OKU Fen Bilimleri Enstitüsü Dergisi
9(1): 254-267, 2026

OKU Journal of The Institute of Science and Technology, 9(1): 254-267, 2026

Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi

Osmaniye Korkut Ata University Journal of The Institute of Science and Technology

Enhancement Political Optimizer Algorithm Based on the Leader Redistribution Mechanism

Manaf Adnan Farhood Al-JARAH[1*], Fatma Özge ÖZKÖK[2*]

[1,2]Erciyes University, Engineering Department, Computer Engineering,38280, Kayseri

[1]https://orcid.org/0000-0003-3580-6269
[2]https://orcid.org/0000-0002-1421-4670
*Corresponding author: manaf.farhood@gmail.com

Research Article

Article History:
Received: 30.09.2024
Accepted: 08.08.2025
Published online:13.01.2026

Keywords:
Optimization algorithm
Political optimizer
Benchmark problems
Uni-model functions
Multi-model functions

ABSTRACT

Optimization algorithms are mathematical steps used to find an optimal solution from a set of possible solutions. These solutions usually require a long time, while optimization algorithms can find solutions quickly. Algorithms are typically combined to overcome algorithm weaknesses or improve performance. In this article, we have enhanced the Political Optimizer (PO) using a mechanism that randomly redistributes the leader's positions to increase the searching efficiency and overcome the problem of stagnation in the local solution. We aimed to redistribute the leaders' locations inside the election campaign process to open the road for the leaders to move to new areas. This idea was applied in critical conditions to have a little effect on the algorithm. The enhancement was promising, and the results improved well for the exploration and exploitation phases. We compared the performance of the enhancement algorithm with the classical political optimizer algorithm beside 13 different state-of-the-art algorithms. The evaluation occurred on 50 benchmark problems, 25 were unimodal and 25 were multi-modal objective functions, while the mean result was completed to achieve the stability of the algorithms. The Enhanced Political Optimizer (EPO) achieved 23 optimal solutions and 13 close to the optimal solutions, while the results of the PO were 17 optimal and ten solutions close to the optimal. In terms of execution time, the algorithm was very promising, where the execution times in one run are (0.79336 ~ 0.16673) for the unimodal functions, while multi-model functions record (1.0874 ~ 0.16953) of the highest and lowest values, respectively.

Lider Yeniden Dağıtım Mekanizmasına Dayalı Geliştirme Politik Optimizasyon Algoritması

Araştırma Makalesi

Makale Tarihçesi:
Geliş tarihi: 30.09.2024
Kabul tarihi:08.08.2025
Online Yayınlanma:13.01.2026

Anahtar Kelimeler:
Optimizasyon algoritması
Politik optimizasyon
Kıyaslama problemleri
Tek modelli fonksiyonlar
Çok modelli fonksiyonlar

ÖZ

Optimizasyon algoritmaları, olası çözümler kümesinden optimum bir çözüm bulmak için kullanılan matematiksel adımlardır. Bu çözümlerin bulunması genellikle çok uzun zaman alırken, optimizasyon algoritmaları çözümleri kısa sürede bulabilir. Algoritmalar genellikle algoritmadaki zayıflıkları gidermek veya performansı iyileştirmek için birleştirilir. Bu makalede, arama verimliliğini artırmak ve yerel çözümdeki durgunluk sorununu aşmak için liderlerin konumlarını rastgele yeniden dağıtan bir mekanizma kullanarak Politik Optimizasyoncuyu (PO) geliştirdik. Liderlerin seçim kampanya süreci içindeki konumlarını yeniden dağıtarak liderlerin yeni alanlara taşınmasına yol açmayı amaçladık. Bu fikir, algoritmaya küçük bir etki yaratmak için kritik koşullarda uygulandı, geliştirme çok umut vericiydi ve sonuçlar hem keşif hem de istismar aşamaları için çok iyi iyileşti. Geliştirme algoritmasının performansını, 13 farklı son teknoloji algoritmanın yanı sıra klasik politik optimizasyon algoritmasıyla karşılaştırdık. Değerlendirme 50 kıyaslama problemi üzerinde yapıldı, 25'i tek modlu ve 25'i çok modelli amaç

fonksiyonlarıydı ve ortalama sonuç algoritmaların kararlılığını elde etmek için hesaplıydı. Gelişmiş Politik Optimizasyon (EPO) 23 en iyi çözüme ve en iyiye yakın 13 çözüme ulaşırken, PO'nun sonuçları 17 en iyi ve en iyiye yakın 10 çözümdü. Yürütme süresi açısından algoritma çok ümit vericiydi, tek modlu fonksiyonlar için bir çalıştırmada yürütme süreleri (0.79336 ~ 0.16673) iken, çok modelli fonksiyonlar sırasıyla en yüksek ve en düşük değerleri (1.0874 ~ 0.16953) kaydetti.

To Cite: Al-Jarah MAF., Özkök FÖ. Enhancement Political Optimizer Algorithm Based on the Leader Redistribution Mechanism. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2026; 9(1): 254-267.

1. Introduction

Optimization algorithms are mathematical methods or procedures used to find the best possible solution to a problem, often under constraints (Tao et al., 2024). These algorithms aim to maximize or minimize an objective function, representing the optimization goal (Zhou et al., 2023). Optimization is critical in various fields, including engineering, economics, machine learning, and operations research (Cheraghi and Hossein Jahangir, 2023). The aim behind inventing optimization algorithms is to solve NP-hard problems that take several years to solve. Optimization algorithms can solve these problems in several steps thanks to their simple mechanics and their avoidance of useless solutions (Chatterjee et al., 2024). Optimization algorithms are divided into two categories: heuristic and meta-heuristic algorithms. Each of these classes is designed to achieve optimal solutions using simple techniques; these techniques usually mimic creatures' behaviors in hunting or any other organized steps to achieve something. Suppose the meta-heuristic algorithm is based on natural behavior. In that case, it is called a Nature-inspired meta-heuristic algorithm, which has been classified into four major groups in the literature: Evolution-based, swarm-based, physics-based, and human behavior-based algorithms (Khan et al., 2024; Ramnath et al., 2024). This study on human behavior-based algorithms called the political optimizer (PO) has improved this algorithm using an adaptation technique to its mechanism. The primary political optimizer simulates the government formation and parliamentary gaining strategies (Askari et al., 2020). The algorithm is designed in four aspects: two involve the exploration phase, the election campaign, and the party switching, and two involve the exploitation phase, which is the election phase and parliamentary affairs. Several improved occurred in the PO in the literature, such as multiple filter political optimizer (Sahu et al., 2022a), political optimizer with interpolation strategy (Zhu et al., 2021), Quasi-oppositional-based political optimizer (Basetti et al., 2021), modeling of chaotic political optimizer (Sunitha et al., 2022) and Dragonfly political optimizer (Muppidi and Kishore, 2023). In this study, we added a heuristic redistributed for the leader's position in the election campaign phase to improve the balance between the exploration and the exploitation phases.

