ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20220113192727
Original scientific paper

Solving Weapon-Target Assignment Problem with Salp Swarm Algorithm

İsa AVCI*, Mehmet YILDIRIM

Abstract: The weapon target problem is a combinatorial optimization problem. It aims to have the weapons on target properly assigned for the intended purposes. When focused on its target, it does things with its effective attack research in mind. It is an ongoing problem program to minimize survivors. This study, using the weapon target assignment model calculates the expected probabilities on the target with the salp model. The nature of this SHA model is designed to be appropriately predicted for this particular use. The Salp Swarm Algorithm (SSA) is a metaheuristic method of methods approaching the solution set as an approximation. Optimum solution or optimum example is in a working example. This study was done with 12 problem examples (200 training and 200 targets with pleasure to observe, to test the efficiency of SSA). In the problem, the iteration resulted in optimum results at the end of the definite usage time. Best value included 48.355 for WTA1, 92.654 for WTA2, 174.432 for WTA3, 155.658 for WTA4, 250.784 for WTA5, 284.967 for WTA6, 247.458 for WTA7, 362.636 for WTA8, 524.732 for WTA9, 548.580 for WTA10, 601.654 for WTA11, and WTA16812. It was obtained by finding in 200,000 iterations and the result value was 50. After 200000 improvements, it was observed to relax to increase iteration. The use of barter when generating new solutions to the problem. To find out the fitness values, mean, best, and worst values were found.

Keywords: metaheuristic optimization; particle-wide optimization; salp swarm algorithm; search weapons; target assignment

1 INTRODUCTION

With the continuous development of technologies, the increase in computer users, and the emergence of complex and difficult problems in different fields, the concept of optimization has emerged. Optimization, which is used in the sense of optimization, means that the problem finds the best, most suitable solution among all possible solutions within the constraints. Optimization has constraints and the problem is handled according to these constraints. For example, the constraint in the traveling salesman problem is that the city you stop by is not visited again [1]. In addition, optimization is used for problems that are difficult or costly to solve, and it also works quickly and finds a solution in a short time. Reasonable time, near-optimal solution, and validity have a very important place in optimization problems [2]. General definitions used in optimization algorithms:

- Problem: The problem, which is the decision-making mechanism in the system, is expressed as system equality.
- Search Space: The whole set of values that the decision variables can take is called the search space.
- Solution Space: It is the expression of the solutions of the inputs in the search space in the problem as output values.
- Global Best: It is the input value for which the best optimum solution is found in the problem.
- Local Best: It is the solution value that is good but not the global best among the neighboring solutions in the search space in the problem.

Optimization algorithms emerge under two headings as deterministic (specific) and stochastic (random) based algorithms. While deterministic algorithms find the best using the derivative of the objective function, stochastic-based algorithms aim to find the best with a certain number of random solutions and optimization steps [3]. Mathematical and heuristic methods are used in solving optimization problems. In problems with a large solution space, the cost is quite high since mathematical algorithms operate on the entire solution space.

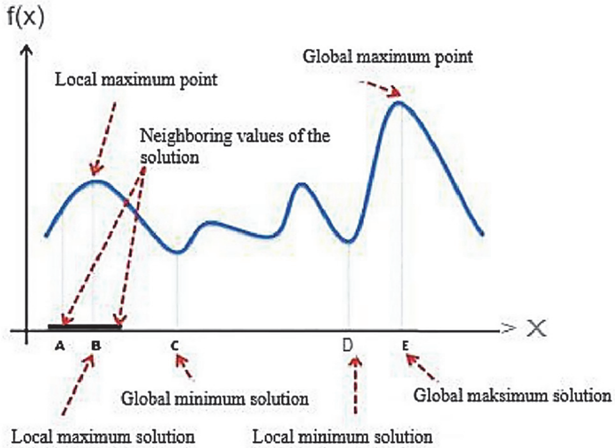
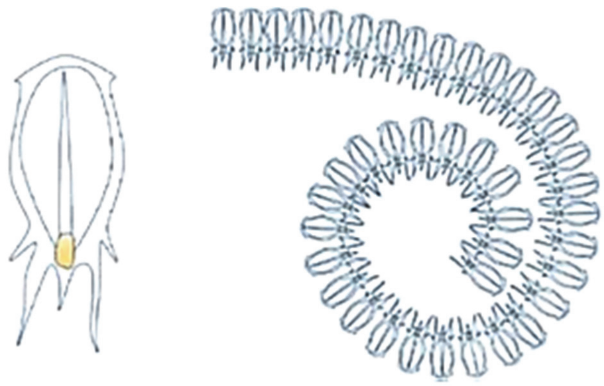


Figure 1 Defines the generals used in optimizations

Heuristic algorithms that can reduce the solution space to a more reasonable level by intuitively navigating the solution space have begun to be used for problems with a large solution space [4]. Heuristic algorithms have come together and metaheuristic algorithms have emerged. Metaheuristic algorithms reach the best solution for the problems with large solution space by considering the cost and reasonable time. The advantage of metaheuristic optimization algorithms is to reach the global optimum solution by exceeding the local optimum points. With a different definition, metaheuristic methods offer balance by exploring new ways to find solutions to unsolved problems, while searching for solutions that can be robust and more efficient in solving accepted optimization problems.

