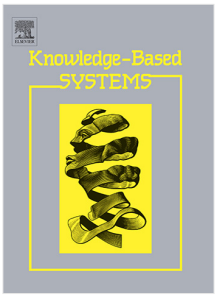
Journal Pre-proof



Political Optimizer: A novel socio-inspired meta-heuristic for global optimization

Qamar Askari, Irfan Younas, Mehreen Saeed

PII: S0950-7051(20)30135-0
DOI: https://doi.org/10.1016/j.knosys.2020.105709
Reference: KNOSYS 105709

To appear in: *Knowledge-Based Systems*

Received date : 13 October 2019
Revised date : 24 February 2020
Accepted date : 28 February 2020

Please cite this article as: Q. Askari, I. Younas and M. Saeed, Political Optimizer: A novel socio-inspired meta-heuristic for global optimization, *Knowledge-Based Systems* (2020), doi: https://doi.org/10.1016/j.knosys.2020.105709.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Published by Elsevier B.V.

Journal Pre-proof

Political Optimizer: A novel socio-inspired meta-heuristic for global optimization

Qamar Askari[a,*], Irfan Younas[a], Mehreen Saeed[a]

[a]*Department of Computer Science, National University of Computer and Emerging Sciences, Lahore, Pakistan*

Abstract

This paper proposes a novel global optimization algorithm called Political Optimizer (PO), inspired by the multi-phased process of politics. PO is the mathematical mapping of all the major phases of politics such as constituency allocation, party switching, election campaign, inter-party election, and parliamentary affairs. The proposed algorithm assigns each solution a dual role by logically dividing the population into political parties and constituencies, which facilitates each candidate to update its position with respect to the party leader and the constituency winner. Moreover, a novel position updating strategy called recent past-based position updating strategy (RPPUS) is introduced, which is the mathematical modeling of the learning behaviors of the politicians from the previous election. The proposed algorithm is benchmarked with 50 unimodal, multimodal, and fixed dimensional functions against 15 state of the art algorithms. We show through experiments that PO has an excellent convergence speed with good exploration capability in early iterations. Root cause of such behavior of PO is incorporation of RPPUS and logical division of the population to assign dual role to each candidate solution. Using Wilcoxon rank-sum test, PO demonstrates statistically significant performance over the other algorithms. The results show that PO outperforms all other algorithms, and consistency in performance on such a comprehensive suite of benchmark functions proves the versatility of the algorithm. Furthermore, experiments demonstrate that PO is invariant to function shifting and performs consistently in very high dimensional search spaces. Finally, the applicability on real-world application is demonstrated by efficiently solving four engineering optimization problems.

Keywords: Optimization, Political optimizer (PO), Meta-heuristics, Socio-inspired algorithm, Past-based position updating, Human behavior-based optimization

1. Introduction

The area of global optimization has attracted considerable attention over the last few decades. The process of optimization becomes harder when discontinuities, incomplete information, dynamicity, and uncertainties are involved. In most of these scenarios, optimization problems become NP-hard and exact algorithms demand exponential time or may not find the optimal solution at all. In such cases, the approximation algorithms come into play and offer a near-optimal solution by dealing with computational intractability. Finding the near-optimal solution for inapproximable problems by using traditional approximation algorithms may become as difficult as finding the exact optimal solution. The nature-inspired meta-heuristics have reasonably dealt with such inapproximable nature of optimization problems and in the last 2 to 3 decades, a lot of research has been done on meta-heuristics [1].

The word "meta-heuristic" was first coined by Fred Glover [2]. There are two major characteristics, which a good meta-heuristic should balance: Intensification (exploitation) and diversification (exploration). Exploration preserves the diversity and finds the promising areas. In the absence of this characteristic, the algorithm may converge prematurely to some local optimum. Exploitation allows the algorithm to search the promising areas discovered in the phase of exploration and in absence of this capability, the algorithm may not even converge. Thus, a very balanced combination of these two capabilities is required for an algorithm to reach the global optimum.

*Corresponding author

Email addresses: l1165502@lhr.nu.edu.pk (Qamar Askari), irfan.younas@nu.edu.pk (Irfan Younas), mehreen.saeed@nu.edu.pk (Mehreen Saeed)

Nature-inspired meta-heuristic algorithms have been classified in 4 major groups in the literature: Evolution-based, swarm-based, physics-based, and human behavior-based algorithms [3, 4]. The evolutionary algorithms are inspired by the concept of natural evolution. Few well-known evolutionary algorithms are Genetic Algorithm (GA) [5], Genetic Programming (GP) [6], Differential Evolution (DE) [7], Biogeography Based Optimizer (BBO) [8], and Evolutionary Strategy (ES) [9]. Swarm-based algorithms are inspired by the collaborative natural behavior of living organisms e.g. how they hunt, how they find their food, how they mate or how they save themselves from their hunters etc. Few well-known swarm-based algorithms are Ant colony optimization (ACO) [10], Particle Swarm Optimization (PSO) [11], Grey Wolf Optimizer (GWO) [12], Whale Optimization Algorithm (WOA) [4], Krill Herd (KH) [13], Salp Swarm Optimization (SSA1) [14], Social Spider Algorithm (SSA2) [15], Butterfly Optimization Algorithm (BOA) [16], Harris Hawk Optimization (HHO) [17], Cuckoo Search (CS) [18], Naked-Mole Rate (NMR) [19], Bald Eagle Search (BES) [20], Sailfish Optimizer (SFO) [21], Artificial Coronary Circulation System (ACCS) [22], Emperor Penguins Colony (EPC) [23], Artificial Feeding Birds (AFB) [24], Sea Lion Optimization (SLnO) algorithm [25], Spider Monkey Optimization (SMO1) [26], Monarch Butterfly Optimization (MBO) [27], Hitchcock bird-inspired algorithm (HBIA) [28], Normative Fish Swarm Algorithm (NFSA) [29], Seagull Optimization Algorithm (SOA) [30], and Squirrel Search Algorithm (SSA3) [31]. Physics-based algorithms are inspired by the laws of physics working behind natural phenomena. Few examples are Simulated Annealing (SA) [32], Gravitational Search Algorithm (GSA) [33], Black Hole (BH) algorithm [34], Sine Cosine Algorithm (SCA) [35], Big-Bang Big-Crunch (BB-BC) optimization algorithm [36], Water Cycle Algorithm (WCA) [37], Artificial Electric Field Algorithm (AEFA) [38], Equilibrium Optimizer (EO) [39], Newton particle optimizer (NwPO) [40], Quantum Approximate Optimization Algorithm (QAOA) [41], Physarum-Energy Optimization algorithm (PEO) [42], and Thermal Exchange Optimization (TEO) [43]. Finally, a lot of optimization algorithms have been proposed in the literature, which are inspired by the social behaviors of the human beings. For example, Teaching-Learning Based Optimization (TLBO) [44], Soccer League Competition (SLC) [45] algorithm, Exchange Market Algorithm (EMA) [46], Socio Evolution and Learning Optimization Algorithm (SELO) [47], Nomadic People Optimizer (NPO) [48], Ludo Game-based Swarm Intelligence (LGSI) [49], Social Mimic Optimization (SMO2) [50], Bus Transportation Algorithm (BTA) [51], and Brain Strom Optimization (BSO) [52].

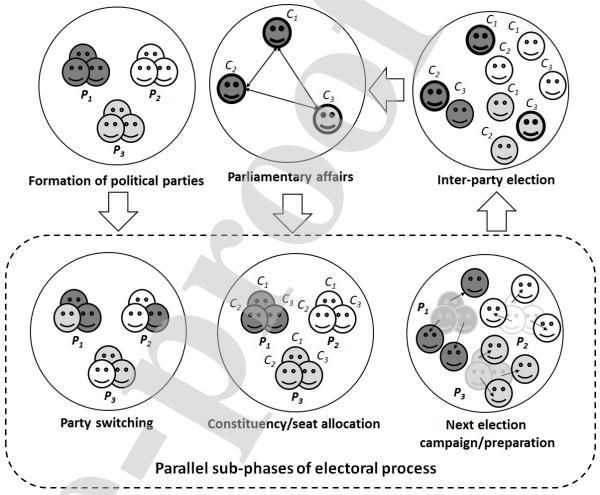


Figure 1: Illustration of the multi-phased political process - A perspective. P_i represents members of i^{th} political party, C_i represents the member(s) contesting election from i^{th} constituency, face color distinguishes members of one party from the other parties, and face in bold denotes the winner from the constituency labeled on that face.

Politics in different contexts holds different meanings. In our work, we use the political system of a country as a reference point and mimic the behavior of politicians to achieve the end goal of optimization. Politics is about the governance of a region, state or country. The party-based political systems can be categorized in four major types: One-party political system, two-party political system, dominant-party political system, and multi-party political system [53, 54]. Each of them is applicable in different countries/states with several variations. Moreover, the governance of a state/country can be practiced through either parliamentary system or presidential system [55]. We have generalized PO by incorporating a few commonalities from these systems, such as the concept of parties and constituencies, association of politicians with political parties, collaboration between the politicians associated with same party and competition between the politicians through inter-party election, change of affiliation of politicians to parties, campaigning before election for votes, and collaboration between the elected members in parliament. The politics being practiced in multi-party democracy is a complex political process, which covers a very wide range of social levels. The process is illustrated in Figure 1 and comprises the following phases:

- Party formation: A political party is formed by the

Journal Pre-proof

individuals having a common agenda. The individuals are called politicians or candidates for public office. In Figure 1, P_i represents members of i^{th} party and face color distinguishes members of one party from the other parties.

- Party switching: Members can switch their affiliation to a party at any time. It is demonstrated in Figure 1 by switching a member from P_1 to P_2, from P_2 to P_3, and from P_3 to P_1.
- Ticket allocation: The members are allocated party tickets to contest an election from a constituency. The constituency may be considered as a group of voters (constituents), who elect a candidate representing a political party. In Figure 1, the label C_i above each face denotes that the member is contesting election from i^{th} constituency.
- Election campaign: The candidates visit their constituencies and convince their voters to elect them which results in goodwill/status updating of the candidates. In Figure 1, it is demonstrated by shifting the positions of the faces.
- Inter-party election: In each constituency, the constituents vote for a candidate to decide a winner. In Figure 1, the winner from each constituency is shown in bold. For example, P_1 wins from C_1 and C_2, P_3 wins from C_3, and P_2 does not win from any constituency.
- Government formation and parliamentary affairs: The elected members (winners) from all parties form the parliament and collaborate with each other to run the government. It is depicted in the phase titled 'parliamentary affairs' of Figure 1.

The main objective of this paper is to propose a new meta-heuristic called Political Optimizer (PO), which is inspired by the multi-phased political process. Although, the idea of using politics as an inspiration for optimization algorithms is not new but politics is a very diverse and complex process, which encourages us to map the inspiration from a totally different perspective. The existing algorithms based on politics are: Greedy Politics Optimization (GPO) [56] inspired by the political strategies adopted during state assembly elections, Parliamentary Optimization Algorithm (POA) [57] inspired by the competition between parliamentarians for head elections, and Election Campaign Optimization (ECO) algorithm [58] inspired by the concept of motivation of candidates in the election campaign. The description of these algorithms is provided in Section 2. However, the following properties make PO distinct and different from other politics-based algorithms.

- The concept is mapped from a totally different perspective and unlike the earlier politics-inspired algorithms, PO is the mathematical mapping of all the major phases of politics, such as party formation, party-ticket/constituency allocation, election campaign and party switching, inter-party election, and parliamentary affairs after government formation.
- A novel position updating strategy called recent past-based position updating strategy (RPPUS) is proposed, which is the mathematical modeling of the learning behavior of politicians from the previous election.
- Each individual solution plays a dual role. It acts as a party member as well as an election candidate. This allows a solution to update its position with respect to 2 improved solutions: the party leader and the constituency winner.
- The phase of parliamentary affairs is incorporated, which allows cooperation between better solutions to further improve the solutions.

The performance of PO is evaluated on a suite of 50 mathematical benchmark test functions against 15 state of the art algorithms. The suite contains unimodal, multimodal and fixed dimensional benchmark functions to demonstrate the versatility of PO. Moreover, we conducted experiments to show that PO is invariant to function shifting and performs well for the high dimensional functions. Finally, in order to evaluate the applicability of PO to real world problems, the performance of PO is evaluated by solving 4 engineering optimization problems. The results exhibit superior optimization capability of PO for engineering problems as well.

The rest of the article is structured as follows. A few well-known human social behavior based algorithms and position updating strategies of a few comparative algorithms are discussed in Section 2. Section 3 presents how the inspiration derives PO and its corresponding mathematical model. The position updating mechanism and time complexity of PO are compared with other well-known and mathematically similar algorithms in Section 4. Experiments and simulations are described in Section 5. The performance of PO is analyzed using four engineering optimization problems in Section 6. The managerial implications are presented in Section 7. Finally, Section 8 presents the concluding remarks and future directions.

Journal Pre-proof

2. Related works

Taking into account the relevance of the proposed research with the field of human behavior-inspired metaheuristics, it is important to discuss some well-known human behavior-inspired optimization algorithms, especially those inspired by the politics. In addition, position updating strategies of a few well-known algorithms, which are contrasted with the proposed algorithm in Section 4, are also discussed in this section.

2.1. Human social behavior inspired algorithms

Society and Civilization Optimization (SCO) [59] maps the notion of social interaction among members of the societies. The group of individuals is called society and it is assumed that there are several societies. The individuals in societies interact with each other in order to enhance their fitness and societies interact with each other in order to map the concept of civilization. Imperialist Competitive Algorithm (ICA) [60] simulates the imperialist nations competition to take the control of weaker colonies, which will further strengthen the stronger imperialists and weaken the weaker nations unless they collapse. League Championship Algorithm (LCA) [61] maps the concept of league matches. The teams compete with each other in league matches over the weeks according to the schedule and the winner is proclaimed the best team at the end of the season. LCA considers teams as solutions, weeks as iterations, team strength as fitness, and the end of the season as a terminating condition. Soccer League Competition (SLC) algorithm [45] is inspired by the soccer league team competition. The population is divided into teams and teams are divided into two types of players: fixed players and substitutes. Each player is considered a candidate solution. Teams compete for top position on the point table and players in a team compete for the better performance than the other. The overall competition results in all teams having the best players (solutions). Social Group Optimization (SGO) [62] is inspired by the social interaction of individuals in a group to solve complex problems. The knowledge of each individual is mapped by its fitness. The algorithm involves two phases: improving phase and acquiring phase. In the improving phase each individual improves knowledge by interacting with the best person (best solution) and in acquiring phase the individuals interact with randomly selected individuals and the best person simultaneously to acquire knowledge.

2.1.1. Recent development in human -behavior inspired algorithms

Nomadic People Optimizer [48] is a recently proposed algorithm inspired by the behavior of nomadic people, which they adopt to search for sources of life and to live for hundreds of years by migrating to their most comfortable zones. Ludo Game-based Swarm Intelligence (LGSI) [49] is also a newly proposed algorithm, which is inspired by the Ludo game playing strategies adopted by the players. Social Mimic Optimization (SMO) [50] is one another recently developed human behavior inspired optimization algorithm, which simulates people's behavior in society through which people try to assimilate to famous people through imitating their behavior. Find-Fix-Finish-Exploit-Analyze (F3EA) [63] is inspired by the targeting process, which was practiced in the war between Iraq and Afghanistan. This process involves five phases: finding the target, precising the location of the target, destroying the target, information gathering from the target area, and evaluating the results. A few more recently developed human-behavior inspired algorithms are: Bus Transportation Algorithm (BTA) [51] inspired by smart behavior of humans in transportation to reach their destination, Deer Hunting Optimization Algorithm (DHOA) [64] inspired by the deer hunting strategy of humans, and Brain Storm Optimization (BSO) [52] inspired by humans brainstorming process.

2.2. Existing politics inspired algorithms

Parliamentary Optimization Algorithm (POA) [57] is inspired by the competitive behavior of the political parties in a parliament. There is a division of population into political groups. There are two stages involved in this algorithm. The regular members (ordinary solutions) are attracted to the candidates (better solutions) in the first stage and in the second stage, groups are merged with a certain probability and the rest of the groups are discarded. Greedy Politics Optimization (GPO) [56] is inspired by the policies adopted during state assembly elections by the political parties. Population is divided into party members and independents. In election, the best on each seat is proclaimed the winner, and those who lose the election are substituted by alternatives that are closely located. Diversity is implemented through a method called scandal that mutates solutions. Election Campaign Optimization (ECO) [58] is influenced by the notion of candidate motivation for the greatest support in an election campaign. The solution space is regarded to consist of global-survey voters, local-survey voters and candidates. The prestige of each individual

Journal Pre-proof

Table 1: Comparison of PO with other algorithms based on their position-updating mechanism and a few well-known performance evaluation metrics. Weaknesses of the other algorithms are highlighted with references from the literature or learned from the results on benchmark functions.

	Algorithms								
Position updating mechanism	**PSO**	**GWO**	**WOA**	**SCA**	**TLBO**	**POA**	**GPO**	**ECO**	**PO**
Utilization of global best position	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No
Utilization of subgroup best	No	No	No	No	No	Yes	Yes	Yes	Yes
Utilization of randomly selected solution	No	No	Yes	No	Yes	No	Yes	No	Yes
History maintenance of solutions	Yes	No	No	No	No	No	No	No	Yes
Interaction between better solutions for further improvement	No	No	No	No	No	No	No	No	Yes
Tunable parameters	3	0	0	0	0	10	5	4	1
Provision of the mechanism to balance exploration and exploitation in paper	No	Yes	Yes	Yes	No	No	No	No	Yes
Evaluation metrics	**PSO**	**GWO**	**WOA**	**SCA**	**TLBO**	**POA**	**GPO**	**ECO**	**PO**
Exploitative capability	Normal	Weak [65]	Weak [66]	Weak	Normal	Weak [56]	Weak	Normal	Good
Exploration capability	Normal	Good	Normal [67]	Normal	Normal	Weak	Weak	Good	Good
Premature convergence avoidance	Weak [68, 69]	Average [70]	Average [71]	Weak [72, 73]	Weak [74–76]	Weak [56]	Weak	Weak	Normal
Convergence speed	Slow [69, 77]	Slow [78]	N/A	Slow [79, 80]	Slow [81]	Slow [56]	Weak	Normal	Fast
Capability to converge on local/global optimum	Weak [82]	Weak [65]	Weak	Weak [79, 80]	Strong	Weak	Weak	N/A	Strong
Dependence on location of the global optimum	No	Yes [65]	Yes	Yes	Yes	No	No	No	No
Parameter tuning overhead	Average	Nil	Nil	Nil	Nil	V.High	High	High	Low

is mapped with the fitness. In the search space, the candidates are generated randomly. Global sample-survey voters are generated in the entire search space evenly and local sample-survey voters are generated in investigated ranges of candidates. The prestige of candidates and voters is compared and a voter replaces the candidate if a voter's prestige is better than the candidate's prestige.

2.3. Position updating strategies of comparative algorithms

Like PSO [11], GWO [12], WOA [4], SCA [35], and TLBO [44], the algorithm proposed in this paper also utilizes the position of the better solution to update the position of a search agent. To compare the position updating strategy of PO with such algorithms, it is required to first discuss their position updating strategies.

PSO is inspired by the social behavior of bird flocks. In PSO, a search agent interacts with the best among all solutions known as the global best and best position the search agent has so far discovered known as the local best. A search agent is attracted to the global best position, the local best position, and the direction in which the search agent is currently moving. GWO is inspired by the hunting behavior of the grey wolves. The search agents update their position with reference to the top three best solutions named as alpha (α), beta (β), and delta (δ). First the next position is calculated in reference to each better solution and then the search agent jumps to the mean of independently calculated positions. WOA is inspired by the hunting behavior of humpback whales. WOA mimics three behaviors of whales known as searching the prey, encircling the prey and the bubble-net attacking method. The encircling behavior is modeled by searching around the best solution in a circular way. The attacking behavior is modeled by searching the space around the best solution in spiral form. Finally, the searching behavior is modeled by exploring the region around a randomly selected solution. Self-adaptive parameters are designed to decide the behavior of a search agent. In SCA too, the search agents interact with global best to update their positions. However, the best solution is just considered a reference position to find the difference, while the search agent explores the vicinity around itself by adding the weighted difference in its own position. Uniqueness of the this algorithm is the utilization of sine and cosine functions to explore and exploit the search space. TLBO is inspired by the teaching-learning behavior in a classroom. TLBO involves two phases: teacher phase and learner phase. In the teacher phase, each search agent updates the position with respect to the best solution (teacher) and mean position of all search agents (class). However, in the learner phase, a search agent interacts with two random solutions to update its position.

The algorithms discussed above have been critically evaluated in the literature and many limitations have been identified in these algorithms. Original version of PSO has a few shortcomings, such as premature convergence [69], inability to escape local optima in complex

functions [68], slow convergence [77], and not guaranteed to converge on local/global optimum [82]. GWO faces slow convergence and low precision in many cases [78]. Although, first half of the iterations allow exploration and rest of the iterations allow exploitation but both are not well-balanced for complex functions [70]. The performance of GWO is also shown to be dependent on the location of the global optimum [65]. WOA faces poor exploration [67], low local search mechanism [66], inability to escape complex local optima [71], and poor performance for the functions having global optimum away from the origin. SCA has a few shortcomings too, such as low optimization precision [72], premature convergence [73], slow convergence, local optima stagnation, skipping of true solutions, and overflow of diversity [79, 80]. TLBO has a few limitations, such as unbalanced exploration and exploitation [74], slow convergence speed [81], dependence on the location of the global optimum, and sticking in local optima [75, 76]. Moreover, the existing politics-based algorithms require tuning of 4 to 10 parameters while the results reported in their respective papers are not very promising. On the contrary, we show in section 5 that PO has excellent exploitative capability with very good convergence speed; however, a balance between exploration and exploitation is also attained to escape local optima and avoid premature convergence. Furthermore, PO is independent of the position of the global optimum unlike GWO, WOA, SCA, and TLBO. Another distinctive feature of PO, which we also demonstrate in section 5, is consistent performance on very high dimensional search spaces. By referring the literature and learning from experiments, we summarize a comparison of PO with comparative algorithms in Table 1 based on the evaluation metrics and position updating mechanisms.