Despite its innovative design, the Political Optimizer (PO) exhibits several mathematical limitations. Its dual-role mechanism can cause interdependent updates that risk premature convergence in complex landscapes. The Recent Past-Based Position Updating Strategy (RPPUS), though effective, introduces search path dependency and may misguide the search in dynamic environments. Additionally, relying on only two reference solutions (party leader and constituency winner) limits directional diversity. The

performance is also sensitive to the party switching rate (λ), which requires careful tuning to balance exploration and exploitation. Lastly, the fixed structure of equal party and constituency sizes reduces adaptability in heterogeneous problem domains.

To handle these limitations, this article proposed periodic random reselection of leaders when no fitness improvement occurs, promoting exploration and preventing stagnation. The structured position-updating logic adapts based on fitness trends, dynamically adjusting search radii to balance exploitation and exploration. This ensures more diverse and adaptive search behavior while preserving the core socio-political structure of the original algorithm. The highlighted points that pushed us to create the article on the enhanced Political Optimizer algorithm include some problem statements, motivation, and the contributions can be summarized in some points:-

- Finding the optimal solution for multi-objective optimization problems by PO is challenging, so the exploration phase should be improved. At the same time, several unimodal problems should be dropped in the local solution.
- Although the PO algorithm represents a complex algorithm, it has a fast running time and short time complexity.
- Our contribution is improving the exploration phase by redistributing the leaders' positions. This idea creates a new search space where the leaders replace the positions with standard agents. This mechanism provides a new search space for the leaders and prevents the algorithm from getting stuck in a local minimum.

After the introduction, this paper begins with a literature review section containing the newer development research of political optimizer algorithms from the highest impact international publishing journals (Elsevier and Springer). Section 3 illustrates the method of the enhancement process, while the fourth section displays the framework's results and implementation steps. This research ends with a discussion and future work, besides highlighting some limitations and gaps State-of-the-art.

2. Literature Review

Dong et al. (Dong et al., 2022) proposed GPOFWA, a hybrid of the Political Optimizer and Fireworks Algorithm, improving subgroup limitations via spark mechanisms. It was validated on 30 benchmark problems, outperforming state-of-the-art algorithms.

Chander et al. (Chander et al., 2023) developed a hybrid Dolphin Echolocation–Political Optimizer to optimize the Tversky index for feature selection. Achieved 0.937 accuracy in classifying big data using clustering techniques.

Sahu et al. (Sahu et al., 2022b) combined multiple filter-based ranking with Political Optimizer to handle high-dimensional, imbalanced DNA microarray data. The hybrid showed strong performance across various machine learning models.

Bashkandi et al. (Bashkandi et al. 2023) introduced a hybrid method using PO, PSO, and CNN for automated brain tumor detection. The PO optimized CNN structure, achieving superior accuracy on preprocessed brain MRI datasets.

Fathy and Rezk (Fathy and Rezk 2022) used the Political Optimizer to estimate SOFC model parameters under steady and dynamic conditions. It achieved high accuracy and robustness, outperforming several modern optimizers.

Kiani et al. (Kiani, Anka, and Erenel 2023) proposed PSCSO, enhancing Sand Cat Swarm Optimization using parliamentary principles to improve exploration and exploitation. Validated on CEC benchmarks and engineering problems, it showed superior performance.

3. Background

3.1. Political optimizer

Politics can be said to be an improvement process from two angles. The first angle is where individuals work to maximize public gain to win elections. In contrast, the other angle pushes parties to secure the most significant number of seats in Parliament to win the formation of a government. The sequence of political improvement includes four stages, each two stages going in one direction: the first direction includes the election campaign and party switching. In contrast, the second direction includes elections between parties and parliamentary affairs. The fifth stage is done only once and includes the formation of the party and the allocation of electoral districts, while the remaining four stages are repeated periodically (Askari et al., 2020).

3.2. Exploration phase

Both the election campaign and the party-switching processes represent the exploration phase in the PO algorithm. In this phase, each agent in the population improves its position concerning the parity leaders or the area winners. The new agent position will update toward the leader or the winner positions locally called (center) according to two conditions and three equations in each condition. See Equations 1 and 2.

$$p^j(t+1) = \begin{cases} m^* + r(m^* - p^j(t)), & \text{if } p^j(t-1) \le p^j(t) \le m^* \text{ or } p^j(t-1) \ge p^j(t) \ge m^* \\ m^* + (2r-1)\lvert m^* - p^j(t)\rvert, & \text{if } p^j(t-1) \le m^* \le p^j(t) \text{ or } p^j(t-1) \ge m^* \ge p^j(t) \\ m^* + (2r-1)\lvert m^* - p^j(t-1)\rvert, & \text{if } m^* \le p^j(t-1) \le p^j(t) \text{ or } m^* \ge p^j(t-1) \ge p^j(t) \end{cases} \quad (1)$$

equation (1) will be active if the solution is improved (i.e., if the fitness of the previous solution is less than the fitness of the current solution), while equation (2) will be active if the solution is collapsed (i.e., the fitness of previous solution is higher than the fitness of current solution) (Askari et al., 2020).

$$p^j(t+1) = \begin{cases} m^* + (2r-1)|m^* - p^j(t)|, & \text{if } p^j(t-1) \le p^j(t) \le m^* \text{ or } p^j(t-1) \ge p^j(t) \ge m^* \\ p^j(t-1) + r(p^j(t) - p^j(t-1)), & \text{if } p^j(t-1) \le m^* \le p^j(t) \text{ or } p^j(t-1) \ge m^* \ge p \\ m^* + (2r-1)|m^* - p^j(t-1)|, & \text{if } m^* \le p^j(t-1) \le p^j(t) \text{ or } m^* \ge p^j(t-1) \ge p^j(t) \end{cases} \quad (2)$$

where

pj (t): Fitness value of the current position of candidate i in party j at iteration t.

