

A Novel Swarm Urochordate Algorithm for Defensive Resource Assignment Problems

Kusum Yadav

University of Hail

Gaurav Dhiman (✉ gdhiman0001@gmail.com)


GBC <https://orcid.org/0000-0002-6343-5197>

Research Article

Keywords: Defence, Optimization, Swarm, WTA.

Posted Date: August 31st, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-850853/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License. [Read Full License](#)

Abstract

The weapon target assignment (WTA) problem is an important task to tactical arrangements in military commitment operations. It describes the optimal method to allocate defenses in opposition to threats in fighting situations. It is an NP-complete issue in which no accurate outcome for all conceivable situations is known. The time performance of created algorithms is a major challenge in modeling the WTA problem, which has only been lately considered in related papers. This article introduces a new algorithm called Swarm Urochordate Algorithm (SUA) which is inspired nature by tunicates to solve the WTA problem. The suggested method is compared to nine metaheuristic approaches recently established for 30 well-known testing benchmarks. Convergence and computer complexity are also examined. The experimental findings show that the method presented works better than previous competing metaheuristic approaches.

1. Introduction

Several previous obstacles to access control over a large number of technical advancements in the sector of military, such as equipment, arm as well as military automobiles, have not been sufficient to avoid the conflicts. The risk reduction in this uncertainty situation is not simple, raises interest in decision support systems which may improve defense outcomes [1–5]. However, defense resources must be deployed with strict criteria, ideally at minimal cost, in order to reduce damage on the protecting area.

Tactical planning, especially for computer-based systems, is a tough job because of many potential situations. Air defense techniques are complex because aerial targets have a high degree of mobility and are able to take very quick and tough choices. Special algorithms have thus been created to tackle WTA issues, which while at times failing to provide the optimal solutions overall, provide a high quality distribution that reduces the danger stage in global world. In this article, we propose an approach named Swarm Urochordate Algorithm (SUA) for the designing of WTA strategy. Good quality answers were achieved even for big situations in testing in a regulated algorithm response time.

The issue of assignment is an example of Waffren-Target Assignment (WTA) problem. Given two different sets, of resources and prey, the waffen problem is used for allocating each of the resources to destination that minimizes the endurance of the commitments. It also maximizes the asset endurance after an adversary attack, thus finding the optimal method to assign the present resources at destination in defense planned scenarios. The authors [6–7] discussed two distinct defense models, an offensive approach and a benefit-based one. The benefit-based model includes planned incensed defense, which requires information on what objectives, while the offensive-based model applicable to that plans which are defined in [30]. Each incoming target gets a value in the target-based model that sums up its degree of danger. If assets has plan, it forms group that can have risk related to damage, a mathematical estimate of the chance that assets may cancel the plan. The danger stage decreases to the anticipated endurance value, representing the assets harmful probability impact on the warning stage [8–15]. For this model, the waffen goal is to reduce the total anticipated value of the objectives surviving all commitments by following a commitment. The development of operations over time is another feature of the WTA model. The static model dubbed the SWTA issue assumes that defensive operations develop in one step. The issue of dynamic weapons target assignment (DWTA) takes many defensive cycles into account during in progress defense operations.

2. Optimization Algorithms Related To Waffren-target

The clear listing approach would provide a worldwide answer to this issue if we could check all potential combinations of arms in a particular situation in an acceptable amount of time. The SWTA issue is a NP-complete problem, which means that a comprehensive explore did not a feasible technique for the questions arisen. Methods that are heuristic, should important because in a SUA period they may offer almost optimum answers. In the middle of the last century, first investigations of the WTA issue emerged in the area of operations research [31–38]. Classical optimization techniques at that time include implied listing, branch and bounding algorithms, integral linear as well as dynamic implementation of algorithms [13]. The author [2] presented the MMR method to the WTA solution. This greedy method is only optimal for a certain kind of WTA issue because it assumes the likelihood of loss to be resource independent. Researcher [37] has improved the MMR with the findings that have been demonstrated in previous works [11]. Naturally inspired heuristic techniques have led to a fresh generation of ideas enhancing WTA issue modeling. The author [36] incorporated dominion-particular information in the crisscross method and a limited search technique based on genetic algorithms (GA). They suggested a hungry eugenic contribution to improved outcomes. Other studies [16–30] focused on hybridizing several bio inspired techniques, integrating them in order to enhance the outcomes by bringing together the benefits provided by each. The optimization related to particle swarm is an algorithm for grouping techniques that were integrated in [39], enhancing PSO locally and globally. The work related to WTA problem solving techniques for PSO and GA is also decribed in [40]. The combination of ACO as well as GA techniques was explored in [41] and in [33]. While these findings significantly reduce the initial degree of danger in battle situations, computing time may be prohibitive as the issue size increases. The decision to take an air danger defense outline needs a response time of one to two sec [10]. Many studies have results produced for hours following operating methods and compromise its practical uses [34–36]. The findings produced by [11], [12] revealed the two distinct kinds of GA, twine-implementation and group-implementation GA. The chromosome in group-implementation GA is the resource order followed by an MMR algorithm process [12][37]. Every GA Algorithm was run 30 different variants of seed and unseeded with 15 stochastic created situations. Seed types of both GA algorithms carry out better as compared to unseeded types, although the GA strings were better suited to aerial situations involving 40 resources and 80 targets in big scenarios. However, the testing was restricted to one instance for every outline size. The authors [9–10] examined heuristic determinisms and non determinisms for WTA solutions [39–42]. Various algorithms calculated the optimal values throughout iterative search in one second; however the experiments were restricted to outlines with up to 30 assets as well as 30 objectives. The authors [1] presented a partition-based Waffren-Target problem creation method in which a neighborhood search technique for very high scale was utilized to handle nonlinear optimizer issues. The research applied a novel version of the Ant Colony System (ACS) [3] to the WTA issue by introducing a simultaneous multi-colony approach for improving performance. GACS has produced excellent outcomes over a fair period of time, including big outlines with up to 80 assets as well as 80 objectives.