Today, metaheuristic algorithm techniques have become very popular. The main reasons for this popularity are known as flexibility, gradient-free mechanism, and escape from local optimum points. Metaheuristic algorithms use random operators. In this way, while solving real problems with local optimum value(s), it also ensures that it gets rid of local solutions. Metaheuristic algorithms are divided into two classes, evolutionary and herd intelligence. Evolutionary algorithms work by imitating evolutionary concepts in nature. Genetic Algorithm (GA), which is the best algorithm in this class, is the most popular of the evolutionary algorithms. Permutation-based optimization GA is one in which a set of random solutions is initialized to solve a particular problem. Before starting the solutions, after evaluating them with the objective function, it changes the variables of the solutions according to their fitness values. It finds a solution by converging with the resulting probability values. This algorithm can work with constraints and find the fitness value by searching according to convergence criteria. Herd intelligence techniques, on the other hand, work by imitating the intelligence of herds in nature. The basis of these algorithms stems from the behavior of a group of living things together [5].

Bat, salp, bee, butterfly, and ant are examples of animals developed as swarms. It has emerged by imitating the eating, drinking, location, position, search features of these animals and by researching these features [6]. Salp Swarm Algorithm (SSA), which has been recently proposed in the field of feature selection, is an optimization algorithm that emerged in 2017, inspired by the fusion of salps.



(a) (b)

Figure 2 (a) Individual salp and (b) Salp chain

The primary goal of the gun targeting problem is to identify interceptors that will minimize the probability of being hit by incoming missiles. This study aims to solve the weapon assignment problem using the Salp Swarm algorithm. The problem of aiming guns was first proposed by Mann in 1958. The purpose of aiming for weapons is to calculate the probability of how we can avoid this attack with minimal damage when certain weapons are attacking [7]. Assigning a target to a weapon is a constrained combinatorial optimization problem. Meta-heuristic Salp Swarm Algorithm (SSA) is used to minimize the damage done to the problem. In previous studies, various heuristic algorithms have been used to solve the problem. This problem has been investigated using swarm algorithms such as sinusoidal cosine (SCA), simulated annealing (SA), particle swarm optimization (PSO), and ant colony optimization (ACO) [8].

Bullet weapons are important tools used in military fields from the past to the present. It aims to assign "m" guns to "n" targets in the best way possible. It aims to maximize the expected probability of damage to targets. In terms of target survival, it tries to minimize the probability of survival [9]. It has enabled the use of weapons in the military field, thanks to its ability to inflict damage from afar. In the 20th century, it has developed at the level of attacking the enemies existing with missile technology, even at a distance, without getting close to the target. To reduce the damage of this distant threat, the concept of air defense has developed. In addition, as a missile's ability to reduce the threat and danger posed increased, the quantity and quality of missiles developed increased accordingly. It has been realized that the increased missiles should be distributed correctly.

Base values were set in the study and these set values were the probability of destruction of the initiator p matrix, the permutation order of the weapons assigned to the target, the *v* value, and the value of the targets. Afterward, parameter adjustments were made for SSA. Random value was used for the initial solution and continued until the most suitable value was found for the solutions. SSA algorithm; The problem takes the parameters of population size, minimum range value, maximum range value, generation, and target function. These parameters are very important as they increase or decrease the quality of SSA. The weapon is sent to the SSA algorithm to find a solution to the target assignment problem.

In this study, a solution was sought by parameter adjustment in SSA. The first solution value is randomly assigned. The results obtained were improved and sent to the algorithm as input again. While weapon target assignment generates integers as a solution, the SSA algorithm operates with continuous values (float). It is necessary to convert the continuous values into values that the Weapon target assignment will process and produce output. As a solution, the Largest Ranked Value (LRV) approach is used. The results were compared with Simulated Annealing and Population-Based Simulated Annealing, which were found in another study, and the process was completed.

2 LITERATURE REVIEW

The weapon targeting problem introduced and first mentioned by Mann in the field of operations research aims to identify compatible interceptors in response to incoming missiles to minimize the probability. While research in the gun targeting literature has focused on a defensive perspective, some studies have taken an offensive perspective into account [10]. The task of aiming for weapons is aimed at providing maximum security so that the enemy does not attack protected objects.