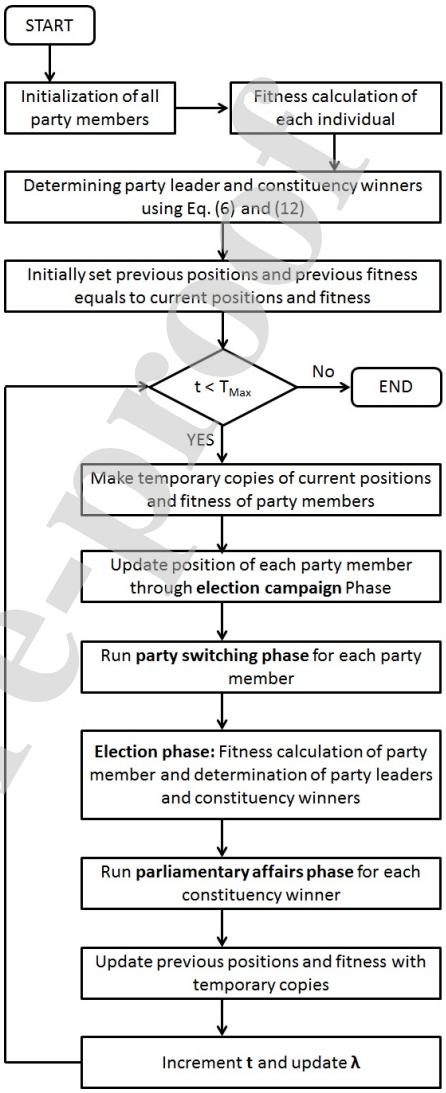


Figure 2: Flowchart of the proposed algorithm.

3. Political optimizer

Politics itself is a process of optimization from two perspectives: each individual optimizes its goodwill to win the election and each party tries to maximize its number of seats in parliament to form a government. These aspects make politics an ideal inspiration for an optimization algorithm because an individual (a party member) may be considered a candidate solution, individual's goodwill is considered the position of the candidate solution in the search space, and goodwill of a political member can be defined by many performance-related parameters which can be mimicked by design variables or components of the position vector of a candidate solution. Election may be considered the evaluation (objective) function and the number of votes obtained by an individual in election is mapped by the fitness of the candidate solution. Furthermore, following are the four major inspirational aspects of the politics that motivate us to propose a new optimization algorithm.

- The electoral process, in which election candidates campaign for votes.
- The intra-party collaboration and inter-party competitiveness.
- The analytical behavior of election candidates to improve their performance based on their experience in the past election.

Journal Pre-proof

- The interaction and cooperation of the winning candidates with each other to run the government in the post-election phase.

Political Optimizer (PO) is structured as a sequence of five phases involving party formation and constituency allocation, election campaign, party switching, inter-party election, and parliamentary affairs. The phase of party formation and constituency allocation runs just once for the sake of initialization and rest of the four phases execute in a loop. The flowchart of the proposed algorithm is presented in Figure 2.

Table 2: List of the variables used in mathematical formulations of the political phases and their descriptions.

Variable	Description
$\mathcal{P}$	set of all political parties (whole population)
$\mathcal{P}_i$	i^{th} political party
$\boldsymbol{p}_i^j$	j^{th} member of i^{th} party
$\mathcal{P}_{i,k}^j$	k^{th} dimension of j^{th} member of i^{th} political party
C	set of all constituencies
C_j	j^{th} constituency
$\boldsymbol{p}_i^*$	leader of i^{th} political party
$\boldsymbol{c}_j^*$	winner of j^{th} constituency
λ	party switching rate
n	number of parties, constituencies, and members in each party
T_{max}	total number of iterations

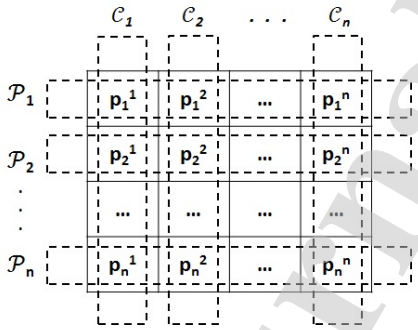


Figure 3: Illustration of the logical division of the population $\mathcal{P}$ in political parties and constituencies.

3.1. Mathematical model and optimization algorithm

In this subsection, it is described how each phase of the multi-party political system is mathematically modeled to propose the new algorithm, which is presented in Algorithm 1. The mapping is kept very simple. However, in Section 8 it is suggested that the variants of PO can be proposed by trying different mappings. Variables used in mathematical formulation of the political phases along with their descriptions are presented in Table 2.

Algorithm 1 Main Framework of the Proposed PO

Input: n (number of constituencies, political parties and party members), λ_{max} (upper limit of the party switching rate), T_{max} (total number of iterations)
Output: final population $\mathcal{P}(T_{max})$
/*Initialization*/
initialize $\mathcal{P}$ as expressed in Eq. (1) to Eq. (5) and depicted in Figure 3
compute and save the fitness of each member p_i^j
compute the set of the party leaders $\mathcal{P}^*$, by using Eq. (6)
compute the set of the constituency winners C^*, by using Eq. (12)
$t = 1$;
$\mathcal{P}(t-1) = \mathcal{P}$;
$f(\mathcal{P}(t-1)) = f(\mathcal{P})$;
$\lambda = \lambda_{max}$;
while $t \leq T_{max}$ **do**
  $\mathcal{P}_{temp} = \mathcal{P}$;
  $f(\mathcal{P}_{temp}) = f(\mathcal{P})$;
  foreach $\mathcal{P}_i \in \mathcal{P}$ **do**
    foreach $\boldsymbol{p}_i^j \in \mathcal{P}_i$ **do**
      $\boldsymbol{p}_i^j$ = ElectionCampaign($\boldsymbol{p}_i^j$, $\boldsymbol{p}_i^j(t-1)$, $\boldsymbol{p}_i^*$, $\boldsymbol{c}_j^*$);
    end
  end
  PartySwitching($\mathcal{P}$, λ);
  /*Election phase*/
  compute and save the fitness of each member p_i^j
  compute the set of the party leaders $\mathcal{P}^*$, by using Eq. (6)
  compute the set of the constituency winners C^*, by using Eq. (12)
  ParliamentaryAffairs(C^*, $\mathcal{P}$);
  $\mathcal{P}(t-1) = \mathcal{P}_{temp}$;
  $f(\mathcal{P}(t-1)) = f(\mathcal{P}_{temp})$;
  $\lambda = \lambda - \lambda_{max}/T_{max}$;
  $t = t + 1$;
end

3.1.1. Party formation and constituency allocation

The population $\mathcal{P}$ is divided in n political parties, as expressed in Eq. (1). Each party $\mathcal{P}_i$ consists of n candidates/members, as shown in Eq. (2). Each j^{th} member $\boldsymbol{p}_i^j$ is considered a potential solution, which is a d-dimensional vector, as presented in Eq. (3). In Eq. (3) the value of d is the number of input variables of the problem being solved and $p_{i,k}^j$ denotes k^{th} dimension of $\boldsymbol{p}_i^j$.

$$\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \ldots, \mathcal{P}_n\} \tag{1}$$

$$\mathcal{P}_i = \{\boldsymbol{p}_i^1, \boldsymbol{p}_i^2, \boldsymbol{p}_i^3, \ldots, \boldsymbol{p}_i^n\} \tag{2}$$

$$\boldsymbol{p}_i^j = [p_{i,1}^j, p_{i,2}^j, p_{i,3}^j, \ldots, p_{i,d}^j]^T \tag{3}$$

Journal Pre-proof

In addition to its role of a party member, a potential solution also acts as an election candidate. It is assumed that there are n constituencies, as illustrated in Eq. (4) and j^{th} member of each party contests election from the j^{th} constituency C_j, as expressed in Eq. (5). The population and its logical division in political parties and constituencies is depicted in Figure 3. This logical division requires the tuning of only one parameter n. In our mapping the number of parties, the number constituencies and the number of candidates in each party are all set to an equal value n. The phase of party formation and constituency allocation by assuming $n = 3$ is illustrated in Figure 4 (a).

$$C = \{C_1, C_2, C_3, \ldots, C_n\} \tag{4}$$

$$C_j = \{\boldsymbol{p_1^j}, \boldsymbol{p_2^j}, \boldsymbol{p_3^j}, \ldots, \boldsymbol{p_n^j}\} \tag{5}$$

The fittest member of a party is declared the party leader, which is decided right after the general election (inter-party election). The selection of the party leader is modeled in Eq. (6), where $\boldsymbol{p_i^*}$ denotes the leader of the i^{th} party and $f(\boldsymbol{p_i^j})$ computes the fitness of $\boldsymbol{p_i^j}$. It is illustrated in Figure 4 (b).

$$q = \underset{1 \le j \le n}{\operatorname{argmin}} f(\boldsymbol{p_i^j}), \qquad \forall i \in \{1, \ldots, n\}$$
$$\boldsymbol{p_i^*} = \boldsymbol{p_i^q} \tag{6}$$

The set of all the party leaders is represented by $\mathcal{P}^*$ as expressed in Eq. (7).

$$\mathcal{P}^* = \{\boldsymbol{p_1^*}, \boldsymbol{p_2^*}, \boldsymbol{p_3^*}, \ldots, \boldsymbol{p_n^*}\} \tag{7}$$

After the election, the winners from all the constituencies become the parliamentarians. In Eq. (8), C^* represents the set of all the parliamentarians and $\boldsymbol{c_j^*}$ denotes the winner of j^{th} constituency. It is also depicted in Figure 4 (b).

$$C^* = \{\boldsymbol{c_1^*}, \boldsymbol{c_2^*}, \boldsymbol{c_3^*}, \ldots, \boldsymbol{c_n^*}\} \tag{8}$$

3.1.2. Election campaign (Exploration and exploitation)

This phase helps candidates to improve their performance in the election. In this paper, we map three aspects of this phase, which are discussed as follows:

1. Learning from the previous election is mapped by proposing a novel position updating strategy called recent past-based position updating strategy (RPPUS), which is mathematically formulated by using Eq. (9) and Eq. (10).
2. The influence of the vote bank of the party leader is mapped by updating the position of the members with reference to the party leader.
3. The comparative analysis with the constituency winner is modeled by updating the position of the candidates with reference to the constituency winner.

The whole process of election campaign is expressed in Algorithm 2, which uses Eq. (9) and Eq. (10) to update the position of a candidate. Depending upon the relationship of current fitness $f(\boldsymbol{p_i^j}(t))$ of a candidate with its previous fitness $f(\boldsymbol{p_i^j}(t-1))$, it is decided whether to update the position by using Eq. (9) or Eq. (10). If the fitness improves, then Eq. (9) is used to update the position, however, if the fitness deteriorates, then Eq. (10) is used. In both scenarios, the position is first updated with reference to the party leader $\boldsymbol{p_i^*}$ and then with reference to the constituency winner $\boldsymbol{c_j^*}$. It should be noted that the r and m^*, which are presented in Algorithm 2, are used in both equations to update the position of the candidate. The variable r is a random number in the range $[0, 1]$ and m^* first holds the value of k^{th} dimension of the party leader $p_{i,k}^*$ then the constituency winner $c_{j,k}^*$.

Both Eq. (9) and Eq. (10) have three cases each, which are graphically illustrated in Figure 5. Parts (a - c) illustrate the cases of Eq. (9) and (d - f) illustrate the cases of Eq. (10). The purpose of these cases is to identify and exploit the most promising region, which is highlighted with grey color in each part of Figure 5. The cases of Eq. (9) are discussed below, while the cases of Eq. (10) can be interpreted in similar way.

$$p_{i,k}^j(t+1) = \begin{cases} m^* + r(m^* - p_{i,k}^j(t)), & \text{if } p_{i,k}^j(t-1) \le p_{i,k}^j(t) \le m^* \text{ or } p_{i,k}^j(t-1) \ge p_{i,k}^j(t) \ge m^* \\ m^* + (2r-1)|m^* - p_{i,k}^j(t)|, & \text{if } p_{i,k}^j(t-1) \le m^* \le p_{i,k}^j(t) \text{ or } p_{i,k}^j(t-1) \ge m^* \ge p_{i,k}^j(t) \\ m^* + (2r-1)|m^* - p_{i,k}^j(t-1)|, & \text{if } m^* \le p_{i,k}^j(t-1) \le p_{i,k}^j(t) \text{ or } m^* \ge p_{i,k}^j(t-1) \ge p_{i,k}^j(t) \end{cases} \tag{9}$$

Journal Pre-proof

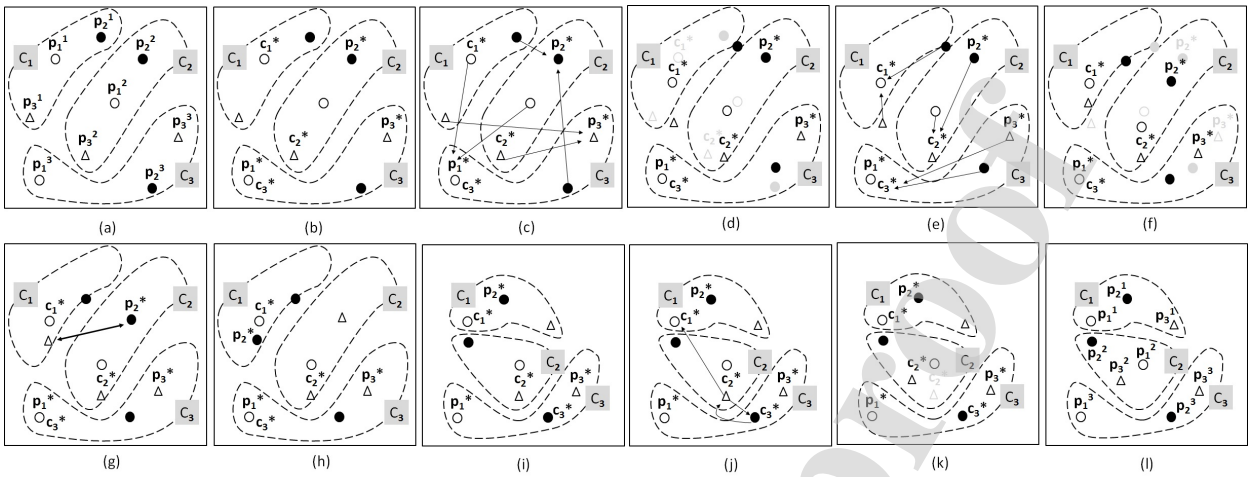


Figure 4: Demonstration of all phases of PO by assuming $n = 3$, where n defines the number of political parties, number of members in each party, and number of constituencies. Dotted regions denote the constituencies and members of each party are represented with different symbols. (a) Initialization of population. (b) Party leaders and constituency winners are decided. (c - d) Position updating with reference to the party leaders. (e - f) Position updating with reference to the constituency winners. (g - h) Party switching, where p_2^2 is swapped with p_3^1. (i) Election phase and reallocation of roles of party leaders and constituency winners. (j - k) Phase of parliamentary affairs - each parliamentarian is attracted towards a random parliamentarian and its position is updated if its fitness is improved. (l) Resultant positions after the whole process.

$$p_{i,k}^{j}(t+1) = \begin{cases} m^* + (2r-1)|m^* - p_{i,k}^{j}(t)|, & \text{if } p_{i,k}^{j}(t-1) \le p_{i,k}^{j}(t) \le m^* \text{ or } p_{i,k}^{j}(t-1) \ge p_{i,k}^{j}(t) \ge m^* \\ p_{i,k}^{j}(t-1) + r(p_{i,k}^{j}(t) - p_{i,k}^{j}(t-1)), & \text{if } p_{i,k}^{j}(t-1) \le m^* \le p_{i,k}^{j}(t) \text{ or } p_{i,k}^{j}(t-1) \ge m^* \ge p_{i,k}^{j}(t) \\ m^* + (2r-1)|m^* - p_{i,k}^{j}(t-1)|, & \text{if } m^* \le p_{i,k}^{j}(t-1) \le p_{i,k}^{j}(t) \text{ or } m^* \ge p_{i,k}^{j}(t-1) \ge p_{i,k}^{j}(t) \end{cases} \tag{10}$$

- Case 1 of Eq. (9): In this case, the current position of the candidate lies between its previous position and the referenced solution. This case is diagrammed in Figure 5 (a), where the most promising region to exploit is highlighted next to the referenced solution and bounded by the distance $\Delta = \mid m^* - p_{i,k}^{j}(t) \mid$.

- Case 2 of Eq. (9): This case occurs when the referenced solution lies between the current and the previous position of the candidate. This case is diagrammed in Figure 5 (b). The most promising region to exploit is highlighted around the position of the referenced solution m^* and the diameter of the region is decided by the distance $\Delta = \mid m^* - p_{i,k}^{j}(t) \mid$.

- Case 3 of Eq. (9): In this case, the previous position of the candidate lies between its current position and the referenced solution. This case is diagrammed in Figure 5 (c). The most promising region to exploit is around the position of the referenced solution m^*, however, in this case the diameter is bounded by the distance $\Delta = \mid m^* - p_{i,k}^{j}(t-1) \mid$ because $f(\boldsymbol{p_i^j}(t-1)) > f(\boldsymbol{p_i^j}(t))$.

The position updating mechanism with reference to the party leader and the constituency winner is depicted in Figure 4 (c - d) and Figure 4 (e - f), respectively.

3.1.3. Party switching (Balancing exploration and exploitation)

In politics, this phase runs in parallel to the election campaign but in PO we run this phase after election campaign phase. An adaptive parameter λ called party switching rate is defined, which starts from λ_{max} and linearly reduces to 0 over the course of iterations. Each member $\boldsymbol{p_i^j}$ is selected with probability λ and switched to some randomly selected party $\mathcal{P}_r$, where it is swapped/exchanged with the least fit member $\boldsymbol{p_r^q}$ of that party $\mathcal{P}_r$. The computation of the index q of the

Journal Pre-proof

Algorithm 2 ElectionCampaign($\boldsymbol{p}_i^j$, $\boldsymbol{p}_i^j(t-1)$, $\boldsymbol{p}_i^*$, $\boldsymbol{c}_j^*$)

Result: $\boldsymbol{p}_i^j(t+1)$ ▷ updated position of $\boldsymbol{p}_i^j$
if $f(\boldsymbol{p}_i^j(t)) \leq f(\boldsymbol{p}_i^j(t-1))$ **then**
  for $k \leftarrow 1$ *to* d **do**
    $m^* \leftarrow p_{i,k}^*$ ▷ where $\boldsymbol{p}_i^*$ is the leader of i^{th} party
    $r \leftarrow$ random number from the interval [0, 1]
    ▷ Update the position with respect to the party leader
    $p_{i,k}^j \leftarrow$ update $p_{i,k}^j(t)$ by using Eq. (9)
    $m^* \leftarrow c_{j,k}^*$ ▷ where $\boldsymbol{c}_j^*$ is the winner of j^{th} constituency
    $r \leftarrow$ random number from the interval [0, 1]
    ▷ Update the position with respect to the constituency winner
    $p_{i,k}^j(t+1) \leftarrow$ update $p_{i,k}^j$ by using Eq. (9)
  end
else
  for $k \leftarrow 1$ *to* d **do**
    $m^* \leftarrow p_{i,k}^*$
    $r \leftarrow$ random number from the interval [0, 1]
    ▷ Update the position w.r.t the party leader
    $p_{i,k}^j \leftarrow$ update $p_{i,k}^j(t)$ by using Eq. (10)
    $m^* \leftarrow c_{j,k}^*$
    $r \leftarrow$ random number from the interval [0, 1]
    ▷ Update the position w.r.t the constituency winner
    $p_{i,k}^j(t+1) \leftarrow$ update $p_{i,k}^j$ by using Eq. (10)
  end
end

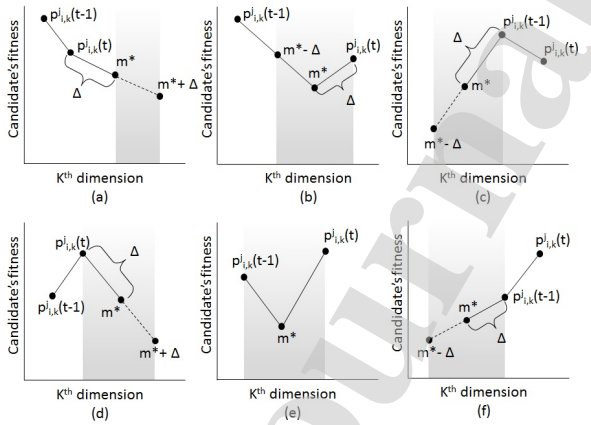


Figure 5: Illustration of RPPUS, which demonstrates all 6 cases of Eq. (9), and Eq. (10). Parts (a - c) depict the cases of Eq. (9) and (d - f) illustrate the cases of Eq. (10). In each case, potential area is highlighted, which depends on the current and previous positions of the candidate and position of the solution being referenced.

least fit member of $\mathcal{P}_r$ is expressed in Eq. (11). This phase is illustrated in Figure 4 (g) and (h), where $\boldsymbol{p}_2^2$ is swapped with $\boldsymbol{p}_3^1$ by assuming $\boldsymbol{p}_3^1$ the least fit member of $\mathcal{P}_3$ and the algorithm of this phase is presented in Algorithm 3.

$$q = \underset{1 \leq j \leq n}{\operatorname{argmax}} f(\boldsymbol{p}_r^j) \tag{11}$$

3.1.4. Election (Fitness evaluation)

The election is mimicked by evaluating the fitness of all the candidates contesting in a constituency and declaring the winner as given by Eq. (12). In Eq. (12) $\boldsymbol{c}_j^*$ denotes the winner of j^{th} constituency (C_j). As mentioned earlier, the party leaders are also updated after the election by using Eq. (6). The reallocation of the roles of constituency winners and party leaders is depicted in Figure 4(i).

$$\begin{aligned} q &= \underset{1 \leq i \leq n}{\operatorname{argmin}} f(\boldsymbol{p}_i^j) \\ \boldsymbol{c}_j^* &= \boldsymbol{p}_q^j \end{aligned} \tag{12}$$

3.1.5. Parliamentary affairs (Exploitation and convergence)

After an inter-party election, the government is formed. The party leaders and the constituency winners/parliamentarians are decided by using Eq. (6) and Eq. (12). The working of this phase is expressed in Algorithm 4. Each parliamentarian $\boldsymbol{c}_j^*$ (the winner of j^{th} constituency) updates its position with reference to a randomly selected parliamentarian $\boldsymbol{c}_r^*$ and if it causes any improvement in fitness of $\boldsymbol{c}_j^*$ then the status and fitness of $\boldsymbol{c}_j^*$ are updated.