3. Heuristic Techniques

Algorithms related to greedy address optimizer issues by picking the option that best matches the target function at that time by finding an optimum local solution for iterations. For comparative reasons, we utilized the gullible technique [37], here called the MMR2, since it is widely known that it produces superior results [11] from [2][3], and called here MMR1. Both methods contain vector V of prey values as well as P (matrix) to die probability as input parameters, and vector S as output. Unlike MMR1, outcome of MMR2 independently searches all prey values as well as assets to find group that presently optimizes threat value reduction. As with all such algorithms, the target value will decrease for the following iteration after assigning a resource to one objective, lowering the danger level defined with the prey. MMR algorithms are extremely fast to give answers, even for huge situations. MMR2 achieves the optimal global optimum more often and is highly successful for non-complex input situations.

4. Swarm Urochordate Algorithm (Sua)

Input

Initialize Population value.

Output

Optimal calculation.

Step 1: Procedure Tunicate technique

Step 2: Introduce different vectors and extreme iterations.

Step 3: Put value of population as 2.

Step 4: Put extreme iteration as 3.

Step 5: The swarm value should be used NIL in the beginning.

Step 6: The while do loop SUA be used in order to go to extreme iterations.

Step 7: For loop is then used for increment swarm values.

Step 8: Now compute the fitness value.

Step 9: The bounded value is used for further calculations in this step.

Step 10: Here the different formulas are used to calculate the force and Swarm behaviour value.

Step 11: Then the random value is being compared in if loop.

Step 12: If loop will end when all the bounded values are used.

Step 13: Then the best optimal solution is calculated using if, for loop and BEST function is used.

Step 14: After lot of iterations the best value is calculated.

Step 15: The return is used at the end.

Step 16: The approach will end when the final optimal values are received.

5. Swarm Urochordate Algorithm (Sua) For Waffen-target Application

The overall heuristics rule of SUA is similar as the defined in Fig. 1. SUA method uses a concurrent approach which improves search and optimises the CPU time. MMR2 finds the optimal pair for every interaction in which the pairings are not sorted according to the original resource sequence. Therefore, we investigated if the search order in step four of the SUA algorithm was changed as defined in Fig. 1. We utilised the same structured graph order of the resource sequence generated by the MMR2 algorithm in the SUA graph instead of following the provided order. The findings indicated that the target survival values were reduced by this design, although slight.

The approach proposes an additional method for limiting repeats of the best values inside each population sequential algorithm, preventing premature convergence. The most used routes get a penalty approach that reduces their appeal by stopping the criteria after a specific time of CPU. This initially speeds up the convergence so that a local optimum may be taken advantage. We decided to generate the optimal value frequently at CPU time. If the CPU time is more than the defined time, the SUA algorithm is reset to their original exploratory stimulation value. The best answers are stored in a memory list to ensure that they are not lost in evolution.

6. Experimental Outcomes And Discussion

In this evaluation, thirty common benchmark tests are used to evaluate the presentation of the developed method.

6.1. Test benchmarking and comparison algorithms

The twenty-nine test formulas (F1 – F29) are used in this work. Eight widely known algorithms are selected for comparison in order to verify the performance of the proposed method. The search agent count for each algorithm is 30. It is noted that there are 30 search agents who are reasonable to solve optimization issues due to the increased number of search representatives, the greater the chance of finding the optimum in global point.

Table 1
Competitor algorithm parameters values

Technique	Numerical	point
TSA	Exploring Factor	79
	Extreme Value	0–3
	Genetically numbering	999
SUA	Exploring Factor	79
	Extreme Value	0–4
	Genetically numbering	999
GWO	Exploring Factor	79
	Extreme Value	1–3
	Genetically numbering	999
PSO	Exploring Factor	79
	Inertia value	.73
	Extreme Value	1-1.6
	Genetically numbering	999
MVO	Exploring Factor	79
	Extreme Value	0–3
	TD	0-4
	Genetically numbering	999
SCA	Exploring Factor	79
	Elite Value	2
	Genetically numbering	999
GSA	Exploring Factor	79
	Extreme Value	0–3
	Alpha Value	18
	Genetically numbering	999
GA	Exploring Factor	79
	Extreme Value	0–3
	Genetically numbering	999
EPO	Exploring Factor	79
	Temp	0-999
	Constant value used	-1.6-1.6
	Cross value	0–3
	Genetically numbering	999

6.2. Experimental installation.

Table 1 lists the parameters of the proposed SUA as well as other metaheuristic techniques named GWO, SCA, GSA, GA, PSO, MFO, MVO, and HS. All these characteristics are defined as per literature provided. It utilized the similar assigning approach in order to differentiate SUA with the metaheuristic methods described previously. The tests and methods are implemented in the version 8.3.0.532 of Matlab R2014a, as well as are implemented in the environment of

Microsoft Windows 8.1, 64 bits on a 2.40 GHz and 4GB Core i-5 CPU. The mean and standard deviation of optimal outcome achieved until final iteration SUA be calculated as presentation tables. To produce as well as present results, SUA method uses 30 individual runs for each benchmark function with 1000 times of iteration each run.

6.3. Comparison of performance

6.3.1. Function Calculation F1 to F7 (exploitation)

F1 – F7 functions are uni-modal and enable the utilization of the metaheuristic based algorithms to be assessed. Table 2 and Fig. 2 indicate that SUA is very cut-throat in relation to next techniques mentioned. SUA has been the most powerful algorithm for F1 to F7 functions, in particular as well as finds the best solution to assess other metaheuristic methods. The SUA algorithm can best be used.

