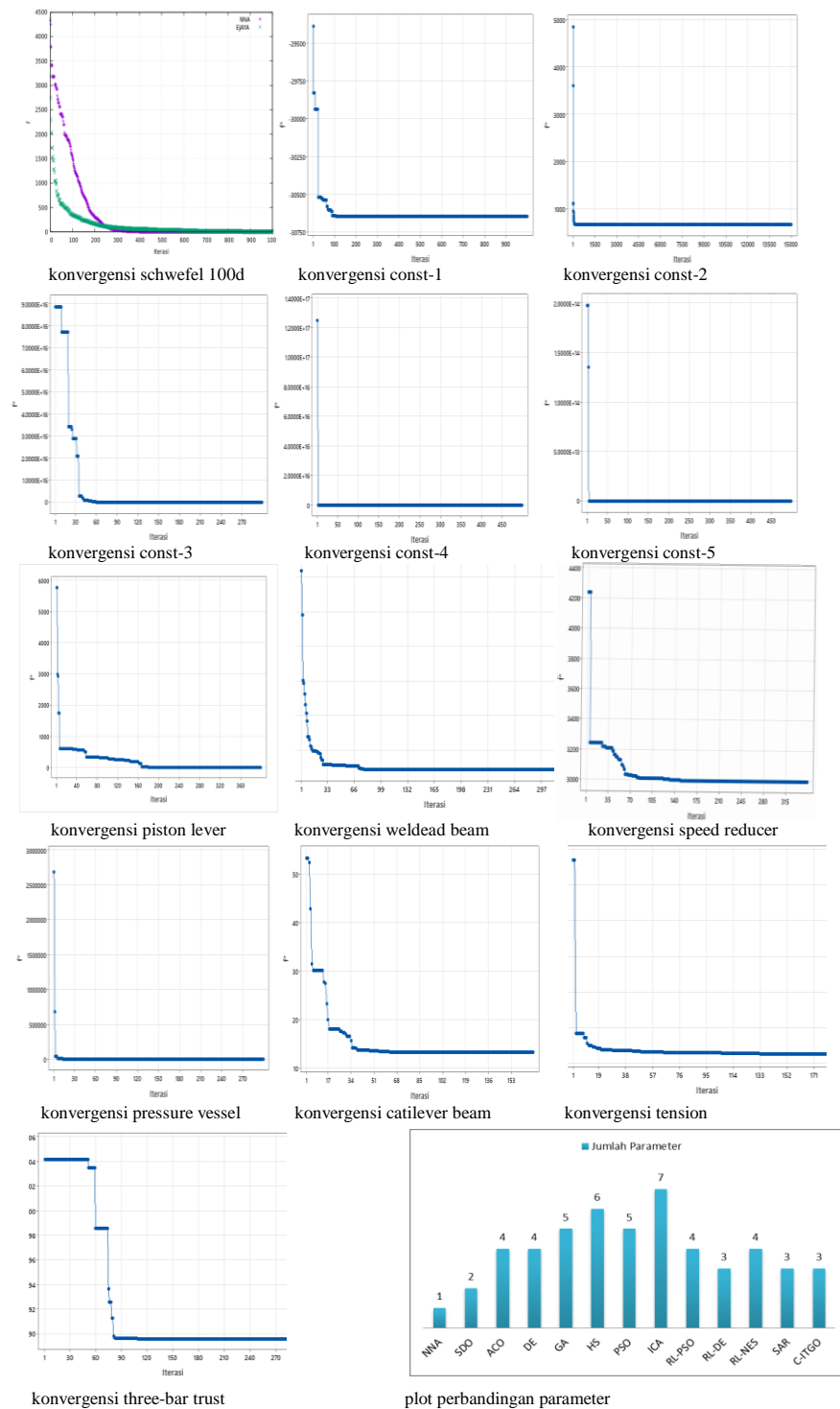
Tabel 1. Perbandingan hasil konvergensi metode NNA vs Spiral Dynamic Optimization (Tamura dkk. 2011)