Pj (t + 1): Fitness value of the previous position of candidate i in party j at iteration $t-1$.

pj: Position vector of candidate i in party j

r: A random number in the range [0,1], used for stochastic updating of positions.

m∗: A value used in the position update; initially holds the kth dimension of the party leader p∗ and then of the constituency winner c∗

3.3. Exploitation phase

This phase involves the election process and Parliamentary affairs. The election process represents the first exploitation process with mathematical representation by evaluating the fitness of each solution and selecting the fitness one. On the other hand, Parliamentary affairs simulate the exploitation and convergence processes.

4. Materials and Methods

4.1. Enhancement idea

By reviewing the multi-objective problem results, it can be predicted that the weakness of the algorithm in achieving the optimal solution means the algorithm's exploration phase needs a mechanism to improve it. It is clear that the algorithm updates the locations of the agents to the leaders or the winners, but this creates monotonous solutions and inhibits the algorithm's ability to find unconventional solutions, rather than switching positions with the most diminutive leader, fitness. Randomly redistributing the leaders' positions creates a high probability of searching for new solutions. It may allow a vanished leader to regain popularity, see Figure 1.

The exploration phase is divided into two steps depending on the previous solution. If the current solution is greater than or equal to the previous solution, Equation 1 will activate; otherwise, Equation 2 will activate. We exploit the first condition (greater than or equal) to divide it into two conditions. The first is (greater than), which activates Equation 1, and the second is (equal), which activates Equation 3.

Algorithm 1 Election campaign part 3 leaders' redistribution *(P, fitness)*

```
for each, I belong to (1, search agent) do.
    if Prefitness = = fitness(i) then each six iterations
        [~, sorted_indices] = sort-fitness
        diversity_threshold = 0.2
        for each i belongs to (1, searchAgent) do.
            Sp = [0, 1]
            if sp < diversity_threshold then
                r = [1, n]
                position (i, :) = position(sorted_indices(r), :)
            end
        end
    end
end
```

$$P(i + 1) = P(sorted_indices(rand(1,N))) \quad (3)$$

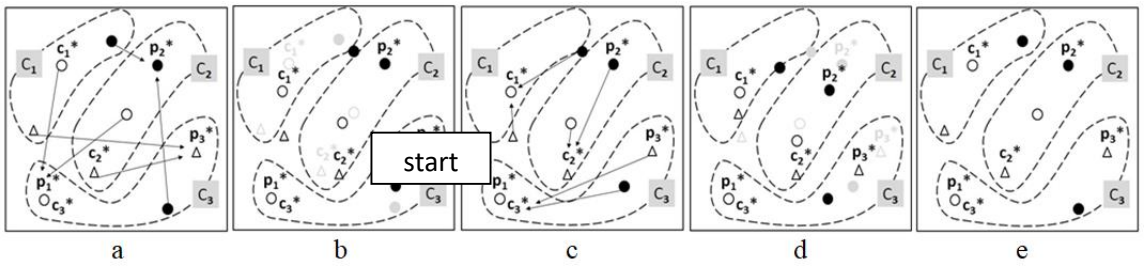


Figure 1. (a, b) shows the position improvement according to the leaders (Equations 1, 2). (c, d) Shows the position improvement according to the constituency winners (Equations 1, 2). (e) Illustrate the new mechanism of the leader's reselection process (Equation 3).

4.2. Total framework

This section explains the total algorithm block diagram, where the algorithm begins by initializing the parity members by creating the population agents. After each individual has been evaluated using the objective function to calculate their fitness, their fitness enables us to select both leaders and winners. After that, the main loop will run until the criteria condition is eliminated (see Figure 2). In the election campaign section, the enhancement statement has been added under this condition (If the previous solution = current solution), then run Equation 3.

4.3. Benchmark functions

To evaluate the performance of the enhancement algorithm, we test it on two sets of benchmark problems from the state-of-the-art; the first set is unimodal-objective functions to test the exploitation capabilities, and the second set is a multimodal-objective function to test the exploration ability. The two sets are explained in two sections below;

