Vol: 2, Issue: 1, 2024
ISSN: 2980-3152
Pages:23-35

CURRENT TRENDS IN COMPUTING
https://dergipark.org.tr/en/pub/ctc
RESEARCH ARTICLE

CURRENT TRENDS IN COMPUTING

SOLVING STATIC WEAPON-TARGET ASSIGNMENT PROBLEM USING MULTI-START LATE ACCEPTANCE HILL CLIMBING

SELIN ALPARSLAN[1], EMRULLAH SONUÇ[1*]

[1] *Department of Computer Engineering, Karabük University, 78050, Karabük, Türkiye*

ABSTRACT. **A challenging methodology predicted in modern military strategies is the unprotected Weapon-Target Assignment (WTA) problem, where weapons under consideration must be assigned to targets in order to minimize the expected survivability attribute against the targets. In this case, this study is interested in the static WTA (SWTA) scenario, where the assignments are made on a one-time basis. Since the SWTA problem has been found to be of NP-complete nature, the more accurate solution techniques can be considered infeasible due to the escalating complexity. In this paper, it is proposed to extend the library of new methods by implementing the multi-start method and the technique called Late Acceptance Hill Climbing (LAHC). Performance comparisons between the Multi-Start Late Acceptance Hill Climbing (MLAHC) and LAHC algorithms, derived from different examples and problem sizes, prove that the MLAHC algorithm yields better quality solutions and higher reliability than the traditional LAHC algorithm for large problems. This strategy can be seen as a revolution in the process of analyzing military resource allocation towards the optimal level.**

1. INTRODUCTION

The Weapon-Target Assignment (WTA) problem is a complex optimization task that is aimed at allocating weapons to targets with the intended objective of either realizing the maximum anticipated damage of or minimizing the expected survival probability of targets. This problem exists in two main forms: static and dynamic. In the static form, weapons are allocated to the targets once way, and both the weapons and targets remain fixed for the duration of their assignment. On the other hand, the dynamic version allows modification of the assignments over some period of time and may allow many assignments [1]. Nonetheless, the Static WTA (SWTA) frameworkfocuses on minimizing optimization where the goal is to attain the most appropriate weapon to target allocation to deter the enemy's projected impact. This approach is based on the assessment of the organisational defensive environment and focuses on the best ways and means of applying the defensive resources available.

The assessment of expected damage for defense assets is carried out after their involvement in a battlefield scenario. A problem for a defensive mission in SWTA problem can generally be formulated as follows [2]:

E-mail address: selinalparslann@gmail.com (S.Alparslan), esonuc@karabuk.edu.tr(*) (E.Sonuç).
Key words and phrases. Combitonarial Optimization, Late Acceptance Hill Climbing, Weapon-Target Assignment Problem.

$$
\begin{aligned}
f(\pi) = \min \sum_{i=1}^{n} V_i \prod (1-p_{ij})^{x_{ij}} \\
s.t. \sum_{i=1}^{n} x_{ij} = w_i, \quad \text{for } i = 1, \cdots, m \\
x_{ij} \in \mathbb{Z}_+, \quad \text{for } i = 1, \cdots, m, \quad j = 1, \cdots, n.
\end{aligned} \tag{1}
$$

There are m types of weapons (w_i represented by $i = 1, ..., m$) available to counter n targets, represented by $j = 1, ..., n$. Each weapon type i is associated with a probability p_{ij} of eliminating target j, while each target j has a destruction value denoted by V_j. The decision variables x_{ij} signify the quantity of weapons of type i allocated to target j.

According to [3], SWTA problem is an NP-Complete. Like other assignment problems, e.g. the quadratic assignment problem [4], it is inherently difficult. In the context of SWTA problem, there are n^m potential permutations for allocating m weapons to n targets. The condition is that all weapons must be allocated. As the number of weapons and targets increases, this process becomes increasingly complex. It can be challenging to explore all possible solutions due to the exponential growth of the problem. Exact solution methods are insufficient for solving the SWTA problem due to its computational complexity. Therefore, metaheuristics, known for their efficiency and efficacy in discover the solution space to address complex problems, are preferred for yielding practical and often nearly optimal solutions.