Problem	Dim	Optimal Solution	Maximum Iteration MAX=100					Maximum Iteration MAX=1000				
			NNA	Spiral 1	Spiral 2	Spiral 3	Spiral 4	NNA	Spiral 1	Spiral 2	Spiral 3	Spiral 4
Griewank	3	Mean	0.022214844	0.06	0.06	0.12	0.13	0.03531911	0.1	0.1	0.03	0.03
		Best	0.000000000	0	0	0.03	0.04	0	0	0	0.2	0.2
		Worst	0.028676760	0.07	0.07	0.2	0.2	0.02291913	0.01	0.2	0.1	0.2
		Std. Dev	0.026666157	0.04	0.04	0.04	0.1	0.06569307	0.04	0.04	0.02	0.03
		Mean	0.127194805	1.6	1	3.3	2	0.065791956	1.5	1	0.8	0.2
	30	Best	0.00141573	1.1	0.7	2.2	1.8	0	1.1	0.3	0.1	0.01
		Worst	1.24733	3.2	1.1	4.7	2.1	0.0221561	3.2	1.1	0.5	0.9
		Std. Dev	0.28465622	0.4	0.1	0.5	0.07	0.007283508	0.4	0.2	0.4	0.2
		Mean	0.83330335	8.6	1.8	18	4.5	0.04441456	8.3	1.7	2.5	1.6
		Best	0.324137	4.6	1.5	15	4.1	2.46822E-09	4.3	1.4	1.3	1.4
100		Worst	4.33859	12	2	20	4.9	0.01971776	11	1.9	4.9	1.9
		Std. Dev	0.0395899	0.1	0	0.1	0.1	0.01971776	1.4	0.1	0.7	0.1
		Mean	0.000000000	0.01	0	0.23	0.16	1.15874E-14	1.3	0	0	0
		Best	8.5265E-14	0	0	0	0	0	0	0	0	0
		Worst	0.00850071	7.86	5.88	8.57	11	1.03029E-13	8	6	3	6
	3	Std. Dev	0.001920621	1.39	1.15	0.23	2.2	2.59262E-14	1.4	1.1	0.7	1.1
		Mean	112.65927	230	88	380	272	33.9604975	209	71	148	55
		Best	49.6442	152	49	304	238	4.97481	133	27	69	23
		Worst	181.811	391	189	456	317	169.204	369	161	260	124
		Std. Dev	35.33386095	40	26	35	15	38.99724651	39	26	39	21
100		Mean	623.88526	1174	628	1622	1032	193.28869	1049	550	777	445
		Best	343.21	193	116	123	100	68.1408	810	432	585	345
		Worst	1374	249	249	1403	1053	481.04	1252	459	1030	689
		Std. Dev	162.228074	70	48	50	27	99.99724697	72	48	70	52
		Mean	5.03806E-05	0	0	0.2	0.3	7.16562E-18	0	0	0	0
	3	Best	2.06335E-17	0	0	0	0.01	3.69364E-94	0	0	0	0
		Worst	0.001070761	0.03	0	0.66	2	5.74098E-17	0.03	0	0	0
		Std. Dev	0.000225308	0.01	0	0.13	0.2	1.45698E-17	0	0	0	0
		Mean	17.78159889	99	20	226	154	3.39594E-17	95	17	13	3
		Best	0.00759054	25	4.1	107	67	4.85108E-87	22	2.9	0.8	1
Schwefel	30	Worst	344.33	265	91	447	487	5.52295E-16	259	86	44	17
		Std. Dev	76.88456242	123	26	204	153	1.23635E-16	81	23	8	2
		Mean	155.065	153	285	204	153	2.70075E-08	1236	235	291	2
		Best	20.4862	618	88	1387	876	2.70075E-08	571	76	281	38
		Worst	15141.6	2777	984	7104	3203	0.00010812	2688	894	1821	243
	100	Std. Dev	3415.009199	394	147	972	466	2.43688E-05	376	137	297	35
		Mean	-234.8853	-229	-221	-224	-211	-234.98605	-229	-221	-234	-222
		Best	-234.997	-235	-235	-235	-234	-234.997	-235	-235	-235	-235
		Worst	-234.905	-174	-178	-198	-168	-234.994	-174	-178	-207	-178
		Std. Dev	0.020519824	14.3	15.3	7	16	0.00074516	14	15	5	15
2' minima	30	Mean	-1892.425	-1798	-1846	-1076	-1286	-2307.3215	-1615	-1888	-1885	-1989
		Best	-2177.11	-2037	-2038	-1575	-1360	-2449.97	-2028	-2057	-2184	-2288
		Worst	-1855	-1855	-1849	-1076	-1360	-2449.97	-2028	-2057	-2184	-2288
		Std. Dev	188.7743799	96	81	150	157	100.835531	98	79	162	159
		Mean	-5001.226	-4373	-4724	-3001	-3816	-7211.546	-4864	-4775	-5317	-5182
	100	Best	-5382.45	-5417	-5074	-2703	-4141	-7465.74	-5553	-5108	-6744	-5546
		Worst	-4086.04	-4055	-4336	-1079	-3483	-8955.42	-4167	-4367	-5942	-4805
		Std. Dev	543.4152607	267	159	311	109	121.9694262	265	160	186	154
		Mean	7.1403E-11	N/A	N/A	N/A	N/A	2.07222E-13	N/A	N/A	N/A	N/A
		Best	6.10769E-33	N/A	N/A	N/A	N/A	5.6091E-125	N/A	N/A	N/A	N/A
Rosenbrock	3	Worst	3.19217E-10	N/A	N/A	N/A	N/A	2.38389E-12	N/A	N/A	N/A	N/A
		Std. Dev	1.17248E-10	N/A	N/A	N/A	N/A	5.4033E-13	N/A	N/A	N/A	N/A
		Mean	0.000000000	N/A	N/A	N/A	N/A	3.19217E-10	N/A	N/A	N/A	N/A
		Best	3.26103E-06	N/A	N/A	N/A	N/A	1.19352E-14	N/A	N/A	N/A	N/A
		Worst	3.55696	N/A	N/A	N/A	N/A	4.37338E-10	N/A	N/A	N/A	N/A
	100	Std. Dev	0.893176067	N/A	N/A	N/A	N/A	1.4448E-10	N/A	N/A	N/A	N/A
		Mean	116.0749954	N/A	N/A	N/A	N/A	6.58085041	N/A	N/A	N/A	N/A
		Best	0.634456	N/A	N/A	N/A	N/A	3.47557E-09	N/A	N/A	N/A	N/A
		Worst	987.914	N/A	N/A	N/A	N/A	131.617	N/A	N/A	N/A	N/A
		Std. Dev	257.0612103	N/A	N/A	N/A	N/A	29.4304558	N/A	N/A	N/A	N/A