I

t is worth mentioning here, each parliamentarian is a party member as well and all the updates are actually applied on position vectors of party members. For example, when $\boldsymbol{c}_j^*$ is updated then corresponding position vector $\boldsymbol{p}_i^j$ is also updated, where i denotes the winning party. This phase is illustrated in Figure 4 (j - k), the winner of the constituency 2 ($\boldsymbol{c}_2^*$) is updated because we are assuming its fitness has been improved and the other two remain unchanged because we are assuming their fitness has not been improved.

4. Comparative analysis of the behavior of PO

In the last 3 decades, the area of nature-inspired optimization algorithms has emerged enormously and hundreds of algorithms have been proposed in literature to solve optimization problems. However, it is also seen that a few well-known algorithms do not have significant difference. For instance, D. Weyland [83] has

Journal Pre-proof

Algorithm 3 PartySwitching($\mathcal{P}$, λ)

foreach $\mathcal{P}_i \in \mathcal{P}$ **do**
  foreach $\boldsymbol{p}_i^j \in \mathcal{P}_i$ **do**
    sp = random number from the interval [0, 1]
    if $sp < \lambda$ **then**
      r = random integer from the range [1, n]
      determine q by using Eq. (11)
      swap ($\boldsymbol{p}_r^q$, $\boldsymbol{p}_i^j$)
    end
  end
end

Algorithm 4 ParliamentaryAffairs(C^*, $\mathcal{P}$)

for $j \leftarrow 1$ *to* n **do**
  $r \leftarrow$ random integer in the range 1 to n, where $r \neq j$
  $a \leftarrow$ random number from the interval [0, 1]
  $\boldsymbol{c}_{new}^* \leftarrow \boldsymbol{c}_r^* + (2a - 1)|\boldsymbol{c}_r^* - \boldsymbol{c}_j^*|$
  compute the fitness of $\boldsymbol{c}_{new}^*$
  if $f(\boldsymbol{c}_{new}^*) \leq f(\boldsymbol{c}_j^*)$ **then**
    $\boldsymbol{c}_j^* \leftarrow \boldsymbol{c}_{new}^*$
    $f(\boldsymbol{c}_j^*) \leftarrow f(\boldsymbol{c}_{new}^*)$
    $i \leftarrow$ party index of the winner of j^{th} constituency
    $\boldsymbol{p}_i^j \leftarrow \boldsymbol{c}_{new}^*$
    $f(\boldsymbol{p}_i^j) \leftarrow f(\boldsymbol{c}_{new}^*)$
  end
end

shown that Harmony Search [84] is a special case of Evolution Strategies [9] and C. Leonardo et al. [85] have shown that Intelligent Water Drop (IWD) [86] is simply a particular instantiation of Ant Colony Optimization [10]. Therefore, a significant difference must be shown between each newly proposed algorithm and similar existing approaches. Considering the resemblance between the position updating mechanism of PO and other state-of-the-art algorithms, such as PSO [11], GWO [12], WOA [4], SCA [35], TLBO [44] etc., it is very important to show how PO is different from them. Although all these algorithms, including PO, use the principle of position updating with respect to the best or comparatively better solution, the position updating mechanism of PO is entirely different from other algorithms, which is shown in this section. Moreover, the time complexity of PO is shown to be equal to other well-known optimization algorithms in this section.

4.1. PO versus comparative algorithms

PO differs from the algorithms enlisted above from the following aspects:

1. Unlike the other optimization algorithms, PO logically divides the population in parties and constituencies to assign a dual role to each search agent. Such a division has the advantage of updating a search agent's position by interacting with an unique pair of better solutions. It is found through experiments that it promotes exploration as well as exploitation.

2. All comparative algorithms use the position of the best solution (global best) to update the position of a search agent; however, PO interacts with party best and constituency best instead of the global best.

3. PSO stores the best position of each search agent (local best) which it utilizes to update their positions; however, PO stores very recent position of each search agent which it utilizes to find the best region to explore in search space.

4. Like TLBO, the position updating mechanism of PO also involves two phases but their working is entirely different. In teaching phase of TLBO, a search agent updates its position by interacting with the global best (teacher) and the mean position of the whole population (class); however, in election campaign phase of PO, the search agent first updates its position with reference to the party leader and then it updates the position with respect to the constituency winner. Similarly, in learner phase of TLBO, the search agent updates its position by interacting with two randomly selected solutions; however, in the phase of parliamentary affairs of PO, each constituency winner further improves its position by interacting with another randomly selected constituency winner. Please note that the constituency winners are best solutions in their constituencies.

5. Another very distinguishing and unique property of PO is its position updating strategy, which involves 6 cases. Based on the position and fitness of the party leader or constituency winner, the current position and fitness of the search agent, and the very recent position and recent fitness of the search agent, the most promising region is identified and searched. We found tremendous improvement in exploitative capability of PO due to this strategy.

6. Because of having multiple better solutions, PO enables the better solutions to further improve by interacting with each other in the phase of parliamentary affairs which other algorithms can not do.

Journal Pre-proof

4.2. Comparative time complexity analysis

The time complexity analysis of most algorithms involves analyses of three components.

1. Time complexity of initialization of the population, generally bounded by $O(MD)$ where M denotes the population size and D denotes the dimensions/design variables of the problem

2. Time complexity of initial fitness evaluation, generally bounded by $O(MC_{0bj})$, where C_{obj} represents the cost of the objective function.

3. Time complexity of the main loop, generally bounded by $O(TMD + TMC_{0bj})$, where T denotes the total number of iterations.

Likewise, the time complexity analysis of PO also requires analyses of these three components:

1. Time complexity of initialization of the population is bounded by $O(MD)$, which is equivalent to other algorithms.

2. Time complexity of initial fitness evaluation is also equivalent to the other algorithms and is bounded by $O(MC_{0bj})$, where $M = n^2$ in PO (n denotes the number of parties, party members, and constituencies). Please note that the computation of party leaders Eq. (8), constituency winners Eq. (12), and least fit member of each party Eq. (11) can be done during the fitness evaluation of the whole population without extra overhead on time complexity of this component.

3. Time complexity of the main loop in Algorithm 1 is $O(TMD + TM + TMC_{0bj} + T\sqrt{M}D)$, where $O(MD)$ is the time complexity of ElectionCampaign() algorithm for whole population, $O(M)$ is the time complexity of PartySwitching() algorithm, $O(MC_{0bj})$ is the time complexity of the election phase in Algorithm 1, $O(\sqrt{M}D)$ is the time complexity of ParliamentaryAffairs() algorithm, and T with each component is for the main loop in Algorithm 1. The time complexity of the main loop can be expressed asymptotically as $O(TMD + TMC_{0bj})$, which is also equivalent to the other algorithms.

We can conclude from the detailed analysis that the time complexity of PO is asymptotically equivalent to the other state-of-the-art algorithms.

5. Experiments

The performance of PO is evaluated on a diverse set of benchmark functions against a good combination of some well-known state of the art algorithms. The proposed algorithm is coded in MATLAB programming software and simulations are run on a Core i7-4650U with 8 GB RAM. The MATLAB code will be released at github and mathworks after acceptance of the paper. In this section, first we present the benchmark functions and state of the art algorithms, and then we discuss the experimental results.

5.1. Benchmark test functions

The benchmark functions are classified based on a few characteristics like modality, dimensionality, separability, continuity and differentiability. In order to evaluate the versatility of PO, a comprehensive suite of benchmark functions having a good mixture of all these characteristics is used. The suite is divided into the following two groups:

1. Unimodal benchmark functions: This group consists of 25 unimodal benchmark functions having a good combination of all aforementioned characteristics. The unimodal functions are preferred for evaluating the exploitation capability of an algorithm because they do not have local optima. The details of these unimodal functions are presented in Table 3 and their mathematical definitions are given in Table A.17 in Appendix A.

2. Multimodal benchmark functions: In this group 25 multimodal benchmark functions are considered. The multimodal functions are used to benchmark the exploration capability of an algorithm because they may have many local optima and to find the global optimum, the algorithm should have the ability to escape from local optima. The details of multimodal functions are presented in Table 4 and their mathematical definitions are given in Table A.18 in Appendix A.

5.2. Algorithms for comparative studies

To compare the performance of PO, a suite of 15 state-of-the-art and recently developed algorithms, including at least 2 algorithms from each category of meta-heuristics discussed in Section 1, is constructed. The names of the algorithms and their parameter settings are presented in Table 5. To achieve the best results the parameters of the algorithms are given the values, which their authors have mentioned in the corresponding papers. Considering the importance of the role

Journal Pre-proof

Table 3: Unimodal benchmark functions for exploitation analysis. *Range* defines the boundary of search space. *Dim* denotes the dimensionality of search space. *Type* defines the characteristics of the functions (Separable (S)/Inseparable (I), Differentiable (D)/Non-differentiable (N), Continuous (C)/Discontinuous (T), and Fixed (F)/Variable (V) dimensional) and F_{min} is the global optimum.

Function	Range	Dim	Type	F_{min}
$F1$– Sphere	$[-100, 100]$	50	[SDCV]	0
$F2$– Quartic Noise	$[-1.28, 1.28]$	50	[SDCV]	0
$F3$– Powell Sum	$[-1, 1]$	50	[SDCV]	0
$F4$– Schwefel's 2.20	$[-100, 100]$	50	[SNCV]	0
$F5$– Schwefel's 2.21	$[-100, 100]$	50	[SNCV]	0
$F6$– Step	$[-100, 100]$	50	[SNTV]	0
$F7$– Stepint	$[-5.12, 5.12]$	50	[SNTV]	$25 - 6n$
$F8$– Schwefel's 1.20	$[-100, 100]$	50	[IDCV]	0
$F9$– Schwefel's 2.22	$[-100, 100]$	50	[IDCV]	0
$F10$– Schwefel's 2.23	$[-10, 10]$	50	[IDCV]	0
$F11$– Rosenbrock	$[-30, 30]$	50	[IDCV]	0
$F12$– Brown	$[-1, 4]$	50	[IDCV]	0
$F13$– Dixon and Price	$[-10, 10]$	50	[IDCV]	0
$F14$– Powell Singular	$[-4, 5]$	50	[IDCV]	0
$F15$– Zakharov	$[-5, 10]$	50	[IDCV]	0
$F16$– Xin-She Yang	$[-20, 20]$	50	[IDCV]	-1
$F17$– Perm 0,D,Beta	$[-d_i, d_i]$	5	[IDCF]	0
$F18$– Three-Hump Camel	$[-5, 5]$	2	[IDCF]	0
$F19$– Beale	$[-4.5, 4.5]$	2	[IDCF]	0
$F20$– Booth	$[-10, 10]$	2	[IDCF]	0
$F21$– Brent	$[-10, 10]$	2	[IDCF]	0
$F22$– Matyas	$[-10, 10]$	2	[IDCF]	0
$F23$– Schaffer N. 4	$[-100, 100]$	2	[IDCF]	0.29257
$F24$– Wayburn Seader 3	$[-500, 500]$	2	[IDCF]	21.35
$F25$– Leon	$[-1.2, 1.2]$	2	[IDCF]	0

Table 4: Multimodal benchmark functions for exploration analysis. *Range* defines the boundary of search space. *Dim* denotes the dimensionality of search space. *Type* defines the characteristics of the functions (Separable (S)/Inseparable (I), Differentiable (D)/Non-differentiable (N), Continuous (C)/Discontinuous (T), and Fixed (F)/Variable (V) dimensional) and F_{min} is the global optimum.

Function	Range	Dim	Type	F_{min}
$F26$– Schwefel's 2.26	$[-500, 500]$	50	[SDCV]	0
$F27$– Rastrigin	$[-5.12, 5.12]$	50	[SDCV]	0
$F28$– Periodic	$[-10, 10]$	50	[SDCV]	0.9
$F29$– Qing	$[-500, 500]$	50	[SDCV]	0
$F30$– Alpine N. 1	$[-10, 10]$	50	[SNCV]	0
$F31$– Xin-She Yang	$[-5, 5]$	50	[SNCV]	0
$F32$– Ackley	$[-32, 32]$	50	[IDCV]	0
$F33$– Trignometric 2	$[-500, 500]$	50	[IDCV]	1
$F34$– Salomon	$[-100, 100]$	50	[IDCV]	0
$F35$– Styblinski-Tang	$[-5, 5]$	50	[IDCV]	$-39.16599 \times n$
$F36$– Griewank	$[-100, 100]$	50	[IDCV]	0
$F37$– Xin-She Yang N.4	$[-10, 10]$	50	[INCV]	-1
$F38$– Xin-She Yang N.2	$[-2\pi, 2\pi]$	50	[INCV]	0
$F39$– Gen. Penalized	$[-50, 50]$	50	[INCV]	0
$F40$– Penalized	$[-50, 50]$	50	[INCV]	0
$F41$– Egg crate	[-5,5]	2	[SDCF]	0
$F42$– Ackley N.3	$[-32, 32]$	2	[IDCF]	-195.629
$F43$– Adjiman	$[-1, 2]$	2	[IDCF]	-2.02181
$F44$– Bird	$[-2\pi, 2\pi]$	2	[IDCF]	-106.7645
$F45$– Camel Six Hump	$[-5, 5]$	2	[IDCF]	-1.0316
$F46$– Branin RCOS	$[-5, 10]$	2	[IDCF]	0.3978873
$F47$– Hartman 3	$[0, 1]$	3	[IDCF]	-3.862782
$F48$– Hartman 6	$[0, 1]$	6	[IDCF]	-3.32237
$F49$– Cross-in-tray	[-10,10]	2	[INCF]	-2.06261218
$F50$– Bartels Conn	$[-500, 500]$	2	[INCF]	1

Table 5: Algorithms used for comparative analysis and their parameter settings. NFEs denotes the number of objective function evaluations.

Algorithm	Parameters setting	NFEs
HHO (2019)	$pSize = 30$	30,000
BOA (2018)	$pSize = 50, c = 0.01, a = 0.1 \rightarrow 0.3, p = 0.8$	30,000
SSA1 (2017)	$pSize = 30$	30,000
WOA (2016)	$p = 30, b = 1, [a, C, l]$ from corresponding eq.	30,000
SCA (2016)	$p = 50, a = 2, [1, r2, r3, r4]$ from corresponding eq.	30,000
SSA2 (2015)	$p = 10, r_a = 1, p_c = 0.7, p_m = 0.1$	30,000
GWO (2014)	$p = 50, [a, C]$ from corresponding eq.	30,000
SLCA (2014)	$nTeams = 5, nFixedPlayer = 11, nSubstitute = 11$	30,000
KH (2012)	$p = 50, v_f = 0.02, D_m = 0.005, N_m = 0.01, S_r = 0$	30,000
TLBO (2011)	$p = 50$	60,000
GSA (2009)	$p = 50, \alpha = 20, G_0 = 100, k = [pSize \rightarrow 1]$	30,000
CS (2009)	$nests = 20, pa = 0.25$	60,000
BBO (2008)	$p = 50, hmp = 1, elit = 2, stSize = 1,$ $mir = 1, mer = 1, mt \geqslant 0.05$	30,000
DE (2004)	$p = 20, CR = 0.5, F = 0.5$	30,000
PSO (1995)	$p = 50, c1 = 2, c2 = 2, w = [0.9 \rightarrow 0.2]$	30,000

of population size in performance of an algorithm, all algorithms are run with population size recommended by their authors. However, for a fair comparison the number of objective function evaluations is set to 30,000 for all algorithms except TLBO [44] and CS [18]. As TLBO and CS perform two evaluations per iteration, they both are set to 60,000 function evaluations per run.

5.3. Parameter settings of PO

PO needs only two main parameters, n (number of parties/constituencies/candidates in each party) and λ (party switching rate). It is found through experiments that lower value of n results in premature convergence, higher value of n improves exploration but at the cost of late convergence, and higher initial value of λ adds extensive shuffling of party members in early iterations which enhances exploration capability of PO in very early iterations. In order to balance the exploration and exploitation, we tune these parameters on several unimodal and multimodal benchmark functions and come up with the conclusion that n should be fixed at 8 and λ should be linearly decreased from 1 to 0 with the course of iterations. To keep NFEs equivalent to the other algorithms PO executes 415 iterations for each benchmark function, which causes 26,560 function evaluations in election phase and 3,320 evaluations in the phase of parliamentary affairs, totaling 29,880 objective function evaluations.

5.4. Results and discussion

A good optimization algorithm should be able to 1) explore the search space, 2) exploit the promising areas, and 3) converge to the best position. In this section, a comparative performance of PO, considering all these three aspects against 15 state of the art algorithms,

Journal Pre-proof

is discussed. Each algorithm including PO is independently run 25 times for each benchmark function and median, mean and standard deviation of the obtained results are reported.

5.4.1. Exploitation capability of PO

To evaluate the capability of PO to exploit the promising regions, 25 unimodal benchmark functions are solved and the results are compared with 15 well-known algorithms in Table 6. The performance of PO for fixed-dimensional functions ($F17 - F25$) is comparable to the other algorithms. PO shares 1^{st} rank with a few other algorithms for almost all fixed-dimensional functions except $F17$ and $F19$; however; the results for these two functions are also comparable to the others. Furthermore, PO outperforms the competitors for all variable-dimensional functions ($F1 - F16$) except $F2$ and $F16$; however, PO comes 2^{nd} for these two functions.

We can conclude our analysis that PO either outperforms the other algorithms or performs equivalently. The true exploitation capability of PO is revealed when high-dimensional functions are solved. The consistent performance of PO for such a comprehensive suite of unimodal benchmarks proves its superior capabilities of exploitation. It should be noted that the good exploitative behavior of PO is due to the position updating strategy (RPPUS) and utilization of two better solutions (party leader and constituency winner) in position updating mechanism of PO. Moreover, the interaction between the constituency winners in parliamentary affairs also improves exploitative behavior.

5.4.2. Exploration capability of PO

To evaluate the capability of PO to explore the search space, 25 multimodal benchmark functions with many local optima are solved. The results are compared with other algorithms in Table 7. Again for the fixed-dimensional functions ($F41 - F50$) the performance of the other algorithms is equivalent to PO and PO shares 1^{st} rank with many others for almost all fixed-dimensional functions. However, the real strength of PO to solve the multimodal functions is revealed for variable-dimensional functions ($F26 - F40$). PO outperforms the other algorithms for all variable-dimensional functions except $F28$, $F29$, and $F37$. Although, the performance for $F28$ and $F37$ is comparable to the others. The consistency in performance proves versatility of PO. The good exploratory behavior of PO is also attributed to its position updating mechanism because each solution explores a different vicinity due to position updating with reference to a unique pair of party leader and constituency winner. Moreover, party switching also promotes exploration by controlling the diversity.

To show that PO has good exploration capability the search history of the search agents is recorded for a few unimodal and multimodal functions. The search history is presented in Figure 6. For the sake of illustration, 2D versions of the benchmark functions are solved by using 4 parties with 4 members each. The scatter of the search agents in space proves that PO well explores the search space in order to find the promising areas, which are then exploited to converge at the global optimum. For further investigation, the trajectories of 8 randomly selected search agents in first two dimensions of these benchmarks are also presented in Figure 7. The zigzag natured paths covering different portions of the search space also proves that PO has good exploration capability. Please note that the polygons (trajectories) are starting from random locations but are ending at the centers (global optimum), which shows that PO is capable to escape local optima.

5.4.3. Convergence of PO

A good optimization algorithm should not converge prematurely to some local optimum. To converge at the global optimum, an algorithm should be able to escape local optima, which requires a good balance between exploration and exploitation. In PO, the balance between the exploration and exploitation is attained through party switching, which uses a parameter λ to control the diversity. It is observed that a higher value of λ allows exploration, so the value of λ is kept higher in the beginning and reduces to 0 with the course of iterations. Moreover, the interaction between the constituency winners in the phase of parliamentary affairs ensures the convergence of PO.

The convergence curves of PO for some unimodal and multimodal benchmark functions are compared with PSO, DE, WOA and GWO in Figure 8. The curves are plotted against the number of function evaluations, which is in hundreds. The figure shows that PO outperforms the other algorithms in all cases. The curves of unimodal functions show that PO has a remarkable property to quickly exploit the promising areas without any disruption. Similarly, the experimental results obtained by PO for multimodal functions (F31-F41 in Figure 8) show the ability of the algorithm to escape local optima very quickly, which is attained by controlling diversity using λ. Another observation is the convergence on global optimum in early iterations. PO is capable of exploring the search space extensively and identifying the most promising region in less iteration because of its position-updating mechanism influenced

Journal Pre-proof

Table 6: Comparison with other state of the art optimization algorithms on unimodal functions. The best results obtained for a function are highlighted.

