

Search-in-forest optimizer: A bio-inspired meta-heuristic algorithm for global optimization problems

Atefeh Amindoust (✉ Atefeh_Amindoust@yahoo.com)

Islamic Azad University Najafabad Branch

Amin Ahwazian

Islamic Azad University Najafabad Branch

Reza Tavakkoli-Moghaddam

Tehran University: University of Tehran

Mehrdad Nikbakhta

Islamic Azad University Najafabad Branch

Research Article

Keywords: search in forest, swarm intelligence, metaheuristic, intra-group search, global optimization problems, transfer of an expert member to the top team

Posted Date: June 29th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-440082/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Search-in-forest optimizer: A bio-inspired meta-heuristic algorithm for global optimization problems

Amin Ahwazian^a, Atefeh Amindoust^{a,*}, Reza Tavakkoli-Moghaddam^b, Mehrdad Nikbakht^a

^a Department of Industrial Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

^b School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

Abstract

The present research proposes a new particle swarm optimization-based metaheuristic algorithm entitled “search in forest (SIF) optimizer” to solve the global optimization problems. The algorithm is designed based on the organized behavior of search teams looking for missing persons in a forest. According to SIF optimizer, a number of teams each including several experts in the search field spread out across the forest and gradually move in the same direction by finding clues from the target until they find the missing person. This search structure was designed in a mathematical structure in the form of intra-group search operators and transferring the expert member to the top team. In addition, the efficiency of the algorithm was assessed by comparing the results to the standard representations and a problem with the genetic, grey wolf, salp swarm, and ant lion optimizers. According to the results, the proposed algorithm was efficient for solving many numerical representations, compared to the other algorithms.

Keywords: search in forest, swarm intelligence, metaheuristic, intra-group search, global optimization problems, transfer of an expert member to the top team

1. Introduction

An optimization process includes finding the best solution from all feasible solutions for a specific problem. With regard to their nature, optimization algorithms can be widely divided into two groups: deterministic algorithms and stochastic intelligent algorithms (Yang 2008). Deterministic algorithms will produce the same output when the solutions to a problem have the same initial values. These methods are classified as gradient restricted methods that move exactly toward the optimal solution. In contrast, deterministic algorithms are generally recognized as gradient-free techniques used in random steps to find the best solution possible. In this method, the optimization process cannot be repeated in any situation (Brownlee 2011). In most cases, however, favorable final solutions can be produced by both

* Corresponding author.

E-mail addresses: am_ahw@yahoo.com (A. Ahwazian), atefeh_amnidoust@yahoo.com (A. Amindoust), tavakoli@ut.ac.ir (R. Tavakkoli-Moghaddam), nikbakht2020@yahoo.com (M. Nikbakht).

techniques. Stochastic intelligent algorithms are divided into two general modes of heuristic and metaheuristic algorithms (Yang et al. 2012).

As the name implies, heuristic algorithms use trial and error to find the solution that works and most of them are applied in various optimization areas such as bat swarm optimization, hill climbing, and simulation annealing.

Many real-world machine learning and AI problems are generally continuous, discrete, finite or infinite (Abbassi et al. 2019). Given these characteristics, some problems cannot be easily solved with ordinary mathematical programming approaches such as conjugate gradient, sequential quadratic programming, fast steepest, and quasi-newton methods (Faris et al. 2019).

Meanwhile, several studies have confirmed the inefficiency or low efficiency of the metaheuristic algorithms in dealing with many large-scale, indirect, and indistinguishable polynomial problems in the real world (Wu et al. 2015). Accordingly, metaheuristic algorithms are designed as a competitive alternative solvent to solve many problems owing to their simplicity and easy implementation process. Moreover, the main operations of these methods do not rely on mathematical features or gradient information. Nevertheless, one of the major drawbacks of metaheuristic algorithms is their sensitivity to setting user-defined parameters. Another problem with these algorithms is their lack of ability to always reach a global optimal solution (Dréo et al. 2006).

Metaheuristic methods are used to solve complicated issues such as scheduling (L. Wang and Zheng 2018), shape design (Rizk-Allah et al. 2017), economic load dispatch (Zou et al. 2016), large-scale data optimization (Yi et al. 2018,2020), infinite impulse response (IIR) systems identification (Zou et al. 2018), malware code detection (Cui et al. 2018), error detection (Li et al. 2013; Yi et al. 2018), forecasting promotion places (Nan et al. 2017), unit commitment (Srikanth et al.2018), classification (Zou et al. 2016,2017), path planning (Wang et al. 2012,2016), vehicle navigation (Chen et al. 2018), knapsack problems (Feng et al. 2018), and cyber-physical systems (Cui et al. 2017), neural network (Pandey et al. 2020), trajectory tracking control of unmanned aerial vehicle (Selma et al. 2020).

Metaheuristic algorithms designed are properly implemented for most optimization issues and can successfully provide a suitable and satisfactory situation. In the area of computer sciences, researchers have proposed a new type of gradient-free technique known as optimization techniques called “genetic algorithm (GA)” by the conceptualization of evolution (Goldberg and Holland 1998). After that, several other techniques have been suggested in the field of optimization, including ant colony optimization (Dorigo and Birattari 2010), monarch butterfly optimization (MBO) (Feng et al. 2018), the Ant Lion Optimizer (Mirjalili 2018), moth search (Wang et al. 2018), artificial bee colony (Wang et al. 2018), evolutionary strategy (Back 1996), harmony search (Geem et al. 2001), imperialist competitive algorithm (Talataheri et al. 2012), earthworm optimization algorithm (Wang et al. 2015), cuckoo search (Wang et al. 2016), biogeography-based optimization (Wang et al.

2014), elephant herding optimization (Wang et al. 2016), and differential evolution (Wang et al. 2012).

In general, metaheuristic algorithms are divided into two main classes (Talbi 2009), single-solution-based (e.g., simulated annealing) and population-based (e.g., GA). In the first class, only one solution is processed in the optimization stage, as the name implies. On the other hand, a set of solutions (i.e., population) evolves with each iteration of the optimization process in the second class. The population-based techniques can often find an optimal or near-optimal solution in the neighborhood. In addition, population-based metaheuristic techniques are usually inspired by natural phenomena, and start the optimization process by producing a set (population) of people, where each person in the population represents a candidate solution to the optimization problem. The population repeatedly evolves with replacing the current population with a new population using some random operators. The optimization process continues until meeting the stopping criteria (i.e., the maximum number of iterations).

2. Literature review

Particle swarm optimization (PSO) is a spirit-based stochastic optimization method proposed by Eberhart and Kennedy (1995) and by Kennedy and Eberhart (1995). Given the inefficiency of the first version of PSO in optimization issues, another version was quickly proposed by Shi and Eberhart (1998). Afterwards, a reproduction operator was incorporated into the algorithm following presenting the concept of a sub-population GA (SPGA) (Suganthan 1999). Another local PSO version was suggested based on the k-means clustering approach known as the social convergence method with hybrid spatial neighborhood and ring topology by Kennedy and Stereotyping (2000). Van Den Bergh (2001) analyzed the definitive version of the PSO algorithm followed by presenting a version of the PSO algorithm with reduction factor after analyzing the convergence behavior of PSO algorithm, which guaranteed convergence and improved its rate (Clerc and Kennedy 2002). The bare-bones PSO (BBPSO) was proposed in as a dynamic model of PSO (Kennedy 2003). In 2004, Emara and Fattah (2004) analyzed the continuous version of PSO algorithm and theoretical analysis was performed on the PSO algorithm for symmetric spherical local neighborhood functions, and numerical tests confirmed the speed characteristics along with reducing variability in the PSO algorithm (Blackwell 2005). Kadiramanathan et al. (2006) analyzed the dynamic stability of particles using Lyapunov stability analysis and the concept of passive system (Kadiramanathan 2006). A marine model was studied to promote search diversity in PSO (Poli et al. 2007). In addition, the social and fully aware particle version of the PSO algorithm was also introduced by Poli (2008) and the stable stochastic analysis encompassed higher-ranking moments, proving its suitability for comprehension of dynamics of particle growth and clarifying the convergence features of PSO (Poli 2009). The modified algorithm presented by Park et al. (2009) introduced the chaotic inertia weight that suddenly falls and simultaneously inclines downward. Zhan et al. (2010) introduced orthogonal learning, in which an orthogonal learning model was applied for efficient sampling.

Li et al. (2011) presented a self-learning PSO algorithm, in which the speed update program can change automatically during the evolution process. Approaches such as search space

compression have been proposed dynamically to determine the search space (Barisal et al. 2013). Garcia-Gonzalo and Fernandez-Martinez (2014) presented a convergent reliability and stability analysis of a series of PSO types and their research was different from the classic PSO in terms of the statistical distribution of PSO parameters. Moreover, Peng and Chen (2015) introduced a coexistence particle optimization algorithm to optimize fuzzy neural networks. Eddaly et al. (2016) presented a hybrid combinatorial particle swarm optimization algorithm (HCPSO) as a resolution technique. An iterated local search algorithm based on probabilistic perturbation is sequentially introduced to the particle swarm optimization algorithm for improving the quality of solution. The computational results showed that their approach was able to improve several best known solutions of the literature. Tanweer et al. (2016) presented a new dynamic particle optimization algorithm and self-resemblance algorithm. Moreover, Wang et al. (2018) proposed a Krill herd algorithm, which is from the category of PSO algorithm. Koyuncu & Ceylan (2018) added a scout bee phase to standard PSO, and formed the Scout Particle Swarm Optimization (ScPSO) to design an efficient technique for continuous function optimization consequently, a robust optimization algorithm was obtained.