Weapon target assignment problem, static weapon target assignment, and dynamic weapon target assignment are also examined in two models. Static weapon target designation was modeled by Manne in 1958. Static weapon target designation is also known as the number of missiles. Some interceptors are limited in number with the possibility of successfully eliminating threats. They can only be used for one exchange. The solution to the problem of static targeting of weapons is to inform the defense about how many weapons of each type there are to fire at each target. In the static model, all the input parameters of the task are set as static, and in this model, the assignment of weapons to targets is performed in one step. This study also used a static model [11]. Weapon target assignment has been solved by precise algorithms and an optimum solution has been obtained. However, the solution of weapon target assignment problems started to be implemented with the development of heuristic algorithms. In the study of Lloyd and Witsenhausen, weapon target assignment has proven to be an NP-complete problem. It has been mentioned that the optimal or optimal solution can be reached as a solution method [12].

In the weapon target assignment problem, the optimum solution is reached quickly. Leboucher made this inference by using the Particle Swarm Algorithm, which is one of the heuristic algorithms, in his study stating that the weapon target assignment is fast to reach the optimum target.

The study of Yanxia et al. solved the weapon target assignment problem with the Ant Colony Algorithm. The path selection, which is one of the parameters used in the Ant Colony Algorithm, was followed by the increase or decrease of the ant pheromone secretions, and the table was created, and the analysis was carried out [13].

Altınoz developed a decision system for the weapon target assignment problem and trained the model with many weapons and target assignment examples with this developed system. It has also been observed that the assignment of weapons to targets is easily done and it makes more accurate assignments compared to other studies. This created system was developed as a result of different pieces of training for weapons and targets and was created with neural networks [14].

The results obtained by Sonuç et al. using Simulated Annealing and Population-Based Simulated Annealing, in solving the weapon target assignment problem, were added to the study. SA, PSA, and SSA Algorithms have been compared and it has been proven that the SSA algorithm gives better results [15].

3 EXPERIMENTAL METHOD
3.1 Salp Swarm Algorithm

SSA, which is a random population-based algorithm, was developed by Mirjalili et al. SSA, an algorithm proposed in 2017, emerged from the swarming behavior of salps as they navigate the oceans and search for food. In the ocean, the rafts cling to each other to form the raft chain. The salp leading in this chain is one of the salps in front of the herd. The remaining salps are called followers.

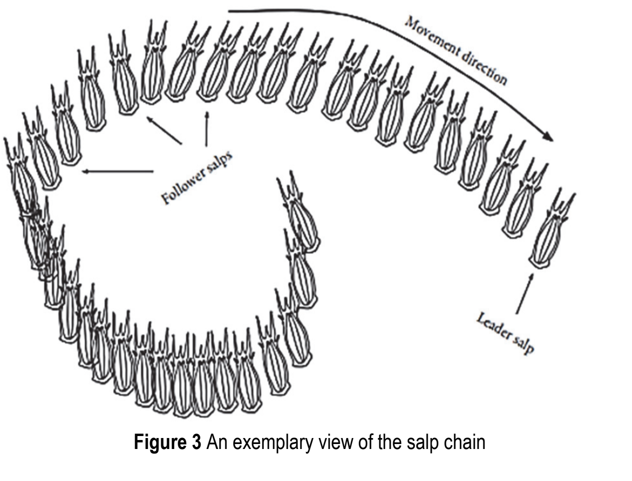


Figure 3 An exemplary view of the salp chain

Techniques found in swarm-based algorithms are also valid in this algorithm. In these techniques, the location of the salps and the number of variables to be used in the problem are defined in the search space (F). The total position of the salps is stored in the z matrix. Since the goal of the salps herd is to reach the food, the P variable was also defined as the food source. The mathematical modeling of SSA is shown in Eq. (2). The leader salp in the herd moves towards the food by changing its position with the given equation to search for food [16].

$$X_i = \begin{bmatrix} x_1^1 & \cdots & x_d^1 \\ \vdots & \ddots & \vdots \\ x_1^n & \cdots & x_d^n \end{bmatrix} \tag{1}$$

In the salp swarm algorithm, the population is started randomly. Individuals are shown as a candidate solution for the problem to be solved. The X seen in Eq. (1) is the two-dimensional matrix representation of the swarm of X consisting of n salps. The goal of the Salp swarm algorithm is to reach the food source P in the search space F.

$$x_j^1 = \begin{cases} F_j + c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right)c_3 \geq 0.5 \\ F_j - c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right)c_3 < 0.5 \end{cases} \tag{2}$$