konvergensi schwefel-3 100d konvergensi 2nminima 100d konvergensi schwefel-2 100d



Tabel 2. Hasil Performa terbaik, terburuk, rata-rata dan standard deviasi dari berbagai metode untuk menyelesaikan permasalahan *Three Bars Truss*

Metode	Worst	Mean	Best	Std
NNA	263.904	263.896	263.896	0.002447545
CS (Gandomi dkk. 2013)	N.A	264.0669	263.971	0.00009
SNS (Bayzidi dkk. 2021)	263.8958561	263.8958462	263.8958434	3.31056E-06
PSO-DE (Liu dkk. 2010)	263.8958434	263.8958434	263.8958434	4.5E10
SaC (Ray dkk. 2003)	263.9697564	263.9033567	263.8958465	1.3E02
MBA (Saddollah dkk. 2013)	263.915983	263.897996	263.895852	3.93E-03
HEAA (Saddollah dkk. 2013)	263.896099	263.895865	263.895843	4.9E-05
DEDS (Saddollah dkk. 2013)	263.895849	263.895843	263.895843	9.7E-07
SC (Saddollah dkk. 2013)	263.969756	263.903356	263.895846	1.3E-02
MVDE (Melo dkk. 2013)	263.8958434	263.8958434	263.8958434	2.576062e7

Tabel 3. Hasil Performa terbaik, terburuk, rata-rata dan standard deviasi dari berbagai metode untuk menyelesaikan permasalahan *Pressure Vessel*

Metode	Worst	Mean	Best	Std
NNA	8249.95	6503.917619	5915.61	579.397517
C-ITGO (Ferreira dkk. 2018)	6059.7143	6059.7143	6059.7143	9.8E-13
MBA (Saddollah dkk. 2013)	6392.5062	6200.64765	5889.3216	160.34
WCA (Eskandar dkk. 2012)	6590.2129	6198.6172	5885.3327	213.049
CPSO (He dkk. 2006)	6363.8041	6147.1332	6061.0777	8.65E+01
IAPSO (Ferreira dkk. 2018)	6090.5314	6068.7539	6059.7143	1.40E+01
CPSO (He dkk. 2006)	6363.8041	6147.1332	6061.0777	8.65E+01
PSO -DE (Liu dkk. 2010)	N.A	6059.714	6059.714	N.A
MVDE (Melo dkk. 2013)	6090.53353	6059.99724	6059.7144	2.9103
HPPO (Hamzehkolaei dkk. 2021)	6288.677	6099.9323	6059.7143	86.20 0 0
UPSO (Ferreira dkk. 2018)	N.A	9032.55	6544.27	9.95E+02
CMA-ES (Ferreira dkk. 2018)	6410.08676	6170.25055	6059.7143	140.4843
LCA (Ferreira dkk. 2018)	6090.6114	6070.5884	6059.8553	11.37534
IAPSO (Guedria 2016)	6090.5314	6068.7539	6059.7143	14.0057
GA-1 (Guedria 2016)	6308.497	6293.8432	6288.7445	7.4133
NM-PSO (Zahara dkk. 2009)	5960.0557	5946.7901	5930.3137	9.161
DE (Melo dkk. 2012)	N.A	6059.770876	6059.714337	0.22613805