This study proposes an improved methodology for solving the SWTA problem. It suggests combining a strategy that involves multiple starting points with late acceptance hill climbing. This approach has been shown to be an alternative option for obtaining quality solutions within reasonable computational timeframes. The multistart approach increases diversity in the search space, while the hill climbing approach focuses on exploiting local optima. The study is significant because it addresses a crucial obstacle in military mission planning: the optimal allocation of weapons to targets. This allocation is essential for operational success. The research is a significant advancement in the field because it highlights the importance of assigning weapons to targets to achieve operational success. The rest of the paper is structured as follows: Section 2 presents the state-of-the-art methods for solving the WTA problem. Section 3 describes the late acceptance hill-climbing algorithm and the proposed approach with its components. Section 4 reports the experimental results and the last section concludes the study and suggests directions for future work.

2. Related Work

In recent years, a large number of exact and approximate algorithms for solving the WTA problem have been studied [5–7]. Due to its complexity, the WTA problem may be too hard for exact algorithms to solve. However, metaheuristics help us to overcome this problem by producing good solutions in a reasonable time. Metaheuristics, which combine several algorithms, are algorithms designed to solve more complex optimization problems and can be applied to different optimization problems. Some metaheuristic algorithms, especially preferred for problems with a large number of solutions, can outperform exact methods and provide an optimal or near optimal solution in a reasonable time [8].

Several approaches to WTA have been studied in the literature. These include genetic algorithms, heuristic methods, and optimization techniques [9]. Exact algorithms based on mathematical programming have computational requirements that grow exponentially with the size of the problem [2]. Therefore, these algorithms are limited by some constraints. Recent research has been directed to dynamic situations and heuristic algorithms [10, 11]. In military operations, the efficient solution of the WTA problem is crucial. However, the complexity of the problem makes real-time optimal solutions impossible. Researchers are therefore working on heuristic algorithms such as genetic algorithms, simulated annealing, ant colony optimization, particle swarm optimization [12].

A branch-and-bound algorithm that combines lower bound methods with a search algorithm is proposed to solve the WTA problem. A combination of exact and heuristic algorithms for solving the WTA problem is presented, providing new methods and approaches for solving WTA in defense-related applications. Computational results are presented that demonstrate the ability to solve moderately large instances optimally and to obtain near-optimal solutions for fairly large instances within a few seconds. The ability to obtain optimal solutions for large instances in a short time is a significant achievement. [13]. A new exact algorithm for solving the WTA problem is presented. The algorithm incorporates new methods called weapon number limitation and weapon dominance to reduce the number of columns to be enumerated. The use of stage-dependent probabilities in WTA problems is proposed to optimize the allocation of weapons between different stages and targets. [14].

For the static version of the WTA problem, three approaches from the literature are presented to linearize the problem and transform it into linear optimization problems with complex numbers. The first approach can only be used as an approximation, the second approach fully linearizes the objective function of the WTA problem but is inferior to the solution time of the assignment problem, and the third approach exactly linearizes the objective function of the WTA problem. A special exact algorithm is proposed that avoids the difficulty of large dimensions. When a larger number of weapons are available for each weapon type, the optimization problems become intractable [15]. The modified Crow Search Algorithm (CSA) presents a new approach with a trial mechanism to improve the solution quality in solving WTA. The results show that the modified CSA performs better than the basic CSA and other state-of-the-art algorithms in most problem instances [16]. Another study has improved the previously proposed multi-objective evolutionary optimization algorithm by introducing an innovative approach. The proposed method consists of a Deep Q-Network (DQN) based mutation operator and a greedy-based matching operator.Experimental results show that the DQN-based mutation operator is successful in effectively identifying promising candidate solutions [17].

The WTA problem plays a central role in the improvement of military strategies and security mechanisms, and is characterized by its complicated nature stemming from the imperative requirement of optimal and competent resource allocation. Recent scientific work has witnessed a growing fascination with metaheuristic methods as a means to address the challenges posed by the WTA problem, as discussed in Kline's study [2]. Metaheuristics are preferred because they provide flexibility and efficiency in solving large and complex problems.

3. The Proposed Method

3.1. **Late Acceptance Hill Climbing:**

The Late Acceptance Hill Climbing (LAHC) Algorithm is a metaheuristic approach designed to address combinatorial optimization problems [18]. It evaluates recent solution history to decide whether to accept a new solution, treating each new solution as an improved version of the current one. The LAHC algorithm has proven effective in various domains, including the traveling salesman problem, scheduling, and timetabling problems. The late acceptance strategy is straightforward. The control parameter for the acceptance condition is derived from the search history. This heuristic resembles Hill Climbing but with a key difference: in Hill Climbing, a candidate solution is compared to the current solution, whereas in LAHC, a candidate solution is compared to a solution from several iterations in the past. LAHC follows an acceptance rule by maintaining a fixed-length list, L_h, which represents the history length and contains previous values of the current cost function. To determine whether to accept a candidate solution, the candidate cost is compared to the final element in the list. If the candidate cost is better, it is accepted. Upon acceptance, the list is updated by inserting the new current cost at the beginning and removing the last element from the end. This process ensures that the added current cost consistently reflects the present cost. The pseudocode of LAHC is outlined in Algorithm 1.