3. Search in forest theory

Every year, thousands of people lose their way in forests and mountain areas, and search-and-rescue operations are carried out to find them and bring them to their destination. To this end, some “search agents” are assigned to find the missing people in the forest after passing special education courses in this regard. These individuals are often grouped in “search teams” during operations and attempt to find the target in the depth of forests. It should be noted that in order to perform search operations, different methods can be implemented according to the opinion of the commander. However, given the standard content presented in educational courses held for police and fire brigade, who are among the main forces searching in the forest, multi-member teams are formed at first to search different parts of the forest. Afterwards, each team that finds a clue of the target informs other teams to adjust their direction. Ultimately, all teams search the direction in which most clues are found. The figure 1 graphically shows how to search the forest in accordance with the above-mentioned issues.

[figure1: insert figure1 here.]

Two techniques are applied to organize teams to conduct effective search operations in all forest areas. The first technique is a local search inside each team, in which the commander of each team allocates parts to each member and gives them the responsibility of searching that part. The commander is informed of any clue found by team members and they will send another team member to that area for more effective search in order to deepen the search. Therefore, all team members attempt to conduct an effective search in the parts allocated to them. In the second technique, the team that finds most clues is enforced by other teams. In fact, commanders of teams send one of their team members to the team with

the most clues in order to help conduct an effective search in the forest. Meanwhile, some members remain in the teams that find the least clues in order to continue the search in the forest in case of any mistake made by the top team. In fact, a team may find several clues that are false and cannot lead to finding the target. In this case, other teams continue the search in the forest. However, their search path is in line with the search path of the top team since most clues have been found on this route. The mathematical structure inspired by this behavior will be explained in the next section to solve global optimization problems.

4. Mathematical model of search in forest optimizer

In this section, the steps related to the implementation of the mathematical model of the forest search (SIF) optimizer are described separately.

4.1. Research team production

First, M teams were determined, each with N members, in order to establish initial order in the formation of search teams. In fact, $(P_n)_m$ population is created, in which m shows the number of search teams and n shows the number of team members. Since team members search the forest space in a stochastic manner and they might find a clue at any time, their movement structure is designed in two sections based on a mathematical model. In the first section, a stochastic step is designed for each m search team in the form of Model 1.

$$X_m^t = \sum_{i=1}^{Max_Itr} \sum_{j=1}^i Sign_j \times \alpha_j,$$

s. t.

$$\alpha_j = \begin{cases} 0, & r1 < 0 \\ 1, & r1 \geq 0 \end{cases} \quad \forall j \in \{1, \dots, Max_Itr\}$$

$$sign_j = \begin{cases} 1 & 0 \leq r2 < 0.33 \\ 0 & 0.33 \leq r2 < 0.66 \\ -1 & r2 \geq 0.66 \end{cases}, \quad \forall j \in \{1, \dots, Max_Itr\} \quad (1)$$

Where r1 and r2 are random numbers in the [0, 1] range. Therefore, each search group (population) can search for the missing person stochastic in the forest space. Model 1 produces various stochastic search structures, which are shown in Figure 2.

[figure2: insert figure2 here.]

4.2. Search space division scheme

Given the fact that each search team searches a specific area of the solution space, a specific movement structure must be determined for each member of each team to ensure a thorough search of the entire solution space. In other words, members of each team must cover the entire solution space based on the angle assigned to them to search using stochastic movements. The search area is assumed to be enclosed in a circle in order to formulate this structure. In addition, it is assumed that all search teams are in the center of the circle at a zero moment. As such, a proper angle of the circle is allocated to each team n based on the number of search teams ($|M|$).

[figure3: insert figure3 here.]

In Figure 3, the solution space is divided into eight equal parts based on the enclosed circle and space is allocated to each of the eight teams based on θ angle. Therefore, the entire solution space will be inspected by the search team members. According to Model 1, all search teams make stochastic movements in the allocated space. In addition, the surrounding space is inspected by the research team while searching the main route of their team's direction. Since the opposite view of θ angle is the sine of the angle, sine function, as shown in Model 2, can be used to mathematically define the movement structure of each search member.

$$Y_{mn}^t = \sum_{i=1}^{Max_Itr} \sum_{j=1}^i Sign_j \times \beta_j,$$

s. t.

$$\beta_j = \begin{cases} \sin \theta & 0 < \theta \leq 90 \\ 1 + \sin \theta & 90 < \theta < 180, \\ 1 & \theta = \{180, 360\} \end{cases} \quad \forall j \in \{1, \dots, Max_Itr\}$$

$$\theta = \frac{360}{M}, \quad \forall j \in \{1, \dots, Max_Itr\}$$

$$sign_j = \begin{cases} 1, & r < 0.5 \\ -1, & r \geq 0.5 \end{cases} \quad \forall j \in \{1, \dots, Max_Itr\}$$
(2)

Where r is a random number in $(0, 1)$ range. Similar to Figure 2, the movement structure produced in Model 2 can create various procedures, which ensures the stochastic movement of the entire members of all search teams. The situation of each search team in the solution space is determined in a random matrix, in which columns are indicative of the number of search teams while rows demonstrate the number of members in each team in the form of matrix (3).

$$Position_{Team} = \begin{bmatrix} T_{11} & \cdots & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{m1} & \cdots & T_{mn} \end{bmatrix}$$
(3)

Where T_{mn} indicates the n th member of the m th team. Clearly, each team member has a set of decision-making variables, which is shown in a matrix in the form of Equation (4).

$$T_{mn} = [M_1, M_2, \dots, M_v] \quad (4)$$

Where M_v is the amount of the v th variable in the n th member of the m th population. The fitness of each team member can be estimated based on the problem's objective function and be considered as Equation 5.

$$Fitness_{Team} = \begin{bmatrix} f(T_{11}) & \dots & f(T_{1n}) \\ \vdots & \ddots & \vdots \\ f(T_{m1}) & \dots & f(T_{mn}) \end{bmatrix} \quad (5)$$

Obviously, $f(T_{mn}) = f(T_{mn}(M_1, M_2, \dots, M_v))$. According to the structure created, it could be expressed that team members have the same role as particles in the PSO algorithm, and the created demographic structure is based on GA.

4.3. Integration of search teams

In algorithm iterations, a number of search teams are dissolved due to failure to find proper clues of the target, and their members are added to other teams. In this regard, each team can attract the members of the dissolved team based on their performance. In fact, the allocation of the members of the dissolved team to other teams depends on the teams' fitness. However, the teams' performance must be assessed and the improper teams must be dissolved at the right time. It is best not to eliminate any team at the beginning of the search due to the need for several teams to inspect the entire solution space. As search operations expand, the search teams can be assessed and dissolved in case of poor performance. In this algorithm, it is assumed that the teams have more time to show their performance in primary iterations in order to formulate this behavior mathematically. In final iterations, however, assessments are carried out in shorter periods. This mathematical structure is shown in Model 6.

$$\text{If } \left(\frac{Best^t - Worst(Fitness_m^t)}{Best^t} \right) \leq 0.4 \rightarrow \text{remove team with the worst fitness}$$

$$Best^t = \frac{Best(Fitness_m^t) + Average(Fitness^t)}{2}$$

$$Average(Fitness^t) = \frac{\sum_{m=1} Fitness_m^t}{M} \quad (6)$$

Where Worst is indicative of the fitness of the weakest population, $Best^t$ indicates the fitness of the best member in the t th iteration, and Average function is the mean fitness of the existing populations in that iteration. Following dissolving the team with the weakest

performance, its members are transferred to other teams. However, the number of members allocated to each of the remaining teams depends on their fitness based on Equation 7.

$$I_m^t = \left\lfloor \frac{Fitness_m^t}{\sum_m Fitness_m^t} \times N_r^t \right\rfloor \quad (7)$$

Where N_r^t the number of members and r is the index of the dissolved team. Therefore, the teams with a weak performance can be eliminated in various algorithm iterations and the remaining teams' ability to search can be strengthened through increasing their members.