Tabel 4. Hasil Performa terbaik, terburuk, rata-rata dan standard deviasi dari berbagai metode untuk menyelesaikan permasalahan *Tension Compression Design*

Metode	Worst	Mean	Best	Std
NNA	0.0150836	0.013439038	0.0126947	0.000789732
CPSO (He dkk. 2006)	0.012924	0.01273	0.0126747	5.20E-05
GA (Coello dkk. 2002)	0.012973	0.012742	1.30E-02	5.90E-05
MSCA (Chen dkk. 2020)	N.A	N.A	0.012667	N.A
CGO (Talatari dkk. 2020)	0.012719055	0.012670085	0.012665246	1.09E-05
CS (Gandomi dkk. 2013)	0.001420272	0.01350052	0.01266522	3.93E-03
SaC (Ray dkk. 2003)	0.0167172	0.0129227	0.0126692	5.1985E-05

Tabel 5. Hasil Performa terbaik, terburuk, rata-rata dan standard deviasi dari berbagai metode untuk menyelesaikan permasalahan *Speed Reducer*

Metode	Worst	Mean	Best	Std
NNA	3097	3049.955714	2996.35	29.14999512
RL-PSO (Hadaideh dkk. 2021)	N.A	N.A	2996.85284	N.A
RL-DE (Hadaideh dkk. 2021)	N.A	N.A	2996.33574	N.A
RL-NES (Hadaideh dkk. 2021)	N.A	N.A	3000.15554	N.A
Artificial Bee Colony (Akay dkk. 2010)	N.A	2997.058412	2997.058412	0.00E+00
SAR (Shabani dkk. 2020)	2994.471066	2994.471066	2994.471066	0.00E+00
PSO-DE (Shabani dkk. 2020)	2996.348204	2996.348174	2996.348167	6.40E-06
MDE (Shabani dkk. 2020)	2996.390137	2996.36722	2996.356689	8.20E-03
EJAYA (Zhang dkk. 2021)	2994.471097	2994.47107	2994.471066	7.1926E-6
CS (Zhang dkk. 2021)	N.A	N.A	3000.981	N.A
SC (Zhang dkk. 2021)	N.A	N.A	2996.669016	N.A
WCA (Eskandar dkk. 2012)	2994.505578	2994.474392	2994.471066	7.4E03
C-ITGO (Ferreira dkk. 2018)	2996.348165	2996.348165	2996.348165	7.5E-13
IAPSO (Ferreira dkk. 2018)	2996.348165	2996.348165	2996.348165	6,88E-13

Tabel 6. Hasil Performa terbaik, terburuk, rata-rata dan standard deviasi dari berbagai metode untuk menyelesaikan permasalahan *Piston Lever*

Metode	Worst	Mean	Best	Std
NNA	295.493	160.8942595	8.41285	77.34948714
CS (Gandomi dkk. 2013)	168.592	40.2319	8.4271	5.91E+01
PSO (Gandomi dkk. 2013)	294	166	1.22E+02	5.17E+01
DE (Gandomi dkk. 2013)	199	187	159	14.2
MSMF (Hamzehkolaei dkk. 2021)	11.6085	9.50055	7.8978	0.32156
MCEO (Hamzehkolaei dkk. 2021)	15.6823	10.7613	8.4516	1.8564
HPSO (Hamzehkolaei dkk. 2021)	197	187	162	13.4
SNS (Bayzidi dkk. 2021)	167.4727747	24.3189743	8.412698349	47.71792646
HPSO (KIM dkk. 2009)	197	187	162	13.4
PSO (KIM dkk. 2009)	294	166	122	51.7

Tabel 7. Hasil Performa terbaik, terburuk, rata-rata dan standard deviasi dari berbagai metode untuk menyelesaikan permasalahan *Weldead Beam*

