A Study of Jamming Resource Allocation Based on a Hyperheuristic Framework

Zelong Hao
Air Force Engineering University
Xing Wang (wang_xing1965@163.com)
Air Force Engineering University
Jundi Wang
Air Force Engineering University

Article

Keywords: Electronic warfare, Radar network, Jamming resource allocation, Hyperheuristics, Evolutionary algorithm

Posted Date: August 5th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1905905/v1

Abstract

There are massive data and rapidly changing battlefield situations in modern electronic warfare, which is a challenge to jamming resource allocation. It is difficult for the existing optimization algorithms to balance optimization capability and calculation speed at the same time. To solve this problem, this study proposes an improved genetic selection electronic warfare operator hyperheuristic (GAEWHH) algorithm. As an emergent optimization algorithm, the hyperheuristic (HH) framework has not previously been applied to the problem of jamming resource allocation. This is a two-level algorithm framework that can isolate problem domains. The high level uses an improved genetic algorithm to search the heuristic space, and four electronic warfare operators (EWOs) based on the problem domain are designed for the low level to search the solution space. Combining different EWOs can change the population diversity, evolution direction and algorithm complexity of the GAEWHH algorithm, which improves the algorithm performance to meet battlefield situation requirements. The experiment shows that for large-scale problems, the GAEWHH algorithm is better than the mainstream evolutionary algorithm in terms of optimization capability and better than Google OR-Tools in terms of calculation speed. In this way, the GAEWHH algorithm achieves a balance between optimization capability and calculation speed.

1 Introduction

Jamming is an important part of electronic warfare, and the allocation of jamming resources directly affects the overall jamming effect, which has always been a key research topic in the field of electronic warfare[1]. With the development of weapons and equipment, there are more types of electronic warfare equipment, larger data scales, and higher real-time requirements, imparting higher requirements for the allocation of jamming resources[2, 3].

Solving the jamming resource allocation problem generally includes two aspects. First, the actual jamming resource allocation task is reasonably simplified and transformed into a combinatorial optimization problem. Second, the combinatorial optimization problem is solved by an optimization algorithm, which is equivalent to solving the jamming resource allocation problem. An optimization algorithm applied to this problem needs to balance the optimization capability and calculation speed because combat involves both aspects. Better allocation of jamming resources than the enemy brings better electronic optimization results, and faster calculation speed accelerates the combat OODA loop, which gains battlefield advantages[4, 5]. Therefore, it is not necessary to obtain the global optimum for the optimization algorithm. An optimization algorithm that balances the calculation speed and optimization results is sufficient to adapt to the battlefield environment.

There are different ways to transform the jamming resource allocation problem into a combinatorial optimization problem (COP)[6], which is related to equipment properties and tactics. The radar network system, which is widely used instead of single radar, connects the sensors in the battlefield to each other, making the topology structure more complex[7]. The corresponding combat model exhibits multilevel and distributed characteristics[8, 9]. The concept and technology of cognitive radar are developing rapidly.

Compared with traditional radar, cognitive radar has faster and more changeable signal patterns and stronger capabilities and can adaptively manage combat resources, thereby making operations intelligent, which also changes the equivalent combat model[10–12].

COPs include vehicle routing problems[13], job shop scheduling problems[14], packing problems[15] and problems in other application fields. The COP of jamming resource allocation is a kind of packing problem and is also an NP-hard problem. Traditional algorithms such as the Hungarian algorithm[16], zero-one programming[17], fuzzy multiattribute dynamic programming[18] and close degree[19] are suitable for solving small-scale jamming resource allocation problems. Along with the fast development of cognitive electronic warfare[20], mosaic warfare[21], multidomain warfare[22] and other tactics, jamming resource allocation faces the search space explosion problem. Traditional optimization algorithms cannot meet the calculation speed requirement of the battlefield. Therefore, researchers have turned to various intelligent optimization algorithms. In a study conducted by Jiang et al.[23], a hybrid quantum-behaved particle swarm optimization and self-adjustable genetic algorithm (HQPSOGA) was employed to solve jamming resource allocation, which effectively generated a better overall jamming capacity. Zhang et al.[24] used particle swarm optimization (PSO) to solve the optimization problem transformed by joint jamming beam selection and power allocation (JJBSPA), which verified the JJBSPA strategy.

However, these traditional heuristics are usually defined for a particular problem domain. If the model, scene or problem domain changes, the heuristic may perform poorly. Hence, these traditional heuristics are problem specific. A hyperheuristic (HH) framework working in the heuristic space rather than the solution space can provide more generalized solutions to the COP. The HH framework is a two-level framework corresponding to two search spaces, of which the high level (HL) is the heuristic space and the low level (LL) is the solution space[25–27]. Thereby, this framework isolates the problem domain. The HL of the HH framework directly employs proposed heuristics in other applications, and LL heuristics that are problem dependent are designed by field experts. The HH framework performs well and has been widely used in various optimization fields for nearly 20 years[28]. In a study by Zhang et al.[29], a new genetic-based HH algorithm was employed to solve the optimization problem in cloud manufacturing. Vela et al.[30] used squared hyperheuristics to solve the job shop scheduling problem and demonstrated the flexibility of this algorithm.