4.4. Intragroup search

In this section, we describe the local search structure conducted inside each group to improve each of its members and ultimately enhance the group's ability to search. As explained, an intra-group search occurs when one of the members of the group finds a clue. For a more detailed search, the commander of the group sends other members to the direction of the mentioned member to reinforce the local search to a certain extent. To this end, $S_m(t)$ is defined as a set of neighbours on the t th variable in $X_m(t)$ solution.

$$S_m(t) = \{W_m^+(t), W_m^-(t)\} \quad (8)$$

$$W_m^+(t) = X_m^t + c \times (u_m(t) - v_m(t)) \quad (9)$$

$$W_m^-(t) = X_m^t + c \times (u_m(t) - v_m(t)) \quad (10)$$

Where $W_m^+(t)$ and $W_m^-(t)$ indicate two neighbours of X_m^t solution, and $u_m(t) = (u_1(t), u_2(t), \dots, u_M(t))$, $m \in \{1, \dots, M\}$ and $v_m(t) = (v_1(t), v_2(t), \dots, v_M(t))$, $m \in \{1, \dots, M\}$ refer to two random solutions selected from P_m population. Moreover, c shows the cluttered factor obtained based on the normal distribution of $N(\mu, \sigma^2)$. While the application of the cluttered factor in the local search algorithm is not considered as a decision-making criterion, it has a slight impact on the improvement process of algorithm implementation. Meanwhile, careful adjustments must be made for u and v factors. However, it is easily justified that u factor must be greater than zero ($u > 0$). The value of the factor must be determined accurately since very large amounts lead to extremely large values for $(u_m(t) - v_m(t))$, which intensifies the cluttered factor in the algorithm. On the other hand, the value of the mentioned equation will be very small if very low amounts of the factor are used. This leads to very small values obtained from multiplying the c factor into the factor's value, very low cluttered factor, and low algorithm convergence. As such, the parameter should be set with great care. In addition, σ parameter's value should be greater than zero. Notably, very large or very low σ values affect the c parameter, and consequently, the convergence of the algorithm. It is suggested that the values of σ and μ be in the $(0, 1)$ range.

The stochastic step created in models 1 and 2 cannot be used directly since the solutions produced by algorithm iterations for each member of the population should be in the permissible range of problem variables. Therefore, necessary numerical conversions should be made using standardization functions.

$$X_m^t = Z_{X_m^t} \quad (11)$$

$$Y_m^t = Z_{Y_m^t} \quad (12)$$

4.5. Transfer of elite members to the top team

In this research, the final solution was improved using the random walk method (Levy flight) based on the equation below:

$$X_m^{t+1} = X_m^t + Levy(d) \times X_m^t \quad (13)$$

Where t is the number of algorithm iteration and d is the dimension of the team members' position vector. The Levy flight value is estimated using the equation below:

$$Levy(X_m^t) = 0.01 \times \frac{r_1 \times \sigma}{|r_2|^{\frac{1}{\beta}}} \quad (14)$$

Where r_1 and r_2 are random values in the (0, 1) range. In addition, β is a constant number (e.g., 1.5). Finally, σ is calculated based on the equation below:

$$\sigma = \left(\frac{\Gamma(1 + \beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1 + \beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right) \quad (15)$$

Where $\Gamma(X_m^t) = (X_m^t - 1)!$ Therefore, the solutions generated are improved with each algorithm iteration, and the best algorithm solution is considered as the final solution.

Algorithm 1 Pseudo-code of SIF

Initialize the parameters $M, N, Pr, c, Max_iteration, ;$
Initialize the positions of Teams and agents using $T_{mn};$
While ($t \leq Max_iteration$)
 Calculate the fitness of all Teams and agents;
 Update bestFitness, T_{mn}
 Team merging using **Model** (6) and $Pr;$

```
For each search agent  
    Run Local search;  
    Run Levy flight;  
End For  
t = t + 1;  
End While  
Return bestFitness, best position;
```

5. Result and discussion

In this section, we evaluate the performance of the proposed algorithm in solving two general categories of optimization problems, including constrained and unconstrained problems. All problems have been run in a 3.2GHz system with 32GB random access memory in Windows 10.

5.1. Introducing standard unconstrained optimization and parameter regulation tests

In this research, we use four popular standardized tests, including classic unimodal (Yao et al. 1999; Digalakis and Margaritis 2001) and multimodal functions (Molga and Smutnicki 2005), unimodal and multimodal CEC2014 functions and CEC2014 combined functions. The mathematical structure, dimensions and the range of values of these functions are described in tables 1 to 4.

[Table1: insert table1 here.]

[Table2: insert table2 here.]

[Table3: insert table3 here.]

A very important issue in the implementation of a metaheuristic algorithm is the accurate adjustment of parameters to increase their efficiency. It is worth noting that there is still no comprehensive approach to the assessment of the performance of metaheuristic algorithms (Yang 2010). Therefore, the solutions provided by the proposed algorithm should be compared to the results obtained from other algorithms. Therefore, we considered the GA (Davis 1991), grey wolf optimizer (GWO) (Mirjalili et al. 2014), salp swarm (SSA) (Mirjalili et al. 2017), ant lion optimizer (ALO) (Mirjalili 2015), gravitational search algorithm (GSA) (Rashidi Cook 2009), differential evolution (DE) (Storn and Price 1997) and PSO (Zhou et al. 2003) algorithms as the base algorithms to compare the results. It is thereby necessary to adjust the parameters of each of the mentioned algorithms based on the table 4.

[Table4: insert table4 here.]

All algorithms are implemented to achieve final results in similar situations. In all algorithms, the population size is estimated at 50 and a total of 1000 iterations are considered. In order to control the performance of all algorithms, the best results are

considered among the 30 independent performances. In addition, a number of diagrams are presented below to evaluate the results of the proposed algorithm in comparison with other metaheuristic algorithms in terms of solving standard numerical representations.

Diagram of search structure: The diagram shows the location of each search agent in the forest space during the implementation of the algorithm. The diagram helps determine the level of success of search agents in the solution space.

Improved trajectory: The diagram shows the amount of target function in the best solution obtained in each implementation of the algorithm. In fact, the diagram shows the improved trajectory of the best member of the search agent population.

Mean solutions: The diagram demonstrates the mean of a fitness function for all members in each iteration. The diagram indicates the algorithm's ability in converging all population members.

Convergence diagram: The diagram shows the convergence of the algorithm toward the best solution in each iteration.

5.2. Numerical results of the limit of classic unimodal and multimodal samples

The diagrams related to the comparison criteria are presented in the figure (4) following solving the classic unimodal and multimodal numerical representations with the help of the proposed algorithm.

[figure4: insert figure 4 here.]

According to the results, the population members are scattered in the space during the optimization process and are concentrated on the optimal solution. This shows that the algorithm performance has balance in diversification and contemplation phases. In addition, the trajectory of the best member of the population in each iteration shows many numerical changes in the first iterations and then convergence on the final solution. In fact, this confirms a balance in the discovery of proper solutions in the initial iterations (diversification) and focusing on the best solution for local search to make an optimized global finding. Evaluation of the convergent diagram shows that the convergence process is carried out with less slope or is discontinued completely in some iterations. This means that the algorithm has failed to find a solution better than the existing solution. However, a sudden significant improvement is observed and the improvement slope has a suitable state again. This shows that the proposed algorithm can properly pass the trap of local optimization solutions and continue the search in other parts of the space using the operators designed in order to pass local optimization and move toward global optimization. Therefore, the proposed algorithm is able to discover global optimization solutions in all standard representations in a shorter period. The results obtained from solving the classic

unimodal and multimodal numerical representations are presented in table 5 with the help of the base algorithms and proposed algorithm.

[Table5: insert table5 here.]

As observed, the SIF optimizer produced better solutions in all numerical representations, compared to GA, SSA, PSO, and GSA algorithms. However, it failed to produce better solutions in F8, F11, F12, and F13 representations, compared to GWA. In fact, given the fact that GWA is currently one of the best algorithms for solving optimization problems, SIF was unable to defeat GWA but produced very close solutions in all numerical representations. Notably, the mean and standard deviation of solutions are considered in the table above. Meanwhile, SIF optimizer produced better solutions in all numerical representations, compared to the other algorithms in independent implementations. In other words, while SIF optimizer failed to produce better mean solutions in standard representations of F8, F11, F12, and F13, compared to GWA, it was able to provide solutions with higher quality in independent implementations. The best and worst values for final solutions obtained from the independent implementation of different algorithms are presented in the table 6 to better show this superiority. In addition, the number of times that the SIF optimizer has been able to generate better solutions, compared to other algorithms, is shown under the heading NBS.

[Table6: insert table6 here.]

As observed, SIF optimizer produced better independent solutions in all standard representations, compared to several algorithms, which shows its capacity to be used as an efficient algorithm in solving optimization problems.

5.3. Numerical results obtained from the limit of CEC2014 unimodal and multimodal representations

In this section, CEC2014 standard representations presented in two unimodal and multimodal versions are solved using SIF optimizer and other comparative algorithms, and the results are presented in the table below. It should be pointed out that owing to higher computational complexity, compared to F1-F13 functions, the samples can challenge the solution discovery power of the proposed algorithm in the exploration phase. This is mainly due to the fact that discovering the accurate location of the global optimization solution in the complicated computational space of these algorithms is extremely difficult, and the algorithm may get stuck in the local optimization trap. In fact, the operators presented in different algorithms can be well evaluated in this category of standard functions. The necessary comparative results are presented in the tables 7 and 8. Notably, solving the two standard samples was assessed in two modes of 1000 and 10000 iterations to evaluate the effect of an increased number of iterations on the final solutions.