Metode	Worst	Mean	Best	Std
NNA	1.80506	1.73188619	1.72486	0.018128157
PSO -DE (Liu dkk. 2010)	1.7248811	1.7248579	1.7248531	4.1E06
SaC (Ray dkk. 2003)	6.3996785	3.0025883	2.39E+00	9.6E01
CDE(Huangdkk. 2007)	1.824105	1.768158	1.733461	0.022194
CPSO (He dkk. 2006)	1.782143	1.748831	1.728024	0.012926
GA (Coello 2000)	1.785835	1.771973	1.748309	0.01122
WCA (Eskandar dkk. 2012)	1.744697	1.726427	1.724856	4.29E03
WCA I (Eskandar dkk. 2012)	1.801127	1.73594	1.724857	1.89E02
RL-PSO (Hadaideh dkk. 2021)	N.A	N.A	1.75005	N.A

RL-DE (Hadaideh dkk. 2021)	N.A	N.A	1.72755	N.A
RL-NES (Hadaideh dkk. 2021)	N.A	N.A	1.7249	N.A
C-ITGO (Ferreira dkk. 2018)	1.7248523	1.7248523	1.7248523	3.65E-12
C-ITGO (Ferreira dkk. 2018)	1.7249215	1.748831	1.728024	7.88E-06
MBA (Saddollah dkk. 2013)	1.724853	1.724853	1.724853	6.94E-19

Tabel 8. Hasil optimum dari beberapa metode dalam menyelesaikan permasalahan *speed reducer*

Item	Metode yang diusulkan	RL-PSO	PSO	ABC	CS
x ₁	3.500000	3.500000	3.500000	3.500000	3.501500
x ₂	0.700000	0.700000	0.700000	0.700000	0.700000
x ₃	17.000000	17.000000	17.000000	17.000000	17.000000
x ₄	7.300270	7.300000	8.067020	7.300000	7.605000
x ₅	7.800030	7.800000	8.086810	7.800000	7.818100
x ₆	3.350220	3.352050	3.900000	3.350220	3.352000
x ₇	5.286680	5.286740	5.322400	5.287800	5.287500
g ₁	-0.073915	-0.07392	-0.07392	-0.07392	-0.074312
g ₂	-0.197999	-0.19800	-0.19800	-0.19800	-0.198342
g ₃	-0.499120	-0.50027	-0.63196	-0.49917	-0.434943
g ₄	-0.901470	-0.90148	-0.89312	-0.90156	-0.900845
g ₅	-0.000004	-0.00165	-0.36524	0.000000	-0.001077
g ₆	0.000002	-0.00003	-0.01994	-0.00063	-0.000460
g ₇	-0.702500	-0.70250	-0.70250	-0.70250	-0.702500
g ₈	-0.583333	0.000000	0.000000	0.000000	-0.583155
g ₉	0.000000	-0.58333	-0.58333	-0.58333	-0.000428
g ₁₀	-0.051360	-0.05095	-0.03930	-0.05133	-0.089020
g ₁₁	-0.010857	-0.01084	-0.04108	-0.01070	-0.013028
f(x)	2996.350000	2996.852840	3198.960720	2997.058410	3000.981000

Tabel 9. Hasil optimum dari beberapa metode dalam menyelesaikan permasalahan *Weldead Beam*

Item	NNA	RL-PSO	PSO	RL-DE	CPSO
x ₁	0.2057270	0.1995300	0.2751700	0.2045200	0.2023690
x ₂	3.4705400	3.6449100	4.3130400	3.4987300	3.5442140
x ₃	9.0366200	9.0233700	9.2512900	9.0405400	9.0482100
x ₄	0.2057300	0.2075400	0.3502500	0.2057400	0.2057230
g ₁	0.0218498	-90.07913	-5236.15894	-10.69933	-13.6555000
g ₂	-0.0265638	-174.30163	-13187.02019	-27.93511	-75.8141000
g ₃	-0.0000030	-0.00801	-0.07508	-0.00123	-0.0033540
g ₄	-3.4329800	-3.40609	-2.13725	-3.42973	-3.4245700
g ₅	-0.0807270	-0.07453	-0.15017	-0.07952	-0.0773690
g ₆	-0.2355400	-0.23560	-0.24208	-0.23556	-0.2355950
g ₇	-0.0298094	-153.88119	-24064.46402	-2.86378	-4.4728600
f(x)	1.7248600	1.7500500	3.2156100	1.7275500	1.7280240

6. Kesimpulan

Pada penelitian ini, kami mengusulkan metode NNA untuk menyelesaikan permasalahan optimasi berkendala, penyelesaian sejumlah permasalahan optimasi berkendala yang ditemukan dalam bidang engineering dan sejumlah fungsi benchmark berdimensi besar. Pada fungsi berkendala, diperlukan konstruksi ulang agar fungsi tersebut menjadi fungsi tidak berkendala. Kami telah mengkonfirmasi/ melakukan validasi terhadap metode yang kami usulkan dalam menyelesaikan sejumlah permasalahan yang ditemukan dari berbagai literatur.