The symbols in the mathematical model of the SSA are the size of the leader's position, the size of the food, the upper and lower limits of these dimensions, the random variables produced, and the number of iterations. The variables ubj and lbj represent upper and lower classes and are j-dimensional. In this mathematical model, c_1, which acts as a balance between exploration and capabilities, is an important parameter in SSA. Eq. (3) shows how the c_1 parameter is calculated.

$$c_1 = 2\mathrm{e}^{\left(\frac{u}{L}\right)^2} \tag{3}$$

Follower salps that follow the lead salp need to be repositioned. This process is as in Eq. (4).

$$Z_n^m = \frac{1}{2}c\mathrm{e}^2 + v_0\mathrm{e} \tag{4}$$

Using Eq. (4), the process of changing the position of the follower salps is performed. Since SSA has time iteration, the contradiction between iterations is "1". The initial speed is considered to be "0". This process continues as seen in Eq. (5).

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right) \tag{5}$$

Using Eq. (5), m. salp a follower, n. size position.

Salp Swarm Algorithm working stages:

1. Initially, a random population of solutions is generated in the SSA. Optimal salp position, iteration number, laps number, and optimal value parameters are determined.
2. The randomly generated solution is evaluated using the objective function based on its population.
3. The food source is determined (P).

4. The c_1 parameter is updated at each step.
5. For each salp.

- If $m == 1$, the salp position corresponding to this value is updated.
- Otherwise, the position of the follower salp is updated.
- The suitability of each salp is evaluated.
- If there is a better solution, it will be updated.
- When the stopping criterion is met, the processes are completed. The best solution is calculated and the search space F is updated.
- The best solution and fitness value are returned.

The flowchart of this working logic is given in Fig. 4.

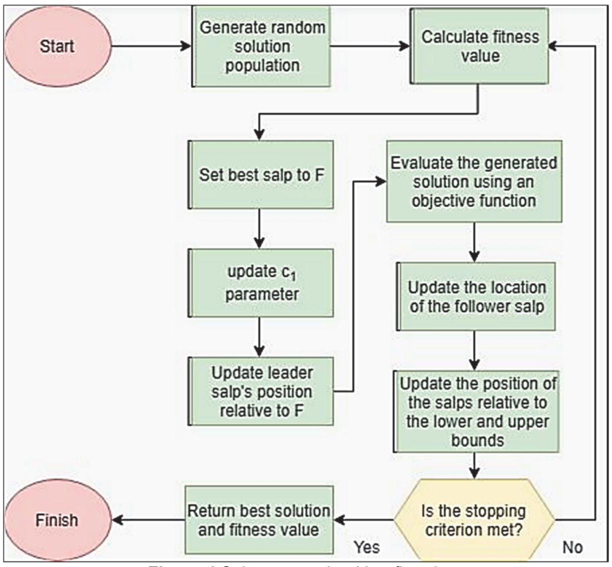


Figure 4 Salp swarm algorithm flowchart

SSA is tested on many mathematical optimization functions to observe the behaviors that are effective in finding optimal solutions and to learn the reality of the behaviors. When the results on the mathematical functions are evaluated, it shows that the SSA algorithm can initially improve the random results and converge to the optimum. With Eq. (2) and Eq. (5) given above, the position of the leader salp and follower salps is updated. While the location update is in progress, if any salp goes out of bounds, that salp is brought back to the limit. Finally, all steps except start are repeated until an end condition is met.

Table 1 Algorithm 1 Pseudo code of the SSA

```
Algorithm 1 Pseudo-code of the SSA
Initialize the swarm x_i (i = 1,2,…, n)
while (end condition is not met) do
    Obtain the fitness of all salps
    Set F as the leader salp
    Update c_1 by Eq. (3)
for (every salp (x_i)) do
if (i == 1) then
        Update the position of leader by Eq. (2)
else
        Update the position of followers by Eq. (5)
    Update the population using the upper and lower limits of variables
    Return salps that violated the bounding restrictions.
Return F
```

Salp Swarm Algorithm emerged as a result of observing the behavior of salps with each other while navigating the oceans and searching for food [17]. The Salp Swarm Algorithm, which is used to minimize Continuous Variable Functions, takes the following parameters as a function. These are;

- problem = It is used as the problem to be solved in the algorithm.
- swarm size = Used to express population size.
- min values = It is used to express the minimum value that the variable(s) in the list can take.
- max values = It is used to express the maximum value that the variable(s) in the list can take.
- generations = It is used to express the total number of iterations.
- target function = Set as function to shrink.