[Table7: insert table7 here.]

As observed, the proposed algorithm was able to provide more efficient solutions in most numerical examples, compared to the other algorithms. This level of superiority is due to the

presence of two different populations, which extremely increases search power in the exploration phase.

[Table8: insert table8 here.]

As observed, the final solutions significantly improved with an increase in the number of iterations, which confirmed the proper implementation of the algorithm's operators to escape the local optimization trap. In fact, an increase in the number of iterations enabled the algorithms to leave the local optimization space and enter the global optimization space by using operators that have the property of causing mutations in the production responses. A 3D structure can be drawn for other functions (see Fig. 5).

[figure5: insert figure 5 here.]

5.4. Numerical results obtained from the limit of CEC2014 combined examples

In this section, six combined numerical examples presented at CEC2014 are considered as evaluation functions and the results of the solution are described as the last numerical analysis presented to test the proposed algorithm using standard examples. These examples can assess the exploitation phase of the proposed algorithm since the space of solutions is designed in a way that it has made the discovery of a global optimization solution in the solution space extremely complicated. In other words, the extend of the solution space leads to the appropriate implementation of the exploration phase during an acceptable period. However, finding a solution to the exploitation phase faces serious challenges. The table 9 presents the numerical results obtained from the implementation of these standard examples.

[Table9: insert table9 here.]

As observed, the proposed algorithm provided better solutions in solving the combined representations, compared to the other algorithms presented in the research. Therefore, it could be expressed that the application of the proposed algorithm can create a suitable condition for finding the final solution in the global optimization solution space. To sum up, the SIF optimizer was able to create a proper situation for a general assessment of the solution space using the mechanism of production of multiple populations in the form of search teams and applying various search factors in each team. Moreover, the algorithm reinforced the discovery phase by integrating search teams in various iterations. In addition, using the Levy flight operator to transfer the elite members resulted in a suitable convergence in the problem-solving procedure. A numerical example corresponding to real-world conditions is solved using this algorithm and the numerical results are fully described to examine the application of the proposed algorithm in solving engineering problems more precisely.

5.5. Solving constrained optimization problems

As mentioned at the beginning of this section, two numerical examples in the area of supply field design are evaluated as constrained optimization problems in order to more accurately evaluate the performance of the proposed algorithm.

5.5.1. Classic supply chain design

In brief, supply chain management is one of the basic issues in the fields of management, economics, industrial engineering, planning, and other related fields. A part of issues in the area of supply chain management is defined as optimization problems and in the field of operations research [79]. In this set of problems, the flow of goods, from the beginning of the production path, which includes the supply of raw materials, manufacturing in production centres, and distribution of the final product, be managed in a way that the final cost of the system is minimized. In this research, the data extracted from the research by (Beamon 1998) is used, which is recognized as one of the most complete classic models in this field. In this research, 20 numerical examples were generated according to the structure of the basic article to more accurately evaluate the behaviour of the SIF optimizer. In table 10 the numerical results obtained from comparison with GWA, SSA, and SIF optimizer were described. Notably, the comparison criteria included the value of the objective function, the amount of deviation of solutions, and the number of iterations required to achieve the final solution.

[Table10: insert table10 here.]

According to Table 10, the SIF optimizer had lower target function values, compared to the other algorithms, which is also observed in the figure 6. In fact, the algorithm is able to conduct a better search in the solution space and report more appropriate results. This is mainly due to the presence of robust operators based on the crossover and mutation in the algorithm.

[figure6: insert figure 6 here.]

Regarding the amount of deviation, the SIF optimizer had the lowest value in this respect, whereas SSA had the highest value, which showed the higher efficiency of the proposed algorithm in solving problems in this field.

[figure7: insert figure 7 here.]

As observed, in figure 7 there was a large gap between the solutions provided by the SIF optimizer and SSA while WGA produced solutions similar to SIF optimizer in some cases. Therefore, it is necessary to more analyse the comparison of the GWA and SIF optimizer, for which statistical tests are used to make a more accurate comparison. Since the tests compared two different societies, the GWO and SIF optimizer, which had the highest

efficiency, was selected as the criterion algorithms and the necessary estimations were carried out. In choosing a statistical test for research, it is necessary to decide on the use of parametric and non-parametric tests in a specific way. In this regard, one of the main criteria for selection is conducting the Kolmogorov-Smirnov test, which shows the normal or abnormal distribution of the data. In general, data has a normal distribution if the mentioned test is rejected, which means that the parametrical statistical tests can be used for research. On the other hand, confirmation of the Kolmogorov-Smirnov test is interpreted as an abnormal distribution of the data, which means that non-parametrical tests should be used in the research. Meanwhile, significant results of the Kolmogorov-Smirnov test ($P\text{-value} \leq 0.05$) show the abnormal distribution of the data and the need for using non-parametric tests since confirmation of the test is defined as non-parametric data. The figure8 diagrams show the result of the assessment of all defined criteria in the test.

[figure8: insert figure 8 here.]

As observed, K-S test results are indicative of the abnormal distribution of all criteria. Therefore, non-parametric tests should be used to compare the two communities. In this research, we used the Kruskal-Wallis test, the results of which are presented in the tables 11 and 12.

[Table11: insert table11 here.]

As observed, the two algorithms are different in terms of all criteria, and SIF optimizer has higher efficiency, compared to GWO, at all times.

[Table12: insert table12 here.]

However, the difference is not significant, and it can be expressed that the two algorithms had similar functions. As such, SIF implementation can lead to highly efficient solutions for problems related to the management chain field.

5.6. Advanced supply chain design problem solving

In this research, we focus on the design of a reverse supply chain to minimize the costs of facility location and material transfer between different categories of the chain in question. Evaluation of this issue is important since determining the proper location of the facility and the proper flow of products transferred between the levels of the chain can improve the performance of the chain and ultimately achieve a suitable profit margin for industry managers. The proposed model is explained in table 13 to 16.

[Table13: insert table13 here.]

[Table14: insert table14 here.]

[Table15: insert table15 here.]

[Table16: insert table16 here.]

The first sentence of the objective function calculates the cost of activation of a supply centre. The second sentence of the objective function shows the cost of activation of producers. In addition, the third sentence estimates the activation cost of collection centres, whereas the fourth sentence calculates the cost of activation of a destruction centre. Moreover, the fifth sentence calculates savings resulted from matching the centres. Notably, the sentence is presented in the target function with a negative symbol since it must be maximized. Furthermore, the sixth sentence shows the transfer of producers from suppliers to producers while the seventh sentence estimates the costs of transfer from producer to retailer. The eighth sentence calculates the cost of transfer of returned products from retailer to collection centres, and the ninth sentence estimates the cost of transferring products from collection centres to destruction centres. Finally, the 10th sentence shows the entire costs of moving a returned product unit from the collection centre to the supplier. Phrases (16) and (17) guarantee that all demands of customers are responded to the direct flow, and all returned products are collected from retailers in the reverse flow. In addition, phases (18-20) are related to flow balance constraints at nodes. In fact, it is guaranteed in these constraints that the products that enter each chain level must exist from the same level. Moreover, constraints (19-26) guarantee that the flow exists only between points where facilitation has been activated and the entire flow does not exceed the capacity of each facilitation. Furthermore, constraints (29-33) guarantee that only one level of capacity is allocated to each facilitation, whereas constraint (31) guarantees that the necessary module exists in the products sent to the desired customer if the modules are mandatory. Additionally, constraint (34) guarantees that customer demand can only be met by a manufacturer when that manufacturer has the ability to add a mandatory customer module. Finally, constraints (35-36) are logical and obvious limitations related to decision variables of a problem.

5.6.1. Comparison of results obtained from supply chain design problem solving values by SIF optimizer and two 2020 algorithm

The supply chain model is solved using the random values by SIF optimizer and two 2020 algorithm Equilibrium Optimizer (EO) (Farmarzi et al. 2020) and Marine Predators Algorithm (MPA) (Farmarzi et al. 2020) in various mode of iteration and the results are presented in the table 18. The results of solved supply chain model by SIF optimizer indicated better optimal values than the MPA and EO. So the SIF is presented to solve

industrial engineering and supply chain problems as an efficient metaheuristic algorithm. The parameters of each of the mentioned algorithms is adjusted based on the table 17.

[Table17: insert table17 here.]

[Table18: insert table18 here.]

5.6.2. Evaluation of results obtained from supply chain design problem solving

In the numerical example of the present model, we assume that a supply chain network is being designed, where four potential points are available for the activation of suppliers, whereas 6, 20, 4, and 4 potential points exist for activation of production centres, customers, the establishment of collection centres and construction of destruction centres respectively. Overall, two suppliers, three producers, two collection centres and two destruction centres must be constructed among the potential points. Also, all centres have three capacity levels. The amount of customer demand is presented in the tables 18 to 31.

[Table19: insert table19 here.]

[Table20: insert table20 here.]

[Table21: insert table21 here.]

[Table22: insert table22 here.]

[Table23: insert table23 here.]

[Table24: insert table24 here.]

[Table25: insert table25 here.]