Fn	Stats	HHO	BOA	SSA1	WOA	SCA	SSA2	GWO	SLC	KH	TLBO	GSA	CS	BBO	DE	PSO	PO
F1	Med	1.9E-196	1.9E-14	9.5E-08	3.7E-159	7.6E+01	8.9E-05	9.9E-30	1.0E-78	1.4E+00	1.1E-92	5.6E-04	1.1E+02	9.6E+01	2.6E-10	4.7E-03	**0.0E+00**
	Mean	7.0E-188	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+02	3.5E-10	9.7E-03	**0.0E+00**
	Std	0.0E+00	9.3E-16	2.6E-08	3.3E-148	1.8E+02	6.1E-05	2.8E-29	5.4E-75	6.4E-01	1.4E-92	1.7E-01	2.6E+01	3.8E+01	3.1E-10	1.3E-02	0.0E+00
F2	Med	**6.1E-05**	9.4E-04	2.7E-01	1.4E-03	1.2E+00	9.6E-02	1.4E-03	5.7E-04	1.2E-01	8.9E-04	7.3E-02	1.6E-01	2.4E-03	4.5E-02	1.9E+01	3.3E-04
	Mean	**7.8E-05**	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	2.2E+01	3.7E-04
	Std	6.9E-05	4.1E-04	1.0E-01	1.6E-03	2.3E+00	9.7E-02	8.3E-04	3.7E-04	5.5E-02	2.9E-04	3.7E-02	5.2E-02	4.4E-03	1.2E-02	1.9E+01	2.3E-04
F3	Med	6.9E-247	5.3E-17	6.8E-07	5.6E-231	1.3E-03	1.9E-29	3.5E-134	2.2E-118	1.1E-07	4.2E-223	1.8E-14	4.9E-12	**0.0E+00**	3.0E-18	3.7E-08	**0.0E+00**
	Mean	5.4E-240	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	**0.0E+00**	4.9E-13	1.3E-07	**0.0E+00**
	Std	0.0E+00	2.6E-16	4.5E-07	0.0E+00	3.7E-03	4.8E-26	5.3E-128	8.1E-112	2.8E-07	0.0E+00	6.1E-13	1.0E-11	0.0E+00	1.9E-12	3.4E-07	0.0E+00
F4	Med	6.6E-100	1.3E-11	3.1E+01	3.0E-106	1.4E+00	4.2E-03	6.8E-17	6.4E-39	2.3E+00	8.1E-46	1.2E+01	6.4E+01	4.2E+01	8.1E-06	2.3E-01	**6.2E-183**
	Mean	4.6E-90	1.3E-11	3.8E+01	3.9E-102	2.0E+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	2.5E-01	**7.7E-179**
	Std	2.3E-89	5.5E-13	2.3E+01	1.5E-101	1.7E+00	1.5E-03	3.2E-17	9.7E-38	7.2E+00	6.2E-46	1.2E+01	6.2E+00	6.6E+00	3.6E-06	1.4E-01	0.0E+00
F5	Med	4.2E-97	1.2E-11	1.7E+01	8.5E+01	6.3E+01	3.8E+01	9.5E-07	7.0E-37	5.1E+00	2.8E-37	7.5E+00	1.5E+01	5.2E+01	1.7E+01	2.4E+00	**1.2E-161**
	Mean	2.1E-89	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	2.3E+00	**2.7E-158**
	Std	1.0E-88	8.4E-13	3.6E+00	2.3E+01	8.8E+00	4.6E+00	1.6E-06	2.2E-35	1.2E+00	1.8E-37	1.2E+00	1.6E+00	8.2E+00	4.4E+00	3.2E-01	1.0E-157
F6	Med	1.2E-05	1.1E+01	7.9E-08	4.4E-01	3.1E+02	9.6E-05	1.8E+00	5.0E-02	1.4E+00	3.8E-06	1.7E-16	9.5E+01	1.0E+02	1.5E-10	4.2E-03	**0.0E+00**
	Mean	4.5E-05	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	9.0E-03	**0.0E+00**
	Std	6.6E-05	8.0E-01	2.7E-08	1.3E-01	3.5E+02	1.0E-04	5.5E-01	5.7E-02	6.8E-01	5.3E-05	2.4E+00	2.3E+01	3.3E+01	1.5E-10	1.1E-02	0.0E+00
F7	Med	**-275.00**	-54.00	-216.00	**-275.00**	-150.00	-48.00	-217.00	**-275.00**	-80.00	-273.00	-180.00	-233.00	-257.00	**-275.00**	-198.00	**-275.00**
	Mean	**-275.00**	-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	-199.76	**-275.00**
	Std	0.00	15.23	17.56	0.00	8.09	12.69	9.36	14.88	9.38	4.32	5.64	5.91	3.38	0.20	20.99	0.00
F8	Med	1.2E-161	1.8E-14	5.3E+03	1.4E+05	3.7E+04	2.4E+04	5.2E-05	1.2E-74	3.4E+03	9.3E-14	1.4E+03	1.2E+04	3.6E+04	8.6E+04	7.5E+02	**6.8E-291**
	Mean	2.6E-140	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	7.9E+02	**1.8E-274**
	Std	1.3E-139	1.2E-15	2.4E+03	2.9E+04	1.3E+04	5.4E+03	5.7E-04	1.6E-71	1.5E+03	2.1E-12	4.4E+02	3.2E+03	7.0E+03	8.3E+03	2.1E+02	0.0E+00
F9	Med	1.4E-99	1.7E+72	3.2E+24	1.4E-105	4.4E-01	1.4E+67	8.3E-17	4.4E-39	6.3E+44	2.5E-45	2.7E+02	1.0E+10	3.6E+01	2.2E-05	2.3E+00	**2.5E-184**
	Mean	1.7E-95	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	2.6E+00	**1.5E-179**
	Std	5.2E-95	7.0E+73	2.5E+37	6.1E-101	1.5E+00	9.0E+68	6.7E-17	1.7E-38	1.8E+60	1.6E-45	3.5E+01	0.0E+00	6.6E+00	8.2E-05	2.2E+00	0.0E+00
F10	Med	**0.0E+00**	1.8E-21	2.7E-10	**0.0E+00**	1.9E+07	1.3E-12	3.1E-98	**0.0E+00**	2.5E-08	**0.0E+00**	6.0E-05	5.7E+02	**0.0E+00**	4.7E-05	1.3E-01	**0.0E+00**
	Mean	**0.0E+00**	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	7.8E-01	**0.0E+00**
	Std	0.0E+00	2.0E-22	7.1E-07	0.0E+00	3.5E+08	1.1E-11	1.5E-92	0.0E+00	8.5E-08	0.0E+00	1.0E-02	1.5E+03	3.3E-01	3.8E+04	1.5E+00	0.0E+00
F11	Med	1.8E-03	4.9E+01	1.0E+02	4.8E+01	1.6E+06	1.8E+02	4.7E+01	4.6E+01	2.1E+02	4.3E+01	1.1E+02	4.3E+03	4.0E+03	4.9E+01	2.1E+02	**0.0E+00**
	Mean	4.3E-03	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	3.1E+02	**0.0E+00**
	Std	4.6E-03	2.5E-02	1.8E+02	5.3E-01	3.1E+06	6.8E+01	7.3E-01	5.5E-01	9.6E+01	6.3E-01	7.8E+01	1.6E+03	2.1E+03	4.5E+01	3.1E+02	0.0E+00
F12	Med	8.2E-200	1.5E-14	2.3E-10	6.6E-161	2.4E-02	4.4E-06	3.2E-32	1.0E-82	1.9E+02	1.8E-95	4.7E+00	1.4E+00	**0.0E+00**	5.1E-13	1.0E+02	**0.0E+00**
	Mean	1.4E-190	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	**0.0E+00**	5.9E-13	1.8E+02	**0.0E+00**
	Std	0.0E+00	1.2E-15	6.4E-11	4.7E-146	1.5E-01	1.1E-05	5.0E-32	6.2E-78	1.1E+02	3.3E-95	2.4E+00	9.8E-01	0.0E+00	4.3E-13	1.7E+02	0.0E+00
F13	Med	0.25	0.99	4.71	0.67	4914.52	8.61	0.67	0.67	9.06	0.67	3.71	67.64	271.50	0.85	19.33	**0.04**
	Mean	0.25	0.99	8.04	0.67	16298.56	9.44	0.67	0.67	9.92	0.67	4.21	67.82	237.28	2.34	7617.05	**0.05**
	Std	0.00	0.00	9.07	0.00	26606.53	4.13	0.00	0.00	6.04	0.00	2.86	22.28	167.45	2.54	26953.37	0.03
F14	Med	5.4E-198	1.7E-14	4.5E+00	1.2E-34	1.7E+02	7.7E-01	1.4E-05	3.7E-80	2.9E+00	6.1E-09	2.0E+00	8.4E+00	4.7E+01	2.1E-01	3.3E+03	**0.0E+00**
	Mean	4.0E-183	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	3.4E+03	**0.0E+00**
	Std	0.0E+00	8.7E-16	4.1E+00	3.3E-13	2.1E+02	1.2E+00	1.8E-05	8.0E-77	9.2E-01	2.5E-07	2.8E+00	2.7E+00	1.3E+01	9.1E-01	1.8E+03	0.0E+00
F15	Med	1.7E-92	1.7E-14	1.3E+02	8.5E+02	9.8E+01	5.6E+02	4.9E-07	9.2E-58	3.6E+02	2.1E-03	1.2E+02	6.0E+02	2.8E+02	5.1E+02	6.9E+02	**3.3E-212**
	Mean	1.7E-67	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	6.9E+02	**9.9E-188**
	Std	6.5E-67	1.2E-15	4.7E+01	2.6E+02	3.7E+01	2.6E+02	4.3E-06	8.7E-54	1.2E+02	8.2E-03	2.1E+01	7.4E+01	5.7E+01	6.6E+01	1.4E+02	0.0E+00
F16	Med	**-1.0E+00**	2.6E-58	2.1E-293	0.0E+00	1.9E-305	0.0E+00	4.2E-207	0.0E+00	8.4E-75	0.0E+00	1.9E-65	0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00
	Mean	**-1.0E+00**	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	0.0E+00	0.0E+00	0.0E+00	-4.8E-01
	Std	0.0E+00	4.5E-47	0.0E+00	4.8E-01	0.0E+00	0.0E+00	0.0E+00	0.0E+00	2.9E-60	0.0E+00	3.4E-53	0.0E+00	0.0E+00	0.0E+00	0.0E+00	5.1E-01
F17	Med	36.63	730.76	11.86	222.23	41.05	29.30	25.85	0.42	14.43	0.47	411.67	2.02	38.90	1.15	**0.12**	0.32
	Mean	65.67	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	505.14	16.76
	Std	87.34	800.03	152.13	585.33	29.43	269.49	1473.63	33.11	262.51	0.74	543.73	1.60	200.14	2.02	1746.47	40.38
F18	Med	1.5E-213	6.4E-18	6.6E-16	5.4E-163	2.3E-97	1.8E-99	**0.0E+00**	1.1E-103	1.2E-11	2.2E-217	6.0E-21	1.0E-26	**0.0E+00**	5.7E-205	8.1E-80	**0.0E+00**
	Mean	3.8E-203	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	**0.0E+00**	2.7E-197	2.7E-78	**0.0E+00**
	Std	0.0E+00	1.2E-17	2.4E-15	8.2E-144	7.6E-91	6.0E-02	0.0E+00	9.0E-100	1.5E-11	0.0E+00	9.4E-21	2.7E-25	0.0E+00	0.0E+00	6.3E-78	0.0E+00
F19	Med	1.6E-14	8.9E-02	1.3E-15	6.0E-12	4.6E-05	3.2E-29	4.6E-08	**0.0E+00**	9.4E-11	**0.0E+00**	1.4E-20	7.9E-22	6.5E-02	**0.0E+00**	**0.0E+00**	2.8E-32
	Mean	2.5E-12	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	**0.0E+00**	**0.0E+00**	1.7E-23
	Std	6.2E-12	2.3E-01	3.0E-15	1.4E-10	1.3E-04	1.8E-01	1.5E-01	0.0E+00	9.6E-11	0.0E+00	1.7E-20	4.8E-20	2.2E-01	0.0E+00	0.0E+00	8.6E-23
F20	Med	5.4E-06	2.3E-03	1.1E-14	1.1E-04	4.2E-04	**0.0E+00**	1.8E-07	**0.0E+00**	6.5E-11	**0.0E+00**	1.5E-20	2.4E-27	**0.0E+00**	**0.0E+00**	**0.0E+00**	**0.0E+00**
	Mean	8.5E-06	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	**0.0E+00**	**0.0E+00**	**0.0E+00**	**0.0E+00**
	Std	1.0E-05	2.1E-01	1.9E-14	1.5E-04	5.9E-04	7.2E-17	1.6E-07	0.0E+00	1.2E-10	0.0E+00	4.8E-20	1.3E-25	0.0E+00	0.0E+00	0.0E+00	0.0E+00
F21	Med	**1.4E-87**	8.3E+00	**1.4E-87**	**1.4E-87**	**1.4E-87**	**1.4E-87**	**1.4E-87**	**1.4E-87**	**1.4E-87**	**1.4E-87**	1.9E-05	**1.4E-87**	**1.4E-87**	**1.4E-87**	**1.4E-87**	**1.4E-87**
	Mean	**1.4E-87**	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**	**1.4E-87**
	Std	4.6E-103	1.8E+01	4.6E-103	4.6E-103	4.6E-103	4.6E-103	4.6E-103	4.6E-103	4.6E-103	4.6E-103	2.7E-04	4.6E-103	4.6E-103	4.6E-103	4.6E-103	4.6E-103
F22	Med	3.8E-277	1.4E-15	1.4E-16	**0.0E+00**	1.4E-83	1.5E-30	1.7E-177	1.3E-106	4.1E-12	5.2E-157	1.0E-21	5.5E-29	**0.0E+00**	7.7E-122	1.1E-59	**0.0E+00**
	Mean	3.5E-245	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	**0.0E+00**	2.8E-117	3.0E-58	**0.0E+00**
	Std	0.0E+00	5.1E-16	1.7E-15	0.0E+00	1.6E-77	1.0E-07	3.4E-161	1.2E-97	8.0E-12	2.1E-154	9.6E-22	4.6E-28	0.0E+00	8.5E-117	1.2E-57	0.0E+00
F23	Med	**0.29**	**0.29**	**0.29**	**0.29**	**0.29**	**0.29**	**0.29**	**0.29**	**0.29**	**0.29**	0.30	**0.29**	0.31	**0.29**	**0.29**	**0.29**
	Mean	**0.29**	**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**	**0.29**
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
F24	Med	**19.11**	19.15	**19.11**	**19.11**	19.13	**19.11**	**19.11**	**19.11**	**19.11**	**19.11**	**19.11**	19.11	19.33	**19.11**	**19.11**	**19.11**
	Mean	**19.11**	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**	**19.11**
	Std	0.00	0.92	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.45	0.00	0.00	0.00
F25	Med	3.0E-07	3.4E-03	1.2E-15	6.1E-06	1.8E-04	4.8E-02	3.0E-07	**0.0E+00**	5.0E-03	1.2E-22	3.4E-03	1.5E-24	**0.0E+00**	**0.0E+00**	1.6E-16	**0.0E+00**
	Mean	5.4E-06	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	**0.0E+00**	**0.0E+00**	2.4E-15	**0.0E+00**
	Std	1.1E-05	2.5E-02	2.3E-15	1.9E-05	1.8E-04	5.3E-02	5.8E-07	4.2E-32	9.5E-03	8.7E-22	6.0E-03	2.9E-22	0.0E+00	0.0E+00	4.8E-15	0.0E+00

Table 7: Comparison with other state of the art optimization algorithms on multimodal functions. The best results obtained for a function are highlighted.