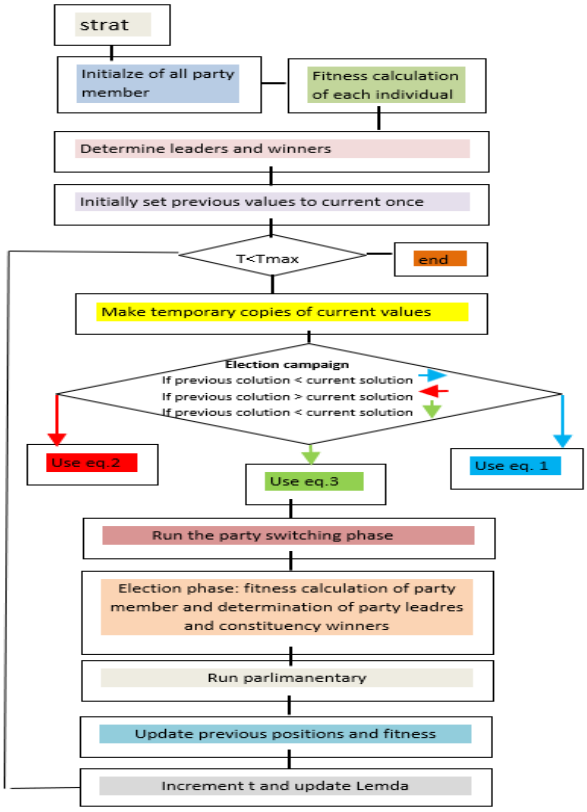


Figure 2. The total framework of the enhancement political optimizer

The unimodel objective functions are (F1 – Sphere, F2 - Quartic Noise, F3 - Powell Sum, F4 - Schwefel's 2.20, F5 - Schwefel's 2.21, F6 – Step, F7 – Stepint, F8 - Schwefel's 1.20, F9 - Schwefel's 2.22, F10 - Schwefel's 2.23, F11 – Rosenbrock, F12 – Brown, F13 - Dixon & Price, F14 - Powell Singular, F15 – Zakharov, F16 - Xin-She Yang, F17 - Perm 0,D,Beta, F18 - Three-hump Camel, F19 – Beale, F20 – Booth, F21 – Brent, F22 – Matyas, F23 - Schaffer N. 4, F24 - Wayburn Seader 3 and F25 – Leon). While the multi-model objective functions are (F1 - Schwefel's 2.26, F2 – Rastrigin, F3 - Periodic, F4 - Qing, F5 - Alpine N. 1, F6 - Xin-She Yang, F7 – Ackley, F8 - Trignometric 2, F9 – Salomon, F10 - Styblinski-Tang, F11 - Griewank, F12 - Xin-She Yang N. 4, F13 - Xin-She Yang N. 2, F14 - Gen. Pendlized, F15 - Pendlized, F16 - Egg Crate, F17 - Ackley N.3, F18 - Adjiman, F19 - Bird, F20 - Camel 6 Hump, F21 - Branin RCOS, F22 - Goldstein Price, F23 - Hartman 3, F24 - Hartman 6, F25 - Cross-in-tray, F26 - Bartels Conn) the optimal values of this functions are mentioned in the results section.

5. Results and Discussion

Acer Nitro PC has implemented all experiments with an Intel CPU 2700G and 6GB GPU. In this section, we arrange the enhancement algorithm to test 50 benchmark problems, with 25 used for the exploration phase and 25 for the exploitation phase. The experiments were run multiple times to determine the optimal parameter values. The experiments were careful not to tamper with the values of the original variables of the basic PO algorithm, such as the population number, the number, and the value of Lambda. However, simultaneously, we implement the experiments several times to obtain the best

values for the new variables, such as the diversity threshold and the iteration period. The experiments found that the best results can be achieved by making the diversity threshold 0.2, and the change occurs every six iterations. See Table 1, which tests more than one case, and select the best values of parameters on the Rosenbrock function.

Table 1. Test several cases to determine the best values of the diversity threshold iteration change

Case number	The diversity threshold	iteration change	Reuslt		
			Best	Worst	main
Case 1	0.1	3	0	0	0
Case 1	0.2	6	0	0	0
Case 3	1	10	0	26.9348	2.6935
Case 4	5	20	0	27.3911	2.7391

Each problem was run ten times, and the final average was calculated to obtain a result with the highest possible stability. At the same time, the results were compared with the original PO algorithm in addition to 13 other algorithms from the literature. Whale Optimization Algorithm (WOA) (Mirjalili and Lewis, 2016), Differential Evolution (DE) (Kukkonen and Lampinen, 2004), Biogeography Based Optimizer (BBO) (Simon, 2008), Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014), Krill Herd (KH) (Gandomi and Alavi, 2012), Salp Swarm Optimization (SSA1) (Mirjalili et al., 2017), Social Spider Algorithm (SSA2) (Yu and Li, 2015), Butterfly Optimization Algorithm (BOA) (Arora and Singh, 2019), Cuckoo Search (CS) (Yang and Suash Deb, 2009), Gravitational Search Algorithm (GSA) (Rashedi, Nezamabadi-pour, and Saryazdi 2009), Sine Cosine Algorithm (SCA) (Mirjalili, 2016), Teaching-Learning Based Optimization (TLBO) (Rao et al., 2011), Soccer League Competition (SLC) (Moosavian and Kasaee Roodsari, 2014) algorithm. In table 2 the results show the ability of EPO to achieve the nearest to the optimal solution was not achieved in the old version where (F2, F4, F5, F8, F9, F13, F15, F16, F21 and F24) This proves the evolution of the algorithm in the exploitation scenarios where ten new nearest to optimal solution achieved. In contrast, the primary PO achieved only two solutions (F7, F9). The EPO algorithm also obtained an optimal solution for (F19 and F23), thus outperforming the basic PO algorithm. In the exploratory phase, the Enhancement algorithm has obtained 13 optimal solutions (F2, F8, F9, F11, F16 – F22, and F24, F25) in addition to what it shares with the old algorithm of solutions in (F2, F8, F9, F11, F16, and F25) see Table 3. By comparing the nearest to the optimal solution, it was also able to find five new nearest optimal solutions (F1, F5–F7, and F12). In terms of execution time, the algorithm was very promising, where the execution times in one run are

(0.79336 ~ 0.16673) for the unimodal functions, while multi-model functions record (1.0874 ~ 0.16953) of the highest and lowest values, respectively. In the summary of the results section, from 50 benchmark problems, the EPO achieved (26, 18) while the PO achieved (17, 6) optimal and nearest to the optimal solutions, respectively.

Table 2. Comparison among the PO and EPO, besides the other 13 optimization algorithms, to evaluate the performance around the uni-model functions