[Table26: insert table26 here.]

[Table27: insert table27 here.]

[Table28: insert table28 here.]

[Table29: insert table29 here.]

[Table30: insert table30 here.]

[Table31: insert table31 here.]

It is possible that all producers are able to upload any module for their customers in order to create more complications in problem-solving process. After solving the problem using the proposed algorithm, the convergence diagram is presented in figure9.

[figure9: insert figure 9 here.]

As observed, the SIF optimizer was suitably converged toward the optimal solution, and the mean of solutions matched the best solution in the final iterations. This shows that the

proposed algorithm has efficiently guided all members in search teams toward the final solution. The structure of the supply chain can also be described in the 32 to 35.

[Table32: insert table32 here.]

The supplier 1 with capacity level 1 and supplier 4 with capacity level 2 is selected to supply the products required by producers. In addition, producers 1, 2, and 4 with capacity level 1 and producer 6 with capacity level 3 are activated. Moreover, the collection centre 1 with capacity level 1, collection centre 3 with capacity level 2, and collection centres 2 and 4 with capacity level 3 are constructed to collect products used by customers. Furthermore, destruction centres 2, 3, and 4 with capacity level 2 are constructed to destroy the non-recyclable part of products. In the management of material flow, 151, 1211, and 84 units are delivered to producer 1, 2, and 3, respectively, by supplier 1. In addition, the supplier 4 sends 410 product units to producer 1. Information related to the flow of products from manufacturers to customers is also in accordance with the table 33.

[Table33: insert table33 here.]

As observed, all demands of each customer are met based on the flow of presented products, which shows the proper performance of the proposed model. The rate of return of products from customers to collection centres is also presented in the table 34.

[Table34: insert table34 here.]

As observed, the necessary allocations are made based on the parameter of the percentage of returned products and cost of transferring products between customers and each centre in order to conduct the product collection operations with the lowest cost possible. The table 35 shows the product flow between the collection centres and the suppliers, aiming at reusing the recycled materials obtained in the collection centres.

[Table35: insert table35 here.]

As observed, the total products delivered to suppliers are equal to the deduction of recyclable producers delivered by customers to collection centres, which makes the return of products into the production cycle possible. However, a part of the products is unrecyclable and is directly sent to destruction centres to carry out the destruction stages. The flow of products is presented in the table 36.

[Table36: insert table36 here.]

6. Conclusions

This article evaluates a new SIF-based structure to present a new optimal problem solution approach as a metaheuristic algorithm. The mathematical structure of the algorithm involves determining a specific number of groups in the form of teams, each encompassing a number of search members. Two different types of search are carried out in each iteration of the algorithm; the first type is an intra-group search, which is adapted from the organizing structure of search teams to accurately inspect the region, and the second type is local search of all teams in the solution space in order not to cover the entire solution space. This guarantees the effective and parallel assessment of the whole solution space. One of the advantages of the present research is the exploitation of parallel search operators in the algorithm structure, which leads to a proper balance in the diversification and contemplation phases of the algorithm. In addition, standard functions presented in two unimodal and multimodal modes are used to evaluate the efficiency of the proposed algorithm (Digalakis and Margaritis 2001; Molga and Smutnicki 2005). According to the numerical results, the SIF optimizer yielded better results in all numerical representations, compared to GA and ALO. Nevertheless, SIF optimizer yielded more fit results only in F8, F9, F11, and F12 representations, compared to GWA. In fact, since GWA is currently one of the best algorithms used in solving optimization problems, SIF optimizer failed to defeat GWA but was able to produce very close solutions in all numerical representations. This shows that SIF optimizer can also be used in optimization problems as an efficient algorithm. Moreover, SIF optimizer yielded far better solutions in solving CEC2014 standard numerical representations, compared to the other algorithms. Furthermore, according to the result obtained from supply chain design problem solving, the algorithm developed in the present research seemed to effectively find optimal solutions in the solution space and converge toward the global optimization solution when solving some practical issues in the field of industrial engineering. As such, the algorithm is proposed to solve optimization problems in various engineering fields. In order to expand the dimensions of research, it is recommended that multi-purpose SIF optimizer be developed, and numerical results obtained from solving various optimization problems be compared to the existing algorithms.

Conflict of interest statement

To: Professors Antonio Di Nola and Raffaele Cerulli

Editors of the soft computing journal

The authors of the manuscript declare that they have completely avoided publishing ethics, including plagiarism, misconduct, data forgery, or double publishing, have been avoided. This is no commercial interests in this article, and the authors have not received any payment for their manuscript. The authors also claims that this manuscript has not been published elsewhere and has not been published in another journal. All rights to use the content, tables, images, etc. have been assigned to the publisher.

Yours sincerely



Atefeh amindoust

Director of research in the school of engineering

Assistant Professor of operation research

*Department of Industrial Engineering, Najafabad Branch, Islamic Azad University, Najafabad,
Iran*

Email: Atefeh_Amindoust@yahoo.com

Tel: +983142292470, +989133696729

Author contributions:

Amin Ahwazian, Atefeh Amindoust, Reza Tavakkoli-Moghaddam, Mehrdad Nikbakht

Study concept and design, Acquisition of data, analysis and interpretation of data, drafting of the manuscript, statistical analysis, Critical revision of the manuscript for important intellectual content: Amin Ahwazian. Supervised the research: Atefeh Amindoust, Reza Tavakkoli-Moghaddam. Research Consultant Mehrdad Nikbakht.

References

- Abbassi R, Abbassi A, Heidari AA, Mirjalili S (2019) An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models. *Energy Conversion and Management* 179:362-72
- Back T (1996) *Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, and genetic algorithms*: Oxford University Press.
- Barisal A (2013) Dynamic search space squeezing strategy based intelligent algorithm solutions to economic dispatch with multiple fuels. *International Journal of Electrical Power & Energy Systems* 45:50-9
- Beamon BM (1998) Supply chain design and analysis: Models and methods. *International Journal of Production Economics* 55:281-94.
- Blackwell TM (2005) Particle swarms and population diversity. *Soft Computing* 9:793-802
- Brownlee J (2011) *clever algorithms: Nature-inspired programming recipes*, Published by Jason Brownlee, Melbourne, Australia
- Chen S, Chen R, Wang G-G, Gao J, Sangaiah AK (2018) An adaptive large neighborhood search heuristic for dynamic vehicle routing problems. *Computers & Electrical Engineering* 67:596-607
- Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation* 6:58-73
- Cui Z, Sun B, Wang G, Xue Y, Chen J (2017) A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber-physical systems. *Journal of Parallel and Distributed Computing* 52:42-17
- Cui Z, Xue F, Cai X, Cao Y, Wang G-g, Chen J (2018) Detection of malicious code variants based on deep learning. *IEEE Transactions on Industrial Informatics* 14:3187-96
- Davis L. (1991) *Handbook of genetic algorithms*.
- Digalakis JG, Margaritis KG (2001) On benchmarking functions for genetic algorithms. *International journal of computer mathematics* 77:481-506.
- Dorigo M, Birattari M (2010) *Ant colony optimization*: Springer.
- Dréo J, Pétrowski A, Siarry P, Taillard E (2006) *Metaheuristics for hard optimization: methods and case studies*: Springer Science & Business Media.

- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. MHS'95 Proceedings of the Sixth International Symposium on Micro Machine and Human Science: IEEE pp 39-43.
- Eddalya M, Jarbouia M, Siarryba P (2016) Combinatorial particle swarm optimization for solving blocking flow shop scheduling problem. *Journal of Computational Design and Engineering* 3: 295–311.
- Emara HM, Fattah HA (2004) Continuous swarm optimization technique with stability analysis. *Proceedings of the American control conference: IEEE* pp 2811-7.
- Faramarzi A, heidarnejad M, Mirjalili M, Gandomi A.H (2020) Marine Predators Algorithm: A Nature-inspired Metaheuristic. *Expert Systems With Applications* 152
- Faramarzi A, heidarnejad M, Stephens B, Mirjalili M (2020) Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems* 191
- Faris H, Ala'M A-Z, Heidari AA, Aljarah I, Mafarja M, Hassonah MA, et al (2019) An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. *Information Fusion* 83:67-42
- Feng Y, Wang G-G, Li W, Li N (2018) Multi-strategy monarch butterfly optimization algorithm for discounted {0-1} knapsack problem. *Neural Computing and Applications* 30:3019-36
- Feng Y, Yang J, Wu C, Lu M, Zhao X-J (2018) Solving 0–1 knapsack problems by chaotic monarch butterfly optimization algorithm with Gaussian mutation. *Memetic Computing* 10:135-50
- García-Gonzalo E, Fernández-Martínez JL (2014) Convergence and stochastic stability analysis of particle swarm optimization variants with generic parameter distributions. *Applied Mathematics and Computation* 249:286-302
- Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. *Simulation* 76:60-8
- Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. *Machine Learning* 3: 95-99
- Kadirkamanathan V, Selvarajah K, Fleming PJ (2006) Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Transactions on Evolutionary Computation* 10:245-55
- Kennedy J (2003) Bare bones particle swarms. *Proceedings of the IEEE Swarm Intelligence Symposium SIS'03 (Cat No 03EX706): IEEE* pp. 80-7
- Kennedy J (2000) Stereotyping: Improving particle swarm performance with cluster analysis. *Proceedings of the Congress on Evolutionary Computation CEC00 (Cat No 00TH8512): IEEE* pp. 1507-12
- Kennedy J, Eberhart R (1995) Particle swarm optimization (PSO). Paper presented at the Proc. IEEE International Conference on Neural Networks, Perth, Australia.
- Koyuncu H, Ceylan R (2018) A PSO based approach: Scout particle swarm algorithm for continuous global optimization problems, *Journal of Computational Design and Engineering* 6: 129–142
- Li C, Yang S, Nguyen TT (2011) A self-learning particle swarm optimizer for global optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 7:62-42
- Li X, Qian J, Wang G-g (2013) Fault prognostic based on hybrid method of state judgment and regression. *Advances in Mechanical Engineering* 5:1495-62
- Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software* 114:163-9
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Advances in Engineering Software* 69:46-61