Fn	Stats	HHO	BOA	SSA1	WOA	SCA	SSA2	GWO	SLC	KH	TLBO	GSA	CS	BBO	DE	PSO	PO
F26	Med	1.3E-03	3.1E+02	1.7E+02	1.2E+02	3.2E+02	3.1E+02	2.3E+02	1.9E+02	2.2E+02	1.8E+02	3.5E+02	1.7E+02	4.0E+00	2.0E+02	2.0E+02	**1.3E-05**
	Mean	5.5E-03	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	4.2E+00	2.0E+02	2.2E+02	**1.3E-05**
	Std	8.7E-03	9.7E+00	1.8E+01	5.4E+01	7.4E+00	1.7E+02	1.7E+01	4.0E+01	2.4E+01	3.8E+01	1.1E+01	5.5E+00	1.2E+00	2.1E+01	4.4E+01	0.0E+00
F27	Med	**0.0E+00**	**0.0E+00**	9.7E+01	**0.0E+00**	6.4E+01	6.4E+01	2.8E-13	**0.0E+00**	2.3E+01	9.0E+00	3.3E+01	2.1E+02	**0.0E+00**	3.0E+02	2.5E+02	**0.0E+00**
	Mean	**0.0E+00**	2.9E+01	9.6E+01	**0.0E+00**	7.8E+01	6.5E+01	2.0E+00	**0.0E+00**	2.5E+01	9.8E+00	3.6E+01	2.1E+02	**0.0E+00**	3.0E+02	2.5E+02	**0.0E+00**
	Std	0.0E+00	1.0E+02	1.7E+01	0.0E+00	4.8E+01	9.1E+00	3.1E+00	0.0E+00	1.2E+01	1.0E+01	8.5E+00	1.7E+01	0.0E+00	1.6E+01	4.6E+01	0.0E+00
F28	Med	**9.0E-01**	1.5E+01	1.0E+00	**9.0E-01**	1.2E+01	3.4E+00	1.5E+00	1.5E+00	3.9E+00	5.1E+00	1.0E+00	2.5E+00	9.0E-01	9.5E+00	5.7E+00	**9.0E-01**
	Mean	**9.0E-01**	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	5.7E+00	9.3E-01
	Std	3.4E-16	9.1E-01	2.7E-10	2.4E-01	2.2E+00	4.1E-01	3.4E-01	6.0E-01	4.6E+00	3.7E+00	5.4E-16	1.2E-01	2.0E-02	5.2E-01	1.9E+00	4.6E-02
F29	Med	2.9E+03	1.4E+04	4.2E+01	5.2E+03	9.4E+08	3.1E+01	8.2E+03	2.4E+02	2.1E+03	**6.1E-03**	2.5E+08	1.0E+10	7.5E+05	1.2E+03	5.3E-01	2.7E+01
	Mean	3.1E+03	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	**8.1E-01**	7.7E+01
	Std	5.3E+02	1.1E+03	9.0E+01	2.3E+03	2.4E+09	5.8E+01	2.3E+03	2.3E+02	9.3E+02	1.4E+01	1.2E+08	0.0E+00	5.0E+05	1.2E+03	1.2E+00	1.5E+02
F30	Med	3.3E-102	1.4E-14	7.6E+00	5.3E-108	3.2E+00	3.1E-01	9.2E-05	1.0E-40	5.9E-02	2.1E-47	6.6E-03	2.5E+01	**0.0E+00**	3.3E-02	1.1E-01	**0.0E+00**
	Mean	1.7E-91	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	1.3E+00	**3.4E-182**
	Std	8.5E-91	1.8E-14	2.4E+00	9.8E-103	5.4E+00	2.3E-01	9.4E-04	7.4E-40	4.4E-01	1.3E-47	1.0E-02	2.2E+00	5.4E-02	8.6E-03	2.3E+00	0.0E+00
F31	Med	5.9E-64	4.3E+24	9.2E+01	3.9E-10	2.0E+03	1.2E+21	1.9E-44	3.9E-44	9.3E+09	1.6E-22	4.3E-02	1.0E+10	1.1E+09	4.4E-02	8.1E+12	**1.4E-199**
	Mean	6.2E-27	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	5.2E+15	**7.2E-196**
	Std	3.1E-26	1.1E+26	5.9E+06	5.2E+04	2.1E+08	1.9E+22	3.3E-22	2.8E-24	1.2E+14	7.8E-15	1.8E-01	0.0E+00	6.9E+15	8.3E+02	1.7E+16	0.0E+00
F32	Med	**-8.9E-16**	1.2E-11	3.1E+00	2.7E-15	2.0E+01	1.8E-03	8.4E-14	**-8.9E-16**	2.8E+00	2.7E-15	6.0E-09	1.0E+01	1.9E+00	3.3E-06	7.8E-01	**-8.9E-16**
	Mean	**-8.9E-16**	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	1.9E+00	3.8E-06	7.9E-01	**-8.9E-16**
	Std	0.0E+00	4.7E-13	9.8E-01	2.1E-15	5.2E+00	4.2E-01	1.3E-14	7.1E-16	7.1E-01	1.8E-15	1.0E-01	1.9E+00	2.8E-01	1.7E-06	6.7E-01	0.0E+00
F33	Med	1.0E+00	1.7E+02	4.1E+02	1.2E+02	6.4E+03	2.1E+01	5.7E+01	1.0E+02	1.3E+02	3.0E+01	7.0E+04	3.1E+03	2.7E+03	5.7E+00	4.9E+01	**1.0E+00**
	Mean	1.0E+00	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	5.1E+01	**1.0E+00**
	Std	7.3E-04	9.9E+00	7.8E+01	3.4E+01	1.2E+04	1.8E+01	1.0E+01	2.6E+01	4.1E+01	5.6E+00	1.0E+04	8.4E+02	7.4E+02	1.7E+01	1.4E+01	0.0E+00
F34	Med	1.4E-97	3.0E-01	2.5E+00	1.0E-01	2.2E+00	1.4E+00	2.0E-01	1.0E-01	2.7E+00	2.0E-01	4.4E+00	4.6E+00	3.6E+00	5.0E-01	7.0E-01	**0.0E+00**
	Mean	6.8E-94	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	3.6E+00	5.2E-01	6.6E-01	**0.0E+00**
	Std	3.3E-93	1.9E-02	4.7E-01	5.7E-02	1.4E+00	4.0E-01	2.8E-02	1.3E-07	4.5E-01	4.9E-02	5.2E-01	4.2E-01	4.2E-01	4.6E-02	5.8E-02	0.0E+00
F35	Med	**-2.0E+03**	NA	-1.6E+03	**-2.0E+03**	-8.9E+02	-8.5E+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	-1.7E+03	**-2.0E+03**
	Mean	**-2.0E+03**	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	-1.7E+03	**-2.0E+03**
	Std	5.1E-03	3.3E+03	4.4E+01	1.1E+02	7.4E+01	1.7E+02	8.7E+01	4.5E+01	5.6E+01	5.7E+01	2.9E+01	2.2E+01	2.7E-01	2.1E+01	5.6E+01	4.6E-13
F36	Med	**0.0E+00**	6.8E-15	7.6E-03	**0.0E+00**	9.9E-01	1.9E-04	**0.0E+00**	**0.0E+00**	1.7E-01	**0.0E+00**	3.0E-02	9.8E-01	1.0E+00	8.0E-12	6.1E-04	**0.0E+00**
	Mean	**0.0E+00**	4.9E-15	7.6E-03	8.8E-03	9.4E-01	1.3E-03	1.6E-03	**0.0E+00**	1.8E-01	**0.0E+00**	5.3E-02	9.8E-01	1.0E+00	5.9E-04	5.2E-03	**0.0E+00**
	Std	0.0E+00	3.5E-15	8.7E-03	3.0E-02	2.3E-01	3.8E-03	4.7E-03	0.0E+00	8.0E-02	0.0E+00	7.3E-02	4.4E-02	4.0E-02	2.0E-03	5.6E-03	0.0E+00
F37	Med	**-1.0E+00**	1.8E-18	1.4E-30	3.9E-21	4.7E-16	1.7E-21	1.1E-23	1.8E-21	6.0E-21	2.0E-23	1.0E-21	4.1E-21	6.9E-22	1.7E-20	2.1E-22	**-1.0E+00**
	Mean	**-1.0E+00**	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	7.2E-22	-6.8E-01
	Std	0.0E+00	1.5E-18	1.9E-30	4.4E-01	8.0E-16	2.0E-22	3.2E-23	4.5E-22	3.1E-19	4.7E-23	4.3E-22	4.5E-22	4.6E-24	3.2E-20	1.1E-21	4.8E-01
F38	Med	**1.2E-20**	1.4E-13	3.2E-19	**1.2E-20**	2.3E-16	4.5E-18	4.3E-15	1.3E-19	2.3E-09	1.7E-19	1.3E-20	1.1E-19	1.0E-19	1.3E-19	1.6E-19	**1.2E-20**
	Mean	**1.2E-20**	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	1.6E-19	**1.2E-20**
	Std	1.0E-23	2.7E-12	8.7E-19	4.2E-21	1.4E-15	8.5E-18	7.2E-12	1.5E-20	1.3E-07	4.2E-21	2.8E-20	3.0E-20	1.1E-20	2.6E-20	5.3E-21	3.4E-35
F39	Med	1.6E-05	5.0E+00	6.0E+01	8.2E-01	8.8E+06	1.6E-04	1.5E+00	1.5E+00	1.9E+01	5.5E-02	2.0E+01	3.0E+01	3.6E+00	1.7E-07	1.4E-02	**1.3E-32**
	Mean	2.9E-05	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	3.7E+00	1.7E-01	1.8E-02	**1.3E-32**
	Std	3.3E-05	8.2E-02	1.8E+01	3.7E-01	2.3E+07	2.2E-01	4.0E-01	1.4E+00	1.3E+01	9.7E-02	5.8E+00	6.1E+00	9.1E-01	7.3E-01	1.6E-02	5.6E-48
F40	Med	8.4E-07	8.3E-01	9.3E+00	8.9E-03	6.8E+05	5.7E-05	5.8E-02	1.3E-03	2.0E+00	2.9E-08	1.5E+00	5.0E+00	5.1E-01	7.8E-09	7.1E-05	**9.4E-33**
	Mean	1.8E-06	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	7.6E-03	**9.4E-33**
	Std	2.4E-06	1.7E-01	3.7E+00	6.4E-03	7.1E+06	6.9E-02	2.0E-02	1.4E-03	8.1E-01	2.1E-07	5.4E-01	7.6E-01	1.7E-01	5.3E-03	2.1E-02	1.4E-48
F41	Med	2.0E-226	3.7E-22	9.2E-15	4.2E-239	1.8E-96	2.0E-118	**0.0E+00**	1.1E-101	5.1E-11	4.3E-218	7.3E-20	1.3E-24	**0.0E+00**	3.3E-234	1.0E-80	**0.0E+00**
	Mean	3.4E-215	9.6E-22	2.8E-14	3.7E-218	1.2E-89	1.2E-117	**0.0E+00**	1.2E-97	1.1E-10	6.6E-217	1.4E-19	1.1E-23	**0.0E+00**	6.1E-228	1.2E-78	**0.0E+00**
	Std	0.0E+00	1.6E-21	5.7E-14	0.0E+00	6.0E-89	2.2E-117	0.0E+00	4.1E-97	1.7E-10	0.0E+00	1.4E-19	2.2E-23	0.0E+00	0.0E+00	4.5E-78	0.0E+00
F42	Med	**-195.63**	NA	**-195.63**	**-195.63**	**-195.63**	-194.36	**-195.63**	**-195.63**	**-195.63**	**-195.63**	**-195.63**	**-195.63**	-194.42	**-195.63**	**-195.63**	**-195.63**
	Mean	**-195.63**	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**	**-195.63**
	Std	0.00	NA	0.00	0.00	0.00	5.45	0.00	0.00	0.00	0.00	0.00	0.00	0.47	0.00	0.00	0.00
F43	Med	**-2.02**	NA	**-2.02**	**-2.02**	**-2.02**	-2.00	**-2.02**	**-2.02**	**-2.02**	**-2.02**	-2.02	**-2.02**	-2.01	**-2.02**	**-2.02**	**-2.02**
	Mean	**-2.02**	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**	**-2.02**
	Std	0.00	NA	0.00	0.00	0.00	0.52	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
F44	Med	**-106.76**	NA	**-106.76**	**-106.76**	-106.73	-90.18	**-106.76**	**-106.76**	**-106.76**	**-106.76**	**-106.76**	**-106.76**	-101.68	**-106.76**	**-106.76**	**-106.76**
	Mean	**-106.76**	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**	**-106.76**
	Std	0.00	NA	0.00	0.00	0.05	21.42	3.89	0.00	3.89	0.00	0.61	0.00	4.95	0.00	0.00	0.00
F45	Med	**-1.03**	NA	**-1.03**	**-1.03**	-1.03	-1.03	**-1.03**	**-1.03**	**-1.03**	**-1.03**	**-1.03**	**-1.03**	-0.96	**-1.03**	**-1.03**	**-1.03**
	Mean	**-1.03**	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**	**-1.03**
	Std	0.00	NA	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.15	0.00	0.00	0.00
F46	Med	**0.40**	0.40	**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**	**0.40**
	Mean	0.40	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**	**0.40**
	Std	0.00	1.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00
F47	Med	-3.86	NA	**-3.86**	-3.86	-3.85	-3.06	-3.86	**-3.86**	**-3.86**	**-3.86**	**-3.86**	**-3.86**	-3.84	**-3.86**	**-3.86**	**-3.86**
	Mean	-3.86	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**	**-3.86**
	Std	0.00	NA	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00
F48	Med	-3.19	NA	-3.20	**-3.32**	-3.01	-1.70	**-3.32**	**-3.32**	**-3.32**	**-3.32**	**-3.32**	**-3.32**	-3.10	**-3.32**	**-3.32**	**-3.32**
	Mean	-3.17	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	-3.25	-3.30
	Std	0.08	NA	0.05	0.07	0.28	0.60	0.07	0.06	0.06	0.00	0.00	0.00	0.18	0.06	0.08	0.04
F49	Med	**-2.06**	NA	**-2.06**	**-2.06**	**-2.06**	**-2.06**	**-2.06**	**-2.06**	**-2.06**	**-2.06**	**-2.06**	**-2.06**	**-2.06**	**-2.06**	**-2.06**	**-2.06**
	Mean	**-2.06**	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**	**-2.06**
	Std	0.00	NA	0.00	0.00	0.00	3.12	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
F50	Med	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	Mean	**1.00**	**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**	**1.00**
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.46	0.00	0.00	0.00

Journal Pre-proof

Journal Pre-proof

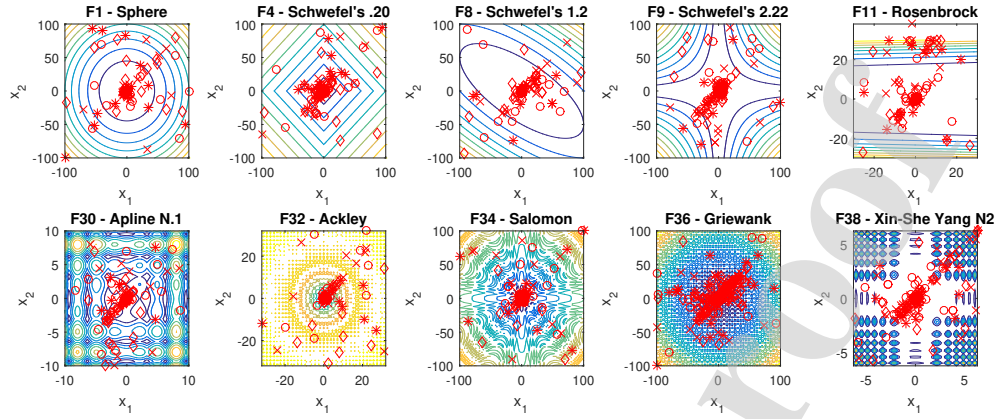


Figure 6: Demonstration of the search history of the search agents. For illustration, 2D version of the benchmark functions is considered and solved by smaller population consisting of 4 parties with 4 members each. Different symbols are used for each party members.

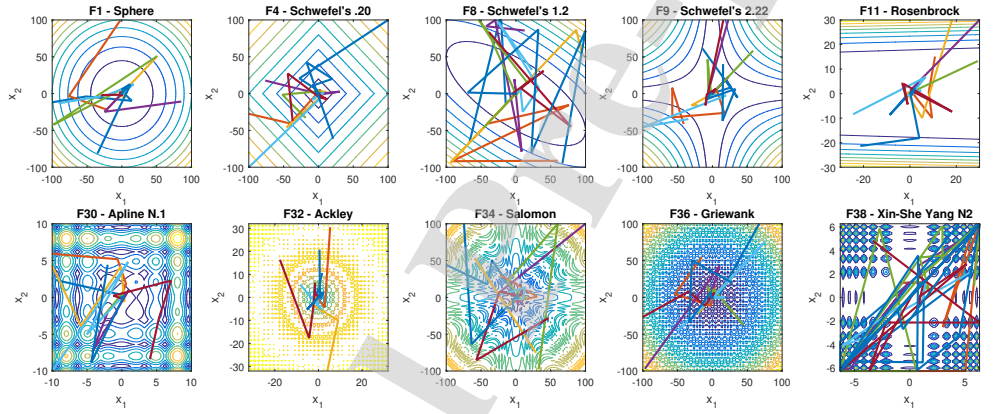


Figure 7: Trajectories of randomly selected 8 search agents in first two dimensions of the respective function. Different color represents the trajectory of different search agent.

by three solutions, higher shuffling of solutions in early iterations, and sufficient size of population. Moreover, PO has capability to converge quickly by exploiting the most promising region discovered at any time because of interaction between the constituency winners in parliamentary affairs, utilization of recent position in position-updating mechanism, and linear reduction in party switching rate with the course of iterations. The convergence at the right location is also evident by the search history of the search agents and trajectories of the randomly selected search agents presented in Figure 6 and 7, respectively. The thicker portion of the contour plots of the functions in the search histories shows the convergence of the search agents in that portion. Since the global optimum is situated in that portion, we can conclude that PO is not converging prematurely. Similarly, the termination of the trajectories in the vicinity of the global optimum also witnesses the convergence at the global optimum.

5.5. *Statistical significance analysis*

In order to evaluate the statistical significance of the comparative results of two or more algorithms, many non-parametric statistical tests have been proposed in the literature [87]. One of the frequently used tests is Wilcoxon rank-sum test, which we use in this paper to statistically evaluate the performance of PO. The results of pair-wise comparison of PO and all other algorithms are presented in Table 8 at 0.05 significance level. At the bottom of the Table 8, "+" indicates the number of functions PO is significantly superior to an algorithm, "≈" denotes the number of functions PO is statistically equivalent to an algorithm, and "−" represents the number of functions PO is statistically inferior to an algo-

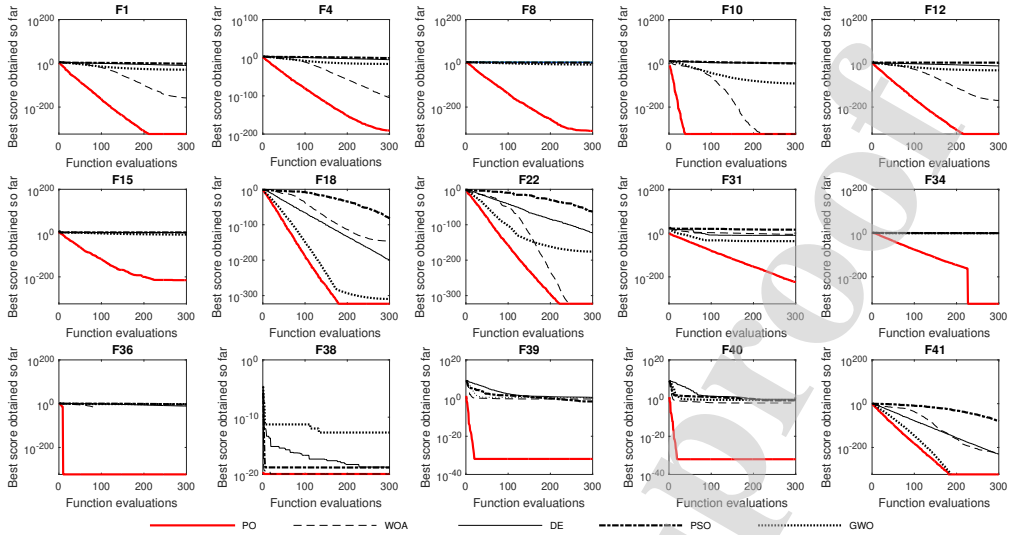


Figure 8: Comparison of convergence curves of PO and few well-known algorithms.

rithm. The larger numbers in "+" row and fewer numbers in "–" row proves that PO demonstrates statistically significant and comparatively better performance over the other algorithms.

5.6. Comparison with other politics based algorithms

The performance of PO is also compared with POA [57], GPO [56] and ECO [58]. In their original papers, POA and GPO were tested on 6 benchmark functions, while ECO was tested on 4 of these functions in [88]. The performance of PO for the same benchmark functions is compared with these algorithms and the results are presented in Table 9. The results show that PO outperforms all other politics-based algorithms.

5.7. Impact of high-dimensionality

One very important characteristic of PO is to solve very high dimensional problems efficiently. To show this capability the performance of PO is evaluated by increasing the number of dimensions of all the scalable benchmark functions to 1000. The results are compared with the results obtained against 50-dimensional versions in Table 10. The results show that the performance remains consistent for all functions except F29. The experimental results illustrate the scalability of the proposed algorithm in terms of the number of variables of the optimization problem.

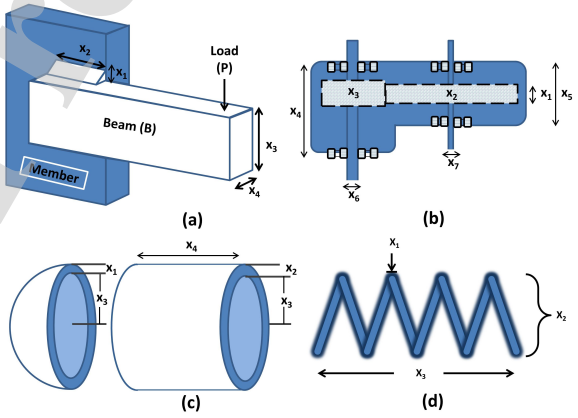


Figure 9: Schematic views of engineering problems. (a) Welded beam design. (b) Speed reducer design. (d) Pressure-vessel design. (d) Tension-compression spring design.

5.8. Invariance to function shifting

The performance of meta-heuristics is supposed to be dependent on the shape and form of the landscape of an optimization function but it should not be dependent on the location of the landscape in the search space. In other words, the performance should remain consistent even if the function is shifted in the search space without changing its shape. PO is executed on shifted versions of 16 benchmark functions from the suite of 50 and the results are compared with the results obtained for original (non-shifted) versions. The mathematical formulation of the shifted functions is given in Table A.19 in

Journal Pre-proof

Table 8: p-values of Wilcoxon rank-sum test at 0.05 significance level for PO against other 15 algorithms for each benchmark function with 25 independent runs. The results having p-values ≥ 0.05 are displayed in bold face and NaN means "Not a Number" returned by the significant test. The symbols "+", "−", and "≈" respectively are the number of PO is significantly better, significantly worse, and statistically similar to that of other compared algorithms.

Fn	HHO	BOA	SSA1	WOA	SCA	SSA2	GWO	SLC	KH	TLBO	GSA	CS	BBO	DE	PSO
F1	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11
F2	1.2E-07	8.3E-07	1.4E-09	1.8E-05	1.4E-09	1.4E-09	1.6E-08	**6.3E-02**	1.4E-09	2.9E-08	1.4E-09	1.4E-09	9.5E-08	1.4E-09	1.4E-09
F3	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	NaN	9.7E-11	9.7E-11
F4	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09
F5	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09
F6	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11
F7	**NaN**	9.7E-11	9.6E-11	**NaN**	9.6E-11	9.5E-11	9.6E-11	5.4E-04	9.6E-11	1.4E-08	9.5E-11	9.6E-11	9.3E-11	**3.4E-01**	8.9E-11
F8	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09
F9	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	9.7E-11	1.4E-09	1.4E-09	1.4E-09
F10	**NaN**	9.7E-11	9.7E-11	**1.6E-01**	9.7E-11	9.7E-11	9.7E-11	**8.1E-02**	9.7E-11	**NaN**	9.7E-11	9.7E-11	2.5E-03	9.7E-11	9.7E-11
F11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11
F12	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	NaN	9.7E-11	9.7E-11
F13	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09
F14	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11
F15	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09
F16	3.5E-05	7.7E-10	7.7E-10	**1.3E-01**	2.2E-07	8.8E-05	7.7E-10	8.8E-05	7.7E-10	8.8E-05	7.7E-10	8.8E-05	8.8E-05	8.8E-05	8.8E-05
F17	8.9E-06	3.3E-09	1.4E-03	3.6E-08	2.0E-05	1.7E-05	7.4E-06	**9.8E-01**	3.3E-04	**6.6E-01**	1.0E-08	3.0E-03	**1.4E-01**	**2.6E-01**	**1.4E-01**
F18	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	4.6E-08	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	**NaN**	9.7E-11	9.7E-11
F19	1.2E-08	1.1E-09	1.1E-09	1.1E-09	1.1E-09	9.4E-03	1.1E-09	1.9E-05	1.1E-09	1.9E-05	1.5E-09	4.3E-09	1.1E-09	1.9E-05	1.9E-05
F20	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	**8.1E-02**	9.7E-11	**NaN**	9.7E-11	**NaN**	9.7E-11	9.7E-11	**NaN**	**NaN**	**NaN**
F21	**NaN**	9.7E-11	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	9.7E-11	**NaN**	**NaN**	**NaN**	**NaN**
F22	9.7E-11	9.7E-11	9.7E-11	**3.4E-01**	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	**NaN**	9.7E-11	9.7E-11
F23	4.6E-09	4.1E-09	4.6E-09	4.6E-09	4.6E-09	3.7E-09	4.6E-09	6.1E-07	4.6E-09	**6.5E-01**	1.9E-10	4.6E-09	4.9E-11	1.8E-03	**1.2E-01**
F24	9.7E-10	9.7E-10	9.7E-10	9.7E-10	9.7E-10	2.7E-03	9.7E-10	**6.4E-02**	9.7E-10	3.8E-07	**2.2E-01**	2.8E-02	2.5E-10	1.5E-08	5.1E-03
F25	3.7E-10	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	4.1E-02	9.7E-11	9.7E-11	9.7E-11	9.7E-11	**NaN**	**NaN**	9.7E-11
F26	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11
F27	**NaN**	1.0E-02	9.7E-11	**NaN**	9.7E-11	9.7E-11	4.5E-08	**NaN**	9.7E-11	7.6E-06	9.7E-11	9.7E-11	**NaN**	9.7E-11	9.7E-11
F28	5.0E-03	5.4E-10	5.4E-10	**2.4E-01**	5.4E-10	5.4E-10	5.4E-10	5.4E-10	5.4E-10	5.4E-10	3.4E-10	5.4E-10	3.0E-02	5.4E-10	5.4E-10
F29	1.4E-09	1.4E-09	**4.8E-01**	1.4E-09	1.4E-09	**4.4E-01**	1.4E-09	9.2E-07	1.8E-09	1.8E-08	1.4E-09	9.7E-11	1.4E-09	3.7E-07	1.4E-09
F30	9.3E-10	9.3E-10	9.3E-10	9.3E-10	9.3E-10	9.3E-10	9.3E-10	9.3E-10	9.3E-10	9.3E-10	9.3E-10	9.3E-10	**1.2E-01**	9.3E-10	9.3E-10
F31	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	9.7E-11	1.4E-09	1.4E-09	1.4E-09
F32	**NaN**	9.7E-11	9.7E-11	2.1E-07	9.7E-11	9.7E-11	8.5E-11	**3.4E-01**	9.7E-11	4.3E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11
F33	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11
F34	9.7E-11	9.7E-11	9.7E-11	9.0E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11
F35	9.7E-11	2.6E-09	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.4E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11
F36	**NaN**	9.7E-11	9.7E-11	**1.6E-01**	9.7E-11	9.7E-11	**8.1E-02**	**NaN**	9.7E-11	**NaN**	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11
F37	2.4E-03	6.1E-10	6.1E-10	6.7E-06	6.1E-10	6.1E-10	6.1E-10	6.1E-10	6.1E-10	6.1E-10	6.1E-10	6.1E-10	4.9E-10	6.1E-10	6.1E-10
F38	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09	1.4E-09
F39	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11
F40	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11
F41	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	1.1E-04	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	NaN	9.7E-11	9.7E-11
F42	3.3E-10	2.8E-12	9.7E-11	9.7E-11	9.7E-11	9.7E-11	9.7E-11	**NaN**	9.7E-11	**NaN**	**3.4E-01**	**NaN**	7.5E-11	**NaN**	**NaN**
F43	1.6E-07	7.8E-04	2.8E-12	2.8E-12	9.7E-11	**4.4E-01**	9.7E-11	**NaN**	9.7E-11	**NaN**	9.7E-11	**NaN**	9.1E-11	**NaN**	**NaN**
F44	7.2E-10	7.2E-10	7.2E-10	7.2E-10	7.2E-10	7.2E-10	7.2E-10	2.0E-02	7.2E-10	2.3E-04	1.2E-02	2.3E-02	7.2E-10	8.9E-10	2.1E-04
F45	9.4E-10	3.8E-10	3.8E-10	3.8E-10	3.8E-10	3.8E-10	3.8E-10	2.1E-02	3.8E-10	2.1E-02	9.2E-05	**7.4E-02**	3.8E-10	2.1E-02	**2.3E-01**
F46	9.7E-11	9.7E-11	9.9E-07	9.7E-11	9.7E-11	**NaN**	9.7E-11	**NaN**	9.7E-11	**NaN**	**NaN**	**1.0E+00**	9.7E-11	**NaN**	**NaN**
F47	2.5E-10	5.8E-09	2.5E-10	2.5E-10	2.5E-10	2.5E-10	2.5E-10	8.1E-02	2.5E-10	**8.1E-02**	1.9E-06	**8.1E-02**	2.5E-10	**8.1E-02**	**8.1E-02**
F48	2.6E-08	**6.3E-02**	2.4E-09	8.6E-07	5.2E-10	3.1E-10	8.6E-07	**3.2E-01**	8.6E-07	**9.6E-01**	**5.6E-01**	1.9E-05	1.0E-08	**1.9E-01**	8.6E-03
F49	9.7E-11	9.7E-11	9.1E-11	9.7E-11	9.7E-11	5.5E-08	9.7E-11	**NaN**	9.7E-11	**NaN**	**NaN**	**3.4E-01**	9.7E-11	**NaN**	**NaN**
F50	**NaN**	**NaN**	9.7E-11	**NaN**	**NaN**	**NaN**	**NaN**	**NaN**	9.7E-11	**NaN**	9.7E-11	1.8E-05	4.1E-02	**NaN**	**NaN**
+	**39**	**46**	**46**	**42**	**48**	**43**	**47**	**35**	**49**	**36**	**45**	**41**	**38**	**38**	**38**
≈	7	2	2	8	2	6	3	15	1	13	5	7	11	12	11
-	4	2	2	0	0	1	0	0	0	1	0	2	1	0	1

Appendix A. and the results are presented in Table 11. The results show that the performance of PO remains unaffected even if the functions are shifted 10 or 100 units in each direction of the search space.