Fn	Stats	Opt.	BOA	SSA1	WOA	SCA	SSA2	GWO	SLC	KH	TLBO	GSA	CS	BBO	DE	PO	EPO
F-1	Mean	0	1.9E-14	9.4E-08	7.4E-149	1.6E+02	9.3E-05	1.9E-29	1.4E-75	1.3E+00	1.6E-92	6.6E-02	1.0E+02	1.1E+023	.5E-10	0.0	0
F-2	Mean	0	9.5E-04	3.0E-01	1.8E-03	2.0E+00	1.2E-01	1.5E-03	5.8E-04	1.3E-01	9.2E-04	8.1E-02	1.6E-01	3.9E-03	4.6E-02	3.7E-04	0.000
F-3	Mean	0	2.0E-16	7.9E-07	5.2E-218	2.6E-03	1.4E-26	1.9E-128	2.4E-112	2.1E-07	6.8E-221	1.7E-13	8.4E-12	0.0	4.9E-13	0.0	0
F-4	Mean	0	1.3E-11	3.8E+01	3.9E-102	20E+00	4.5E-03	7.1E-17	4.3E-38	5.3E+00	1.1E-45	1.5E+01	6.4E+01	4.1E+01	8.6E-06	7.7E-179	3.46e-180
F-5	Mean	0	1.2E-11	1.7E+01	7.5E+01	6.2E+01	3.7E+01	1.6E-06	8.6E-36	5.1E+00	3.1E-37	7.4E+00	1.6E+01	5.3E+01	1.8E+01	2.7E-158	1.42e-158
F-6	Mean	0	1.1E+01	8.9E-08	4.2E-01	3.6E+02	1.3E-04	1.7E+00	7.0E-02	1.4E+00	1.8E-05	8.3E-01	9.7E+01	1.0E+02	2.0E-10	0.0	0
F-7	Mean	25 – 6n	-58.60	-212.88	-275.00	-149.80	-51.08	-219.12	-265.72	-79.52	-271.64	-180.40	-233.04	-257.64	-274.96	-275.00	-275
F-8	Mean	0	1.8E-14	5.8E+03	1.3E+05	3.9E+04	2.4E+04	3.1E-04	5.6E-72	3.7E+03	6.5E-13	1.5E+03	1.3E+04	3.6E+04	8.6E+04	1.8E-274	1.80e-281
F-9	Mean	0	3.0E+73	5.0E+36	1.5E-101	1.3E+00	4.1E+68	1.1E-16	1.2E-38	3.5E+59	2.7E-45	2.7E+02	1.0E+10	3.6E+01	6.7E-05	1.5E-179	1.93e-179
F-10	Mean	0	1.9E-21	2.7E-07	0.0E+00	1.7E+08	5.8E-12	3.9E-93	0.0E+00	5.8E-08	0.0E+00	3.5E-03	9.3E+02	1.2E-01	7.6E+03	0.0	0
F-11	Mean	0	4.9E+01	1.7E+02	4.8E+01	2.5E+06	1.7E+02	4.7E+01	4.6E+01	2.5E+02	4.3E+01	1.3E+02	4.5E+03	4.3E+03	7.7E+01	0.0	0
F-12	Mean	0	1.5E-14	2.4E-10	9.3E-147	9.7E-02	7.8E-06	4.5E-32	1.3E-78	2.0E+02	3.1E-95	5.5E+00	1.6E+00	0.0	5.9E-13	0.0	0
F-13	Mean	0	0.99	8.04	0.67	16298.5	9.44	0.67	0.67	9.92	0.67	4.21	67.82	237.28	2.34	0.05	0.03
F-14	Mean	0	1.7E-14	6.4E+00	7.1E-14	2.3E+02	1.2E+00	1.7E-05	2.2E-77	2.8E+00	7.6E-08	3.1E+00	8.6E+00	4.5E+01	5.8E-01	0.0	0
F-15	Mean	0	1.7E-14	1.4E+02	9.3E+02	9.3E+01	6.3E+02	2.2E-06	2.5E-54	4.0E+02	5.4E-03	1.2E+02	6.0E+02	2.9E+02	4.9E+02	9.9E-188	1.008e-189
F-16	Mean	-1	9.1E-48	6.9E-276	-3.2E-01	6.8E-290	0.0E+00	1.2E-165	0.0E+00	5.8E-61	0.0E+00	6.7E-54	0.0E+00	0.0	0.0	-4.8E-01	-0.2
F-17	Mean	0	939.27	44.34	514.73	44.40	131.14	571.13	19.78	97.61	0.72	578.76	2.40	135.89	1.70	16.76	0.9
F-18	Mean	0	1.1E-17	1.5E-15	1.6E-144	2.3E-91	1.2E-02	1.8E-290	2.9E-100	1.7E-11	5.9E-214	1.0E-20	9.3E-26	0.0	2.7E-197	0.0	0
F-19	Mean	0	2.1E-01	2.3E-15	4.4E-11	1.2E-04	5.3E-02	3.0E-02	0.0E+00	1.1E-10	0.0E+00	1.9E-20	1.4E-20	1.8E-01	0.0	1.7E-23	0
F-20	Mean	0	8.5E-02	1.7E-14	1.7E-04	6.0E-04	1.4E-17	1.8E-07	0.0E+00	9.9E-11	0.0E+00	3.1E-20	5.0E-26	0.0	0.0	0.0	0
F-21	Mean	0	1.6E+01	1.4E-87	1.4E-87	1.4E-87	1.4E-87	1.4E-87	1.4E-87	1.4E-87	1.4E-87	1.3E-04	1.4E-87	1.4E-87	1.4E-87	1.4E-87	1.383e-87
F-22	Mean	0	1.2E-15	5.8E-16	0.0E+00	3.3E-78	2.8E-08	6.7E-162	2.4E-98	7.3E-12	7.2E-155	1.2E-21	2.8E-28	0.0	2.8E-117	0.0	0
F-23	Mean	0.29257	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.31	0.29	0.30	0.29	0.29	0.29258
F-24	Mean	.21.35	19.48	19.11	19.11	19.13	19.11	19.11	19.11	19.11	19.11	19.11	19.11	20.44	19.11	19.11	19.1059
F-25	Mean	0	1.5E-02	1.7E-15	1.3E-05	2.1E-04	5.4E-02	4.8E-07	1.4E-32	8.5E-03	4.4E-22	5.1E-03	6.1E-23	0.0	0.0	0.0	0

Optimal solution Nearest to optimal

Table 3. Comparison among the PO and EPO, besides the other 13 optimization algorithms, to evaluate the performance around the multi-model functions.