Table 2
Uni-modal benchmark results

Fun	TSA		SUA		GWO		PSO		MVO		SCA		GSA		GA		EPO	
	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St
		Di		Di		Di		Di		Di		Di		Di		Di		Di
F1	7.70	7.01	0.01	0.01	4.58	7.36	4.96	1.39	2.90	1.11	1.7	1.16	6.1	2.01	2.01	8.0	7.99	3.5
F2	8.46	5.90	0.01	0.01	1.17	1.28	7.27	1.82	3.97	1.41	8.5	1.70	8.99	6.66	4.9	6.1	.99	3.2
F3	1.16	6.69	0.01	0.01	1.01	4.07	1.41	7.12	4.27	9.01	4.0	4.01	1.25	8.01	7.3	9.0	6.01	5.01
F4	1.30	1.14	7.73	8.94	2.00	2.40	5.58	1.70	8.78	1.11	8.21	1.02	10.0	10.0	5.99	9.1	1.1	2.11
F5	5.11	4.67	8.54	5.50	2.75	1.82	3.37	3.86	1.16	1.41	2.11	4.12	3.5	5.99	3.98	9.2	3.1	7.12
F6	7.07	1.11	2.44	1.76	6.53	3.36	9.21	1.76	3.12	9.01	10.1	.99	3.99	3.10	10.1	8.1	2.01	5.11
F7	3.71	5.06	3.26	2.40	7.76	3.81	6.90	2.84	2.05	3.11	5.99	8.01	3.01	6.59	3.12	5.6	2.11	4.21

6.3.2. Function calculation F8 to F21 (exploration)

Multimodal formulas contain numerous regional optima that rapidly grow. Testing tasks may assess the optimization algorithm exploration. The findings for F8 – F21 are SUA in Tables 3 and 4 which may suggest that SUA has excellent exploration capabilities. The Figs. 3 and 4 indicated that in seven testing issues (i.e. F9, F11, F15, F16, F17, F18, F21). SUA is effective and competitive in other experiment difficulties as well. Indeed, in most of these test situations the SUA method is the most efficient approach. These findings indicate that the exploratory SUA algorithm is worthwhile.

Table 3
Multimodal benchmark results

Fun	TSA		SUA		GWO		PSO		MVO		SCA		GSA		GA		EPO	
	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St
		Di		Di		Di		Di		Di		Di		Di		Di		Di
F8	-4.6	7.3	-4.9	2.6	3.1	6.01	4.96	1.39	2.90	1.11	1.7	1.16	6.1	2.01	2.01	8.0	7.99	3.5
F9	2.2	1.3	7.3	3.5	3.4	2.1	7.27	1.82	3.97	1.41	8.5	1.70	8.99	6.66	4.9	6.1	.99	3.2
F10	3.1	4.00	1.4	8.99	4.10	3.99	1.41	7.12	4.27	9.01	4.0	4.01	-4.6	7.3	-4.9	2.6	3.1	5.01
F11	8.3	2.41	6.00	2.99	7.0	.99	5.58	1.70	8.78	1.11	8.21	1.02	2.2	1.3	7.3	3.5	3.4	2.11
F12	2.5	2.71	7.12	2.01	7.12	4.01	3.37	3.86	1.16	1.41	2.11	4.12	3.1	4.00	1.4	8.99	4.10	7.12
F13	2.50	3.3	4.99	8.54	7.55	1.89	9.21	1.76	3.12	9.01	10.1	.99	8.3	2.41	6.00	2.99	7.0	5.11
F14	-4.6	7.3	-4.9	2.6	3.1	6.01	6.90	2.84	2.05	3.11	5.99	8.01	2.5	2.71	7.12	2.01	7.12	4.21

Table 4
Outcomes related to multimodal (benchmark) function of a given dimension

Fun	TSA		SUA		GWO		PSO		MVO		SCA		GSA		GA		EPO	
	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St
		Di		Di		Di		Di		Di		Di		Di		Di		Di
F15	6.01	4.96	1.39	2.90	1.11	1.7	1.16	6.1	2.90	1.11	1.7	1.16	6.1	2.01	2.01	8.0	7.99	3.5
F16	2.1	7.27	1.82	3.97	1.41	8.5	1.70	8.99	3.97	1.41	8.5	1.70	8.99	6.66	4.9	6.1	.99	3.2
F17	3.99	1.41	7.12	4.27	9.01	4.0	4.01	-4.6	4.27	9.01	4.0	4.01	-4.6	7.3	-4.9	2.6	3.1	5.01
F18	.99	5.58	1.70	8.78	1.11	8.21	1.02	2.2	8.78	1.11	8.21	1.02	2.2	1.3	7.3	3.5	3.4	2.11
F19	4.01	3.37	3.86	1.16	1.41	2.11	4.12	3.1	1.16	1.41	2.11	4.12	3.1	4.00	1.4	8.99	4.10	7.12
F20	1.89	9.21	1.76	3.12	9.01	10.1	.99	8.3	3.12	9.01	10.1	.99	8.3	2.41	6.00	2.99	7.0	5.11
F21	6.01	6.90	2.84	2.05	3.11	5.99	8.01	2.5	2.05	3.11	5.99	8.01	2.5	2.71	7.12	2.01	7.12	4.21

6.3.3. Function calculation F22 to F30

Optimization for benchmark functions related to composite is a highly cut-throat job as the balance among exploration as well as operation that may prevent the local issue of optima. Due to the huge amount of local optimum, local prohibition of a method may detect in these test formulas.

Table 5 and Fig. 5 indicates SUA method that is best and in other instances highly competitive optimizer for F26, F28 and F30 functions. The findings for F1-F30 functions indicate that the SUA technique is the most efficient compared to other optimization approaches.