LAHC has a wide range of applications in various domains. Its primary application has been in course scheduling, where it optimizes the quality of schedules by significantly reducing the final solution value, demonstrating an ability to effectively handle complex scheduling constraints [18]. LAHC has been used to solve the unrelated parallel machine scheduling problem [19], the general lot sizing and scheduling problem with rich constraints [20], and the traveling salesman problem [21]. Furthermore, LAHC has been applied in the context of drone trajectory planning algorithms, where it demonstrates superior performance compared to conventional approaches by incorporating local search operators to improve the efficiency of path determination [22]. In addition, the use of LAHC has been integrated into the feature selection process, thereby enhancing the ability to utilize metaheuristic algorithms to reduce dimensionality in machine learning tasks [23]. These examples highlight the adaptability and effectiveness of LAHC in tackling complex optimization and trajectory planning problems.

3.2. **Multi-Start Late Acceptance Hill Climbing:**

The Multistart Late Acceptance Hill Climbing Algorithm (MLAHC) is one of the most advanced optimization techniques which is an enhancement of the existing LAHC as it not only considers single start points of the search space but also in using multiple starts points in the exploration domain. While, in LAHC, new solutions are accepted after a specific time-interval based on their fitness value, and thus, navigate away from local optima, MLAHC enhances this by beginning the search process with different random initial solutions. This, in turn, enhances the prospects of visiting different regions of the exploration space in pursuit of the near-optimum solutions through what has been referred to as the

Algorithm 1 The pseudocode of LAHC.

```
Input: maxIterations, L (length of history list), initialSolution
Output: bestSolution
Initialisation:
currentSolution ← initialSolution
bestSolution ← currentSolution
currentValue ← Evaluate(currentSolution)
historyList ← Array of size L initialized with currentValue
Loop for a fixed number of iterations:
for i ≤ maxIterations do
    neighborSolution ← GenerateNeighbor(currentSolution)
    neighborValue ← Evaluate(neighborSolution)
    if neighborValue ≤ currentValue or neighborValue ≤ historyList[i % L] then
        currentSolution ← neighborSolution
        currentValue ← neighborValue
    end if
    if currentValue ≤ Evaluate(bestSolution) then
        bestSolution ← currentSolution
    end if
    historyList[i % L] ← currentValue
end for
return bestSolution
```

multi-start strategy. In other words, MLAHC is different from the basic LAHC in that it have multiple initial solutions as opposed to LAHC's single-start nature, meaning that the exploration of the search space is going to be better and wider with multiple LAHC.

The MLAHC begins by initializing the number of iterations, acceptance period, and restarts, followed by generating an initial solution and setting up the acceptance history. The algorithm then enters the multistart loop, where at each restart it sets the initial solution as the current solution and resets the acceptance history. Within each restart, the iteration loop generates neighboring solutions, calculates their costs, and compares these costs to those in the acceptance history. If a neighboring solution's cost is less than or meets the acceptance criteria, it becomes the new current solution and the acceptance history is updated accordingly. The algorithm tracks the best solution from each restart and updates the global best solution when a superior solution is found. This process continues until all restarts and iterations are complete, ultimately returning the global best solution as the optimal solution found by the algorithm. By using multiple starting points, MLAHC enhances its ability to explore the search space more extensively than traditional LAHC. The flowchart of the MLAHC is shown in Figure 1.