In this study, the function of the SSA algorithm is written. This function takes the parameters of the problem (weapon target assignment has been sent to the function as a problem since the weapon target assignment problem will be resolved), swarm size, min. values, max. values, generations, target function. The number of iterations was determined as 200000 and the population number as 50. While the weapon target assignment generates integers as a solution, the SSA algorithm operates with continuous values. It is necessary to convert the continuous values into values that the weapon target assignment will process and produce output. This is perceived as a problem. As a solution, the Largest Ranked Value (LRV) approach is used. LRV assigns the largest value to the first index value and the other values to other indexes, respectively.

3.2 Weapon Target Assignment Problem (WTA)

A permutation-based optimization problem is the basis of the weapon target assignment problem. The purpose of weapon target assignment is to give weapons that complement them according to targets in the environment and reduce the survivability of the sum of targets. Weapon target designation is often used in military fields [18].

The way it is used in the military field is to minimize the survival value of the enemy by having different weapons and assigning them to different targets as this target [19]. It is preferred to solve this problem with heuristic algorithms. The reason for this is that it has an integer decision variable and contains nonlinear functions. SSA was used as the solution for weapon target assignment.

The parameters of the weapon target assignment problem are given below:

- n, number of targets (1, 2, …, n),
- m, number of weapons (1, 2, …, m),
- v_i, the value of i target,
- q_{ij}, i. target j. probability of destruction by throwing weapons,
- $x\ =\ [x_{ij}]$ nxm decision variable with matrix.

$$x_{ij} = \begin{cases} 1, \text{ if weapon } j \text{ is assigned to target } i \\ 0, \text{ otherwise} \end{cases} \tag{6}$$

Objective function of weapon target assignment:

$$\text{Min } Z = \sum_{j=1}^{n} V_j \prod_{i=1}^{m} q_{ij}^{x_{ij}} \tag{7}$$

The number of n targets in the objective function of weapon target assignment, and m denotes the variety of weapons. Z is the values included in the total chance of survival (probability) of the targets and aims to minimize it. Constraints of weapon target assignment:

$$\sum_{j=1}^{n} x_{ij} \leq W_i \ , i = 1, 2, \ldots, m \tag{8}$$

$$x_{ij} \in Z_+ \tag{9}$$

As with the optimization problems, there are also limitations in the weapon target assignment problem. The constraint above refers to the set that W_i, which is the value set of weapon types, and \emph{x\textsubscriptij, which is the decision variable, can be taken as values. In the weapon target assignment problem, all the weapons found must be assigned to the targets. In this study, operations were carried out considering that the number of targets and weapons was equal.

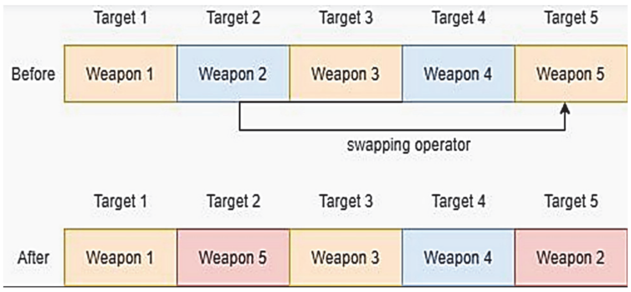


Figure 5 Displacement display of targets using displacement operator in weapon assignments

In the steps given above for the solution of the weapon target assignment problem, it was mentioned that the Salp Swarm Algorithm started the solution by randomly assigning the parameters in the first study and then looked for new solutions. It performs the new solution generation process by using swapping (replacement operator). Fig. 5 includes weapons and targets. The location of the displacement operator and the weapons are changed between each other, and the salp is assigned as a parameter to the herding algorithm and the most suitable solution is tried to be found. The initial solution is improved in this way [19].

3.3 Dataset Information

For the weapon target assignment problem, there are not many data set examples in the literature for comparison. For this reason, the data set created for testing was used. This data set was created as 12 problems and studies were carried out on them.

These target values, which were created by taking random numbers from the uniform between 25-100, were set according to the probability of extinction. The dimensions of these created problems are different. Dimension examples are shown in Tab. 2 [20].

Table 2 The states and dimensions created for the WTA problem

Instance	Weapon	Target
SHA1	5	5
SHA2	10	10
SHA3	20	20
SHA4	30	30
SHA5	40	40
SHA6	50	50
SHA7	60	60
SHA8	70	70
SHA9	80	80
SHA10	90	90
SHA11	100	100
SHA12	200	200

4 RESULTS AND DISCUSSION

Objective values were obtained for 50 populations in the first 500 iterations of the Salp Swarm Algorithm in Fig. 6a.

The algorithm shows a downward trend starting from values close to 720. It has been determined that the best value is fixed after converging to 640. The graph in Fig. 8 shows the best results of the SSA algorithm for all weapon target designation populations.