Table 5: Composite benchmarking results

Fun	TSA		SUA		GWO		PSO		MVO		SCA		GSA		GA		EPO	
	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St	Mean	St
		Di		Di		Di		Di		Di		Di		Di		Di		Di
F22	3.1	4.96	1.39	2.90	1.11	1.7	1.16	6.1	2.90	1.11	1.7	1.16	6.1	2.01	2.01	8.0	7.99	3.5
F23	3.4	7.27	1.82	3.97	1.41	8.5	1.70	8.99	3.97	1.41	8.5	1.70	8.99	6.66	4.9	6.1	.99	3.2
F24	3.1	1.41	7.12	4.27	9.01	4.0	4.01	-4.6	4.27	9.01	4.0	4.01	-4.6	7.3	-4.9	2.6	3.1	5.01
F25	8.3	5.58	1.70	8.78	1.11	8.21	1.02	2.2	8.78	1.11	8.21	1.02	2.2	1.3	7.3	3.5	3.4	2.11
F26	2.5	3.37	3.86	1.16	1.41	2.11	4.12	3.1	1.16	1.41	2.11	4.12	3.1	4.00	1.4	8.99	4.10	7.12
F27	2.50	9.21	1.76	3.12	9.01	10.1	.99	8.3	3.12	9.01	10.1	.99	8.3	2.41	6.00	2.99	7.0	5.11
F28	-4.6	6.90	2.84	2.05	3.11	5.99	8.01	2.5	2.05	3.11	5.99	8.01	2.5	2.71	7.12	2.01	7.12	4.21
F29	1.89	9.21	1.76	3.12	9.01	10.1	.99	8.3	3.12	9.01	10.1	.99	8.3	2.41	6.00	3.11	5.99	8.01
F30	6.01	6.90	2.84	2.05	3.11	5.99	8.01	2.5	2.05	3.11	5.99	8.01	2.5	2.71	7.12	9.01	10.1	.99

6.4 Convergence analysis: The suggested SUA approach's convergence curves are given in Fig. 6. SUA is found to be quite competitive when compared to benchmark functions. The SUA approach includes three different convergence tendencies. SUA approach's convergence efficiency in a given search space is higher at the beginning of the iteration phase. When the last iteration is reached, it may be seen that SUA displays three distinct convergence tendencies while optimizing the test functions. The convergence behaviour can be seen from the early stage of iterations may be seen in the final iteration process. SUA approach establishes a balance between exploration and exploitation, according to the results. These findings demonstrate that the SUA method stabilized an exploration as well as exploitation balance in order to reach optimum. These section findings have various aspects of the developed SUA method as well as the succeed rate of SUA method seems to be max to solve optimized issues. Due to the difficulty of these test routines, this is the case. SUA explore full domain instead to rest in iterations but gives the complete solution. The sample points are distributed around the genuine optimal solution, ensuring that it can be exploited. As a result, SUA is capable of both finding and utilizing.

6. 6. Research into scalability: Using the proposed TSA, the influence of capability of iterations on different factors. The test functions extent is ranges from 30, 48, 78, and 99. The presentation of the TSA method on capable test functions is also SUA and it gives best results. If the capability of exploring factors is incremented or decremented the TSA won't suffer. The results of SUA, increase the dimensionality of the search space has the least impact on TSA performance.

a. Statistical analysis: The Analysis of Variance (ANOVA) test was used with respect to normal statistical analysis such as mean and standard deviation. The ANOVA test is used to see if the suggested approach's results are statistically significant different from those of competitors' approaches. With a 95%

confidence interval, the sample size for an ANOVA test is 30. A p value is used to establish whether or not an approach is statistically significant. If the related approach's p value is less than 0.05, the approach is statistically significant. As a result, the suggested TSA differs statistically from competing approaches. ANOVA tests were performed in addition to the basic mean and standard to compare the above described approaches. ANOVA testing was carried out. The ANOVA test was carried out to depart from the results of the proposed methods and determines whether the results of other competing methods differ statistically substantially. We have taken 30 ANOVA samples. ANOVA testing used 94.5% of the faith. Results reveal that, the suggested technique is statistically significant compared to other competitive approaches. TSA technique use different tasks to perform different techniques which gives a high approach to engineering.

7. Sua Related To Engineering

SUA was tested for constrained as well as one uncontrolled design problem named welded beam. The issue has distinct restrictions and we must use the restricted management technique to optimize these problems. There are several kinds of formula to address restriction problems [8]:

- The static penalty does not relay latest generated value and stays fixed throughout whole cycle of calculation.
- Dynamic penalty is calculation of the comparable penalty components in the present generation
- The coefficients related to penalty of rinsing are modified only one time throughout several iteration courses. Only active restrictions, which are not entangled in regional optima, are evaluated at iterations.
- The penalty related to adaptive is based on input from the previous search and changes only if the viable/infallible option in the population is deemed better.
- Co-evolutionary punishment divided in two numbers to determine which constraint is breached and the associated infringement amounts.
- Penalty related to death handles a solution that breaches restriction as well as gives a 0 fitness value. During optimization, the infeasible solutions may be discarded. However, the approach does not use knowledge about inviolable solutions to address the issues in dominated, unfeasible areas. Because of its simplicity and cheap calculative cost, in this part, the SUA method has the death penalty function to manage restrictions.

7.1. Design of the welded beam

The primary aim of this issue is to reduce price of manufacture of sold beam illustrated in Fig. 7 Optimized limitations include shear as well as bending stress, hitch spring balance in bar and end un reflecting of beam as SUAwn in fig..

The four optimized values for this problem are:

- Sulfur thickness (h) • Clamped bar length (l) • Bar height (t) • Bar thickness (b)

The optimization techniques used to deal with this issue in particular. The best solution comparison achieved with these methods is given Table 6. The calculated findings are compared in Table 7. The outcomes explain that SUA performs best in mean, standard deviation, and needs low analysis for identify most optimum diagram.