6. Classical engineering problems

The proposed algorithm (PO) is also tested on four well-known constrained engineering design problems: Welded beam design (WBD), speed reducer design (SRD), pressure vessel design (PVD) and tension/compression spring design (TCSD). The mathematical formulations of the problems are presented in Appendix A. The schematic views of the problems are depicted in Figure 9 [89] and the variables involved in these problems are presented in Table 12. The table also presents the settings used for the parameters of PO to solve the problems. As the engineering problems are

Table 9: Comparison of PO with other politics inspired algorithms by using same functions which have been used by those algorithms.

Algo		Sphere	Rosenbrock	Ackley	Griewank	Rastrigin	Schwefel
POA	Avg.	21.0104	51.3701	2.4587	0.8011	3.6	2.11E+03
	Std.	78.6304	1.21E+02	0.0027	0.0576	0.8	6.65E-02
GPO	Avg.	0.3688	0.8564	0.0073	0.0246	3.51E-06	2.05E+03
	Std.	16.6683	0.0841	0.9525	0.0403	7.15E-12	1.03E+02
ECO	Avg.	N/A	4.58E-08	3.32E-03	1.50E-03	1.00E-07	N/A
	Std.	N/A	5.56E-08	4.16E-03	1.68E-03	1.89E-07	N/A
PO	Avg.	**0**	**0**	**-8.88E-16**	**0**	**0**	**1.27E-05**
	Std.	**0**	**0**	**0**	**0**	**0**	**0**

Table 10: Impact of high-dimensionality on the performance of PO.

	Dimensions = 50			Dimensions = 1000		
Fn	**Med**	**Avg**	**Std**	**Med**	**Avg**	**Std**
F1	0	0	0	0	0	0
F2	0.000229	0.000242	0.000155	0.00023	0.000238	0.000167
F3	0	0	0	0	0	0
F4	6.2E-183	7.7E-179	0	6.7E-182	1.5E-177	0
F5	1.2E-161	2.7E-158	0	6.5E-160	1.3E-157	0
F6	0	0	0	0	0	0
F7	-275	-275	0	-5975	-5975	0
F8	6.8E-291	1.8E-274	0	3.2E-221	5.6E-208	0
F9	2.5E-184	1.5E-179	0	2.3E-180	2.1E-179	0
F10	0	0	0	0	0	0
F11	0	0	0	0	0	0
F12	0	0	0	0	0	0
F13	0.03066	0.035802	0.019449	0.235565	0.47818	0.359758
F14	0	0	0	0	0	0
F15	3.3E-212	9.9E-188	0	6.2E-124	9.9E-114	3.1E-113
F26	1.27E-05	1.27E-05	0	1.27E-05	1.27E-05	0
F27	0	0	0	0	0	0
F28	0.9	0.929167	0.046431	1	0.98	0.042164
F29	17.4695	18.42905	10.8334	66145934	65405523	2032182
F30	1.4E-203	2.8E-196	0	2.4E-202	3.1E-198	0
F31	1.9E-218	6.7E-214	0	2.9E-215	1.9E-209	0
F32	-8.9E-16	-8.9E-16	0	-8.9E-16	-8.9E-16	0
F33	1	1	0	1	1	0
F34	0	0	0	0	0	0
F35	-1958.31	-1958.31	4.65E-13	-39166.2	-39166.2	0
F36	0	0	0	0	0	0
F37	-1	-0.83333	0.380693	-1	-1	0
F38	1.21E-20	1.27E-20	3.05E-21	0	0	0
F39	1.35E-32	1.35E-32	5.59E-48	1.35E-32	1.35E-32	2.88E-48
F40	9.42E-33	9.42E-33	0	4.71E-34	4.71E-34	9.02E-50

Table 11: Exhibition of invariance to function shifting on 16 shiftable benchmark functions, where m is the shift length, which defines how much a function is shifted in all directions of the search space.

	m=0		**m=10**		**m=100**	
	Avg	Std	Avg	Std	Avg	Std
F1	0	0	0	0	0	0
F3	0	0	0	0	0	0
F4	7.9E-180	0	0	0	0	0
F7	-275	0	-275	0	-275	0
F9	5.9E-181	0	0	0	0	0
F10	0	0	0	0	0	0
F12	0	0	0	0	0	0
F14	0	0	0	0	0	0
F18	0	0	0	0	0	0
F21	1.38E-87	2.4E-103	1.38E-87	2.4E-103	1.38E-87	2.4E-103
F22	0	0	0	0	0	0
F27	0	0	0	0	0	0
F30	2.9E-180	0	1.04E-16	1.68E-16	4.55E-15	4.8E-15
F32	-8.9E-16	0	-8.9E-16	0	-8.9E-16	0
F36	0	0	0	0	0	0
F38	1.21E-20	4.75E-35	1.21E-20	8.1E-36	1.21E-20	5.66E-35

constrained optimization problems, a simple constraint handling technique called death penalty (scalar penalty function) [90] is used. In this approach, solutions which violate any of the constraints are penalized by a large fitness value (in case of minimization).

Table 12: Variables of engineering optimization problems and parameter settings of PO for these problems. n denotes the number of parties, members in each party and constituencies. λ is the party switching rate.

Problem	Parameters	PO parameters
WBD	Weld thickness (x_1) Weld/bar length (x_2) Beam height (x_3) Beam thickness (x_4)	n = 12 λ = 0.05 to 0 T_{max} = 100
SRD	Face width (x_1) Teeth module (x_2) Number of teeth (x_3) First shaft length (x_4) Second shaft length (x_5) First shaft diameter (x_6) Second shaft diameter (x_7)	n = 8 λ = 0.1 to 0 T_{max} = 75
PVD	Head thickness (x_1) Shell thickness (x_2) Inner radius (x_3) Cylindrical section length (x_4)	n = 18 λ = 0.1 to 0 T_{max} = 60
TCSD	Wire diameter (x_1) Mean coil Diameter (x_2) Active coils (x_3)	n = 14 λ = 0.1 to 0 T_{max} = 50

6.1. Welded beam design problem (WBD)

In this problem, the optimal cost of welding a beam with a strong member is determined. The statistical results generated from 25 randomly started independent executions of PO are compared with some good algorithms in the literature and reported in Table 13. The table shows that PO finds the best statistical results in a fewer number of objective function evaluations (NFEs) as compared to the others. The values of the constraints and the design variables obtained for the best solution are presented and compared with the literature in Table A.20 in Appendix A. The results show that PO performs better than the others except NMPSO [91]. However, NMPSO does not fulfill the third constraint $g3(x)$ as precisely as PO does.

6.2. Speed reducer design problem (SRD)

In SRD, the objective is to minimize the weight of a speed reducer subject to the constraints on stresses in the shafts, transverse deflection of the shafts, surface stress and bending stress of the gear teeth. The proposed algorithm is statistically compared with a wide range of other algorithms in the literature and the results are presented in Table 14. The results show the superiority of PO over the others, while using a fewer number of function evaluations. The values of the variables obtained for the best solution are compared with other algorithms in Table A.21 in Appendix A. The results show that PO performs as good as the best algorithms from the literature do.

Journal Pre-proof

Table 13: Comparison of the statistical results of PO with a wide range of other algorithms in the literature for the welded beam design problem. NFEs denotes the number of objective function evaluations.

Algorithm	Worst	Mean	Best	SD	NFEs
PO	**1.724852**	**1.724851**	**1.724851**	2.53E-07	**15,600**
PSO-DE [92]	**1.724852**	1.724852	1.724852	6.7E-16	66,600
COMDE [93]	**1.724852**	1.724852	1.724852	1.6E-12	20,000
DELC [94]	**1.724852**	1.724852	1.724852	4.1E-13	20,000
MADE [95]	**1.724852**	1.724852	1.724852	9.6E-16	18,000
AMDE [96]	**1.724852**	1.724852	1.724852	1.1E-15	18,000
MDDE [97]	1.725000	1.725000	1.725000	1.0E-15	24,000
NM–PSO [91]	1.733393	1.726373	1.724717	0.003500	80,000
WCA [37]	1.744697	1.726427	1.724856	0.004290	46,450
WOA [4]	NA	1.732000	NA	0.022600	9,900
PSO [11]	NA	1.742200	NA	0.012750	13,770
CPSO [98]	1.782143	1.748831	1.728024	0.012900	240,000
HPSO [99]	1.814295	1.749040	1.724852	0.040100	81,000
CDE [100]	N.A.	1.768150	1.733460	N.A.	NA
DE [7]	1.824105	1.768158	1.733461	0.022100	204,800
GA2 [101]	1.785835	1.771973	1.748309	0.011200	900,000
GA3 [102]	1.993408	1.792654	1.728226	0.074700	80,000
GA4 [103]	1.993408	1.792654	1.728226	0.074700	NA
CAEP [104]	3.179709	1.971809	1.724852	0.443000	50,020
CS [18]	NA	2.014400	0.081300	1.946000	NA
BA [105]	NA	2.142200	0.051200	2.084000	NA
CGWO [106]	2.435700	2.428900	1.725450	1.357800	NA
UPSO [107]	N.A.	2.837210	1.921990	0.683000	NA
GWO [12]	2.913600	2.859400	1.942100	2.690800	NA
GSA [33]	NA	3.576100	NA	1.287400	10,750

Table 14: Comparison of the statistical results of PO with a wide range of other algorithms in the literature for the speed reducer design problem. NFEs denotes the number of objective function evaluations.

Algorithm	Worst	Mean	Best	SD	NFEs
PO	**2994.471057**	**2994.471051**	**2994.471047**	0.000003	**5,400**
WCA [37]	2994.505578	2994.474392	2994.471066	0.007400	15,150
MDE [108]	NA	2996.367220	2996.356689	0.008200	24,000
DELC [94]	2994.471066	2994.471066	2994.471066	0.000000	30,000
DEDS [109]	2994.471066	2994.471066	2994.471066	0.000000	30,000
ABC [110]	NA	2997.058000	2997.058000	0.000000	30,000
HEAA [111]	2994.752311	2994.613368	2994.499107	0.070000	40,000
PSO–DE [92]	2996.348204	2996.348174	2996.348167	0.000006	54,350
SC [59]	3009.964736	3001.758264	2994.744241	4.000000	54,456

6.3. Pressure vessel design problem (PVD)

The objective in this problem is to optimize the total cost to design a pressure vessel, which includes material, formation and welding of a cylindrical pressure vessel. The statistical performance of PO is compared with other well-known algorithms and results are presented in Table 15. The results show that PO outperforms the others in all three cases (worst, mean and best), while the number of function evaluations (NFEs) are significantly less than the others. The values of the design variables obtained for the best solution and the constraints are presented and compared in Table A.22 in Appendix A. The results show that PO performs better than the others and finds the best optimal solution by satisfying all constraints.

Table 15: Comparison of the statistical results of PO with a wide range of other algorithms in the literature for the pressure vessel design problem. NFEs denotes the number of objective function evaluations.

Algorithm	Worst	Mean	Best	SD	NFEs
PO	**5908.0250**	**5891.8068**	**5885.3997**	**8.4746**	**20,520**
NMPSO [91]	5960.0557	5946.7901	5930.3137	9.1610	80,000
PSO-DE [92]	NA	6059.7140	6059.7140	NA	NA
TLBO [44]	NA	6059.7140	6059.7140	NA	NA
WOA [4]	NA	6068.0500	NA	65.6519	6,300
CDE [100]	6371.0455	6085.2303	6059.7340	43.0130	204,800
HPSO [99]	6288.6770	6099.9323	6059.7143	86.2000	81,000
CPSO [98]	6363.8041	6147.1332	6061.0777	86.4500	240,000
GWO [12]	6395.3600	6159.3200	6051.5630	379.6740	NA
GA4 [103]	6469.3220	6177.2530	6059.9460	130.9290	NA
GA3 [102]	6469.3220	6177.2533	6059.9463	130.9297	80,000
G-QPSO [112]	7544.4925	6440.3786	6059.7208	448.4711	8,000
QPSO [112]	8017.2816	6440.3786	6059.7209	479.2671	8,000
PSO [11]	NA	6531.1000	NA	154.3716	14,790
UPSO [107]	9387.7700	8016.3700	6154.7000	745.8690	NA
PSO [11]	14076.3240	8756.6803	6693.7212	1492.5670	8,000
GSA [33]	NA	8932.9500	NA	683.5475	7,110

6.4. Tension/compression spring design problem

This is the problem of optimally designing a tension/compression spring having minimum weight. The statistical results are compared with some good algorithms in the literature and reported in Table 16. The statistics show that NMPSO [91] outperforms PO and other algorithms. However, it is shown in Table A.23 in Appendix A that NMPSO does not fulfill the first and second constraints as precisely as PO and other equivalent algorithms do. Hence, we can conclude that PO satisfies all the constraints of the problem and finds the comparable results in significantly less iterations than the other algorithms.

Table 16: Comparison of the statistical results of PO with a wide range of other algorithms in the literature for the tension/compression spring design problem. NFEs denotes the number of objective function evaluations.

Algorithm	Worst	Mean	Best	SD	NFEs
NM–PSO [91]	0.0126	0.0126	0.0126	0.0000	80,000
PO	0.0128	0.0127	0.0127	0.0000	**10,500**
DELC [94]	0.0127	0.0127	0.0127	0.0000	20,000
HEAA [111]	0.0127	0.0127	0.0127	0.0000	24,000
PSO–DE [92]	0.0127	0.0127	0.0127	0.0000	24,950
MADE [95]	0.0127	0.0127	0.0127	0.0000	20,000
AMDE [96]	0.0127	0.0127	0.0127	0.0000	20,000
DELC [94]	0.0127	0.0127	0.0127	0.0000	20,000
DEDS [109]	0.0127	0.0127	0.0127	0.0000	24,000
DE [7]	0.0128	0.0127	0.0127	0.0000	204,800
HPSO [99]	0.0127	0.0127	0.0127	0.0000	81,000
ABC [110]	NA	0.0127	0.0127	0.0128	30,000
CPSO [98]	0.0129	0.0127	0.0127	0.0005	240,000
GA3 [102]	0.0130	0.0127	0.0127	0.0001	80,000
WCA [37]	0.0130	0.0127	0.0127	0.0001	11,750
G-QPSO [112]	0.0178	0.0135	0.0127	0.0013	2,000
CAEP [104]	0.0151	0.0136	0.0127	0.0008	50,020
GSA [33]	NA	0.0136	NA	0.0026	4,980
QPSO [112]	0.0181	0.0139	0.0127	0.0013	2,000
PSO [11]	0.0718	0.0196	0.0129	0.0117	2,000

Journal Pre-proof

7. Managerial implications

The main objective of this paper is to propose a novel nature-inspired meta-heuristic for global optimization. The outstanding performance of the proposed algorithm on a comprehensive suite of benchmarks against well-known state-of-the-art algorithms proves that PO has all those properties which are required in a good optimization algorithm. For example, the heuristics practitioners may incorporate RPPUS in their algorithms to improve their exploitative skill and convergence speed. The concept of logical division of population to assign a dual role to each individual solution would also give a new direction to algorithm designers to map their inspiration in this way. Furthermore, the interaction between better solutions (constituency winners in this paper) can also be incorporated in algorithms which subdivide the population.

Moreover, the applicability of the proposed algorithm is demonstrated on three well-known engineering optimization problems. The outstanding performance of the proposed algorithm against very well-known other meta-heuristics would encourage managers to apply PO to solve other industrial and real-world optimization problems, such as identification of key determinants for sustainable transportation planning [113], optimizing the design of virtual factory systems [114], optimizing the lot-sizing [115–117], supply-chain network designing, optimizing, and coordinating [118–122], optimization of multi-product constrained and integrated economic production quantity (EPQ) [123], maintenance scheduling [124], sustainable transport policy evaluation [125, 126], and solving joint replenishment problems [127–131].

8. Conclusion

This study presents a novel socio-inspired meta-heuristic called Political Optimizer (PO) by mapping different phases of the political process, such as party formation and constituency allocation, election campaign, party switching, inter-party election, and parliamentary affairs. PO logically divides the population into the political parties and constituencies, which enables a candidate solution to update its position with respect to two good solutions: the party leader and the constituency winner. The exclusivity of this position updating strategy is that each solution updates its position with respect to a unique pair consisting of a party leader and a constituency winner, which enhances exploration if the solutions are distant and exploitation when solutions start converging. Moreover, to further enhance exploitation, this paper proposes a new position updating strategy called recent past-based position updating strategy (RPPUS). The concept of party switching is also incorporated, which helps to control the diversity of the solutions. Finally, the interaction between the constituency winners aids to further improve their position and promotes convergence.

To evaluate the performance of PO, a comprehensive suite of 50 benchmark functions and 15 state of the art algorithms are used. The empirical investigation and statistical results show that PO either outperforms the others or performs equivalently. Moreover, the applicability of PO on real world problems is also tested by solving 4 constrained engineering design problems and the results show the versatility of PO. Furthermore, the impact of high-dimensionality on the performance of PO is investigated by optimizing 30 scalable benchmark functions and it is shown that the performance of PO remains consistent even if the dimensions of functions are increased to 1000. Finally, it is shown that PO is invariant to function shifting and its performance remains consistent if the function landscape is shifted in the search space.

Besides the above-mentioned qualities, PO has some weak points which should be highlighted. First,PO adds a little time complexity overhead due to party switching and parliamentary affairs phases but that overhead does not increase the cost of PO asymptotically. Second, we presume in this paper that total number of parties, number of party members and total number of constituencies are equal. This assumption makes the mapping very simple and also reduces the overhead of tunable parameters. Nonetheless, we believe that PO's overall performance can be improved by easing this assumption but at the cost of additional parameters and proposition of a different constituency allocation scheme which we leave as a future direction. Third, to achieve best results it is necessary to well-tune the upper limit of party switching rate (λ_{max}) and the population size (n^2).

This study opens up a wide range of possibilities for future research. The concept of logical division of the population and assigning dual role to candidate solutions may be applied in other algorithms. RPPUS can be incorporated in other existing or new optimization algorithms to improve their behavior, especially their exploitation capability. Many options can be tried to come up with a variant of PO like investigating different logical divisions of the population and incorporation of independent candidates. An independent candidate is the one who does not belong to any political party. In future, we would like to merge this strategy with PSO. The consistent performance of PO for the high-

dimensional functions opens a door for the researchers to investigate the applicability of PO to large-scale optimization problems. Moreover, based on the good performance for the engineering problems the capability of PO to solve highly-constrained problems should also be investigated. Furthermore, in politics each political party has an agenda and an agenda can be mapped by an objective function, which may allow each political party to optimize different objective function(s). It would be interesting to extend PO to solve multi-objective optimization problems. Moreover, we would also apply our proposed algorithm to complex real-life applications to check its effectiveness.