The best working values for the populations in all weapon target assignment problems of the SSA algorithm are given in Fig. 6a and Fig. 6b. When these values are compared, the success of the algorithms is very close to each other.

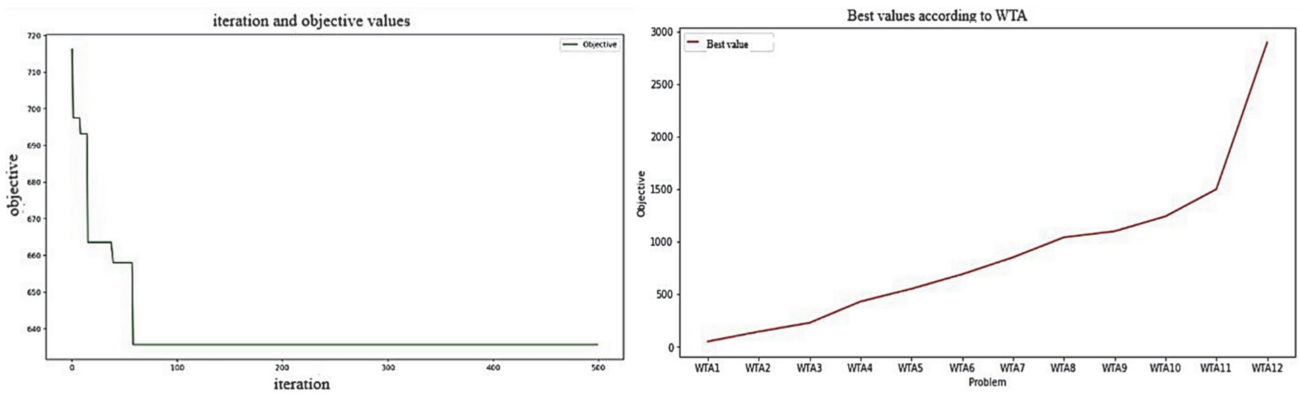


Figure 6 (a) Results for 50 populations in the first 500 iterations; (b) Results for 50 populations in the first 500 iterations

According to the results given in Tab. 3, the trial and error method was used to make the best assignment to the target in the solution of the weapon target assignment problem and the best solution was found by making changes on the number of iterations.

The results given in Tab. 4 are the best results for the scenario case when the number of iterations is 200000. It has been observed that the values get better when the number of iterations is increased.

Table 3 Comparison of the gains obtained in the study

Instance	Weapon	Target	10000 iterations	30000 iterations	60000 iterations	80000 iterations	Best (200000) iterations	2500000 iterations
SHA1	5	5	60.640	58.456	55.742	50.78	48.355	48.355
SHA2	10	10	102.475	109.89	107.75	106.82	92.654	92.654
SHA3	20	20	211.238	207.85	192.55	186.21	174.432	174.432
SHA4	30	30	397.878	384.39	380.85	254.78	155.658	155.658
SHA5	40	40	539.596	536.46	510.034	451.74	250.784	250.784
SHA6	50	50	646.432	610.61	452.10	385.42	284.967	284.967
SHA7	60	60	782.104	779.2	620.541	580.65	247.458	247.458
SHA8	70	70	979.535	964.13	922.505	863.74	362.636	362.636
SHA9	80	80	1038.167	1045.5	840.25	830.42	524.732	524.732
SHA10	90	90	1195.046	1198.24	950.56	756.25	548.580	548.580
SHA11	100	100	1391.808	1405.60	1102.25	812.76	601.654	601.654
SHA12	200	200	2793.153	2733.52	1028.47	958.54	895.168	895.168

Table 4 Representation of the best solution values in the SSA algorithm of the WTA problem

Instance	Weapon	Target	Algorithm	Best	Mean	Median
SHA1	5	5	SSA	48.355	48.35	48.36
SHA2	10	10	SSA	92.654	92.65	92.62
SHA3	20	20	SSA	174.432	174.45	174.43
SHA4	30	30	SSA	155.658	155.65	155.64
SHA5	40	40	SSA	250.784	250.74	250.77
SHA6	50	50	SSA	284.967	284.95	284.95
SHA7	60	60	SSA	247.458	247.46	247.46
SHA8	70	70	SSA	362.636	362.75	362.73
SHA9	80	80	SSA	524.732	524.73	524.73
SHA10	90	90	SSA	548.580	548.66	548.65
SHA11	100	100	SSA	601.654	601.65	601.67
SHA12	200	200	SSA	895.168	895.18	895.16

Table 5 The best solution values of the weapon target assignment problem found in the PSA and SA algorithms