Table 6
Outcomes of diagram of a welded beam

Approach	Best Value				Best price
	h	T	l	B	
SUA	.21	3.4	9.0	.21	1.72
PSO	.20	3.5	9.1	.20	1.72
MFO	.15	3.2	9.0	.15	1.8
MVO	.20	3.4	10.0	.20	1.88
SCA	.20	4.01	10.1	.20	2.1
GSA	.15	4.1	9.9	.15	1.78
GA	.20	3.2	9.2	.20	1.89
HS	.20	3.3	9.5	.22	1.83

Table 7
Statistical outcome of different techniques for welded beam issue

Approach	Optimal	Average	Poor	St. Dev	Median
SUA	1.77	1.77	1.6	.002	1.72
PSO	1.6	1.6	1.66	.001	1.72
MFO	1.55	1.66	1.64	.32	1.6
MVO	1.56	1.64	3.12	.012	1.54
SCA	1.32	1.56	1.32	.004	1.1
GSA	1.6	1.53	1.6	.25	1.52
GA	1.67	1.6	1.55	.03	1.12
HS	1.56	1.62	1.56	.135	1.35

8. Conclusion

In this article, we have suggested to solve a static weapon-target assignment issue using a novel version of the conventional Ant Colony System method, called SUA. In an implementation that incorporates the problem-specific initialization of pheromone and heuristic information and particular selection method. Simulations of military defensive operations now concentrate on the allocation of resources in complicated situations. In large scale issues, we evaluated SUA and compared its performance with 2 greedy heuristics and an ACS method. Even modest gains may make the difference in important simulations that justify the use of time intensive methods with a restricted time requirement. This article introduced a novel Swarm Optimizer method dubbed the Swarm Urochordate Algorithm (SUA). Basic ideas motivating the technique are appreciated by social and the camping nature of detected hyena. It suggested in this article that the SUA algorithm SUA be used to solve 30 test formulas to evaluate exploration, exploitation, regional optimum prevention and convergence nature. Six challenges of real world diagram are also used for evaluate further the effectiveness of the SUA algorithm. The findings SUA offers highly competitive outcomes in relation to other well-known heuristics. The statistical findings, based on comparisons among the developed SUA as well as optimized techniques, indicate that method being presented may handle different kinds of limits as well as provide best outcomes than existing optimizers. The suggested approach utilized to solve issues of real-life optimization may take less effort to discover the best answers. Several guidelines for future work may be suggested. Problems of optimization may be resolved in various areas. In addition, this method may be considered as a future contribution in order to tackle multi-objective as well as many-object if issues.

9. Declarations

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Funding Details

None.

Conflict of interest Author

Authors declare that they have no conflict of interest.

Informed Consent

Informed consent was obtained from all subjects involved in the study.

10. References

1. Ahuja, R. K., A. Kumar, K. C. Jha, and J. B. Orlin. 2007. Exact and heuristic algorithms for the weapon-target assignment problem. *Operations Research* 55:1136–46. doi:10.1287/opre.1070.0440.
2. Den Broeder, G. G., and R. E. Ellison. 1959. On optimum target assignment. *Operations Research* 7:322–26.
3. Dorigo, M., and T. Stützle. 2004. *Ant Colony Optimization*. 1. Cambridge, MA: MIT Press.
4. Eckler, A. R., and E. S. A. Burr. 1972. Mathematical models of target coverage and missile allocation. Technical Report DTIC:AD-A953517, Alexandria, VA: Military Operations Research Society.
5. Gao, D., G. Gong, L. Han, and N. Li. 2010. Application of multi-core parallel ant colony optimization in target assignment. Taiyuan: In: Proceedings of the International Conference on Computer Application and System Modeling (ICCSM), IEEE 514–18.
6. Goldberg, D. E., and J. R. Lingle. 1985. Alleles, loci, and the traveling salesman problem. Proceedings of the First International Conference on Genetic Algorithms and their Applications. NJ: Lawrence Erlbaum Associates, 154–59.