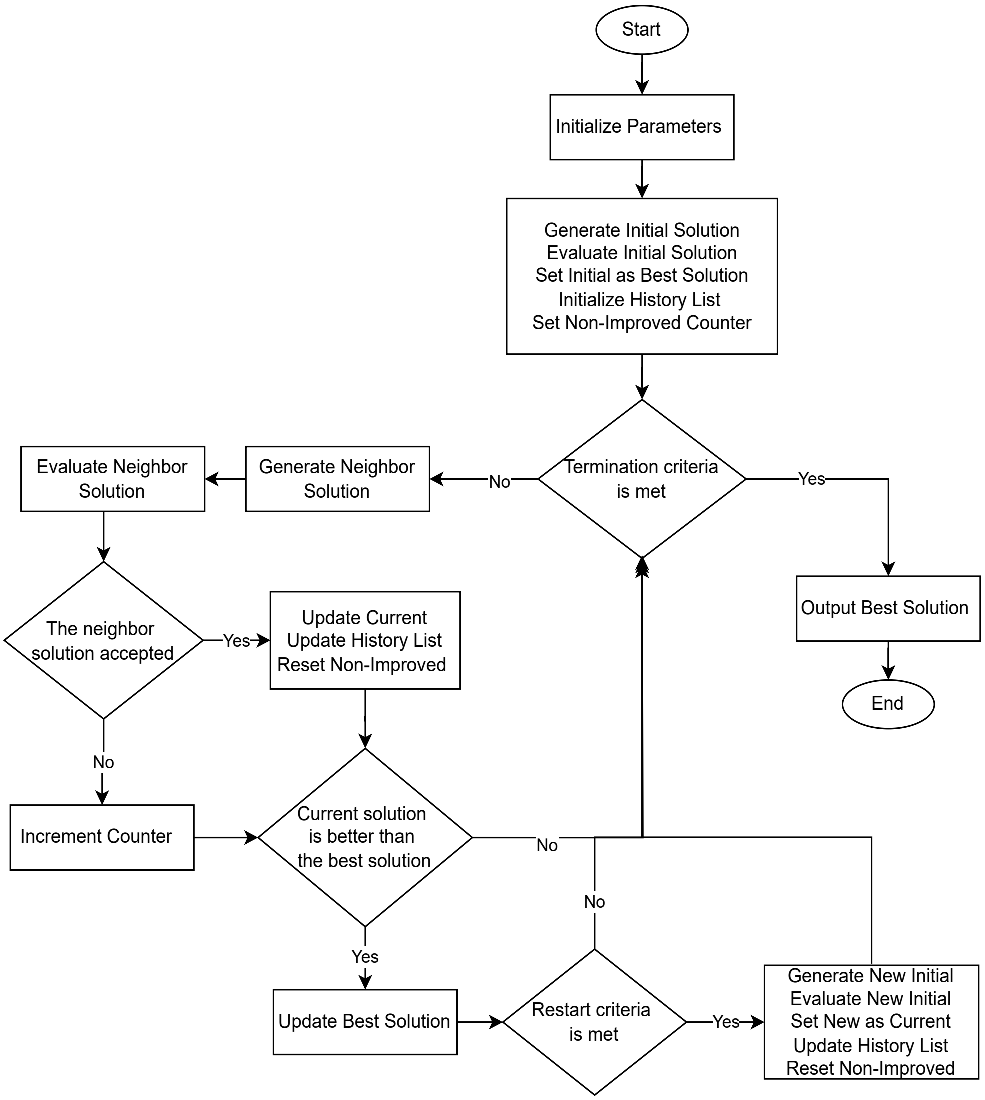

FIGURE 1. The flowchart of MLAHC.

4. EXPERIMENTAL RESULTS

MLAHC is tested on 12 WTA problem instances [7]. The results are given in different metrics: best, mean, median, worst and standard deviation (SD). Sizes of problem instances are in the range 5 and 200 and shown in Table 1. Results were collected from 10 independent runs. The numerical experiments were performed on a PC with 8.00 GB of RAM, MacOS 14.4.1 operating system. The MLAHC codes were written in the C programming language using CLion IDE v2023.3.4.

TABLE 1. WTA problem instances.

Instance	Number of Weapons	Number of Targets
WTA1	5	5
WTA2	10	10
WTA3	20	20
WTA4	30	30
WTA5	40	40
WTA6	50	50
WTA7	60	60
WTA8	70	70
WTA9	80	80
WTA10	90	90
WTA11	100	100
WTA12	200	200

Table 2 presents the results on small-scale WTA instances. The results of the MLAHC show that for WTA1, WTA2, and WTA3, the algorithm consistently achieves identical best, worst, mean, and median objective values, with an SD of 0.0000 across all configurations, indicating highly stable performance. In contrast, WTA4 shows more variability, especially with a history length of 1 and no restarts, resulting in higher and more variable target values and an SD of 6.4358. This means that there is a direct relation between the number of identified parameter configurations and the overall performance of the algorithm, notably when it is faced with more complex instances. Although the algorithm shows consistent performance on small instances (WTA1, WTA2, and WTA3) regardless of historical length and restart values, it requires careful parameter tuning when faced with more complex instances such as WTA4. In particular, adopting larger historical lengths and using restarts as enhancements could be seen as effective on the grounds of stability and optimality to these particular cases.