Fn	Stats	Opt.	BOA	SSA1	WOA	SCA	SSA2	GWO	SLC	KH	TLBO	GSA	CS	BBO	DE	PO	EPO
F-1	Mean	0	3.1E+02	1.8E+02	9.4E+01	3.2E+02	3.5E+02	2.2E+02	2.0E+02	2.2E+02	1.9E+02	3.5E+02	1.7E+02	4.2	2.0E+02	1.3E-05	1.27E-05
F-2	Mean	0	2.9E+01	9.6E+01	0.0	7.8E+01	6.5E+01	2.0E+00	0.0	2.5E+01	9.8E+00	3.6E+01	2.1E+02	0.0	3.0E+02	0.0	0
F-3	Mean	0.9	1.5E+01	1.0E+00	1.0E+00	1.1E+01	3.5E+00	1.6E+00	1.7E+00	5.3E+00	5.6E+00	1.0E+00	2.5E+00	9.0E-01	9.4E+00	9.3E-01	0.96
F-4	Mean	0	1.4E+04	8.1E+01	5.3E+03	2.1E+09	5.0E+01	8.4E+03	3.0E+02	2.2E+03	2.8E+00	2.7E+08	1.0E+10	9.0E+05	1.5E+03	7.7E+01	41.01
F-5	Mean	0	2.2E-14	7.5E+00	2.0E-103	5.4E+00	3.3E-01	5.1E-04	4.5E-40	2.0E-01	2.5E-47	8.6E-03	2.5E+01	1.6E-02	3.3E-02	3.4E-182	2.7e-183
F-6	Mean	0	3.9E+25	1.2E+06	1.0E+04	4.2E+07	9.4E+21	6.6E-23	8.0E-25	3.0E+13	1.7E-15	1.2E-01	1.0E+10	1.4E+15	2.3E+02	7.2E-196	6.36e-196
F-7	Mean	0	1.2E-11	3.4E+00	2.0E-15	1.8E+01	2.3E-01	8.6E-14	-7.5E-16	2.8E+00	4.2E-15	3.0E-02	1.0E+01	1.9	3.8E-06	-8.9E-16	-4.4409e-16
F-8	Mean	1	1.7E+02	4.2E+02	1.3E+02	1.0E+04	2.4E+01	5.7E+01	9.8E+01	1.4E+02	3.0E+01	7.4E+04	3.3E+03	2.8E+03	1.5E+01	1.0	1
F-9	Mean	0	3.1E-01	2.6E+00	1.4E-01	2.5E+00	1.6E+00	2.1E-01	1.0E-01	2.7E+00	1.6E-01	4.4E+00	4.7E+00	3.6	5.2E-01	0.0	0
F-10	Mean	-39.16599	NA	-1.6E+03	-1.9E+03	-8.9E+02	-9.0E+02	-1.4E+03	-1.7E+03	-1.6E+03	-1.7E+03	-1.8E+03	-1.6E+03	-2.0E+03	-1.9E+03	-2.0E+03	-1958.3083
F-11	Mean	0	4.9E-15	7.6E-03	8.8E-03	9.4E-01	1.3E-03	1.6E-03	0.0	1.8E-01	0.0	5.3E-02	9.8E-01	1.0	5.9E-04	0.0	0
F-12	Mean	-1	2.2E-18	2.3E-30	-2.4E-01	7.3E-16	1.7E-21	2.2E-23	1.7E-21	8.4E-20	3.7E-23	8.9E-22	4.1E-21	6.9E-22	2.7E-20	-6.8E-01	-0.7
F-13	Mean	0	9.8E-13	7.1E-19	1.4E-20	7.6E-16	7.3E-18	2.7E-12	1.4E-19	4.6E-08	1.7E-19	2.7E-20	1.1E-19	9.9E-20	1.4E-19	1.2E-20	1.20e-20
F-14	Mean	0	5.0E+00	6.2E+01	7.8E-01	1.6E+07	4.7E-02	1.5E+00	1.9E+00	2.1E+01	9.0E-02	2.2E+01	3.1E+01	3.7	1.7E-01	1.3E-32	1.3e-32
F-15	Mean	0	8.6E-01	9.7E+00	1.1E-02	4.3E+06	3.6E-02	5.9E-02	1.6E-03	2.0E+00	1.4E-07	1.5E+00	5.0E+00	5.4E-01	1.1E-03	9.4E-33	9.4e-33
F-16	Mean	0	9.6E-22	2.8E-14	3.7E-218	1.2E-89	1.2E-117	0.0	1.2E-97	1.1E-10	6.6E-217	1.4E-19	1.1E-23	0.0	6.1E-228	0.0	0
F-17	Mean	-195.629	NA	-195.63	-195.63	-195.63	-192.62	-195.63	-195.63	-195.63	-195.63	-195.63	-195.63	-194.57	-195.63	-195.63	-195.629
F-18	Mean	-2.02181	NA	-2.02	-2.02	-2.02	-2.19	-2.02	-2.02	-2.02	-2.02	-2.02	-2.02	-2.01	-2.02	-2.02	-2.0218
F-19	Mean	-106.7645	NA	-106.76	-106.76	-106.71	-80.59	-105.99	-106.76	-105.99	-106.76	-106.47	-106.76	-100.41	-106.76	-106.76	-106.7645
F-20	Mean	-1.0316	NA	-1.03	-1.03	-1.03	-1.01	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03	-0.90	-1.03	-1.03	-1.0316
F-21	Mean	0.3978873	0.90	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.47	0.40	0.40	0.39789
F-22	Mean	-3.862782	NA	-3.86	-3.86	-3.86	-3.10	-3.86	-3.86	-3.86	-3.86	-3.86	-3.86	-3.82	-3.86	-3.86	-3.8628
F-23	Mean	-3.32237	NA	-3.22	-3.27	-2.95	-1.66	-3.28	-3.28	-3.28	-3.32	-3.32	-3.32	-3.08	-3.28	-3.30	-3.2982
F-24	Mean	-2.06261218	NA	-2.06	-2.06	-2.06	-2.85	-2.06	-2.06	-2.06	-2.06	-2.06	-2.06	-2.05	-2.06	-2.06	-2.0626
F-25	Mean	1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.19	1.00	1.00	1