7. Hossein, P. A. 1990. A class of dynamic nonlinear resource allocation problems. PhD. Thesis, Massachusetts Institute of Technology, Massachusetts, USA.
8. Johansson, F. 2010. Evaluating the performance of TEWA systems. Phd.Thesis, University of Skövde, Sweden.
9. Johansson, F., and G. Falkman. 2009. An empirical investigation of the static weapon-target allocation problem. In Proceedings of the 3rd skövde workshop on information fusion topics (SWIFT 2009), Edited by J. Laere, and J. M. R. Johansson, 63–67. Skövde: CSREA Press.
10. Johansson, F., and G. Falkman. 2010. A suite of metaheuristic algorithms for static weapon-target allocation. Proceedings of the 2010 International Conference on Genetic and Evolutionary Methods. CSREA Press, 132–38.
11. Johansson, F., and G. Falkman. 2011. Real-time allocation of firing units to hostile target. Journal of Advances in Information Fusion 6:187–99.
12. Julstrom, B. A. 2009. String-and permutation-coded genetic algorithm for the static weapon-target assignment problem. In: Procs of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference (GECCO2009). New York: ACM, 2553-2558.
13. Karasakal, O. 2008. Air defense missile-target allocation models for a naval task group. Computers & Operations Research 35:1759–70. Kennedy, J., and R. Eberhart. 1995.
14. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, 1942–48.
15. Formato RA. Central force optimization: a new deterministic gradientlike optimization metaheuristic. Opsearch 2009;46(1):25–51. doi:10.1007/ s12597-009-0003-4.
16. Gandomi AH. Interior search algorithm (isa): a novel approach for global optimization. ISA Trans 2014;53(4):1168–83.doi:10.1016/j.isatra.2014.03.018.
17. Gandomi AH, Yang X-S. Benchmark problems in structural optimization. Springer Berlin Heidelberg; 2011. p. 259–81. doi:10.1007/978-3-642-20859-1_12.
18. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation 2001;76(2):60–8. doi:10.1177/003754970107600201.
19. Ghorbani N, Babaei E. Exchange market algorithm. Application Soft Computer 2014;19:177–87. doi:10.1016/j.asoc.2014.02.006. <http://www.sciencedirect.com/science/article/pii/S156849461400074X>.
20. Glover F. Tabu search-part i. ORSA J Comput 1989;1(3):190–206.
21. Glover F. Tabu search-part ii. ORSA J Comput 1990;2(1):4–32.
22. Hatamlou A. Black hole: a new heuristic optimization approach for data clustering. Inf Sci 2013;222:175–84.doi:10.1016/j.ins.2012.08.023.
23. He S, Wu QH, Saunders JR. A novel group search optimizer inspired by animal behavioural ecology. In: IEEE International Conference on Evolutionary Computation; 2006. p. 1272–8. doi:10.1109/CEC.2006.1688455.
24. He S, Wu QH, Saunders JR. Group search optimizer: an optimization algorithm inspired by animal searching behavior. IEEE Trans Evol Comput 2009;13(5):973–90. doi:10.1109/TEVC.2009.2011992.70 G. Dhiman, V. Kumar/ Advances in Engineering Software 114 (2017) 48–70
25. Ilany A, Booms AS, Holekamp KE. Topological effects of network structure on long-term social network dynamics in a wild mammal. Ecol Lett 2015;18(7):687–95. doi:10.1111/ele.12447.
26. Bhatia, Sarjit Singh, Anurag Rai, and Harsimran Kaur. "An architectural framework for the implementation of ERP using cloud computing in SMEs: A literature survey." In " International Journal of Science and Research"6, no. 2, 11-18, (2017).
27. A Rai, S Bhatia, H Kaur, " Cloud Based Ranking Prediction Framework: A Selection Criteria", in " Journal of Advanced Research in Dynamical and Control Systems", 120-125, 10 (01), 2018.
28. Kaur, H, Rai A., and Bhatia S. S., "An analysis of QoS ranking prediction framework techniques", in " Modern Physics Letters B", 01 ,1850420, 2019.
29. Kaur, A. Rai, S. S. Bhatia, and G. Dhiman, "MOEPO: a novel multi-objective emperor penguin optimizer for global optimization: special application in ranking of cloud service providers," Engineering Applications of Artificial Intelligence , vol. 96, p. 104008, 2020.
30. Murphey, R. A. 2000. Target-based weapon target assignment problems. In Nonlinear assignment problems. vol. 7, in combinatorial optimization, edited by P. M. Pardalos, and L. S. Pitsoulis, 39–53.
31. New York: Springer US. Wang, S., and W. Chen. 2012. Solving weapon-target assignment problems by cultural particle swarm optimization. In: 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC).
32. Nanchang, Jiangxi: IEEE, 141–44. Zeng, X., Y. Zhu, L. Nan, K. Hu, B. Niu, and X. He. 2006. Solving weapon-target assignment problem using discrete particle swarm optimization. Proceedings of the 6th World Congress on Intelligent Control and Automation. Dalian: IEEE, 3562-3565.
33. Zhang, J., W. Xiaojing, and X. Chuanqing. 2012. ACGA algorithm of solving weapon - target assignment problem. Open Journal of Applied Sciences 2:74–77
34. Lee, Z. J., C. Y. Lee, and S. F. Su. 2002b. Parallel ant colonies with heuristics applied to weapon-target assignment problems. In: Proceedings of the 7th Conference on Artificial Intelligence and Applications, 201–06. Taichung, Taiwan.
35. Lee, Z.-J., C.-Y. Lee, and S.-F. Su. 2002a. An immunity-based ant colony optimization for solving weapon target assignment problem. Applied Soft Computing 2:39–47.
36. Lee, Z. J., S. F. Su, and C. Y. 2003. Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. IEEE Transactions on Systems, Man and Cybernetics, Part B 33:113–21.
37. Kolitz, S. E. 1988. Analysis of a maximum marginal return assignment algorithm. Procs. of the 27th conference on Decision and Control, 2431–36.
38. Manne, A. S. 1958. A target-assignment problem. Operations Research 6:346–
39. Wang, S., and W. Chen. 2012. Solving weapon-target assignment problems by cultural particle swarm optimization. In: 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC). Nanchang, Jiangxi: IEEE, 141–44.

40. Zeng, X., Y. Zhu, L. Nan, K. Hu, B. Niu, and X. He. 2006. Solving weapon-target assignment problem using discrete particle swarm optimization. Proceedings of the 6th World Congress on Intelligent Control and Automation. Dalian: IEEE, 3562-3565.
41. Lee, Z.-J., and W. L. Lee. 2005. A hybrid search algorithm with heuristics for resource allocation problem. Information Sciences 173:155–67.
42. Kennedy, J., and R. Eberhart. 1995. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, 1942–48.