Ada beberapa hal yang perlu dipertimbangkan dalam melakukan/ menyelesaikan setiap permasalahan yang diusulkan pada penelitian ini, pada kasus fungsi berdimensi besar dan multimodal sebaiknya nilai NPOP dibuat dalam rentang/ range (100-140) dan maksimum iterasi MAX = (1000-1500), sedangkan pada fungsi yang memiliki kendala, banyaknya kendala yang ada dan sifat nonlinear dari fungsi tersebut juga dapat dijadikan pertimbangan dalam memilih NPOP dan MAX.

2 Referensi

- [1] H. Chen, Y. Xu, M. Wang, dan X. Zhao, "A balanced whale optimization algorithm for constrained engineering design problems," *Applied Mathematical Modelling*, vol. 71, hlm. 45–59, Jul 2019, doi: 10.1016/j.apm.2019.02.004.
- [2] M. P. Ferreira, M. L. Rocha, A. J. Silva Neto, dan W. F. Sacco, "A constrained ITGO heuristic applied to engineering optimization," *Expert Systems with Applications*, vol. 110, hlm. 106–124, Nov 2018, doi: 10.1016/j.eswa.2018.05.027.
- [3] W. Zhu dan H. Fan, "A discrete dynamic convexized method for nonlinear integer programming," *Journal of Computational and Applied Mathematics*, vol. 223, no. 1, hlm. 356–373, Jan 2009, doi: 10.1016/j.cam.2008.01.023.
- [4] H. Chen, M. Wang, dan X. Zhao, "A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems," *Applied Mathematics and Computation*, vol. 369, hlm. 124872, Mar 2020, doi: 10.1016/j.amc.2019.124872.
- [5] P. Jangir dan N. Jangir, "A new Non-Dominated Sorting Grey Wolf Optimizer (NS-GWO) algorithm: Development and application to solve engineering designs and economic constrained emission dispatch problem with integration of wind power," *Engineering Applications of Artificial Intelligence*, vol. 72, hlm. 449–467, Jun 2018, doi: 10.1016/j.engappai.2018.04.018.
- [6] J. Cho, B. Craig, M. Hur, dan G. J. Lim, "A novel port call optimization framework: A case study of chemical tanker operations," *Applied Mathematical Modelling*, vol. 102, hlm. 101–114, Feb 2022, doi: 10.1016/j.apm.2021.09.037.
- [7] R. Tao, Z. Meng, dan H. Zhou, "A self-adaptive strategy based firefly algorithm for constrained engineering design problems," *Applied Soft Computing*, vol. 107, hlm. 107417, Agu 2021, doi: 10.1016/j.asoc.2021.107417.
- [8] Q. He dan L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, hlm. 89–99, Feb 2007, doi: 10.1016/j.engappai.2006.03.003.
- [9] W. Long, X. Liang, Y. Huang, dan Y. Chen, "An effective hybrid cuckoo search algorithm for constrained global optimization," *Neural Comput & Applic*, vol. 25, no. 3–4, hlm. 911–926, Sep 2014, doi: 10.1007/s00521-014-1577-1.
- [10] B. Ghasemishabankareh, X. Li, dan M. Ozlen, "Cooperative coevolutionary differential evolution with improved augmented Lagrangian to solve constrained