Appendix A. Supplementary data

Appendix A.1. Mathematical formulation of WBD

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4\,(14 + x_2) \\
\text{subject to} \quad & g_1(x) = \tau(x) - \tau_{\max} \leqslant 0 \\
& g_2(x) = \sigma(x) - \sigma_{\max} \leqslant 0 \\
& g_3(x) = x_1 - x_4 \leqslant 0 \\
& g_4(x) = 0.10471x_1^2 + \\
& \qquad 0.04811x_3x_4\,(14 + x_2) - 5 \leqslant 0 \\
& g_5(x) = 0.125 - x_1 \leqslant 0 \\
& g_6(x) = \delta(x) - \delta_{\max} \leqslant 0 \\
& g_7(x) = P - P_c(x) \leqslant 0 \\
\text{range} \quad & 0.1 \leqslant x_i \leqslant 2 \quad i = 1, 4 \\
& 0.1 \leqslant x_i \leqslant 10 \quad i = 2, 3 \\
\text{where} \quad & \tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \\
& \tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J} \\
& M = P\left(L + \frac{x_2}{2}\right), \\
& R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \\
& J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \\
& \sigma(x) = \frac{6PL}{x_4x_3^2}, \quad \delta(x) = \frac{4PL^3}{Ex_3^3x_4} \\
& P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \\
& P = 6000lb, \quad L = 14in, \quad E = 30 \times 10^6\text{psi}, \\
& G = 12 \times 10^6\text{psi}, \quad \tau_{\max} = 13,600psi \\
& \sigma_{\max} = 30,000\text{psi}, \quad \delta_{\max} = 0.25in
\end{aligned}
\tag{A.1}
$$

Appendix A.2. Mathematical formulation of SRD

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & f(x) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) \\
& - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) \\
& + 0.7854\left(x_4x_6^2 + x_5x_7^2\right) \\
\text{subject to} \quad & g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leqslant 0 \\
& g_2(x) = \frac{397.5}{x_1x_2^2x_2^2} - 1 \leqslant 0 \\
& g_3(x) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leqslant 0 \\
& g_4(x) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leqslant 0 \\
& g_5(x) = \frac{\left[(745x_4/x_2x_3)^2 + 16.9 \times 10^6\right]^{1/2}}{110x^3} - 1 \leqslant 0 \\
& g_6(x) = \frac{\left[(745x_5/x_2x_3)^2 + 157.5 \times 10^6\right]^{1/2}}{85x_7^3} - 1 \leqslant 0 \\
& g_7(x) = \frac{x_2x_3}{40} - 1 \leqslant 0 \\
& g_8(x) = \frac{5x_2}{x_1} - 1 \leqslant 0 \\
& g_9(x) = \frac{x_1}{12x_2} - 1 \leqslant 0 \\
& g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leqslant 0 \\
& g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leqslant 0 \\
\text{where} \quad & 2.6 \leqslant x_1 \leqslant 3.6, \quad 0.7 \leqslant x_2 \leqslant 0.8, \quad 17 \leqslant x_3 \leqslant 28, \\
& 7.3 \leqslant x_4 \leqslant 8.3, \quad 7.3 \leqslant x_5 \leqslant 8.3, \quad 2.9 \leqslant x_6 \leqslant 3.9, \\
& 5.0 \leqslant x_7 \leqslant 5.5
\end{aligned}
\tag{A.2}
$$

Appendix A.3. Mathematical formulation of PVD

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 \\
& + 19.84x_1^2x_3 \\
\text{subject to} \quad & g_1(x) = -x_1 + 0.0193x \\
& g_2(x) = -x_2 + 0.00954x_3 \leqslant 0 \\
& g_3(x) = -\pi x_3^2x_4 - 4/3\pi x_3^3 + 1296,000 \leqslant 0 \\
& g_4(x) = x_4 - 240 \leqslant 0 \\
\text{where} \quad & 0 \leqslant x_i \leqslant 100 \quad i = 1, 2 \\
& 10 \leqslant x_i \leqslant 200 \quad i = 3, 4
\end{aligned}
\tag{A.3}
$$

Journal Pre-proof

Appendix A.4. Mathematical formulation of TCSD

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & f(x) = (x_3 + 2)\, x_2 x_1^2 \\
\text{subject to} \quad & g_1(x) = 1 - x_2^3 x_3 / 71,785 x_1^4 x_1^4 \leqslant 0 \\
& g_2(x) = 4x_2^2 - x_1 x_2 / 12,566 \left(x_2 x_1^3 - x_1^4\right) \\
& \quad + 1/5108 x_1^2 - 1 \leqslant 0 \\
& g_3(x) = 1 - 140.45 x_1 / x_2^2 x_3 \leqslant 0 \\
& g_4(x) = x_2 + x_1/1.5 - 1 \leqslant 0 \\
\text{where} \quad & 0.05 \leqslant x_1 \leqslant 2.00 \\
& 0.25 \leqslant x_2 \leqslant 1.30 \\
& 2.00 \leqslant x_3 \leqslant 15.00
\end{aligned}
\tag{A.4}
$$

Table A.17: Mathematical definitions of unimodal benchmark functions.

Function definitions
$F1 = \Sigma_{i=1}^{n} x_i^2$
$F2 = \Sigma_{i=1}^{n} i x_i^4 + random[0,1]$
$F3 = \Sigma_{i=1}^{D} \lvert x_i \rvert^{i+1}$
$F4 = \Sigma_{i=1}^{n} \lvert x_i \rvert$
$F5 = \max_i(\lvert x_i \rvert, 1 \leq i \leq n)$
$F6 = \Sigma_{i=1}^{n} ([x_i + 0.5])^2$
$F7 = 25 + \Sigma_{i=1}^{n} (\lfloor x_i \rfloor)$
$F8 = \Sigma_{i=1}^{n} (\Sigma_{j=1}^{i} x_j)^2$
$F9 = \Sigma_{i=1}^{n} \lvert x_i \rvert + \Pi_{i=1}^{n} \lvert x_i \rvert$
$F10 = \Sigma_{i=1}^{n} x_i^{10}$
$F11 = \Sigma_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
$F12 = \Sigma_{i=1}^{n-1} (x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{x_i^2+1}$
$F13 = (x_1 - 1)^2 + \Sigma_{i=2}^{D} i(2x_i^2 - x_{i-1})^2$
$F14 = \Sigma_{i=1}^{D/4} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$
$F15 = \Sigma_{i=1}^{n} x_i^2 + (\Sigma_{i=1}^{n} 0.5 i x_i)^2 + (\Sigma_{i=1}^{n} 0.5 i x_i)^4$
$F16 = exp(-\Sigma_{i=1}^{n} (x_i/\beta)^{2m}) - 2exp(-\Sigma_{i=1}^{n} x_i^2) \Pi_{i=1}^{n} cos^2(x_i)$
$F17 = \Sigma_{i=1}^{d} [\Sigma_{j=1}^{d} (j + \beta)(x_j^i - \frac{1}{j^i})]^2$
$F18 = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1 x_2 + x_2^2$
$F19 = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$
$F20 = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
$F21 = (x_1 + 10)^2 + (x_2 + 10)^2 + e^{-x_1^2 - x_2^2}$
$F22 = 0.26(x_1^2 + x_2^2) - 0.48 x_1 x_2$
$F23 = 0.5 + \frac{cos^2(sin(\lvert x^2 - y^2 \rvert)) - 0.5}{(1 + 0.001(x^2 + y^2))^2}$
$F24 = 2\frac{x_1^3}{3} - 8x_1^2 + 33x_1 - x_1 x_2 + 5 + [(x_1 - 4)^2 + (x_2 - 5)^2 - 4]^2$
$F25 = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

Table A.18: Mathematical definitions of Multimodal benchmark functions.

Function definitions
$F26 = 418.9829n - \Sigma_{i=1}^{n} - x_i sin(\sqrt{\lvert x_i \rvert})$
$F27 = \Sigma_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$
$F28 = 1 + \Sigma_{i=1}^{n} sin^2(x_i) - 0.1 e^{(\Sigma_{i=1}^{n} x_i^2)}$
$F29 = \Sigma_{i=1}^{n} (x^2 - i)^2$
$F30 = \Sigma_{i=1}^{n} \lvert x_i sin(x_i) + 0.1 x_i \rvert$
$F31 = \Sigma_{i=1}^{n} \epsilon_i \lvert x_i \rvert^i$
$F32 = -20exp(-0.2\sqrt{1/n \Sigma_{i=1}^{n} x_i^2}) - exp(1/n \Sigma_{i=1}^{n} cos(2\pi x_i)) + 20 + e$
$F33 = \Sigma_{i=1}^{n} 8\sin^2[7(x_i - 0.9)^2] + 6\sin^2[14(x_1 - 0.9)^2] + (x_i - 0.9)^2$
$F34 = 1 - cos(2\pi \sqrt{\Sigma_{i=1}^{n} x_i^2}) + 0.1\sqrt{\Sigma_{i=1}^{n} x_i^2}$
$F35 = \frac{1}{2} \Sigma_{i=1}^{n} (x_i^4 - 16x_i^2 + 5x_i)$
$F36 = 1/4000 \Sigma_{i=1}^{n} x_i^2 - \Pi_{i=1}^{n} cos(x_i/\sqrt{i}) + 1$
$F37 = \left(\Sigma_{i=1}^{n} sin^2(x_i) - e^{-\Sigma_{i=1}^{n} x_i^2}\right) e^{-\Sigma_{i=1}^{n} sin^2 \sqrt{\lvert x_i \rvert}}$
$F38 = (\Sigma_{i=1}^{n} \lvert x_i \rvert) exp(-\Sigma_{i=1}^{n} sin(x_i^2))$
$F39 = 0.1\{sin^2(3\pi x_1) + \Sigma_{i=1}^{n} (x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \Sigma_{i=1}^{n} u(x_i, 5, 100, 4)$
$F40 = \pi/n\{10sin(\pi y_i) + \Sigma_{i=1}^{n-1} [(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \Sigma_{i=1}^{n} u(x_i, 10, 100, 4)$
$F41 = x^2 + y^2 + 25(sin^2(x) + sin^2(y))$
$F42 = -200e^{-0.2\sqrt{x^2+y^2}} + 5e^{cos(3x)+sin(3y)}$
$F43 = cos(x) sin(y) - \frac{x}{y^2+1}$
$F44 = sin(x) e^{(1-cos(y))^2} + cos(y) e^{(1-sin(x))^2} + (x - y)^2$
$F45 = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 X_2 - 4x_2^2 + 4x_2^4$
$F46 = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x1 - 6)^2 + 10(1 - \frac{1}{8\pi}) cos x_1 + 10$
$F47 = -\Sigma_{i=1}^{4} c_i exp(-\Sigma_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$
$F48 = -\Sigma_{i=1}^{4} c_i exp(-\Sigma_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$
$F49 = -0.0001(\lvert sin(x) sin(y) exp(\lvert 100 - \frac{\sqrt{x^2+y^2}}{\pi} \rvert) \rvert + 1)^{0.1}$
$F50 = \lvert x^2 + y^2 + xy \rvert + \lvert sin(x) \rvert + \lvert cos(y) \rvert$

Journal Pre-proof

Table A.19: Mathematical definitions of shifted versions of 16 benchmark test functions selected from the suite of 50 unimodal and multimodal functions. Shifted means the landscapes are shifted m units in all directions of the search space, where m denotes the shift length.

Function	Mathematical formulation	Range	F_{min}
$F1-$ Shifted Sphere	$\Sigma_{i=1}^{n}(x_i - m)^2$	$[-100 + m, 100 + m]$	0
$F3-$ Shifted Powell Sum	$\Sigma_{i=1}^{n}\|(x_i - m)\|^{i+1}$	$[-1 + m, 1 + m]$	0
$F4-$ Shifted Schwefel's 2.20	$\Sigma_{i=1}^{n}\|(x_i - m)\|$	$[-100 + m, 100 + m]$	0
$F7-$ Shifted Stepint	$25 + \Sigma_{i=Shifted1}^{n}(\lfloor(x_i - m)\rfloor)$	$[-5.12 + m, 5.12 + m]$	$25 - 6n$
$F9-$ Shifted Schwefel's 2.22	$\Sigma_{i=1}^{n}\|(x_i - m)\| + \Pi_{i=1}^{n}\|(x_i - m)\|$	$[-100 + m, 100 + m]$	0
$F10-$ Shifted Schwefel's 2.23	$\Sigma_{i=1}^{n}(x_i - m)^{10}$	$[-10 + m, 10 + m]$	0
$F12-$ Shifted Brown	$\Sigma_{i=1}^{n-1}((x_i - m)^2)^{((x_{i+1}-m)^2+1)} + ((x_{i+1} - m)^2)^{(x_i-m)^2+1}$	$[-1 + m, 4 + m]$	0
$F14-$ Shifted Powell Singular	$\Sigma_{i=1}^{n/4}((x_{4i-3} - m) + 10(x_{4i-2} - m))^2 + 5((x_{4i-1} - m) - (x_{4i} - m))^2$ $+((x_{4i-2} - m) - (x_{4i-1} - m))^4 + 10((x_{4i-3} - m) - (x_{4i} - m))^4$	$[-4 + m, 5 + m]$	0
$F18-$ Shifted Three-Hump Camel	$2(x_1 - m)^2 - 1.05(x_1 - m)^4 + \frac{(x_1-m)^6}{6} + (x_1 - m)(x_2 - m) + (x_2 - m)^2$	$[-5 + m, 5 + m]$	0
$F21-$ Shifted Brent	$((x_1 - m) + 10)^2 + ((x_2 - m) + 10)^2 + e^{-(x_1-m)^2-(x_2-m)^2}$	$[-10 + m, 10 + m]$	0
$F22-$ Shifted Matyas	$0.26((x_1 - m)^2 + (x_2 - m)^2) - 0.48(x_1 - m)(x_2 - m)$	$[-10 + m, 10 + m]$	0
$F27-$ Shifted Rastrigin	$\Sigma_{i=1}^{n}[(x_i - m)^2 - 10cos(2\pi(x_i - m)) + 10]$	$[-5.12 + m, 5.12 + m]$	0
$F30-$ Shifted Alpine N. 1	$\Sigma_{i=1}^{n}\|(x_i - m)sin((x_i - m)) + 0.1(x_i - m)\|$	$[-10 + m, 10 + m]$	0
$F32-$ Shifted Ackley	$-20exp(-0.2\sqrt{1/n\Sigma_{i=1}^{n}(x_i - m)^2}) - exp(1/n\Sigma_{i=1}^{n}cos(2\pi(x_i - m))) + 20 + e$	$[-32 + m, 32 + m]$	0
$F36-$ Shifted Griewank	$1/4000\Sigma_{i=1}^{n}(x_i - m)^2 - \Pi_{i=1}^{n}cos((x_i - m)/\sqrt{i}) + 1$	$[-100 + m, 100 + m]$	0
$F38-$ Shifted Xin-She Yang N. 2	$(\Sigma_{i=1}^{n}\|(x_i - m)\|)exp(-\Sigma_{i=1}^{n}sin((x_i - m)^2))$	$[-2\pi + m, 2\pi + m]$	0

Table A.20: Comparison of the best solution obtained from the previous algorithms for the welded beam design problem.

Algorithm	X1(h)	X2(l)	X3(t)	X4(b)	g1(X)	g2(X)	g3(X)	g4(X)	g5(X)	g6(X)	g7(X)	f(X)
NM–PSO [91]	0.205830	3.468338	9.036624	0.205730	-0.025250	-0.053122	0.000100	-3.433169	-0.080830	-0.235540	-0.031555	1.724717
PO	0.205730	3.470472	9.036624	0.205730	-0.000560	-0.003697	0.000000	-3.432985	-0.080730	-0.235540	-0.002159	**1.724851**
CAEP [104]	0.205700	3.470500	9.036600	0.205700	1.988676	4.481548	0.000000	-3.433213	-0.080700	-0.235538	2.603347	1.724852
HGA [132]	0.205700	3.470500	9.036600	0.205700	1.988676	4.481548	0.000000	-3.433213	-0.080700	-0.235538	2.603347	1.724852
MADE	0.205730	3.470489	9.036624	0.205730	-1.81E-12	0.000000	0.000000	-3.432984	-0.080730	-0.235540	-2.72E-12	1.724852
WCA [37]	0.205728	3.470522	9.036620	0.205729	-0.034128	-0.000035	-0.000001	-3.432980	-0.080728	-0.235540	-0.013503	1.724856
CGWO [106]	0.343891	1.883570	9.031330	0.212121	NA	NA	NA	NA	NA	NA	NA	1.725450
GWO [12]	0.205676	3.478377	9.036810	0.205778	NA	NA	NA	NA	NA	NA	NA	1.726240
CPSO [98]	0.202369	3.544214	9.048210	0.205723	-13.655547	-78.814077	-0.003350	-3.424572	-0.077369	-0.235595	-4.472858	1.728024
GA3 [102]	0.205986	3.471328	9.020224	0.206480	-0.103049	-0.231747	-0.000500	-3.430044	-0.080986	-0.235514	-58.646888	1.728226
WOA [4]	0.205396	3.484293	9.037426	0.206276	NA	NA	NA	NA	NA	NA	NA	1.730499
GSA [33]	0.182129	3.856979	10.000000	0.202376	NA	NA	NA	NA	NA	NA	NA	1.879952
CS [18]	0.182200	3.795100	9.998100	0.211100	NA	NA	NA	NA	NA	NA	NA	1.946000
BA [105]	0.154300	5.736100	8.862700	0.229700	NA	NA	NA	NA	NA	NA	NA	2.084000
PSO [11]	0.183200	4.717100	10.000000	0.218900	NA	NA	NA	NA	NA	NA	NA	2.146100
HS [84]	0.244200	6.223100	8.291500	0.244300	NA	NA	NA	NA	NA	NA	NA	2.380700

Journal Pre-proof

Table A.21: Comparison of the best solution obtained from the previous algorithms for the speed reducer design problem.

Algorithm	X1	X2	X3	X4	X5	X6	X7	f(X)
PO	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2867	**2994.471**
DEDS [109]	3.5+9	0.7+9	17.0000	7.3+9	7.7153	3.3502	5.2867	**2994.471**
DELC [94]	3.5+9	0.7+9	17.0000	7.3+9	7.7153	3.3502	5.2867	**2994.471**
WCA [37]	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2867	**2994.471**
HEAA [111]	3.5000	0.7000	17.0000	7.3004	7.7154	3.3502	5.2867	2994.499
PSO-DE [92]	3.5000	0.7000	17.0000	7.3000	7.8000	3.3502	5.2867	2996.348
MDE [108]	3.5000	0.7000	17.0000	7.3002	7.8000	3.3502	5.2867	2996.357

Table A.22: Comparison of the best solution obtained from the previous algorithms for the pressure vessel design problem.

Algorithm	X1	X2	X3	X4	g1(X)	g2(X)	g3(X)	g4(X)	f(X)
PO	0.7782	0.3847	40.3215	199.9733	0.0000	0.0000	-0.2602	-40.0267	**5885.3997**
NMPSO [91]	0.8036	0.3972	41.6392	182.4120	0.0000	0.0000	-1.5914	-57.5879	5930.3137
GWO [12]	0.8125	0.4345	42.0892	176.7587	NA	NA	NA	NA	6051.5639
ACO [10]	0.8125	0.4375	42.1036	176.5727	NA	NA	NA	NA	6059.0888
HPSO [99]	0.8125	0.4375	42.0984	176.6366	0.0000	-0.0358	3.1226	-63.3634	6059.7143
G-QPSO [112]	0.8125	0.4375	42.0984	176.6372	0.0000	-0.0358	-0.2179	-63.3628	6059.7208
CDE [100]	0.8125	0.4375	42.0984	176.6376	0.0000	-0.0358	-3.7051	-63.3623	6059.7340
DE [7]	0.8125	0.4375	42.0984	176.6377	NA	NA	NA	NA	6059.7340
WOA [4]	0.8125	0.4375	42.0983	176.6390	NA	NA	NA	NA	6059.7410
ES [9]	0.8125	0.4375	42.0981	176.6405	NA	NA	NA	NA	6059.7456
GA3 [102]	0.8125	0.4375	42.0974	176.6540	-0.0020	-0.0358	-24.7593	-63.3460	6059.9463
GA [5]	0.8125	0.4375	42.0974	176.6541	NA	NA	NA	NA	6059.9463
CPSO [98]	0.8125	0.4375	42.0913	176.7465	0.0000	-0.0004	-118.7687	-63.2535	6061.0777
PSO [11]	0.8125	0.4375	42.0913	176.7465	NA	NA	NA	NA	6061.0777
GSA [33]	1.1250	0.6250	55.9887	84.4542	NA	NA	NA	NA	8538.8359

Table A.23: Comparison of the best solution obtained from the previous algorithms for the tension/compression spring design problem.

Algorithm	X1	X2	X3	g1(X)	g2(X)	g3(X)	g4(X)	f(X)
NM–PSO [91]	0.05162	0.35550	11.33327	0.00101	0.00099	-4.06186	-0.72859	0.01263
PO	0.05248	0.37594	10.24509	-0.00002	0.00000	-4.09004	-0.71439	**0.01267**
DEDS [109]	0.05169	0.35672	11.28897	0.00000	0.00000	-4.05379	-0.72773	0.01267
HEAA [111]	0.05169	0.35673	11.28829	0.00000	0.00000	-4.05381	-0.72772	0.01267
DELC [94]	0.05169	0.35672	11.28897	0.00000	0.00000	-4.05379	-0.72773	0.01267
WCA [37]	0.05168	0.35652	11.30041	0.00000	0.00000	-4.05340	-0.72786	0.01267
MADE [95]	0.05169	0.35672	11.28897	0.00000	0.00000	-4.05379	-0.72773	0.01267
GWO [12]	0.05169	0.35674	11.28885	NA	NA	NA	NA	0.01267
DE [7]	0.05161	0.35471	11.41083	NA	NA	NA	NA	0.01267
HS [84]	0.05115	0.34987	12.07643	NA	NA	NA	NA	0.01267
CPSO [98]	0.05173	0.35764	11.24454	-0.00083	-0.00003	-4.05131	-0.72709	0.01267
PSO [11]	0.05173	0.35764	11.24454	NA	NA	NA	NA	0.01267
WOA [4]	0.05121	0.34522	12.00403	NA	NA	NA	NA	0.01268
GA3 [102]	0.05199	0.36397	10.89052	-0.00126	-0.00003	-4.06134	-0.72270	0.01268
ES [9]	0.05199	0.36397	10.89052	NA	NA	NA	NA	0.01268
GSA [33]	0.05028	0.32368	13.52541	NA	NA	NA	NA	0.01270

Journal Pre-proof

References

[1] M. Diver, Lilamani: A Study in Possibilities, Oxford University Press, 2004.