At present, few studies have been performed on the use of the HH framework in the jamming resource allocation problem. This paper proposes a genetic selection electronic warfare operator hyperheuristic (GAEWHH) algorithm based on the HH framework to solve jamming resource allocation, which aims to balance optimization capability and calculation speed. The following section presents the jamming resource allocation model. The HH framework and the GAEWHH algorithm designed based on this framework are introduced in Section 3. Section 4 describes a contrast experiment and performance analysis based on the GAEWHH algorithm. Section 5 summarizes the findings and further possible extensions of this study.

2 Jamming Resource Allocation Problem

2.1 Problem Description

The task of jamming resource allocation is to formulate a strategy that determines the correspondence between jammers and radars. The goal of jamming is to maximize the whole effect. The working modes of electronic warfare equipment can be divided into three types according to the number of relevant participants, namely, one-to-one, many-to-one and one-to-many, which are analyzed separately below[31].

Assume that there are m jammers and n radars in the battlefield. A one-to-one model means that there exists a one-to-one correspondence relationship between jammers and radars, namely, one jammer can interfere with only one radar, and one radar is interfered with by only one jammer. Therefore, $m = n$.

A many-to-one model means that one radar is jammed by many jammers at the same time. Blinking jamming generally involves two jammers working together to jam a radar. Distributed noise jamming generally occurs when more than two jammers cooperate to jam a radar.

In the one-to-many model, one jammer jams multiple radars at the same time. Distant support jamming in air-to-ground warfare uses noise jamming, which has high power and a wide range to jam multiple radars on the ground. Another example is the jammer using phased array technology, which jams many radar targets by multiple beams.

The one-to-one model is the general model. The one-to-many or many-to-one working mode can be transformed into an equivalent one-to-one model according to combat scenarios, spatial position relationships, equipment types and other specific situations. Therefore, this paper considers a one-to-one equivalent model.

2.2 Model Assumptions

Electronic warfare is complex and has changed rapidly. To build mathematical models and analyze the problem, the following three reasonable assumptions are proposed.

First, we assume that the information on all jammers and radars in the battlefield is known. This is difficult to do in a real war because both sides of the war are trying their best to hide their information. Although battlefield information is asymmetric, in specific local scenarios, more comprehensive information can be obtained through reconnaissance activities.

Second, the analyzed equipment parameters are unchanged in a time window. The battlefield information changes rapidly over time. For the convenience of analysis, we assume that the parameters remain unchanged in a small space range and over a short period of time.

Third, we establish a one-to-one model; that is, the number of equivalent jammers and equivalent radars are equal. According to the previous analysis, both the one-to-many model and the many-to-one model can be transformed into a one-to-one model.

2.3 Mathematical Model

Assume that the jamming benefit matrix *E* consists of *m* jammers and *n* radars.

$$E = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ e_{m1} & e_{m2} & \cdots & e_{mn} \end{bmatrix}$$

1

where e_{ij} represents the jamming effectiveness between the *i*-th equivalent jammer and the *j*-th equivalent radar.

The objective function is defined as follows:

$$S = \max \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} e_{ij}$$

2

The purpose is to find the maximum value of the objective function, where x_{ij} is the weight between the *i*-th equivalent jammer and the *j*-th equivalent radar.

The constraint conditions are the following formulas:

$$\begin{cases} \sum_{i=1}^{m} x_{ij} = 1 & i = 1, 2, \ldots, m \\ \sum_{i=1}^{m} x_{ij} = 1 & j = 1, 2, \ldots, n \\ x_{ij} = 1 \quad or \quad x_{ij} = 0 \end{cases} \qquad (3)$$

In this way, the problem of jamming resource allocation is transformed into the COP. The decision variable is x_{ij}, a value of 0 indicates that the *i*-th equivalent jammer does not jam the *j*-th equivalent radar, and a value of 1 indicates that jamming is implemented.

3 Hh Framework

The HH framework is categorized into four classifications, namely, selection perturbative, selection constructive, generation perturbative, and generation constructive[27]. The GAWEHH algorithm in this study is a type of selection constructive algorithm.

Pillay et al.[27] introduce a general mathematical framework for describing the HH framework. Therefore, the GAEWHH algorithm in this paper is described by this mathematical model, and the two-level framework is presented in Fig. 1.

The HH framework includes HL configurations and LL configurations.

HL configurations:

- *P* - The optimization problem domain of the HL, which is problem independent.
- *h* – The HL heuristic configuration and solution of *P*. Here, *h** is the best of *h*, and *H* is the search space of *h*.
- *L* - A set of LL heuristics.
- *HLH* - The HL heuristic algorithm that creates *h* on *L*.
- *F*(*h*)->*R* The objective function of the HL used to evaluate the corresponding performance.

LL configurations:

- *p* - An optimization problem for a particular problem domain.
- *s* - The solution of p. Here, *s** is the optimal solution, and *S* is the solution search space.
- *f*(*s*)->*R* - An objective function used to evaluate the performance.
- *M*: *f*(*s*)->*F*(*h*) - A mapping function from *f* to *F*.