- optimisation problems,” *Information Sciences*, vol. 369, hlm. 441–456, Nov 2016, doi: 10.1016/j.ins.2016.06.047.
- [11] N. Kumar, S. K. Mahato, dan A. K. Bhunia, “Design of an efficient hybridized CS-PSO algorithm and its applications for solving constrained and bound constrained structural engineering design problems,” *Results in Control and Optimization*, vol. 5, hlm. 100064, Des 2021, doi: 10.1016/j.rico.2021.100064.
 - [12] Y. Zhang, A. Chi, dan S. Mirjalili, “Enhanced Jaya algorithm: A simple but efficient optimization method for constrained engineering design problems,” *Knowledge-Based Systems*, vol. 233, hlm. 107555, Des 2021, doi: 10.1016/j.knosys.2021.107555.
 - [13] V. V. de Melo dan G. L. C. Carosio, “Evaluating differential evolution with penalty function to solve constrained engineering problems,” *Expert Systems with Applications*, vol. 39, no. 9, hlm. 7860–7863, Jul 2012, doi: 10.1016/j.eswa.2012.01.123.
 - [14] Z. Cheng, H. Song, J. Wang, H. Zhang, T. Chang, dan M. Zhang, “Hybrid firefly algorithm with grouping attraction for constrained optimization problem,” *Knowledge-Based Systems*, vol. 220, hlm. 106937, Mei 2021, doi: 10.1016/j.knosys.2021.106937.
 - [15] Z. Cheng, H. Song, J. Wang, H. Zhang, T. Chang, dan M. Zhang, “Hybrid firefly algorithm with grouping attraction for constrained optimization problem,” *Knowledge-Based Systems*, vol. 220, hlm. 106937, Mei 2021, doi: 10.1016/j.knosys.2021.106937.
 - [16] A. Sadollah, A. Bahreininejad, H. Eskandar, dan M. Hamdi, “Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems,” *Applied Soft Computing*, vol. 13, no. 5, hlm. 2592–2612, Mei 2013, doi: 10.1016/j.asoc.2012.11.026.
 - [17] M. I. Radaideh dan K. Shirvan, “Rule-based reinforcement learning methodology to inform evolutionary algorithms for constrained optimization of engineering applications,” *Knowledge-Based Systems*, vol. 217, hlm. 106836, Apr 2021, doi: 10.1016/j.knosys.2021.106836.
 - [18] A. Shabani, B. Asgarian, M. Salido, dan S. Asil Gharebaghi, “Search and rescue optimization algorithm: A new optimization method for solving constrained engineering optimization problems,” *Expert Systems with Applications*, vol. 161, hlm. 113698, Des 2020, doi: 10.1016/j.eswa.2020.113698.
 - [19] Y. H. Liu, J. M. Chuang, dan M.-J. Hwang, “Solving a nonlinear integer program for allocating resources,” *Mathematical and Computer Modelling*, vol. 44, no. 3–4, hlm. 377–381, Agu 2006, doi: 10.1016/j.mcm.2006.01.023.
 - [20] J. Jiang, Y. Liu, dan Z. Zhao, “TriTSA: Triple Tree-Seed Algorithm for dimensional continuous optimization and constrained engineering problems,” *Engineering Applications of Artificial Intelligence*, vol. 104, hlm. 104303, Sep 2021, doi: 10.1016/j.engappai.2021.104303.
 - [21] Y. Belkourchia, L. Azrar, dan E.-S. M. Zeriab, “A Hybrid Optimization Algorithm for Solving Constrained Engineering Design Problems,” dalam 2019 5th International Conference on Optimization and Applications (ICOA), Kenitra, Morocco, Apr 2019, hlm. 1–7. doi: 10.1109/ICOA.2019.8727654.
 - [22] W. Li dan B. Wu, “A Modified Differential Evolution Algorithm for Constrained Optimization Problems,” dalam 2019 2nd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM), Shanghai, China, Nov 2019, hlm. 69–72. doi: 10.1109/WCMEIM48965.2019.00021.
 - [23] F. Hamza, D. Ferhat, H. Abderazek, dan M. Dahane, “A new efficient hybrid approach for reliability-based design optimization problems,” *Engineering with Computers*, Okt 2020, doi: 10.1007/s00366-020-01187-5.