Optimal solution Nearest to optimal

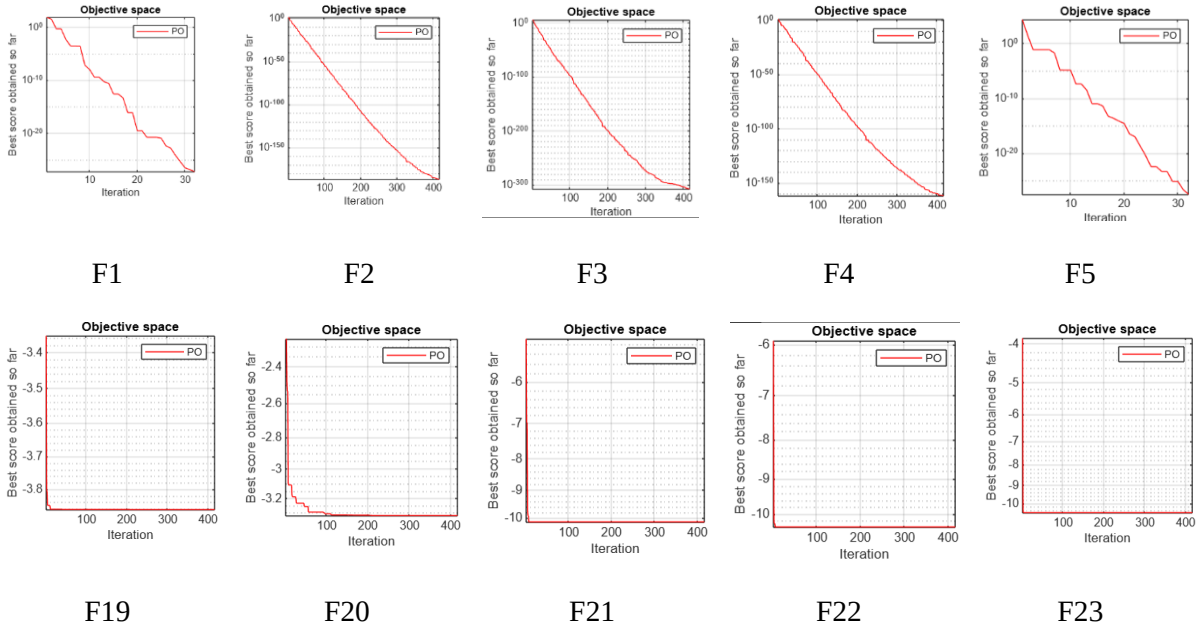


Figure 3: The convergence of the classic political optimizer

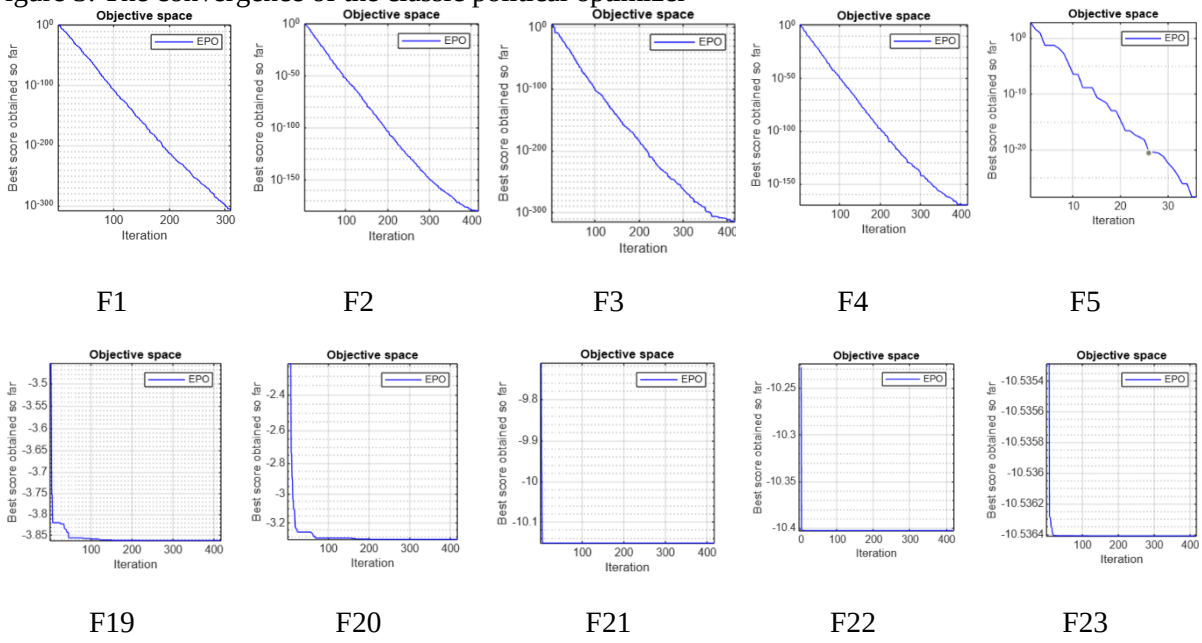


Figure 4. The convergence of the classic Enhancement political optimizer

6. Conclusion

This study presents enhancements to the political optimizer algorithm. The basic optimizer involves the election campaign and party switching mechanism in the exploration technique, and gathers each election phase and parliamentary affairs for the exploitation phase. The algorithm suffered from a weakness in the exploration phase, where the multi-objective functions were not solved, or a good solution could not be found. Also, the exploitation phase missed the optimized solutions repeatedly. Thus, the enhancement idea introduces the basic algorithm to rescue it from these dangers. We aimed to redistribute the leaders' locations inside the election campaign process to open the road for the leaders

to move to new areas. This idea was applied in critical conditions, and each of the six iterations with a threshold value had little effect on the algorithm; the enhancement was very promising, and the result improved very well for both the exploration and exploitation phases. We compared the performance of the enhancement algorithm with the classical political optimizer algorithm beside 13 different state-of-the-art algorithms. The evaluation occurred on 50 benchmark problems, 25 were unimodal and 25 were multimodal objective functions, while the mean result was completed to achieve the stability of the algorithms. Moreover, for future enhancement, the algorithm needs more balance mechanisms to achieve higher results, where there are some functions with alarming results, which means that the algorithm is sometimes stuck in a local minimum, such as F1, F2, and F4. Better results can be achieved by balancing the exploration and exploitation phases.

Conflict of Interest Statement

The article's author declares that there is no conflict of interest.

Contribution Rate Statement Summary of Researchers

The author declares that he has contributed 100% to the article.

References

Arora S., Singh S. Butterfly optimization algorithm: A novel approach for global optimization. Soft Computing 2019; 23(3):715–734. doi:10.1007/s00500-018-3102-4.

Askari Q., Younas I., Saeed M. Political optimizer: A novel socio-inspired meta-heuristic for global optimization. Knowledge-Based Systems. 2020; 195: 105709. doi:10.1016/j.knosys.2020.105709.

Basetti V., Rangarajan SS., Shiva CK., Pulluri H., Kumar R., Collins RE., Senjyu T. Economic emission load dispatch problem with valve-point loading using a novel quasi-oppositional-based political optimizer. Electronics 2021; 10(21): 2596. doi:10.3390/electronics10212596.

Bashkandi AH., Sadoughi K., Aflaki F., Alkhazaleh HA., Mohammadi H., Jimenez G. Combination of political optimizer, particle swarm optimizer, and convolutional neural network for brain tumor detection. Biomedical Signal Processing and Control 2023; 81: 104434. doi:10.1016/j.bspc.2022.104434.

Chander S., Vijaya P., Fernandes R., Rodrigues AP., Maheswari R. Dolphin-political optimized Tversky index based feature selection in Spark architecture for clustering big data. Advances in Engineering Software. 2023; 176: 103331. doi:10.1016/j.advengsoft.2022.103331.