Figure 1 shows the working process of the HH framework. In the LL, *s* is a solution of *p* and belongs to the set *S*. The optimal solution is *s**. The function *f* maps *s* to a value that is used to evaluate the quality of *s* and takes an extreme value at *s**. In the HL, *h* is a solution of *P* and belongs to the set *H*. The optimal solution is *h**. The function *F* maps the solution *h* of the HL to a value *F*(*h*) used to evaluate the quality of *h* and takes an extreme value at *h**. The set of LL heuristic operators *L* can map *h* to *s*. The algorithm *HLH* in the HL constructs *h* from the set *L* by executing a series of operations. All *h* constitute the heuristic search space *H*. In this way, each *h* can obtain the corresponding *s*. The function of mapping *M* is to map the LL *f*(*s*) to the HL *F*(*h*), that is, *F*(*h*) = *M*(*f*(*s*)) or *M*: *f*(*s*)->*F*(*h*). Thus, the HL and LL are connected. In fact, the distinction between *M*, *F*, and *f* is convenient only to formally describe the HH framework. In practical applications, *M* and *F* can be omitted, and *f* is used instead.

In the HH framework, the HL plays a leading role. Therefore, the change in *F*(*h*) reflects the optimization direction of the HH framework. When *h** is found by *HLH* in *H*, *s** is obtained in the corresponding solution space *S*.

4 Gaewhh Algorithm Based On The Hh Framework

Based on the general HH framework in the previous section, the GAEWHH algorithm is innovatively designed for the jamming resource allocation problem. The HL uses an improved genetic algorithm to search the heuristic space by genetic operations. EWOs are heuristics that are problem domain dependent. The main work of this research is to design four EWOs related to the electronic warfare field, which constitute the LL algorithm framework. In addition, the HL part is improved to suit the EWOs. The pseudocode of the GAEWHH algorithm is provided in Fig. 2.

4.1 HL Configurations in the GAEWHH algorithm

Figure 3 shows the pseudocode of the HL. An improved genetic algorithm (GA) is used as the HLH in the HL. There are two main differences between the improved GA and the traditional GA. First, the chromosome of the traditional GA is a fixed length, while the chromosome length of the improved GA in this study can be changed. Second, the traditional GA is problem dependent, directly searching the solution space, and the chromosome itself is the solution of the problem. The improved GA searching the heuristic space is problem independent, and the chromosome is not the solution. Then, the replication, mutation and crossover operations of the improved GA are introduced in detail.

Replication

Roulette wheel selection is used to choose a parent chromosome from the population and then copy it. This is the same as the traditional GA replication operation.

Mutation

A chromosome is selected in the population by roulette wheel selection. We randomly select a gene on this chromosome as the mutated gene and then change it. This process is different from traditional genetic algorithm mutation, which acts on the problem domain. Chromosomal encoding is determined by the problem, and genes in the chromosome are thus the variables of question. In the HH framework, the chromosome of the improved GA is a combination of LL heuristic operators. Genes are heuristics that act on the problem domain.

Crossover

Tournament selection is used to choose two chromosomes as parents. One gene is randomly selected as the crossover point in each parent, and two offspring are generated. To ensure that all chromosomes are the same length, the two crossover points of the traditional GA are in the same position. However, the two crossover points of the improved GA are generally not the same, which results in different lengths of the two offspring. The improved GA is more flexible than the traditional GA.

4.2 LL Configurations in the GAEWHH algorithm

This section describes the key work of this study. The heuristic space is searched with the improved GA in the HL, and the heuristic combination h is obtained. In the LL, h needs to be decoded by the set of heuristic operators L to obtain the solution s of the problem domain p. Field experts need to design a heuristic operator for the set L according to the characteristics of the problem, which is related to the optimization ability of the HH algorithm. In this study, four heuristic EWOs are designed for set *L* according to the characteristics of jamming resource allocation. Then, the fitness function $f(s)$ of the solution s is designed on Formula (2), which is also the evaluation function in the HL. The algorithm process is presented in Fig. 4.

The heuristic operators in set *L* ensure not only optimality and the selection of better individuals in the population but also diversity of individuals so that the algorithm does not fall into a local optimum. Four EWOs were designed in this study, as described in detail below.

Best operator (B) - A global optimization operator that selects the best jammer–radar combination from all the available radars and jammers. The algorithm complexity of this operator is O(n^2).

First operator (F) - A local optimum operator that selects a jammer from the available jammers to jam the current radar so that the jamming effect for the current radar is the best. The algorithm complexity of this operator is O(n).

Suboptimal operator (S) - A local search operator. For the currently jammed radar, we select a jammer with the suboptimal jamming effect from the available jammers to jam it. The algorithm complexity of this operator is O(2n).

Random operator (R) - A jammer is randomly assigned to jam the current radar. This operator introduces random selection, thus guaranteeing diversity. The algorithm complexity of this operator is O(n).

The design idea of these operators is analyzed below. The B and F operators search the solution space from the global and local perspectives, respectively, which can search for excellent individuals from the population. The S and R operators are also necessary. Although they cannot search for the optimal individual, they can increase the diversity of the population. If only the B and F operators are used, the population loses diversity, which causes the algorithm to fall into a local optimum. If only the S and R operators are used, the population can maintain diversity, but there is a lack of excellent individuals in the group, which causes the algorithm to converge slowly or fail to converge. The following experiments verify that this idea is reasonable.