Figures

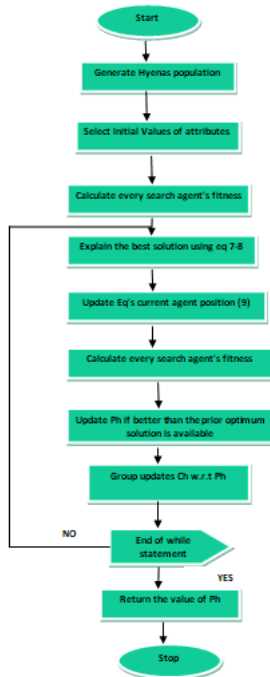


Figure 1

Diagram of the SUA suggested

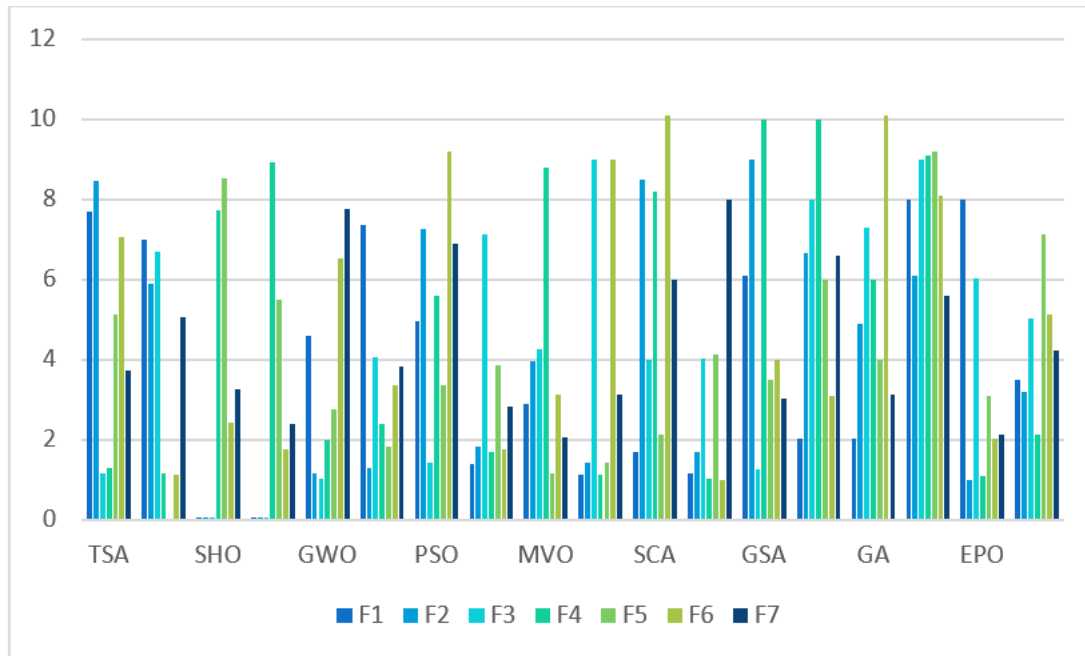


Figure 2

Iterations from F1 - F7

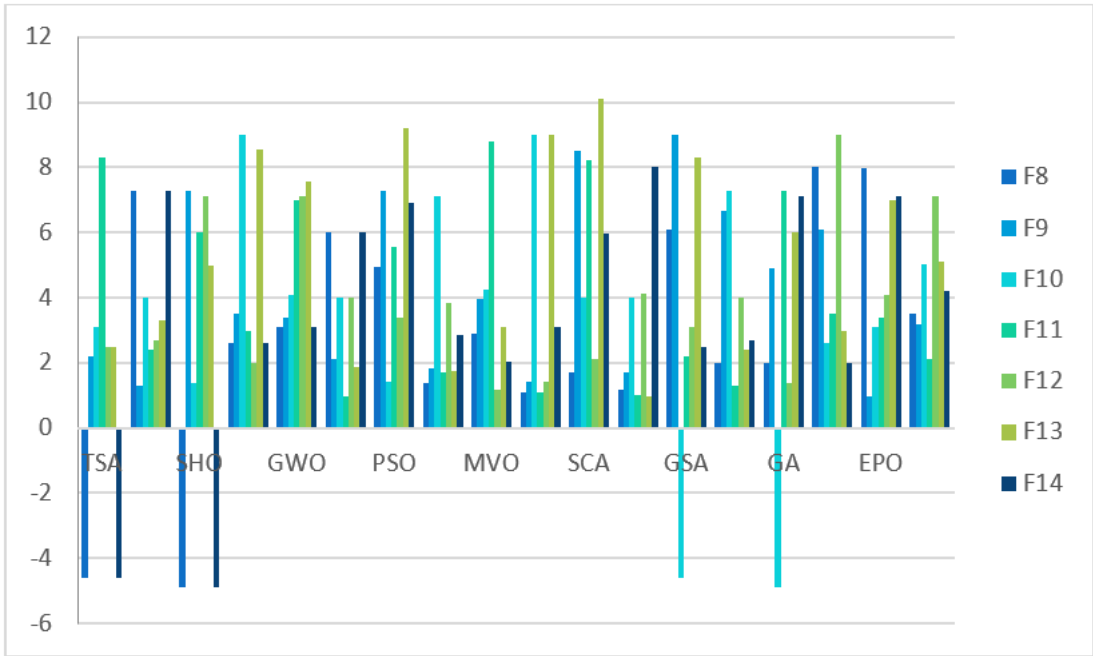


Figure 3

Iterations from F8 - F14

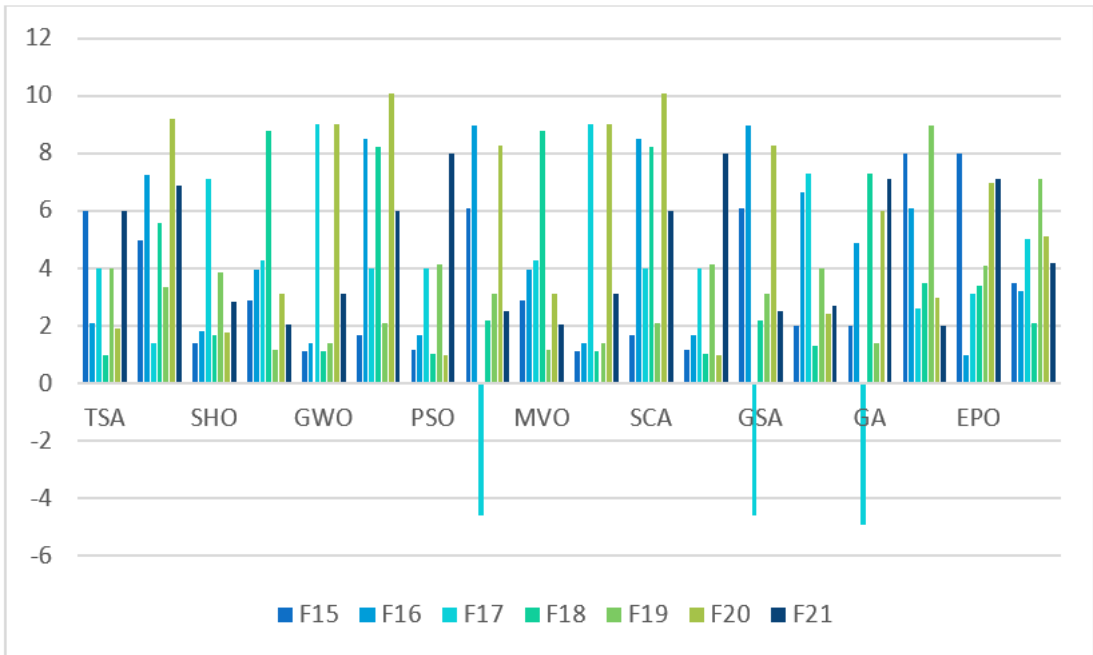