- Mirjalili, Seyedali (2015) The Ant Lion Optimizer. *Advances in Engineering Software* 83: 80–98
- Molga M, Smutnicki C (2005) Test functions for optimization needs. *Test functions for optimization needs* 101
- an X, Bao L, Zhao X, Zhao X, Sangaiah A, Wang G-G, et al. (2017) EPuL: an enhanced positive-unlabeled learning algorithm for the prediction of population sites. *Molecules* 22:1463
- Pandey, A. Singh Panwar V. Ehtesham Hasan, M and R Parhi (2020) DV-REP-based navigation of automated wheeled robot between obstacles using PSO-tuned feed forward neural network. *Journal of Computational Design and Engineering* 7:427–434
- Park J-B, Jeong Y-W, Shin J-R, Lee KY (2009) An improved particle swarm optimization for nonconvex economic dispatch problems. *IEEE Transactions on Power Systems* 25:156-66
- Peng C-C, Chen C-H (2015) Compensatory neural fuzzy network with symbiotic particle swarm optimization for temperature control. *Applied Mathematical Modelling* 39:383-95
- Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization. *Swarm intelligence* 1:33-57
- Poli R (2008) Dynamics and stability of the sampling distribution of particle swarm optimizers via moment analysis. *Journal of Artificial Evolution and Applications* 15
- Poli R (2009) Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. *IEEE Transactions on Evolutionary Computation* 13:712-21
- Rashidi P, Cook DJ (2009) Keeping the Resident in the Loop: Adapting the Smart Home to the User. *IEEE Trans Systems, Man, and Cybernetics, Part A* 39:949-59
- Rizk-Allah RM, El-Sehiemy RA, Deb S, Wang G-G (2017) A novel fruit fly framework for multi-objective shape design of tubular linear synchronous motor. *The Journal of Supercomputing* 73:1235-56
- Selma B, Chouraqi S, Abouaïssa H (2020) Fuzzy swarm trajectory tracking control of unmanned aerial vehicle. *Journal of Computational Design and Engineering* 7:435–447
- Shi Y, Eberhart R (1998) A modified particle swarm optimizer. *IEEE international conference on evolutionary computation proceedings IEEE world congress on computational intelligence (Cat No 98TH8360): IEEE pp. 69-73*
- Srikanth K, Panwar LK, Panigrahi BK, Herrera-Viedma E, Sangaiah AK, Wang G-G (2018) Meta-heuristic framework: quantum inspired binary grey wolf optimizer for unit commitment problem, *Computers & Electrical Engineering* 70:243-60
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11:341-59
- Suganthan PN (1999) Particle swarm optimizer with neighborhood operator. *Proceedings of the Congress on Evolutionary Computation-CEC99 (Cat No 99TH8406): IEEE pp. 1958-62*
- Talatahari S, Azar BF, Sheikholeslami R, Gandomi A (2012) Imperialist competitive algorithm combined with chaos for global optimization. *Communications in Nonlinear Science and Numerical Simulation* 17:1312-9
- Talbi E-G (2009) *Metaheuristics: From design to implementation*: John Wiley & Sons
- Tanweer MR, Suresh S, Sundararajan N (2016) Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems. *Information Sciences* 326:1-24.
- Van Den Bergh, F (2001) *An analysis of particle swarm optimizers*. University of Pretoria South Africa
- Wang G, Guo L, Duan H, Liu L, Wang H, Shao M (2012) Path planning for uninhabited combat aerial vehicle using hybrid meta-heuristic DE/BBO algorithm. *Advanced Science, Engineering and Medicine* 4:550-64

- Wang G-G, Chu HE, Mirjalili S (2016) Three-dimensional path planning for UCAV using an improved bat algorithm. *Aerospace Science and Technology* 49:231-8
- Wang G-G, Deb S, Gandomi AH, Zhang Z, Alavi AH (2016) Chaotic cuckoo search. *Soft Computing* 20:334-296
- Wang G-G, Gandomi AH, Alavi AH (2014) An effective krill herd algorithm with migration operator in biogeography-based optimization. *Applied Mathematical Modelling* 38:2454-62
- Wang G-G (2018) Moth search algorithm: a bio-inspired meta-heuristic algorithm for global optimization problems. *Memetic Computing* 10:151-64
- Wang H, Yi J-H (2018) An improved optimization method based on krill herd and artificial bee colony with information exchange. *Memetic Computing* 10:177-98
- Wang L, Fu X, Mao Y, Menhas MI, Fei M (2012) A novel modified binary differential evolution algorithm and its applications. *Neurocomputing* 98:55-75
- Wang L, Zheng X-l (2018) A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem *Swarm and Evolutionary Computation* 38:54-63
- Wang G-G, Bai D, Gong W, Ren T, Liu X, Yan X (2018) Particle-swarm Krill Herd Algorithm. Paper presented at the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)
- Wang G-G, Deb S, Coelho L.D.S (2015) Earthworm optimization algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *International Journal of Bio-Inspired Computation* 7:1-23
- Wang G-G, Deb S, Gao X-Z, Coelho L.D.S A new metaheuristic optimisation algorithm motivated by elephant herding behaviour. *International Journal of Bio-Inspired Computation* 8(6): 394-409
- Wu G, Pedrycz W, Suganthan PN, Mallipeddi R. (2015) A variable reduction strategy for evolutionary algorithms handling equality constraints. *Applied Soft Computing* 37:774-86.
- Yang X-S, Gandomi AH, Talatahari S, Alavi AH (2012) *Metaheuristics in water, geotechnical and transport engineering*, Newnes, Elsevier
- Yang X-S (2008) *Introduction to mathematical optimization: From linear programming to metaheuristics*. Cambridge International Science Publishing Ltd, Cambridge, U.K
- Yang, X.-S (2010) *Nature-inspired metaheuristic algorithms*: L univier press
- Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation* 3:82-102
- Yi J-H, Deb S, Dong J, Alavi AH, Wang G-G (2018) An improved NSGA-III Algorithm with adaptive mutation operator for big data optimization problems. *Future Generation Computer Systems* 88:571-85
- Yi J-H, Wang J, Wang G-G (2016) Improved probabilistic neural networks with self-adaptive strategies for transformer fault diagnosis problem. *Advances in Mechanical Engineering* 8:1687814015624832
- Yi J-H, Xing L-N, Wang G-G, Dong J, Vasilakos AV, Alavi AH, et al. (2020) Behavior of crossover operators in NSGA-III for large-scale optimization problems. *Information Sciences* 509:87-47
- Zhan Z-H, Zhang J, Li Y, Shi Y-H (2010) orthogonal learning particle swarm optimization. *IEEE Transactions on Evolutionary Computation* 15:832-47
- Zhou, C, GAO H.B, Gao L , ZHANG W.G (2003) Particle Swarm Optimization (PSO) Algorithm [J]. *Application Research of Computers* 12: 7-11

- Zou D, Li S, Wang G-G, Li Z, Ouyang H (2016) An improved differential evolution algorithm for the economic load dispatch problems with or without valve-point effects. *Applied Energy* 181:375-90
- Zou D, Wang G-G, Sangaiah AK, Kong X (2017) A memory-based simulated annealing algorithm and a new auxiliary function for the fixed-outline floor planning with soft blocks. *Journal of Ambient Intelligence and Humanized Computing* 1-12
- Zou D-X, Deb S, Wang G-G (2018) Solving IIR system identification by a variant of particle swarm optimization. *Neural Computing and Applications* 30:685-98
- Zou D-x, Wang G-g, Pan G, Qi H-w (2016) A modified simulated annealing algorithm and an excessive area model for floor planning using fixed-outline constraints. *Frontiers of Information Technology & Electronic Engineering* 17:1228-44