Table 3 presents the results on medium-scale WTA instances. For the WTA5 instance , the best, worst, mean, and median objective values show some variation across different history lengths and restart values, indicating that the algorithm's performance is somewhat sensitive to these parameters. The lowest mean objective value is 306.8923 for a history length of 1000 and a restart value of 500. For the WTA6 instance, the results also vary across configurations. The lowest mean objective value is 355.6795 for a history length of 1000 and a restart value of 1000. For the WTA7 instance , there is variability in the results, with the lowest mean objective value of 419.5174 achieved with a history length of 1000 and restart value of 1000. For the WTA8 instance, the results show significant variability, especially with a history length of 1 and no restart, leading to much higher objective values and standard deviation. The lowest mean objective value is 502.9574 with a history length of 1000 and a restart value of 500. As a result, for instances WTA5, WTA6, and WTA7, MLAHC consistently achieves better and more stable results with larger history lengths and higher restart values, suggesting that these configurations help the algorithm explore the solution space more effectively. For the WTA8 instance, the variability in results is more pronounced, especially with shorter history lengths and no restarts, resulting in higher objective values

TABLE 2. Experimental results on small-scale problem instances.

Instance	History Length	Restart	Best	Worst	Mean	Median	SD
WTA1	1000	1000	48.3640	48.3640	**48.3640**	48.3640	0.0000
	500	500	48.3640	48.3640	**48.3640**	48.3640	0.0000
	500	1000	48.3640	48.3640	**48.3640**	48.3640	0.0000
	1000	500	48.3640	48.3640	**48.3640**	48.3640	0.0000
	500	-	48.3640	48.3640	**48.3640**	48.3640	0.0000
	1000	-	48.3640	48.3640	**48.3640**	48.3640	0.0000
	1	-	48.3640	48.3640	**48.3640**	48.3640	0.0000
WTA2	1000	1000	96.3123	96.3123	**96.3123**	96.3123	0.0000
	500	500	96.3123	96.3123	**96.3123**	96.3123	0.0000
	500	1000	96.3123	96.3123	**96.3123**	96.3123	0.0000
	1000	500	96.3123	96.3123	**96.3123**	96.3123	0.0000
	500	-	96.3123	96.3123	**96.3123**	96.3123	0.0000
	1000	-	96.3123	96.3123	**96.3123**	96.3123	0.0000
	1	-	96.3123	96.3123	**96.3123**	96.3123	0.0000
WTA3	1000	1000	142.1070	142.1070	**142.1070**	142.1070	0.0000
	500	500	142.1070	142.1070	**142.1070**	142.1070	0.0000
	500	1000	142.1070	142.1070	**142.1070**	142.1070	0.0000
	1000	500	142.1070	142.1070	**142.1070**	142.1070	0.0000
	500	-	142.1070	150.2510	144.7579	144.0702	2.3100
	1000	-	142.1070	144.4690	143.3774	143.2416	0.7843
	1	-	164.5723	178.6062	173.3124	174.3449	4.5964
WTA4	1000	1000	248.0285	248.5817	**248.3479**	248.4051	0.1936
	500	500	248.2730	249.3956	248.5891	248.4222	0.3427
	500	1000	248.3312	249.0275	248.5460	248.4222	0.2646
	1000	500	248.0285	248.8386	248.3717	248.3476	0.2605
	500	-	249.9979	256.5385	253.2718	253.4581	2.2874
	1000	-	250.4865	257.0525	253.6127	253.7566	2.0373
	1	-	327.0574	346.8141	339.7976	340.0045	6.4358

and standard deviations. This suggests that for more complex or larger instances, a longer history length and the ability to restart the search process are critical to achieving optimal and consistent solutions.

Table 4 presents the results on medium-scale WTA instances. For the WTA9 instance, the best, worst, mean, and median objective values show some variation across different history lengths and restart values. The lowest mean objective value is 539.2292 with a history length of 1000 and a restart value of 1000. The SD values are relatively low for most configurations, indicating stable performance, except for configurations with shorter history lengths and no restarts. For the WTA10 instance, the results also vary, with the lowest mean objective value of 599.2728 achieved with a history length of 1000 and restart value of 1000. For the WTA11 instance, the results indicate variability, with the lowest mean objective value of 704.6850 with a history length of 1000 and restart value of 1000. For WTA10 and WTA11 instances, the SD values are also low for most configurations, indicating stable performance, but higher for shorter history lengths and no restarts. For the WTA12 instance, the variability in results is more pronounced, especially for a history length of 1 and no restart, leading to much higher objective values

TABLE 3. Experimental results on medium-scale problem instances.