Chatterjee Y., Bourreau E., Rančić MJ. Solving various NP-hard problems using exponentially fewer qubits on a quantum computer. Physical Review A. 2024; 109(5): 052441. doi:10.1103/PhysRevA.109.052441.

Cheraghi R., Jahangir MH. Multi-objective optimization of a hybrid renewable energy system supplying a residential building using NSGA-II and MOPSO algorithms. Energy Conversion and Management. 2023; 294: 117515. doi:10.1016/j.enconman.2023.117515.

Dong J., Zou H., Li W., Wang M. A hybrid greedy political optimizer with fireworks algorithm for numerical and engineering optimization problems. Scientific Reports. 2022; 12(1): 13243. doi:10.1038/s41598-022-17076-4.

Fathy A., Rezk H. Political optimizer based approach for estimating SOFC optimal parameters for static and dynamic models. Energy 2022; 238: 122031. doi:10.1016/j.energy.2021.122031.

Gandomi AH., Alavi AH. Krill herd: a new bio-inspired optimization algorithm. Communications in Nonlinear Science and Numerical Simulation. 2012; 17(12): 4831–4845. doi:10.1016/j.cnsns.2012.05.010.

Khan S., Yairi T., Tsutsumi S., Nakasuka S. A review of physics-based learning for system health management. Annual Reviews in Control. 2024; 57: 100932. doi:10.1016/j.arcontrol.2024.100932.

Kiani F., Anka FA., Erenel F. PSCSO: enhanced sand cat swarm optimization inspired by the political system to solve complex problems. Advances in Engineering Software 2023; 178: 103423. doi:10.1016/j.advengsoft.2023.103423.

Kukkonen S., Lampinen J. An extension of generalized differential evolution for multi-objective optimization with constraints. In: Proceedings of the 2004 Congress on Evolutionary Computation (CEC); 2004 Jun 20-23; Portland, OR, USA. IEEE; 2004. p. 752–61.

Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems. Knowledge-Based Systems. 2016; 96: 120–133. doi:10.1016/j.knosys.2015.12.022.

Mirjalili S., Gandomi AH., Mirjalili SZ., Saremi S., Faris H., Mirjalili SM. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Advances in Engineering Software 2017; 114: 163–191. doi:10.1016/j.advengsoft.2017.07.002.

Mirjalili S., Lewis A.The whale optimization algorithm. Advances in Engineering Software. 2016; 95: 51–67. doi:10.1016/j.advengsoft.2016.01.008.

Mirjalili S., Mirjalili SM., Lewis A. Grey wolf optimizer. Advances in Engineering Software. 2014; 69: 46–61. doi:10.1016/j.advengsoft.2013.12.007.

Moosavian N., Kasaee Roodsari B. Soccer league competition algorithm: a novel meta-heuristic algorithm for optimal design of water distribution networks. Swarm and Evolutionary Computation 2014; 17: 14–24. doi:10.1016/j.swevo.2014.02.002.

Muppidi S., Prakash PG., Kishore B. Dragonfly political optimizer algorithm-based rider deep long short-term memory for soil moisture and heat level prediction in IoT. The Computer Journal. 2023; 66(6): 1350–1365. doi:10.1093/comjnl/bxab215.

Ramnath GS., Harikrishnan R., Muyeen SM., Kukker A., Pohekar SD., Kotecha K. A peer-and self-group competitive behavior-based socio-inspired approach for household electricity conservation. Scientific Reports 2024; 14(1): 17245. doi:10.1038/s41598-024-56926-1.

Rao RV., Savsani VJ., Vakharia DP. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design 2011; 43(3): 303–315. doi:10.1016/j.cad.2010.12.015.

Rashedi E., Nezamabadi-Pour H., Saryazdi S. GSA: a gravitational search algorithm. Information Sciences. 2009; 179(13): 2232–2248. doi:10.1016/j.ins.2009.03.004.

Sahu B., Panigrahi A., Rout SK., Pati A. Hybrid multiple filter embedded political optimizer for feature selection. In: 2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP); 2022; 1–6. IEEE.

Sahu B., Panigrahi A., Rout SK., Pati A. Hybrid multiple filter embedded political optimizer for feature selection. In: 2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP); 2022b; 1–6. IEEE.

Simon D. Biogeography-based optimization. IEEE Transactions on Evolutionary Computation. 2008; 12(6): 702-713. doi:10.1109/TEVC.2008.919004.

Shubham S., Garg H. Recent advances in linear and integer programming models for decision making. Mathematics. 2023; 11(9): 2150. doi:10.3390/math11092150.

Sunitha G., Pushpalatha MN., Parkavi A., Boyapati P., Walia R., Kohar R., Qureshi K. Modeling of chaotic political optimizer for crop yield prediction. Intelligent Automation & Soft Computing 2022; 34(1): 423-437. doi:10.32604/iasc.2022.024757.

Tao H., Abba SI., Al-Areeq AM., Tangang F., Samantaray S., Sahoo A., Siqueira HV., Maroufpoor S., Demir V., Bokde ND., Goliatt L., Jamei M., Ahmadianfar I., Bhagat SK., Halder B., Guo T., Helman DS., Ali M., Sattar S., Al-Khafaji Z., Shahid S., Yaseen ZM. Hybridized artificial intelligence models with nature-inspired algorithms for river flow modeling: a comprehensive review, assessment, and possible future research directions. Engineering Applications of Artificial Intelligence 2024; 129: 107559. doi:10.1016/j.engappai.2023.107559.

Yang XS., Deb S. Cuckoo search via Lévy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC); 2009; 210–214. IEEE.

Yu JJQ., Li VOK. A social spider algorithm for global optimization. Applied Soft Computing 2015; 30: 614–627. doi:10.1016/j.asoc.2015.02.014.

Zhou W., Liu Y., Li M., Wang Y., Shen Z., Feng L., Zhu Z. Dynamic multi-objective optimization framework with interactive evolution for sequential recommendation. IEEE Transactions on Emerging Topics in Computational Intelligence 2023; 7(4): 1228–1241. doi:10.1109/TETCI.2023.3251352.

Zhu A., Gu Z., Hu C., Niu J., Xu C., Li Z. Political optimizer with interpolation strategy for global optimization. PLoS One 2021; 16(5): e0251204. https://doi.org/10.1371/journal.pone.0251204