[2] F. Glover, Future paths for integer programming and links to artificial intelligence, Computers & Operations Research 13 (5) (1986) 533–549.

[3] F. Fausto, A. Reyna-Orta, E. Cuevas, Á. G. Andrade, M. Perez-Cisneros, From ants to whales: metaheuristics for all tastes, Artificial Intelligence Review (Jan. 2019).

[4] S. Mirjalili, A. Lewis, The whale optimization algorithm, Advances in Engineering Software 95 (2016) 51–67.

[5] J. H. Holland, Genetic algorithms, Scientific American 267 (1) (1992) 66–73.

[6] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems), A Bradford Book, 1992.

[7] J. Lampinen, R. Storn, Differential evolution, in: New Optimization Techniques in Engineering, Springer Berlin Heidelberg, 2004, pp. 123–166.

[8] D. Simon, Biogeography-based optimization, IEEE Transactions on Evolutionary Computation 12 (6) (2008) 702–713.

[9] A. Huning, ARSP: Archiv für Rechts- und Sozialphilosophie / Archives for Philosophy of Law and Social Philosophy 62 (2) (1976) 298–300.

[10] M. Dorigo, G. D. Caro, Ant colony optimization: a new metaheuristic, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), IEEE.

[11] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN95 - International Conference on Neural Networks, IEEE.

[12] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, Advances in Engineering Software 69 (2014) 46–61.

[13] A. H. Gandomi, A. H. Alavi, Krill herd: A new bio-inspired optimization algorithm, Communications in Nonlinear Science and Numerical Simulation 17 (12) (2012) 4831–4845.

[14] 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, Advances in Engineering Software 114 (2017) 163–191.

[15] J. J. Yu, V. O. Li, A social spider algorithm for global optimization, Applied Soft Computing 30 (2015) 614–627.

[16] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, Soft Computing 23 (3) (2018) 715–734.

[17] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Generation Computer Systems 97 (2019) 849–872.

[18] X.-S. Yang, S. Deb, Cuckoo search via levy flights, in: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), IEEE, 2009.

[19] R. Salgotra, U. Singh, The naked mole-rat algorithm, Neural Computing and Applications 31 (12) (2019) 8837–8857.

[20] H. A. Alsattar, A. A. Zaidan, B. B. Zaidan, Novel meta-heuristic bald eagle search optimisation algorithm, Artificial Intelligence Review (Jul. 2019).

[21] S. Shadravan, H. Naji, V. Bardsiri, The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems, Engineering Applications of Artificial Intelligence 80 (2019) 20–34.

[22] A. Kaveh, M. Kooshkebaghi, Artificial coronary circulation system; a new bio-inspired metaheuristic algorithm, Scientia Iranica (Apr. 2019).

[23] S. Harifi, M. Khalilian, J. Mohammadzadeh, S. Ebrahimnejad, Emperor penguins colony: a new metaheuristic algorithm for optimization, Evolutionary Intelligence 12 (2) (2019) 211–226.

[24] J.-B. Lamy, Artificial feeding birds (AFB): A new metaheuristic inspired by the behavior of pigeons, Springer International Publishing, 2018, pp. 43–60.

[25] R. Masadeh, B. A., A. Sharieh, Sea lion optimization algorithm, International Journal of Advanced Computer Science and Applications 10 (5) (2019).

[26] H. Sharma, G. Hazrati, J. C. Bansal, Spider monkey optimization algorithm, in: Studies in Computational Intelligence, Springer International Publishing, 2018, pp. 43–59.

[27] G.-G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, Neural Computing and Applications 31 (7) (2015) 1995–2014.

[28] R. G. Morais, L. M. Mourelle, N. Nedjah, Hitchcock birds inspired algorithm, in: Computational Collective Intelligence, Springer International Publishing, 2018, pp. 169–180.

[29] W.-H. Tan, J. Mohamad-Saleh, Normative fish swarm algorithm (NFSA) for optimization, Soft Computing (May 2019).

[30] G. Dhiman, V. Kumar, Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, Knowledge-Based Systems 165 (2019) 169–196.

[31] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: Squirrel search algorithm, Swarm and Evolutionary Computation 44 (2019) 148–175.

[32] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[33] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A gravitational search algorithm, Information Sciences 179 (13) (2009) 2232–2248.

[34] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, Information Sciences 222 (2013) 175–184.

[35] S. Mirjalili, SCA: A sine cosine algorithm for solving optimization problems, Knowledge-Based Systems 96 (2016) 120–133.

[36] O. K. Erol, I. Eksin, A new optimization method: Big bang–big crunch, Advances in Engineering Software 37 (2) (2006) 106–111.

[37] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems, Computers & Structures 110-111 (2012) 151–166.

[38] Anita, A. Yadav, AEFA: Artificial electric field algorithm for global optimization, Swarm and Evolutionary Computation 48 (2019) 93–108.

[39] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: A novel optimization algorithm, Knowledge-Based Systems (2019) 105190.

[40] S. Jeong, P. Kim, A population-based optimization method using newton fractal, Complexity 2019 (2019) 1–9.

[41] A. Gilyén, S. Arunachalam, N. Wiebe, Optimizing quantum optimization algorithms via faster quantum gradient computation, in: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2019, pp. 1425–1444.

[42] X. Feng, Y. Liu, H. Yu, F. Luo, Physarum-energy optimization algorithm, Soft Computing (Sep. 2017).

[43] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: Thermal exchange optimization, Advances in Engineering Software 110 (2017) 69–84.

[44] R. Rao, V. Savsani, D. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, Computer-Aided Design 43 (3) (2011) 303–315.

Journal Pre-proof

[45] N. Moosavian, B. K. Roodsari, Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks, Swarm and Evolutionary Computation 17 (2014) 14–24.
[46] N. Ghorbani, E. Babaei, Exchange market algorithm, Applied Soft Computing 19 (2014) 177–187.
[47] M. Kumar, A. J. Kulkarni, S. C. Satapathy, Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology, Future Generation Computer Systems 81 (2018) 252–272.
[48] S. Q. Salih, A. A. Alsewari, A new algorithm for normal and large-scale optimization problems: Nomadic people optimizer, Neural Computing and Applications (Oct. 2019).
[49] P. R. Singh, M. A. Elaziz, S. Xiong, Ludo game-based metaheuristics for global and engineering optimization, Applied Soft Computing 84 (2019) 105723.
[50] S. Balochian, H. Baloochian, Social mimic optimization algorithm and engineering applications, Expert Systems with Applications 134 (2019) 178–191.
[51] M. Bodaghi, K. Samieefar, Meta-heuristic bus transportation algorithm, Iran Journal of Computer Science 2 (1) (2018) 23–32.
[52] M. El-Abd, Global-best brain storm optimization algorithm, Swarm and Evolutionary Computation 37 (2017) 27–44.
[53] G. Sartori, Parties and Party Systems: A Framework for Analysis (ECPR Press Classics), ECPR Press, 2005.
URL `https://www.xarg.org/ref/a/0954796616/`
[54] G. V. Golosov, The effective number of parties, Party Politics 16 (2) (2009) 171–192.
[55] J. M. Carey, Presidential versus parliamentary government, in: Handbook of New Institutional Economics, Springer Berlin Heidelberg, 2008, pp. 91–122.
[56] J. L. Melvix, Greedy politics optimization: Metaheuristic inspired by political strategies adopted during state assembly elections, in: 2014 IEEE International Advance Computing Conference (IACC), IEEE, 2014.
[57] A. Borji, A new global optimization algorithm inspired by parliamentary political competitions, in: MICAI 2007: Advances in Artificial Intelligence, Springer Berlin Heidelberg, pp. 61–71.
[58] W. Lv, C. He, D. Li, S. Cheng, S. Luo, X. Zhang, Election campaign optimization algorithm, Procedia Computer Science 1 (1) (2010) 1377–1386.
[59] T. Ray, K. Liew, Society and civilization: an optimization algorithm based on the simulation of social behavior, IEEE Transactions on Evolutionary Computation 7 (4) (2003) 386–396.
[60] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007.
[61] A. H. Kashan, League championship algorithm (LCA): An algorithm for global optimization inspired by sport championships, Applied Soft Computing 16 (2014) 171–200.
[62] S. Satapathy, A. Naik, Social group optimization (SGO): a new population evolutionary optimization technique, Complex & Intelligent Systems 2 (3) (2016) 173–203.
[63] A. H. Kashan, R. Tavakkoli-Moghaddam, M. Gen, Find-fix-finish-exploit-analyze (f3ea) meta-heuristic algorithm: An effective algorithm with new evolutionary operators for global optimization, Computers & Industrial Engineering 128 (2019) 192–218.
[64] G. Brammya, S. Praveena, N. S. N. Preetha, R. Ramya, B. R. Rajakumar, D. Binu, Deer hunting optimization algorithm: A new nature-inspired meta-heuristic paradigm, The Computer Journal (May 2019).
[65] P. Niu, S. Niu, N. liu, L. Chang, The defect of the grey wolf optimization algorithm and its verification method, Knowledge-Based Systems 171 (2019) 37–43.
[66] W. zhen Sun, J. sheng Wang, X. Wei, An improved whale optimization algorithm based on different searching paths and perceptual disturbance, Symmetry 10 (6) (2018) 210.
[67] S. M. Bozorgi, S. Yazdani, IWOA: An improved whale optimization algorithm for optimization problems, Journal of Computational Design and Engineering 6 (3) (2019) 243–259.
[68] J. Liang, A. Qin, P. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Transactions on Evolutionary Computation 10 (3) (2006) 281–295.
[69] F. Chen, X. Sun, D. Wei, Y. Tang, Tradeoff strategy between exploration and exploitation for PSO, in: 2011 Seventh International Conference on Natural Computation, IEEE, 2011.
[70] H. Joshi, S. Arora, Enhanced grey wolf optimization algorithm for global optimization, Fundamenta Informaticae 153 (3) (2017) 235–264.
[71] M. Tubishat, M. A. M. Abushariah, N. Idris, I. Aljarah, Improved whale optimization algorithm for feature selection in arabic sentiment analysis, Applied Intelligence 49 (5) (2018) 1688–1707.
[72] M. H. Suid, M. A. Ahmad, M. R. T. R. Ismail, M. R. Ghazali, A. Irawan, M. Z. Tumari, An improved sine cosine algorithm for solving optimization problems, in: 2018 IEEE Conference on Systems, Process and Control (ICSPC), IEEE, 2018.
[73] M. Belazzoug, M. Touahria, F. Nouioua, M. Brahimi, An improved sine cosine algorithm to select features for text categorization, Journal of King Saud University - Computer and Information Sciences (Jul. 2019).
[74] X. Ji, H. Ye, J. Zhou, Y. Yin, X. Shen, An improved teaching-learning-based optimization algorithm and its application to a combinatorial optimization problem in foundry industry, Applied Soft Computing 57 (2017) 504–516.
[75] H. F. Farahani, F. Rashidi, An improved teaching-learning-based optimization with differential evolution algorithm for optimal power flow considering HVDC system, Journal of Renewable and Sustainable Energy 9 (3) (2017) 035505.
[76] X. Qu, B. Liu, Z. Li, W. Duan, R. Zhang, W. Zhang, H. Li, A novel improved teaching-learning based optimization for functional optimization, in: 2016 12th IEEE International Conference on Control and Automation (ICCA), IEEE, 2016.
[77] G. Ciuprina, D. Ioan, I. Munteanu, Use of intelligent-particle swarm optimization in electromagnetics, IEEE Transactions on Magnetics 38 (2) (2002) 1037–1040.
[78] J. Luo, H. Chen, K. Wang, C. Tong, J. Li, Z. Cai, LGWO: An improved grey wolf optimization for function optimization, in: Lecture Notes in Computer Science, Springer International Publishing, 2017, pp. 99–105.
[79] S. Gupta, K. Deep, Improved sine cosine algorithm with crossover scheme for global optimization, Knowledge-Based Systems 165 (2019) 374–406.
[80] C. Qu, Z. Zeng, J. Dai, Z. Yi, W. He, A modified sine-cosine algorithm based on neighborhood search and greedy levy mutation, Computational Intelligence and Neuroscience 2018 (2018) 1–19.
[81] X. Li, P. Niu, G. Li, J. Liu, H. Hui, Improved teaching-learning-based optimization algorithms for function optimization, in: 2015 11th International Conference on Natural Computation (ICNC), IEEE, 2015.
[82] M. B. Ghalia, Particle swarm optimization with an improved exploration-exploitation balance, in: 2008 51st Midwest Symposium on Circuits and Systems, IEEE, 2008.
[83] D. Weyland, A rigorous analysis of the harmony search algo-

Journal Pre-proof

rithm, International Journal of Applied Metaheuristic Computing 1 (2) (2010) 50–60.

[84] K. S. Lee, Z. W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Computer Methods in Applied Mechanics and Engineering 194 (36-38) (2005) 3902–3933.

[85] C. L. Camacho-Villalón, M. Dorigo, T. Stützle, Why the intelligent water drops cannot be considered as a novel algorithm, in: Lecture Notes in Computer Science, Springer International Publishing, 2018, pp. 302–314.

[86] H. Shah-Hosseini, Intelligent water drops algorithm, International Journal of Intelligent Computing and Cybernetics 1 (2) (2008) 193–212.

[87] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (1) (2011) 3–18.

[88] W. Lv, Q. Xie, P. Tang, X. Zhang, S. Luo, S. Cheng, An experimental study of benchmarking functions for election campaign algorithm, in: 2010 International Conference on Measuring Technology and Mechatronics Automation, IEEE, 2010.

[89] Z. Liu, Z. Li, P. Zhu, W. Chen, A parallel boundary search particle swarm optimization algorithm for constrained optimization problems, Structural and Multidisciplinary Optimization 58 (4) (2018) 1505–1522.

[90] C. A. C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Computer methods in applied mechanics and engineering 191 (11-12) (2002) 1245–1287.

[91] E. Zahara, Y.-T. Kao, Hybrid nelder–mead simplex search and particle swarm optimization for constrained engineering design problems, Expert Systems with Applications 36 (2) (2009) 3880–3886.

[92] H. Liu, Z. Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, Applied Soft Computing 10 (2) (2010) 629–640.

[93] A. W. Mohamed, H. Z. Sabry, Constrained optimization based on modified differential evolution algorithm, Information Sciences 194 (2012) 171–208.

[94] L. Wang, L. po Li, An effective differential evolution with level comparison for constrained engineering design, Structural and Multidisciplinary Optimization 41 (6) (2009) 947–963.

[95] F. Hamza, H. Abderazek, S. Lakhdar, D. Ferhat, A. R. Yıldız, Optimum design of cam-roller follower mechanism using a new evolutionary algorithm, The International Journal of Advanced Manufacturing Technology 99 (5-8) (2018) 1267–1282.

[96] H. Abderazek, D. Ferhat, A. Ivana, Adaptive mixed differential evolution algorithm for bi-objective tooth profile spur gear optimization, The International Journal of Advanced Manufacturing Technology 90 (5-8) (2016) 2063–2073.

[97] E. Mezura-Montes, C. A. C. Coello, J. Velázquez-Reyes, L. Muñoz-Dávila, Multiple trial vectors in differential evolution for engineering design, Engineering Optimization 39 (5) (2007) 567–589.

[98] R. A. Krohling, L. dos Santos Coelho, Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 36 (6) (2006) 1407–1416.

[99] Q. He, L. Wang, A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization, Applied Mathematics and Computation 186 (2) (2007) 1407–1422.

[100] F. zhuo Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization, Applied Mathematics and Computation 186 (1) (2007) 340–356.

[101] C. A. C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Computers in Industry 41 (2) (2000) 113–127.

[102] C. A. C. Coello, E. M. Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, Advanced Engineering Informatics 16 (3) (2002) 193–203.

[103] S. Gupta, R. Tiwari, S. B. Nair, Multi-objective design optimisation of rolling bearings using genetic algorithms, Mechanism and Machine Theory 42 (10) (2007) 1418–1443.

[104] C. A. C. Coello, R. L. Becerra, Efficient evolutionary optimization through the use of a cultural algorithm, Engineering Optimization 36 (2) (2004) 219–236.

[105] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer Berlin Heidelberg, 2010, pp. 65–74.

[106] M. Kohli, S. Arora, Chaotic grey wolf optimization algorithm for constrained optimization problems, Journal of Computational Design and Engineering 5 (4) (2018) 458–472.

[107] K. E. Parsopoulos, M. N. Vrahatis, Unified particle swarm optimization for solving constrained engineering optimization problems, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 582–591.

[108] E. Mezura-Montes, J. Velazquez-Reyes, C. C. Coello, Modified differential evolution for constrained optimization, in: 2006 IEEE International Conference on Evolutionary Computation, IEEE.

[109] M. Zhang, W. Luo, X. Wang, Differential evolution with dynamic stochastic selection for constrained optimization, Information Sciences 178 (15) (2008) 3043–3074.

[110] D. Karaboga, B. Basturk, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 789–798.

[111] Y. Wang, Z. Cai, Y. Zhou, Z. Fan, Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique, Structural and Multidisciplinary Optimization 37 (4) (2008) 395–413.

[112] L. dos Santos Coelho, Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems, Expert Systems with Applications 37 (2) (2010) 1676–1683.

[113] R. Sayyadi, A. Awasthi, A simulation-based optimisation approach for identifying key determinants for sustainable transportation planning, International Journal of Systems Science: Operations & Logistics 5 (2) (2016) 161–174.

[114] Y. Hao, P. Helo, A. Shamsuzzoha, Virtual factory system design and implementation: integrated sustainable manufacturing, International Journal of Systems Science: Operations & Logistics 5 (2) (2016) 116–132.

[115] A. Gharaei, S. A. H. Shekarabi, M. Karimi, Modelling and optimal lot-sizing of the replenishments in constrained, multi-product and bi-objective EPQ models with defective products: Generalised cross decomposition, International Journal of Systems Science: Operations & Logistics (2019) 1–13.

[116] S. A. H. Shekarabi, A. Gharaei, M. Karimi, Modelling and optimal lot-sizing of integrated multi-level multi-wholesaler supply chains under the shortage and limited warehouse space: generalised outer approximation, International Journal of Systems Science: Operations & Logistics 6 (3) (2018) 237–257.

[117] A. Gharaei, M. Karimi, S. A. H. Shekarabi, Joint economic lot-sizing in multi-product multi-level integrated supply chains: Generalized benders decomposition, International Journal of

Journal Pre-proof

Systems Science: Operations & Logistics (2019) 1–17.
[118] Y.-C. Tsao, Design of a carbon-efficient supply-chain network under trade credits, International Journal of Systems Science: Operations & Logistics 2 (3) (2015) 177–186.
[119] M. Rabbani, N. Foroozesh, S. M. Mousavi, H. Farrokhi-Asl, Sustainable supplier selection by a new decision model based on interval-valued fuzzy sets and possibilistic statistical reference point systems under uncertainty, International Journal of Systems Science: Operations & Logistics 6 (2) (2017) 162–178.
[120] S. Yin, T. Nishi, G. Zhang, A game theoretic model for coordination of single manufacturer and multiple suppliers with quality variations under uncertain demands, International Journal of Systems Science: Operations & Logistics 3 (2) (2015) 79–91.
[121] B. Giri, S. Bardhan, Coordinating a supply chain with backup supplier through buyback contract under supply disruption and uncertain demand, International Journal of Systems Science: Operations & Logistics 1 (4) (2014) 193–204.
[122] N. H. Shah, U. Chaudhari, L. E. Cárdenas-Barrón, Integrating credit and replenishment policies for deteriorating items under quadratic demand in a three echelon supply chain, International Journal of Systems Science: Operations & Logistics (2018) 1–12.
[123] A. Gharaei, S. A. H. Shekarabi, M. Karimi, E. Pourjavad, A. Amjadian, An integrated stochastic EPQ model under quality and green policies: generalised cross decomposition under the separability approach, International Journal of Systems Science: Operations & Logistics (2019) 1–13.
[124] C. Duan, C. Deng, A. Gharaei, J. Wu, B. Wang, Selective maintenance scheduling under stochastic maintenance quality with multiple maintenance actions, International Journal of Production Research 56 (23) (2018) 7160–7178.
[125] A. Awasthi, H. Omrani, A goal-oriented approach based on fuzzy axiomatic design for sustainable mobility project selection, International Journal of Systems Science: Operations & Logistics 6 (1) (2018) 86–98.
[126] R. Sayyadi, A. Awasthi, An integrated approach based on system dynamics and ANP for evaluating sustainable transportation policies, International Journal of Systems Science: Operations & Logistics (2018) 1–10.
[127] L. Cui, J. Deng, L. Wang, M. Xu, Y. Zhang, A novel locust swarm algorithm for the joint replenishment problem considering multiple discounts simultaneously, Knowledge-Based Systems 111 (2016) 51–62.
[128] L. Cui, Y. Zhang, J. Deng, M. Xu, A novel multi-item joint replenishment problem considering multiple type discounts, PloS one 13 (6) (2018).
[129] L. Cui, J. Deng, R. Liu, D. Xu, Y. Zhang, M. Xu, A stochastic multi-item replenishment and delivery problem with lead-time reduction initiatives and the solving methodologies, Applied Mathematics and Computation 374 (2020) 125055.
[130] L. Cui, J. Deng, Y. Zhang, G. Tang, M. Xu, Hybrid differential artificial bee colony algorithm for multi-item replenishment-distribution problem with stochastic lead-time and demands, Journal of Cleaner Production (2020) 119873.
[131] L. Cui, J. Deng, Y. Zhang, Z. Zhang, M. Xu, The bare-bones differential evolutionary for stochastic joint replenishment with random number of imperfect items, Knowledge-Based Systems (2019) 105416.
[132] Q. Yuan, F. Qian, A hybrid genetic algorithm for twice continuously differentiable NLP problems, Computers & Chemical Engineering 34 (1) (2010) 36–41.

Journal Pre-proof