Instance	History Length	Restart	Best	Worst	Mean	Median	SD
WTA5	1000	1000	306.5564	308.1392	307.2418	307.2133	0.4812
	500	500	306.2859	307.7959	306.9699	307.0481	0.4560
	500	1000	306.1562	308.5417	307.1399	306.9142	0.6765
	1000	500	306.0912	307.5720	**306.8923**	307.1220	0.5953
	500	-	308.0490	319.2780	313.5596	314.6806	3.8503
	1000	-	311.2144	320.5870	314.4265	313.8629	2.5515
	1	-	461.5889	488.0803	476.8728	477.8627	7.9504
WTA6	1000	1000	354.0916	356.8551	**355.6795**	355.6409	0.8638
	500	500	355.3909	356.7601	356.2280	356.3174	0.0536
	500	1000	355.1825	358.1991	356.6363	356.7981	0.8159
	1000	500	354.6224	356.2821	355.6847	355.7897	0.5000
	500	-	360.1330	370.9876	364.8432	364.1965	3.6817
	1000	-	357.4125	364.0606	360.0826	360.0549	1.9652
	1	-	545.3753	596.1006	577.8765	579.7290	14.9065
WTA7	1000	1000	418.5731	420.5899	**419.5174**	419.5085	0.5927
	500	500	417.1001	422.0817	419.6406	419.7593	0.0613
	500	1000	417.1754	421.1842	419.8538	420.4126	1.3062
	1000	500	417.4177	421.3109	419.7452	419.7255	1.1660
	500	-	425.9927	432.3662	428.6032	428.5293	2.0944
	1000	-	422.9075	435.2173	426.7079	426.1469	3.5601
	1	-	700.1578	732.6472	715.8937	713.3675	10.0763
WTA8	1000	1000	500.0615	504.5395	503.0515	503.4774	1.3633
	500	500	499.9063	505.1909	503.3437	503.7197	1.5449
	500	1000	501.4779	507.3864	504.4214	504.8288	2.0883
	1000	500	499.9062	504.4010	**502.9574**	503.6857	1.6086
	500	-	508.2395	519.8273	514.9432	515.8243	3.7444
	1000	-	504.3269	520.1396	511.7618	512.2577	5.7482
	1	-	864.0853	898.3761	884.8409	884.3876	10.5878

and standard deviation. The lowest mean objective value is 1,306.1270 with a history length of 1000 and a restart value of 1000. The SD is higher, especially for the history length of 1 and no restart, where the SD is 20.4028, indicating less stable performance and greater variability in results.

The performance of MLAHC on large instances is strongly influenced by the history length and restart parameters. For instances WTA9, WTA10, and WTA11, MLAHC consistently yields better and more stable results with larger history lengths and higher restart values. This suggests that these configurations help the algorithm explore the solution space more effectively. For the WTA12 instance, the variability in results is more significant, especially with shorter history lengths and no restarts, resulting in higher objective values and standard deviations. This suggests that for more complex or larger instances, using a longer history length and allowing the search process to restart are critical to achieving optimal and consistent solutions. Overall, adjusting these parameters is key to improving the performance of the algorithm. Longer history lengths and restarts are generally recommended for large instances to improve results.

TABLE 4. Experimental results on large-scale problem instances.