4.3 Move Acceptance Techniques

The hill climbing algorithm used in this study to accept offspring is an effective acceptance technique. The offspring produced in each iteration may be better or worse than the parent. To improve the evolutionary direction, the hill-climbing algorithm is used to accept the offspring. If the offspring is better than its parent, the offspring is returned; otherwise, the operation is repeated. If the specified iteration limit is exceeded, the last offspring is returned.

5 Experiment And Analysis

5.1 Experimental Environment and Configuration

Hardware configuration: 4.0 GHz, 4-core CPU, 8 GB memory.

The parameters of the GAEWHH algorithm are listed in Table 1.

Table 1
Parameters of the GAEWHH algorithm

Parameter	Value
Population Size	150
Tournament Size	5
No. of Generations	50
Mutation Rate	0.75
Crossover Rate	0.25
Initial Max. Length	10
Offspring Max. Length	20
Mutation Length	5
Iteration Limit	30

5.2 Methodology

Due to the different results of the evolutionary algorithm each time, the GAEWHH algorithm is evaluated by the Monte Carlo method, which is a type of statistical method.

In the experiment, 8 algorithms are compared to analyze their performance. There have been many studies on large-scale combinatorial optimization problems, from which we select two representative algorithms. First, OR-Tools developed by Google is a stable and effective optimization algorithm, which has been verified by a large number of experiments. Second, PSO is a typical evolutionary algorithm that is a hotspot of current research. Finally, based on the four EWOs, six operator combinations are designed for the LL, and the performance of these combinations is analyzed.

5.3 Data Sets

This study conducts experiments on data sets of different scales to examine the computational speed and optimality of the algorithm. To illustrate the experimental results, three groups of typical experimental data are displayed, namely, Data Set 1 (5×5), Data Set 2 (10×10), and Data Set 3 (400×400). Data Set 1 and Data Set 2 are reserved for comparison with the existing literature, and Data Set 3 is the maximum data scale allowed by the experimental equipment of this study.

5.4 Results and Analysis

The results of the comparative experiment are presented in Fig. 2.

Figure 5 consists of 9 subgraphs in 3 rows and 3 columns. Each column presents the results of the different data sets. The first row of subgraphs presents the convergence rate of the different algorithms to solve the problem. The second row shows the optimization capability of the algorithms. The third row presents the running time of the algorithms, which is used to analyze the complexity. Therefore, the experiments are analyzed from three aspects, namely, the convergence rate, optimization capability and algorithm complexity.

5.4.1 Convergence Rate

Convergence speed is one of the indicators for evaluating evolutionary algorithms; here, OR-Tools is not considered. The first row of subgraphs shows that when the data scale is relatively small, all algorithms can converge quickly. As the data scale grows, evolutionary algorithms require more iterations to converge. The average number of iterations of the GAEWHH algorithm, which includes FBSR, FSR, F, B, S and R, is less than 50, and the average number of iterations of the PSO algorithm is more than 150. Therefore, the convergence speed of the GAEWHH algorithm is generally better than that of PSO.

The main reason for this phenomenon is that operators in the HH framework can be flexibly designed for the LL according to the problem domain without compromising for the HL. Thus, the HH framework can better search the problem space. In the design of traditional heuristics, it is necessary to consider both the search behavior characteristics and the coding method of the problem domain. Therefore, the convergence ability of the HH framework is not worse than that of traditional heuristic methods.

5.4.2 Optimization Capability

The optimization capability is the most important index to evaluate the optimization algorithm. The results of each running of the evolutionary algorithm are generally different. Experiments were repeated and analyzed by statistical methods. Table 2 shows the results of 100 repeated experiments.

In the table below, a represents the mean, B represents the maximum value, and C represents the standard deviation.

Table 2
Optimization Capability

	Data Set 1			Data Set 2			Data Set 3		
	V_A	V_M	V_s	V_A	V_M	V_s	V_A	V_M	V_s
PSO	3.65	3.65	0.00	7.05	7.26	0.10	7063.46	7360.14	91.23
OR-Tools	3.65	3.65	0.00	7.26	7.26	0.00	11947.90	11947.90	0.00
FBSR	3.65	3.65	0.00	7.26	7.26	0.00	11876.86	11884.15	2.41
FSR	3.65	3.65	0.00	7.15	7.26	0.08	11846.20	11854.49	4.06
F	3.65	3.65	0.00	6.71	6.71	0.00	11812.36	11812.36	0.00
B	3.51	3.51	0.00	6.95	6.95	0.00	11851.68	11851.68	0.00
S	2.75	2.75	0.00	5.62	5.62	0.00	11650.63	11650.63	0.00
R	3.65	3.65	0.00	7.03	7.26	0.10	6704.33	6840.56	47.52

Here, V_A is the mean value, V_M is the maximum value, and V_s is the standard deviation.