Problem	PSA	SA Best	SA Worst	SA Mean	SA Median
SHA1	48.3640	48.3640	48.3640	48.3640	48.3640
SHA2	96.3123	96.3123	96.3123	96.3123	96.3123
SHA3	142.1070	142.1070	142.1070	142.1070	142.1070
SHA4	248.0285	248.0285	248.0285	248.0285	248.0285
SHA5	305.5016	305.5016	305.5016	305.5016	305.5016
SHA6	353.4801	353.0767	353.5702	353.3112	353.2610
SHA7	414.7340	415.0528	415.7079	415.4068	415.4371
SHA8	497.2972	498.1049	499.0167	498.5918	498.5860
SHA9	535.5422	534.4408	536.2618	535.4559	535.5937
SHA10	595.3730	594.0639	596.1228	595.3277	595.6466
SHA11	699.4143	699.8357	702.1189	701.0054	701.2495
SHA12	1307.0154	1306.9126	1309.4616	1308.3382	1308.5187

In Tab. 5, the results obtained by Sonuç et al. in solving the weapon target assignment problem using Simulated Annealing and Population-Based Simulated Annealing are added to the study. The results of SA, PSA, and SSA algorithms are compared in Tab. 3 and Tab. 4.

5 CONCLUSIONS

In this study, solving the weapon target assignment problem with the Salp Swarm Algorithm is analyzed. Parameters sent to Salp Swarm Algorithm are defined as problem, swarm size, minimum value, maximum value, number of iterations and target function. The number of iterations was determined as 2000.00 and the population number as 50. While weapon target assignment generates integers as a solution, Salp Swarm Algorithm operates with continuous values (float). It is necessary to convert the continuous values into values that the weapon target assignment will process and produce output. The largest Ranked Value (LRV) approach was used as a solution to this problem. In the Salp Swarm Algorithm, the food and coordinate positions (locations) were first randomly assigned. Afterward, it was updated and searched until the best values were found.

The discovery feature of the Salp Swarm Algorithm, which is one of the metaheuristic algorithms, and the solutions of the weapon target assignment problem are examined. In addition, the iteration number and population number of the weapon target assignment problem were increased, and the optimum solution was found in the solution of this problem. The values found in the SSA algorithm gave better results than the Simulated Annealing and Population-Based Simulated Annealing algorithms, which were previously used in the literature. The results of the SSA algorithm are given in Tab. 3, and the results of SA and PSA, which are the algorithms used in the study of Okul et al., are given in Tab. 4. As a result, it was observed that Salp Herd Algorithm gave better (higher quality) results among 3 different algorithms performed on the same data set. The best result was found by changing the number of iterations in the problem. 48.355 for WTA1, 92.654 for WTA2, 174.432 for WTA3, 155.658 for WTA4, 250.784 for WTA5, 284.967 for WTA6, 247.458 for WTA7, 362.636 for WTA8, 524.732 for WTA9, 548.580 for WTA10, 601.654 for WTA11 and 8 for WTA12. Good results were obtained at 200000 iterations and a population value of 50. After 200000 iterations, it was observed that there was no change in the improvement value when the number of iterations was increased. It was observed that only a value of 0.00001% increased, which did not affect the result when rounded. It is known that when 50000 iterations are increased in the study, there is an increase of 0.00001% and this value is negligible in terms of efficiency because the prolongation (loss) in the working time is higher than the efficiency obtained (0.00001%).

After the successful results of the Salp Swarm algorithm, it is used in the dynamic weapon targeting problem, which is another model of the weapon targeting problem. In this model, optimum assignment is provided. By assigning the most suitable weapons to the target defense on the offensive side, it weakens the target's defense. It is foreseen to achieve defense success by being further developed and used in multiple defense fields and the defense industry in the future.

6 REFERENCES

[1] Eröz, E. & Tanyıldızı, E. (2018). Güncel Metasezgisel Optimizasyon Algoritmalarının Performans Karşılaştırılması. *International Conference on Artificial Intelligence and Data Processing (IDAP)*, 28-30.

[2] Alataş, S. & Akyol, B. (2012). Güncel Sürü Zekâsı Optimizasyon Algoritmaları. *Nevşehir Üniversitesi Fen Bilimleri Enstitü Dergisi*, *1*, 36-50.

[3] Gong, W., Cai, W., Ling, C. X., & Li, H. (2010). A real-coded biogeography-based optimization with mutation. *Applied Mathematics and Computation*, *216*(9), 2749-2758. https://doi.10.1016/j.amc.2010.03.123.