Figure 4

Iterations from F15-F21



Figure 5

Iterations from F22-F30

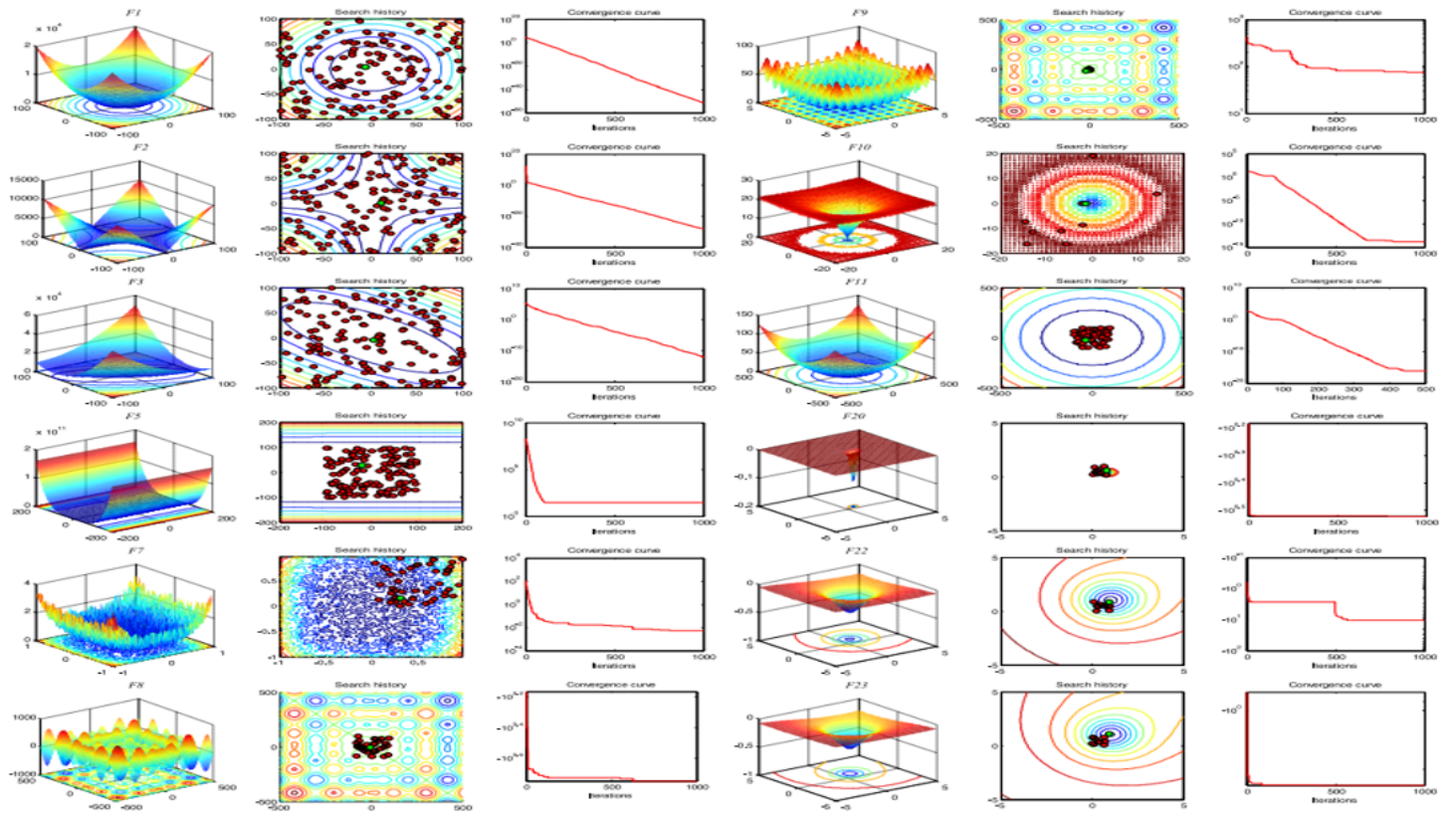


Figure 6

Fluctuations in benchmark operations

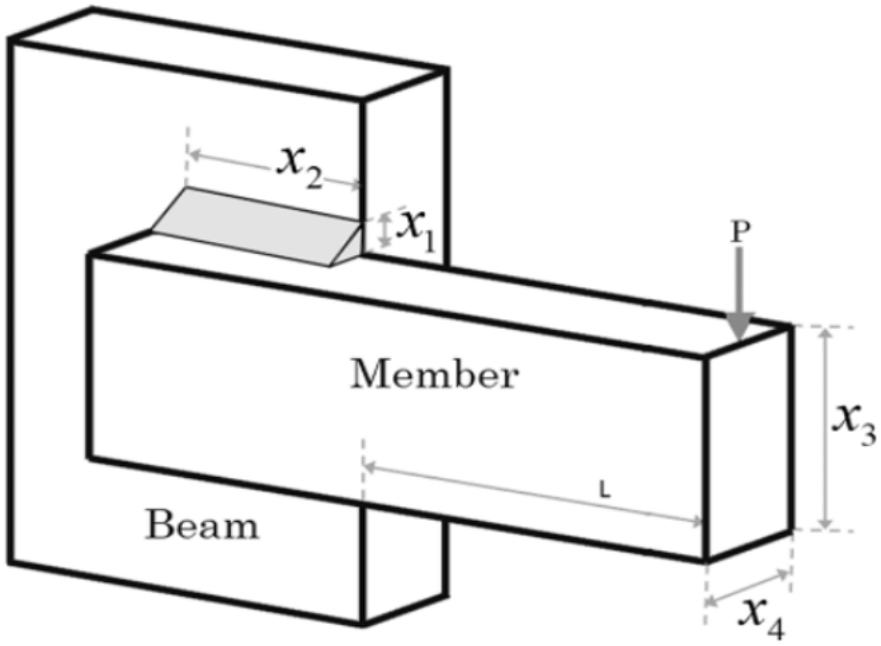


Figure 7

Schematic illustration of the issue of welded beam