As illustrated in Table 2, the best solution is found each time on Data Set 1 by the six algorithms, namely, PSO, OR-Tools, FBSR, FSR, F, and R. Because of the problem of falling into the local optimum, the GAEWHH algorithm with the B or S operator does not obtain the best solution. In the test of Data Set 2, only the OR-Tools and FBSR algorithms can obtain the global optimal solution every time they run. The PSO, FSR, and R algorithms can find the optimal solution in multiple runs, and the results fluctuate. The mean value of the FSR algorithm is better than that of PSO and more stable. In the third group of large-scale test data, only OR-Tools can obtain the global optimal solution, the optimization performance of FBSR and FSR is slightly weaker, and the mean and stability of FBSR are better than those of FSR. The mean value of PSO is small, and the results are unstable. In the third group of large-scale test data, only OR-Tools can obtain the global optimal solution, the optimization performance of FBSR and FSR is slightly weaker, and the mean and stability of FBSR are better than those of FSR. The mean value of PSO is small, only slightly better than that of the R algorithm, and the solution of each run is not stable.

Three sets of experiments show that evolutionary algorithms can find optimal solutions in small-scale problems but generally cannot find optimal solutions in large-scale problems and use feasible solutions instead. In large-scale problem solving, the optimization ability of the GAEWHH algorithm is approximately 1.6 times that of PSO and can reach 99.4% that of OR-Tools. The repeatability of the GAEWHH algorithm is also much better than that of PSO.

Table 2 also shows that the design of LL heuristic operators has a great influence on the optimization capability. The optimization capability of combining the four EWOs is always better than that of using one EWO alone, which proves that the design of the EWO is reasonable. The B and F operators search the solution space from the global and local perspectives, respectively, which can search for excellent individuals from the population. The S and R operators can increase the diversity of the population.

5.4.3 Algorithm Complexity

The algorithm complexity is analyzed by counting the running time of different algorithms to repeatedly solve the optimization problem. The statistical results are presented in Table 3.

Table 3
Time consumption statistics

	Data Set 1			Data Set 2			Data Set 3		
	V_A	V_M	V_s	V_A	V_M	V_s	V_A	V_M	V_s
PSO	0.464	0.636	0.019	0.468	0.717	0.026	8.250	8.400	0.047
OR-Tools	0.001	0.003	0.000	0.003	0.008	0.001	130.050	139.392	1.032
FBSR	0.055	0.157	0.016	0.101	0.234	0.019	1076.481	1096.387	8.277
FSR	0.052	0.140	0.014	0.086	0.181	0.015	47.146	48.468	0.563
F	0.047	0.145	0.014	0.073	0.156	0.014	43.669	46.486	0.317
B	0.052	0.133	0.014	0.098	0.186	0.015	1460.057	1463.537	1.810
S	0.052	0.220	0.021	0.080	0.171	0.016	58.394	58.927	0.175
R	0.058	0.125	0.012	0.100	0.192	0.016	45.297	46.266	0.308

Table 3 and the third row in Fig. 2 show that the GAEWHH algorithm and PSO as evolutionary algorithms need population iteration operations, and the computation time is longer than that of OR-Tools in the small-scale data set. However, the speed advantage of the evolutionary algorithm emerges in the large-scale Data Set 3. Then, the difference in the time consumption between the GAEWHH algorithm and PSO is explained. The GAEWHH algorithm is associated with the HH framework and is problem independent. The PSO algorithm is dependent on the problem, and it uses decimal encoding specifically for the problem domain to adapt it to the PSO search mode. Therefore, in large-scale data sets, PSO takes less time than the GAEWHH algorithm with the same iterations. However, in small-scale data sets, the calculation work is low, and PSO mainly spends time in the data preparation stage, which takes longer than the GAEWHH algorithm.

The LL of the GAEWHH algorithm includes six EWO combinations, namely, FBSR, FSR, F, B, S and R. The time consumption of different operator combinations in the three experiments is consistent with the theoretical analysis of the EWO complexity. Taking the experiment of Data Set 3 as an example, the algorithm complexity of the B operator is the highest, which is O(n^2), and takes the longest time with a mean time of 1460.057 s. The algorithm complexity of the S operator is O(2n), which is much less time consuming at 58.394 s than the B operator. The algorithmic complexity of the F and R operators is O(n), and the corresponding running time is also smaller than (but similar to) that of the S operator but. FSR is a mixture of three operators, whose algorithm complexity is greater than O(n) but less than O(2n). The running time of FSR should also be shorter than that of the S operator and longer than that of the F or R operator. The experimental results support this conclusion. Similarly, the algorithm complexity of FBSR is greater than O(2n) but less than O(n^2), and the time consumption should be slightly less than that of the individual B operator. Because of the characteristics of the evolutionary algorithm, the solutions of each run are different; that is, the heuristic combinations are different. The algorithmic complexity of the B operator is much larger than that of other operators, which causes great fluctuations in the running time with a standard deviation of 8.277 s. The experiment also reflects this feature. Therefore, the experimental results are consistent with the algorithmic complexity of the theoretical analysis.

5.5 Experimental Summary

The GAEWHH algorithm can effectively solve the jamming resource allocation problem. The convergence rate of this algorithm is not worse than that of the common heuristic algorithms.

In this study, four LL heuristic EWOs, namely, F, B, S, and R, are first proposed for jamming resource allocation. The EWOs can efficiently search the problem domain, and the optimization capability of combined operators is better than that of the individual operators.