Figures

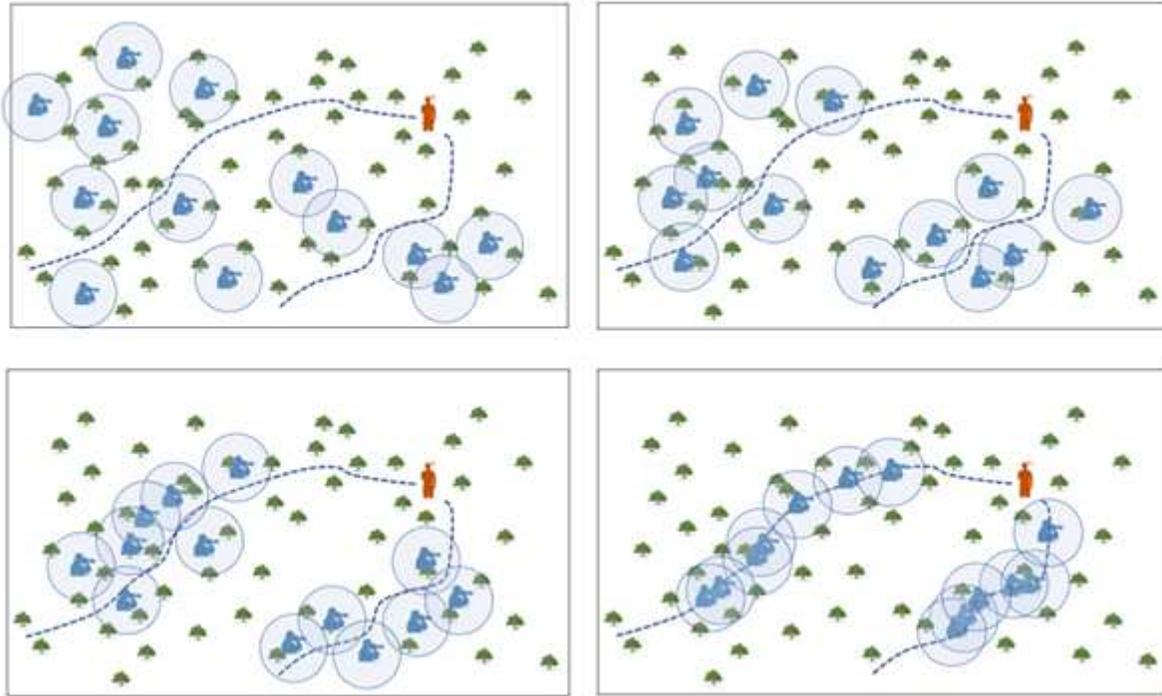


Figure 1

Convergence of search teams to find targets in the forest

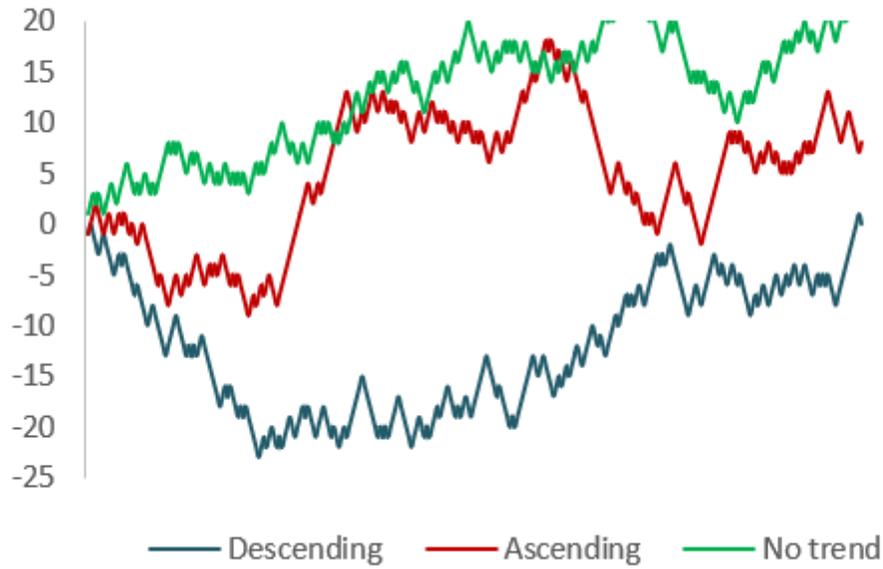


Figure 2

Movement structure of search teams in the solution space

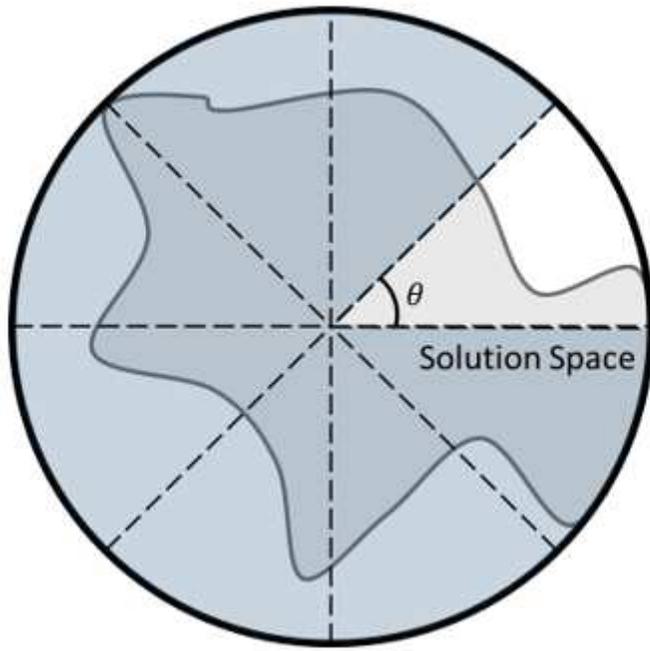


Figure 3

Division of solution space based on the number of search teams

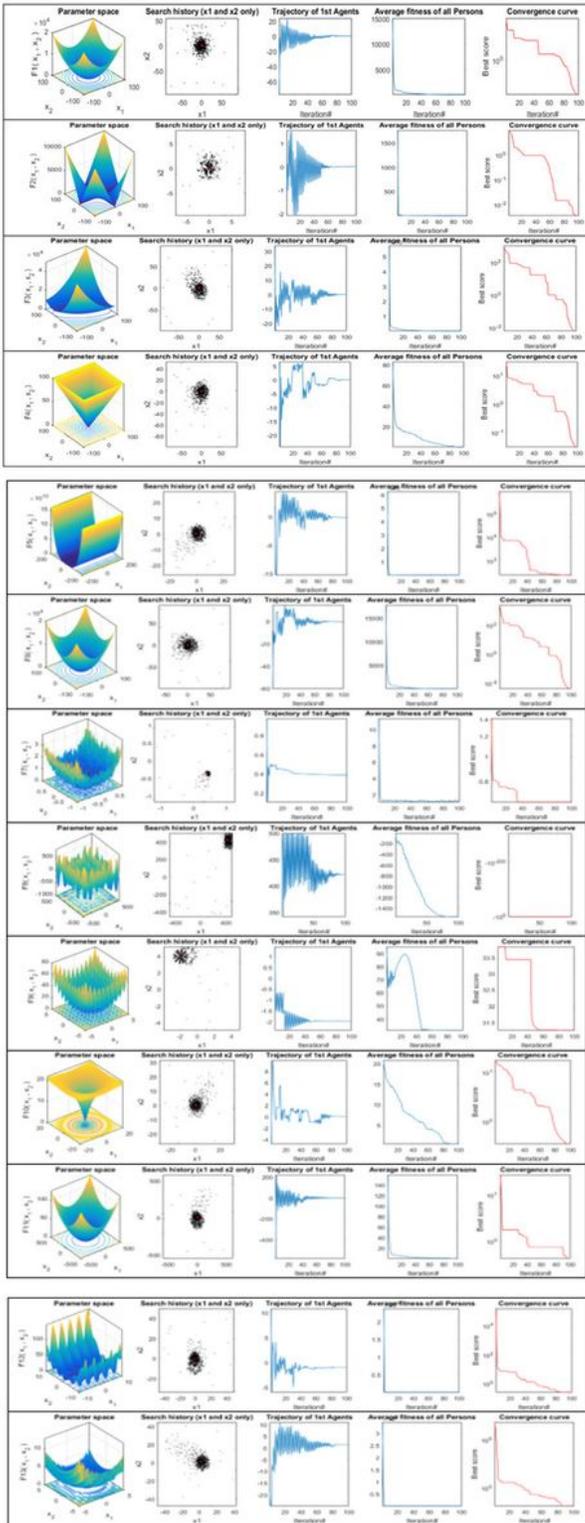


Figure 4

Behavior of the proposed algorithm in solving the standard numerical representation