[4] Çelik, Y., Yıldız, İ., & Karadeniz, A. (2019). Son Üç Yılda Geliştirilen Metasezgisel Algoritmalar Hakkında Kısa Bir İnceleme. *European Journal of Science and Technology*, 463-477. https://doi.org/10.31590/ejosat.638431

[5] See https://www.veribilimiokulu.com/genetik-algoritma/

[6] Mirjalili, S. M., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, *114*, 163-191. https://doi.org/10.1016/j.advengsoft.2017.07.002

[7] Valdez, F., Castillo, O., & Melin, P. (2021). Bio-Inspired Algorithms and its Applications for Optimization in Fuzzy Clustering. *Algorithms*, 14, 122. https://doi.org/10.3390/a14040122

[8] Kline, A., Ahner, D., & Hill. R. (2019). The Weapon-Target Assignment Problem, *Computers & Operations Research*, *105*, 226-236. https://doi.org/10.1016/j.cor.2018.10.015

[9] Cao, M. & Fang, W. (2020). Swarm Intelligence Algorithms for Weapon-Target Assignment in a Multilayer Defense Scenario: A Comparative Study. *Symmetry*, *12*(5), 824. https://doi.org/10.3390/sym12050824

[10] Lu, Y. & Chen, D. Z. (2021). A new exact algorithm for the Weapon-Target Assignment problem. *Omega, Elsevier*, *98*(C). https://doi.org/10.1016/j.omega.2019.102138

[11] Yanxia, W., Longjun,Q., Zhi, G., & Lifeng, M. (2008). Weapon target assignment problem satisfying expected damage probabilities based on ant colony algorithm. *Journal of Systems Engineering and Electronics*, *19*(5), 939-944. https://doi.org/10.1016/S1004-4132(08)60179-6

[12] Altınöz, O. T. (2021). Evolving model for synchronous weapon target assignment problem. *2021 International Conference on Innovations in Intelligent SysTems and Applications (INISTA)*. https://doi.org/10.1109/INISTA52262.2021.9548606

[13] Hocaoğlu, M. F. (2021). Agent-based target evaluation and fire doctrine: an aspect-oriented programming view. *The Journal of Defense Modeling and Simulation*. https://doi.org/10.1177/15485129211040369

[14] Lloyd, S. P. & Witsenhause, H. S. (1986). Weapon allocation is NP-Complete. *Proceeding of the IEEE Summer Simulation Conference*, 1054-1058.

[15] Sonuç, E., Şen, B., & Bayır, Ş. (2017). A Parallel Simulated Annealing Algorithm for Weapon-Target Assignment Problem. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *8*(4). https://doi.org/10.14569/IJACSA.2017.080412

[16] Leboucher, C., Shin, H. S., Siarry, P., Chelouah, R., Le Ménec, S., & Tsourdos, A. (2013). A two-step optimisation method for dynamic weapon target assignment problem. *Recent Advances on Meta-Heuristics and Their Application to Real Scenarios, InTech.* https://doi.org/10.5772/53606

[17] Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., Al-Zoubi, A. M., Mirjalili, S., & Fujita, H. (2018). An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems*, *154*, 43-67. https://doi.org/10.1016/j.knosys.2018.05.009

[18] See https://www.mathworks.com/matlabcentral/fileexchange/63745-ssa-salp-swarm-algorithm

[19] Pala, O. (2021). Silah hedef atama problemi için uyarlanmış yerel arama ile yeni bir melez genetik algoritma. *Gümüşhane Üniversitesi Fen Bilimleri Dergisi*, *11*(2), 348-356. https://doi.org/10.17714/gumusfenbil.691176

[20] Ahuja, R. K., Kumar, A., Jha, K. C., & Orlin, J. B. (2007). Exact and Heuristic Algorithms for the Weapon-Target Assignment Problem. *Operations Research, INFORMS*, *55* (6), 1136-1146. https://doi.org/10.1287/opre.1070.0440

[21]Şahin, Y. (2019). Sezgiselve Metasezgisel Yöntemlerin Gezgin Satıcı Problemi Çözüm Performanslarının Kıyaslanması. *Bolu Abant İzzet Baysal Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, *19*(4), 911-932 . https://doi.org/10.11616/basbed.v19i51339.558208

Contact information:

İsa AVCI, Assistant Professor, PhD
(Corresponding author)
Karabuk University,
Kastamonu Yolu Demir Çelik Kampüsü,
78050 Kılavuzlar, Karabük, Türkiye
E-mail: isaavci@karabuk.edu.tr

Mehmet YILDIRIM, Student, MsC
Karabuk University,
Kastamonu Yolu Demir Çelik Kampüsü,
78050 Kılavuzlar, Karabük, Türkiye
E-mail: myildirim@ogrenci.karabuk.edu.tr