The running time of the GAEWHH algorithm is related to the algorithmic complexity of the LL operator combination. The complexity of the LL operator combination is between that of the minimum-complexity operator and the maximum-complexity operator. The appropriate operator combination is selected according to the problem demand. For example, in the large-scale Data Set 3, choosing a combination of FSR operators can well balance optimization capability and calculation speed.

6 Conclusion

This study is the first to extend the HH framework to jamming resource allocation for electronic warfare. No relevant previous studies have been found thus far. The proposed GAEWHH algorithm can effectively solve the jamming resource allocation problem. Four problem-dependent EWOs are designed for the LL, namely, B, F, S and R. Among them, the B and F operators improve the evolutionary direction, and the S and R operators increase the diversity of the population.

The GAEWHH algorithm uses a two-level framework, which allows domain experts to focus on LL heuristic operator design for the problem domain and select the mature and proven HLH algorithms for the HH framework. This feature facilitates the rapid development of the optimization algorithm.

The GAEWHH algorithm can effectively balance the optimization capability and calculation speed of the optimization algorithm when solving large-scale problems. However, once the traditional optimization algorithm is developed for a specific problem, its optimization capability and calculation speed are fixed and cannot be changed flexibly. The jamming resource allocation problem always needs to flexibly adjust the performance of these two aspects according to the battlefield information. Therefore, the GAEWHH algorithm is suitable for the problem.

The HH framework enables a novel and rapidly developing algorithm, but its application in electronic warfare is still lacking. The GAEWHH algorithm is a type of selection constructive algorithm of the HH framework. Future work is planned that will involve other applications of the HH framework in this field and improve the mathematical model of jamming resource allocation to make it more suitable for the actual battlefield.

Declarations

Data availability

All data generated or analyzed during this study are included in this published article and its supplementary information files.

References

1. Spezio, A. E., Electronic warfare systems. Ieee Transactions on Microwave Theory and Techniques **2002**, *50* (3), 633–644.
2. Riihonen, T.; Korpi, D.; Rantula, O.; Rantanen, H.; Saarelainen, T.; Valkama, M., Inband Full-Duplex Radio Transceivers: A Paradigm Shift in Tactical Communications and Electronic Warfare? IEEE Commun. Mag. 2017, *55* (10), 30–36.
3. Shao, G. Q.; Chen, Y. S.; Wei, Y. S., Convolutional Neural Network-Based Radar Jamming Signal Classification With Sufficient and Limited Samples. IEEE Access 2020, *8*, 80588–80598.
4. Huang, J. M.; Gao, D. P.; Chuqin In *Research on Combat SD Model based on OODA Loop*, Conference on Systems Science, Management Science and System Dynamics, Shanghai, PEOPLES R CHINA, May 29–31; Publishing House Electronics Industry: Shanghai, PEOPLES R CHINA, 2009; pp 13–18.
5. Revay, M.; Liska, M. In *OODA Loop in Command & Control Systems*, Communication and Information Technologies (KIT), Vysoke Tatry, SLOVAKIA, Oct 04–06; Ieee: Vysoke Tatry, SLOVAKIA, 2017; pp 127–130.
6. Peres, F.; Castelli, M., Combinatorial Optimization Problems and Metaheuristics: Review, Challenges, Design, and Development. Appl. Sci.-Basel 2021, *11* (14), 39.
7. Griffiths, H.; Farina, A.; Ieee In *Multistatic and Networked Radar: Principles and Practice*, IEEE Radar Conference (RadarConf), IEEE Atlanta Sect, ELECTR NETWORK, May 08–14; Ieee: IEEE Atlanta Sect, ELECTR NETWORK, 2021.
8. Wang, X. T.; Huang, T. Y.; Liu, Y. M., Resource Allocation for Random Selection of Distributed Jammer Towards Multistatic Radar System. IEEE Access 2021, *9*, 29048–29055.
9. Wu, Z.; Hu, S.; Luo, Y.; Li, X., Optimal distributed cooperative jamming resource allocation for multi-missile threat scenario. IET Radar, Sonar & Navigation 2022, *16* (1), 113–128.