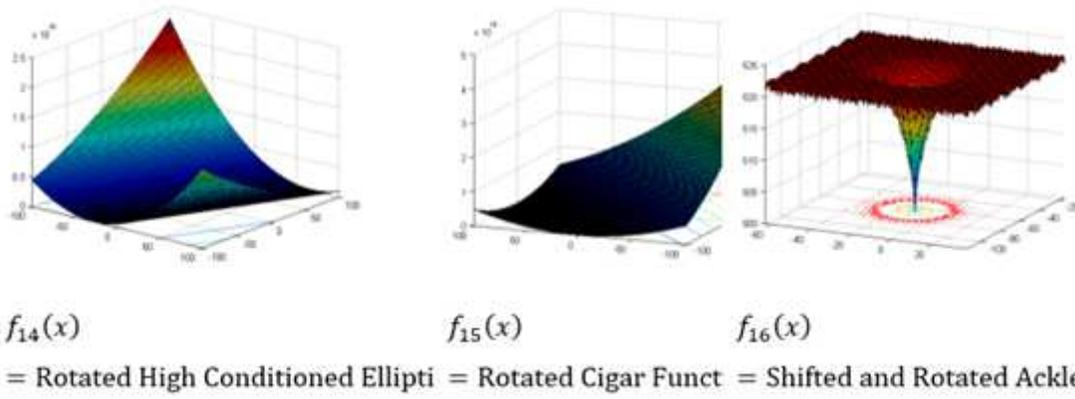


Figure 5

3D structure of F14, F15, and F16 functions

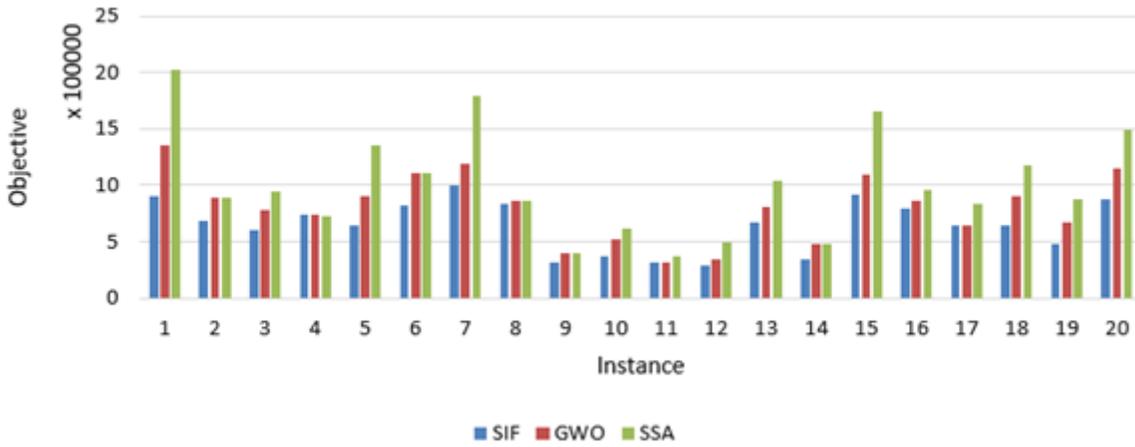


Figure 6

Comparison of target function values of GWA, SSA, and SIF

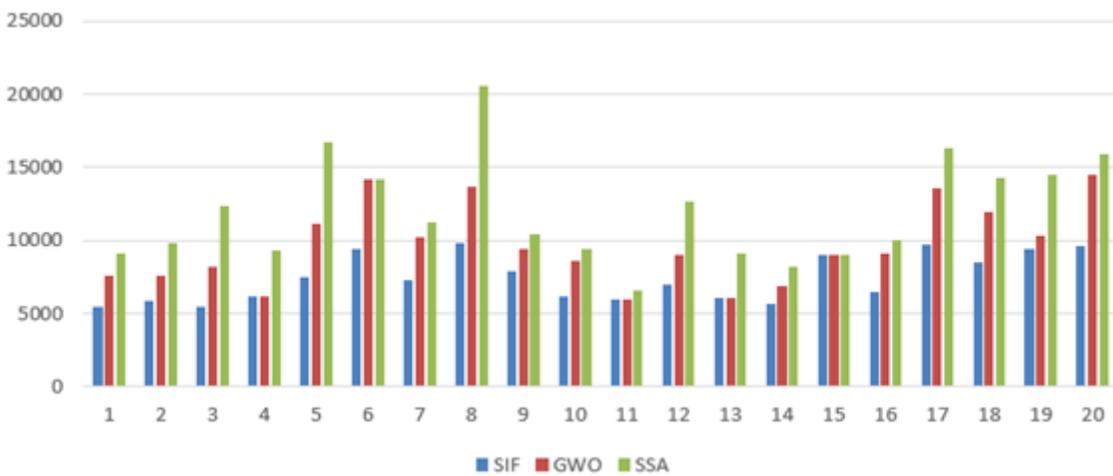


Figure 7

Comparison of deviation amounts of GWA, SSA, and SIF

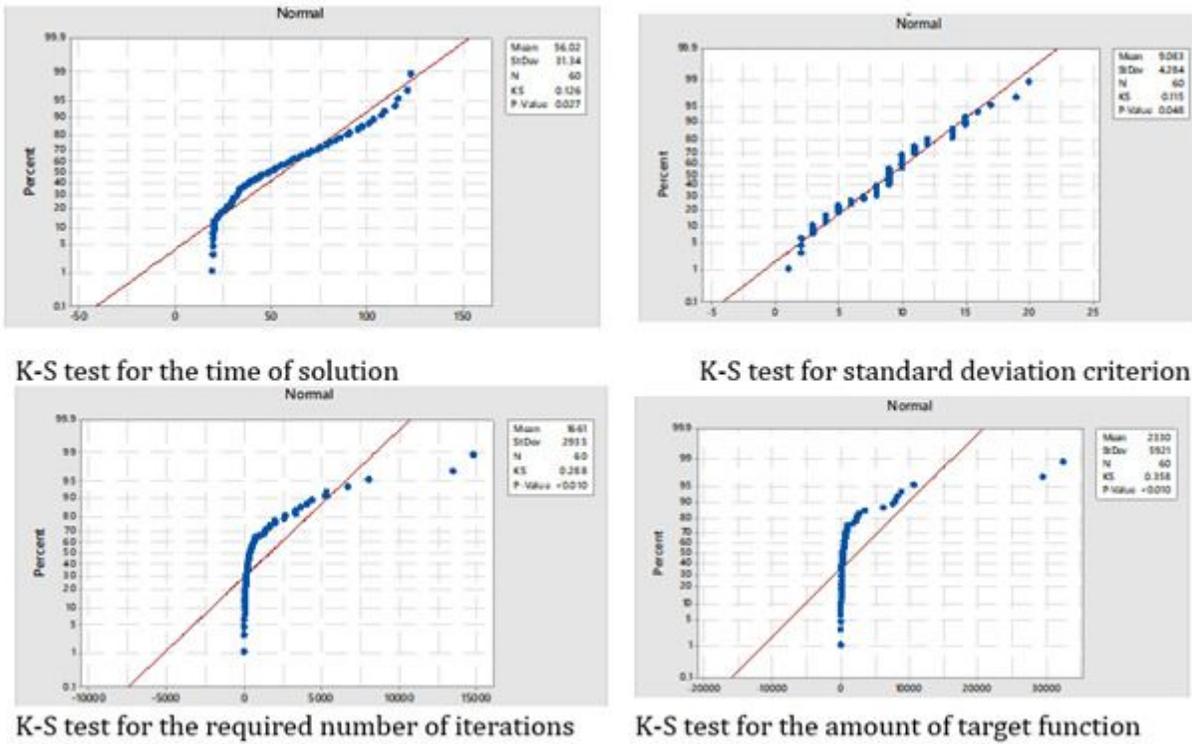


Figure 8

K-S test diagrams for various criteria

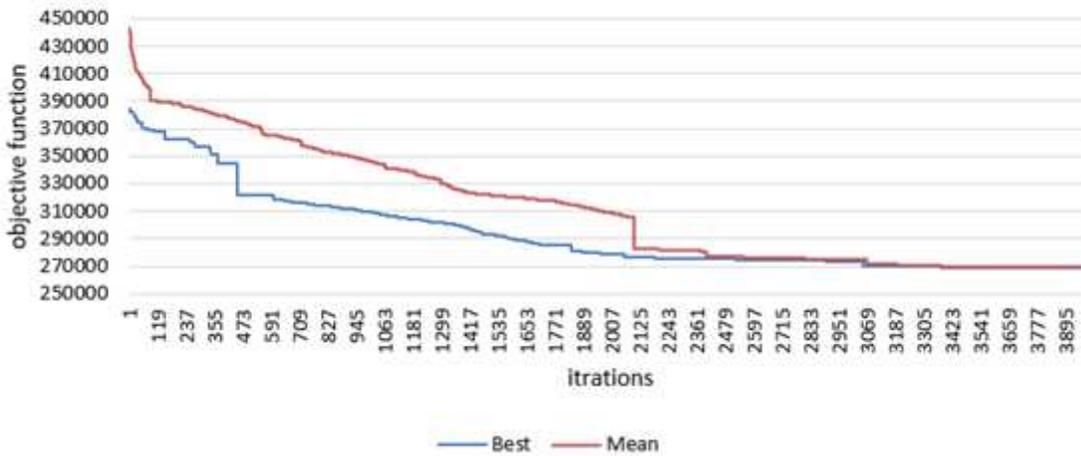


Figure 9

Diagram of convergence obtained from supply chain design problem-solving

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Tables.pdf
- SIF.rar