- [24] Q. Li dan L.-Y. Yuan, "A novel filled function method for solving constrained optimization problems," dalam 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, Jun 2020, hlm. 2221–2225. doi: 10.1109/ITNEC48623.2020.9084669.
- [25] H. Wang, Z. Hu, Y. Sun, Q. Su, dan X. Xia, "A novel modified BSA inspired by species evolution rule and simulated annealing principle for constrained engineering optimization problems," *Neural Comput & Applic*, vol. 31, no. 8, hlm. 4157–4184, Agu 2019, doi: 10.1007/s00521-017-3329-5.
- [26] V. K. Pathak dan A. K. Srivastava, "A novel upgraded bat algorithm based on cuckoo search and Sugeno inertia weight for large scale and constrained engineering design optimization problems," *Engineering with Computers*, vol. 38, no. 2, hlm. 1731–1758, Apr 2022, doi: 10.1007/s00366-020-01127-3.
- [27] S. Khalilpourazari dan S. Khalilpourazary, "An efficient hybrid algorithm based on Water Cycle and Moth-Flame Optimization algorithms for solving numerical and constrained engineering optimization problems," *Soft Comput*, vol. 23, no. 5, hlm. 1699–1722, Mar 2019, doi: 10.1007/s00500-017-2894-y.
- [28] I. Brajević dan J. Ignjatović, "An upgraded firefly algorithm with feasibility-based rules for constrained engineering optimization problems," *J Intell Manuf*, vol. 30, no. 6, hlm. 2545–2574, Agu 2019, doi: 10.1007/s10845-018-1419-6.
- [29] A. H. Gandomi, X.-S. Yang, dan A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, hlm. 17–35, Jan 2013, doi: 10.1007/s00366-011-0241-y.
- [30] G. Steven, "Evolutionary algorithms for single and multicriteria design optimization. A. Osyczka. Springer Verlag, Berlin, 2002," hlm. 1.
- [31] N. Jangir, M. H. Pandya, I. N. Trivedi, R. H. Bhesdadiya, P. Jangir, dan A. Kumar, "Moth-Flame optimization Algorithm for solving real challenging constrained engineering optimization problems," dalam 2016 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, Mar 2016, hlm. 1–5. doi: 10.1109/SCEECS.2016.7509293.
- [32] K. H. Raj dan R. Setia, "Quantum seeded evolutionary computational technique for constrained optimization in engineering design and manufacturing," *Struct Multidisc Optim*, vol. 55, no. 3, hlm. 751–766, Mar 2017, doi: 10.1007/s00158-016-1529-8.
- [33] Q. Zhao dan C. Li, "Two-Stage Multi-Swarm Particle Swarm Optimizer for Unconstrained and Constrained Global Optimization," *IEEE Access*, vol. 8, hlm. 124905–124927, 2020, doi: 10.1109/ACCESS.2020.3007743.
- [34] X. L. Sun dan D. Li, "Value-Estimation Function Method for Constrained Global Optimization," *Journal of Optimization Theory and Applications*, vol. 102, no. 2, hlm. 385–409, Agu 1999, doi: 10.1023/A:1021736608968.
- [35] A. O. Topal dan O. Altun, "A novel meta-heuristic algorithm: Dynamic Virtual Bats Algorithm," *Information Sciences*, vol. 354, hlm. 222–235, Agu 2016, doi: 10.1016/j.ins.2016.03.025.
- [36] A. Kaveh dan A. Dardas, "A novel meta-heuristic optimization algorithm: Thermal exchange optimization," *Advances in Engineering Software*, vol. 110, hlm. 69–84, Agu 2017, doi: 10.1016/j.advengsoft.2017.03.014.
- [37] J. de J. Rubio, M. A. Islas, G. Ochoa, D. R. Cruz, E. Garcia, dan J. Pacheco, "Convergent newton method and neural network for the electric energy usage prediction," *Information Sciences*, vol. 585, hlm. 89–112, Mar 2022, doi: 10.1016/j.ins.2021.11.038.

- [38] A. Kaveh dan A. Zolghadr, "Cyclical Parthenogenesis Algorithm for guided modal strain energy based structural damage detection," *Applied Soft Computing*, vol. 57, hlm. 250–264, Agu 2017, doi: 10.1016/j.asoc.2017.04.010.
- [39] V. K. Patel dan V. J. Savsani, "Heat transfer search (HTS): a novel optimization algorithm," *Information Sciences*, vol. 324, hlm. 217–246, Des 2015, doi: 10.1016/j.ins.2015.06.044.
- [40] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, dan S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Future Generation Computer Systems*, vol. 101, hlm. 646–667, Des 2019, doi: 10.1016/j.future.2019.07.015.
- [41] J. Deepho, A. B. Abubakar, M. Malik, dan I. K. Argyros, "Solving unconstrained optimization problems via hybrid CD-DY conjugate gradient methods with applications," *Journal of Computational and Applied Mathematics*, vol. 405, hlm. 113823, Mei 2022, doi: 10.1016/j.cam.2021.113823.
- [42] S. Mirjalili dan A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, hlm. 51–67, Mei 2016, doi: 10.1016/j.advengsoft.2016.01.008.
- [43] A. Sadollah, H. Sayyaadi, dan A. Yadav, "A dynamic metaheuristic optimization model inspired by biological nervous systems: Neural network algorithm," *Applied Soft Computing*, vol. 71, hlm. 747–782, Okt 2018, doi: 10.1016/j.asoc.2018.07.039.
- [44] A. Kania dan K. A. Sidarto, "Solving mixed integer nonlinear programming problems using spiral dynamics optimization algorithm," Bandung, Indonesia, 2016, hlm. 020004. doi: 10.1063/1.4942987.
- [45] K. Tamura, K. Yasuda, dan Tokyo Metropolitan University, 1-1 Minamiosawa, Hachioji, Tokyo 192-0397, Japan, "Spiral Dynamics Inspired Optimization," *J. Adv. Comput. Intell. Intell. Inform.*, vol. 15, no. 8, hlm. 1116–1122, Okt 2011, doi: 10.20965/jaciii.2011.p1116.