10. Haykin, I., Cognitive radar - A way of the future. IEEE Signal Process. Mag. 2006, *23* (1), 30–40.
11. Gurbuz, S. Z.; Griffiths, H. D.; Charlish, A.; Rangaswamy, M.; Greco, M. S.; Bell, K., An Overview of Cognitive Radar: Past, Present, and Future. IEEE Aerosp. Electron. Syst. Mag. 2019, *34* (12), 6–18.
12. Li, K.; Jiu, B.; Wang, P. H.; Liu, H. W.; Shi, Y. C., Radar active antagonism through deep reinforcement learning: A Way to address the challenge of mainlobe jamming. Signal Process. 2021, *186*, 15.
13. Golden, B.; Raghavan, S.; Wasil, E., THE VEHICLE ROUTING PROBLEM: LATEST ADVANCES AND NEW CHALLENGES Preface. In *Vehicle Routing Problem: Latest Advances and New Challenges*, Golden, B.; Raghavan, S.; Wasil, E., Eds. Springer: New York, 2008; Vol. 43, pp V-VI.
14. Vela, A.; Cruz-Duarte, J. M.; Ortiz-Bayliss, J. C.; Amaya, I., Tailoring Job Shop Scheduling Problem Instances Through Unified Particle Swarm Optimization. IEEE Access 2021, *9*, 66891–66914.
15. Zhou, S. C.; Li, X. P.; Zhang, K. K.; Du, N., Two-dimensional knapsack-block packing problem. Appl. Math. Model. 2019, *73*, 1–18.
16. Gang, W.; Zhenyu, C.; Ni, P., Implementation of Hungarian algorithm for radar jamming resource allocation. Ship Electronic Engineering 2011, *31* (01), 87–91 + 94.
17. Yang, S.; Yongguang, C.; Xiuhe, L. I., Research on Optimal Distribution of Radar Jamming Resource Based on Zero-one Programming. Acta Armamentarii 2007, *28* (5), 528–532.
18. Ren, S.; Si, C.-z.; Lei, J., Fuzzy multi-attribute dynamic programming model for radar's disturbing machine allocation. Syst. Eng. Electron. (China) 2008, *30* (10), 1909–1913.
19. Lu, Y.-s.; Wang, S.-z.; Wang, X.-w.; Wang, J.-f., Study on the allocation tactics for radar jamming resources based on close degree. Syst. Eng. Electron. (China) 2005, *27* (11), 1893–1974.
20. Mitola, J.; Maguire, G. Q., Cognitive radio: Making software radios more personal. IEEE Pers. Commun. 1999, *6* (4), 13–18.
21. Jasper, S.; Hollingshead, T., Mosaic Warfare Networks Can Serve Naval Expeditionary Forces. Signal 2022, *76* (5).
22. Flack, N.; Reith, M. In *Self-Directed Learning Tools in USAF Multi-Domain Operations Education*, 18th European Conference on Cyber Warfare and Security (ECCWS), Univ Coimbra, Coimbra, PORTUGAL, Jul 04–05; Acad Conferences Ltd: Univ Coimbra, Coimbra, PORTUGAL, 2019; pp 752–759.
23. Jiang, H.; Zhang, Y.; Xu, H., Optimal allocation of cooperative jamming resource based on hybrid quantum-behaved particle swarm optimisation and genetic algorithm. IET Radar, Sonar & Navigation **2017**, *11* (1), 185–192.
24. Zhang, D. L.; Sun, J.; Yi, W.; Yang, C. X.; Wei, Y. Q.; Ieee In *Joint Jamming Beam and Power Scheduling for Suppressing Netted Radar System*, IEEE Radar Conference (RadarConf), IEEE Atlanta Sect, ELECTR NETWORK, May 08–14; Ieee: IEEE Atlanta Sect, ELECTR NETWORK, 2021.
25. Burke, E.; Kendall, G.; Newall, J.; Hart, E.; Ross, P.; Schulenburg, S., Hyper-heuristics: An emerging direction in modern search technology. In *Handbook of metaheuristics*, Springer: 2003; pp 457–474.
26. Burke, E. K.; Gendreau, M.; Hyde, M.; Kendall, G.; Ochoa, G.; Ozcan, E.; Qu, R., Hyper-heuristics: a survey of the state of the art. J. Oper. Res. Soc. 2013, *64* (12), 1695–1724.

27. Pillay, N.; Qu, R., *Hyper-heuristics: theory and applications*. Springer: 2018.
28. Sanchez, M.; Cruz-Duarte, J. M.; Ortiz-Bayliss, J. C.; Ceballos, H.; Terashima-Marin, H.; Amaya, I., A Systematic Review of Hyper-Heuristics on Combinatorial Optimization Problems. IEEE Access 2020, *8*, 128068–128095.
29. Zhang, S.; Xu, Y. B.; Zhang, W. Y., Multitask-oriented manufacturing service composition in an uncertain environment using a hyper-heuristic algorithm. J. Manuf. Syst. 2021, *60*, 138–151.
30. Vela, A.; Cruz-Duarte, J. M.; Ortiz-Bayliss, J. C.; Amaya, I., Beyond Hyper-Heuristics: A Squared Hyper-Heuristic Model for Solving Job Shop Scheduling Problems. IEEE Access 2022, *10*, 43981–44007.
31. Cui, B. F., *Radar Anti-jamming Effectiveness Evaluation*. Beijing: PHEI: 2017; p 604.