Instance	History Length	Restart	Best	Worst	Mean	Median	SD
WTA9	1000	1000	537.7873	541.1868	**539.2292**	539.2745	1.0036
	500	500	538.7035	15.2201	541.2645	541.5898	1.3981
	500	1000	539.4272	542.8226	541.1484	541.3882	1.3585
	1000	500	536.7075	541.1680	539.5932	539.9091	1.4354
	500	-	543.9957	554.6371	548.5667	548.8087	3.2537
	1000	-	540.7142	553.3719	545.6256	544.8100	3.8780
	1	-	935.5836	987.4912	969.9581	976.2804	16.8421
WTA10	1000	1000	598.0171	602.8778	599.8728	599.5441	1.4008
	500	500	597.1049	604.1977	601.0149	601.3492	0.0621
	500	1000	599.9423	602.5693	601.2927	601.2210	0.9187
	1000	500	597.2714	602.1404	**599.4435**	599.5426	1.4863
	500	-	606.5543	614.2934	610.2164	610.1721	2.5238
	1000	-	603.0792	615.5761	607.9658	607.2778	3.9888
	1	-	1,089.2605	1,134.0014	1,118.0351	1,123.5720	14.6743
WTA11	1000	1000	702.4334	706.9329	**704.6850**	704.7220	1.5023
	500	500	705.8282	709.3710	707.8575	707.7180	0.0810
	500	1000	704.9791	708.7655	707.3339	707.3468	1.1818
	1000	500	702.8853	707.9533	705.0482	704.8346	1.5613
	500	-	712.9811	721.9246	716.4566	715.2716	3.3886
	1000	-	704.3536	720.3480	712.7618	713.0143	5.0992
	1	-	1,297.7713	1,338.8655	1,325.7621	1,329.8928	13.7453
WTA12	1000	1000	1,304.0334	1,308.1425	1,306.1527	1,306.2038	1.2910
	500	500	1,306.3751	1,311.0284	1,308.4978	1,308.6416	0.1910
	500	1000	1,305.1196	1,310.4764	1,308.7313	1,309.3100	1.7343
	1000	500	1,300.8688	1,309.1603	**1,304.7562**	1,304.9484	2.4993
	500	-	1,311.4409	1,324.3238	1,319.0374	1,321.3748	4.6725
	1000	-	1,307.7223	1,318.4404	1,312.3279	1,312.8594	3.3418
	1	-	2,664.1240	2,727.3161	2,696.9456	2,698.5258	20.4028

5. DISCUSSION

WTA problem experimentation under small, medium, and large problem instances indicate that the effectiveness of MLAHC depends on the history length and restart parameters. In regard to small scenarios, the specific algorithm seems to be highly stable in terms of yielding near-optimal solutions and is almost inert to these settings. However, these parameters when complex, require adjustments that are more relevant, with the size of the problem advancing. In medium and large scenarios, the use of larger history length and inclusion of restarts, generally enhance and stabilize the performance by attaining better lower objective values with less variability. In the most difficult problem instances, a history length of 1000 combined with frequent restarts consistently produces the best results. Thus, for more complicated and extensive tasks, it regimens stable and longer histories, as well as organize restarts when using the MLAHC algorithm. It also helps give enhanced solution quality and reliability since the best possible solution is chosen from numerous different solutions.

A comprehensive evaluation over a range of problem sizes highlights the effectiveness and stability of the algorithm. However, the study's comparison is limited to traditional LAHC and lacks broader comparisons with other state-of-the-art algorithms. It focuses specifically on SWTA, which may limit generalizability, and does not investigate scalability or computational requirements for extremely large or real-time applications. The performance of the MLAHC algorithm is highly dependent on the history length and restart parameters, which require careful tuning, especially for larger and more complex problem instances. While the algorithm shows consistent performance on small instances, it requires more precise parameter settings to achieve efficient solutions as the problem size increases. In addition, the study focuses primarily on the static version of the WTA problem, and although it suggests potential applications in dynamic WTA problems, these areas are not explored in this paper. Despite these limitations, the practical relevance of the study to military resource allocation and the novel approach presented are significant contributions.

6. CONCLUSION

This paper aims to develop a new heuristic approach for solving the Static Weapon-Target Assignment (SWTA) problem incorporating the multistart strategy and Late Acceptance Hill Climbing. This new technique called Multistart Late Acceptance Hill Climbing (MLAHC) enhances the search mechanism coupled with optimization in the local optima, and they deliver the best quality solutions with high performance. As one can observe from tests on various scenarios of WTA problems, it can been seen that the MLAHC approach performs well. The simulation results indicate that this technique is also more efficient than conventional versions of LAHC and this becomes more evident when applied to larger and complicated problems.

This research also found that except for the history length and restart parameters, MLAHC has significantly high dependency on these two factors. For the small levels, MLAHC holds an ideal and constant performance in all environment settings. However, as problem size increase, it becomes paramount to tweak these parameters within the system. History length is longer and it has more restarts which prove that it provides better and improved results and it underlines the point that there should be much proper setting required to get the efficient solutions in the complex problems.

Thus, the MLAHC algorithm gives a strong and versatile method to the SWTA problem, which will significantly adds its value to the scopes of computational combinatorial optimization in general and the military operations study in particular. If this algorithm is applied in the dynamic WTA problem and other optimization problems in defence and other fields, then future research can be on these areas. Further enhancement of this algorithm can be done by combining several metaheuristic algorithms and hybridization of the strategies.