Figures

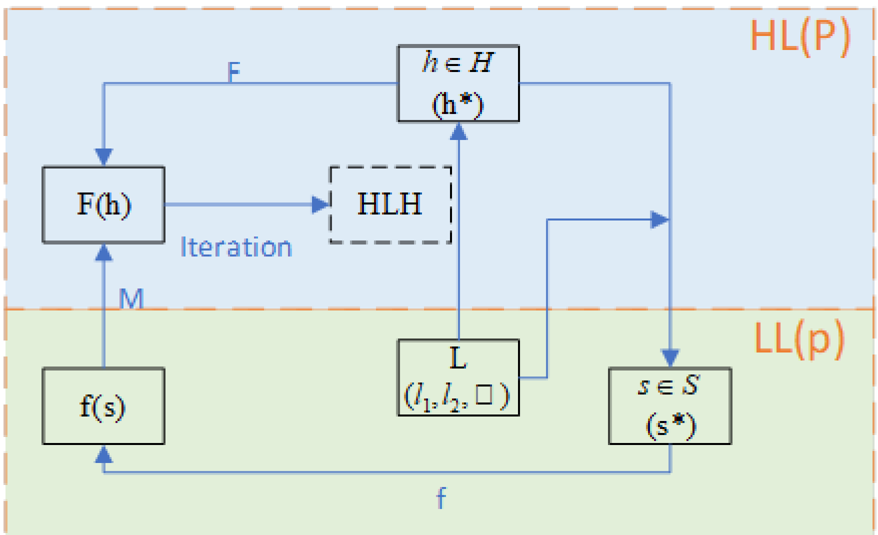

Figure 1

Two-level framework of the HH framework

Algorithm 1 GAHH

Create an initial population P
repeat
  Evaluate the population P:
  for $i = 1 \rightarrow length(P)$ **do**
    Initialize an empty solution s
    for $j = 1 \rightarrow length(h_i)$ **do**
      Apply the EW Operator l_j in h_i to extend the solution s
    end for
    Generate the low-level solution s
    Evaluate s
    Evaluate the corresponding high-level chromosome h_i by a mapping function M
  end for
  Select parents by Tournament method
  Apply genetic operators to the parents to create offspring
  Accept offspring P_{new} by hill-climbing algorithm
  $P \Leftarrow P_{new}$
until Terminate criteria are met

Figure 2

Pseudocode of the overall algorithm

Algorithm 2 High Level Algorithmic Framework by Genetic Algorithm

Create an initial population P
repeat
  Evaluate the population P:
  Select parents by Tournament method
  Apply genetic operators to the parents to create offspring P_{new}
  $P \Leftarrow P_{new}$
until Terminate criteria are met

Figure 3

Pseudocode of the HL algorithm framework

Algorithm 3 Low Level Algorithmic Framework with EW Operator

```
procedure CREATESOLUTION(h)
    Initialize an empty solution s
    for j = 1 → length(h) do
        Apply the EW Operator l_j in h to extend the solution s
    end for
    Evaluate the low-level solution s
    return s
end procedure
```

Figure 4

LL algorithm framework

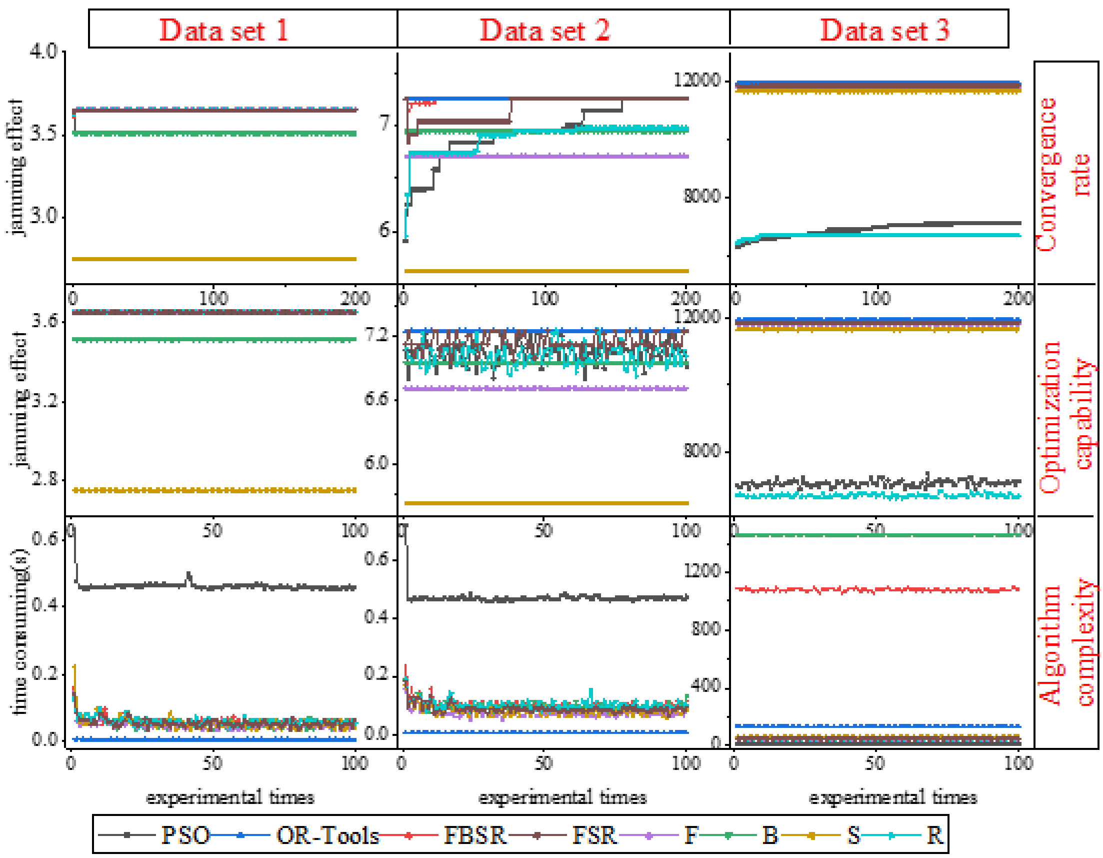

Figure 5

Comparative experiment of the 8 algorithms

Supplementary Files

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

- rawdata.zip