Data Statement WTA problem instances are available at `https://doi.org/10.17632/jt2ppwr62p.1`
Conflict of Interest The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.

REFERENCES

[1] R. K. Ahuja, A. Kumar, K. C. Jha, J. B. Orlin, Exact and heuristic algorithms for the weapon-target assignment problem, Operations research 55 (6) (2007) 1136–1146.

[2] A. Kline, D. Ahner, R. Hill, The weapon-target assignment problem, Computers & Operations Research 105 (2019) 226–236.

[3] S. P. Lloyd, H. S. Witsenhausen, Weapons allocation is np-complete., in: 1986 summer computer simulation conference, 1986, pp. 1054–1058.

[4] E. Sonuc, B. Sen, S. Bayir, A cooperative gpu-based parallel multistart simulated annealing algorithm for quadratic assignment problem, Engineering Science and Technology, an International Journal 21 (5) (2018) 843–849.

[5] Ö. Tolga, E. BOZKAYA, An evaluation on weapon target assignment problem, Journal of Naval Sciences and Engineering 18 (2) (2022) 305–332.

[6] H. Xing, Q. Xing, An air defense weapon target assignment method based on multi-objective artificial bee colony algorithm., Computers, Materials & Continua 76 (3) (2023).

[7] E. Sonuc, B. Sen, S. Bayir, A parallel simulated annealing algorithm for weapon-target assignment problem, International Journal of Advanced Computer Science and Applications 8 (4) (2017) 87–92.

[8] B. Chopard, M. Tomassini, An introduction to metaheuristics for optimization, Springer, 2018.

[9] A. Toet, H. de Waard, The Weapon-Target Assignment Problem, Citeseer, 1995.

[10] C. Wang, G. Fu, D. Zhang, H. Wang, J. Zhao, et al., Genetic algorithm-based variable value control method for solving the ground target attacking weapon-target allocation problem, Mathematical Problems in Engineering 2019 (2019).

[11] D. Guo, Z. Liang, P. Jiang, X. Dong, Q. Li, Z. Ren, Weapon-target assignment for multi-to-multi interception with grouping constraint, IEEE Access 7 (2019) 34838–34849.

[12] M. D. Rezende, B. S. P. De Lima, S. Guimarães, A greedy ant colony system for defensive resource assignment problems, Applied Artificial Intelligence 32 (2) (2018) 138–152.

[13] Exact and heuristic algorithms for the weapon-target assignment problem (2007). doi:10.1287/OPRE.1070.0440.

[14] A new exact algorithm for the weapon-target assignment problem (2021). doi:10.1016/J.OMEGA.2019.102138.

[15] A. C. Andersen, K. Pavlikov, T. A. Toffolo, Weapon-target assignment problem: Exact and approximate solution algorithms, Annals of Operations Research 312 (2) (2022) 581–606.

[16] E. Sonuç, A modified crow search algorithm for the weapon-target assignment problem, An International Journal of Optimization and Control: Theories & Applications (IJOCTA) 10 (2) (2020) 188–197.

[17] S. Zou, X. Shi, S. Song, Moea with adaptive operator based on reinforcement learning for weapon target assignment, Electronic Research Archive 32 (3) (2024) 1498–1532.

[18] E. K. Burke, Y. Bykov, The late acceptance hill-climbing heuristic, European Journal of Operational Research 258 (1) (2017) 70–78.

[19] M. Terzi, T. Arbaoui, F. Yalaoui, K. Benatchba, Solving the unrelated parallel machine scheduling problem with setups using late acceptance hill climbing, in: Asian Conference on Intelligent Information and Database Systems, Springer, 2020, pp. 249–258.

[20] A. Goerler, E. Lalla-Ruiz, S. Voß, Late acceptance hill-climbing matheuristic for the general lot sizing and scheduling problem with rich constraints, Algorithms 13 (6) (2020) 138.

[21] S. Clay, L. Mousin, N. Veerapen, L. Jourdan, Clahc-custom late acceptance hill climbing: First results on tsp, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2021, pp. 1970–1973.

[22] B. S. Shihab, H. N. Abdullah, L. A. Hassnawi, Improved artificial bee colony algorithm-based path planning of unmanned aerial vehicle using late acceptance hill climbing., International Journal of Intelligent Engineering & Systems 15 (6) (2022).

[23] B. Chatterjee, T. Bhattacharyya, K. K. Ghosh, P. K. Singh, Z. W. Geem, R. Sarkar, Late acceptance hill climbing based social ski driver algorithm for feature selection, IEEE Access 8 (2020) 75393–75408.