



OPEN Public security patrol path planning recommendation method based on wolf-pack optimization algorithm using DAF and BRS

Dongxing Wang^{1,2}, Gaomin Zhang³✉, Fangbo Wu¹, Zhiyang Huang¹ & Yiping Chen¹

The public security patrol path planning plays a key role in public security work, however, existing ones has varying degrees of shortcomings. To address these shortcomings, this paper proposes a public security patrol path planning recommendation method based on an improved wolf-pack optimization algorithm (S3PRM-DAF-BRS-CWOA). Firstly, an optimization objective function regarding the public security patrol path planning (S3P-Function) was abstracted from the actual situation; Moreover, this paper proposed an improved wolf-pack optimization algorithm named DAF-BRS-CWOA using Dynamic-Adjustment-Factor(DAF) and Balanced-Raid-Strategy (BRS), and DAF devoted to adjust the overall wolf-pack running strategy by dynamically adjusting the number of airdropped wolves during the stage of Summon-Raid while BRS with symmetric property was to improve both the algorithm's global exploration as well as the local development capabilities by increasing the number of checking locations, that means not only checking the reverse position of the current wolf, but also the positions generated according to certain rules between the reverse position of the current wolf and the current optimal wolf during the stage of Summon-Raid; Finally, DAF-BRS-CWOA was adopted to optimize S3P-Function, forming a public security patrol path planning recommendation method based on DAF-BRS-CWOA (S3PRM-DAF-BRS-CWOA). 30 independent numerical experiments were conducted on 20 public classical datasets by DAF-BRS-CWOA compared with GA, PSO, WDX-WPOA. The results indicated that the time-spent of DAF-BRS-CWOA was shortened and the optimization accuracy is the highest as well as S3PRM-DAF-BRS-CWOA has the best effect and is more time-efficient for solving public safety patrol path planning.

Keywords Public security patrol, Path planning, Swarm intelligence, Wolf optimization

With the rapid development of human civilization, the public safety situation is also facing new and severe challenges. To create a good public safety environment, the public security patrol path planning plays an important role, which considers factors such as time, road conditions and distance, aiming to provide the shortest security patrol route for police to achieve higher duty efficiency. At present, the core challenges of public security patrol path optimization revolve around “resource allocation” and “integration of technology”¹. Effective patrol strategies are essential to manage limited police resources while minimizing emergency response times¹, therefore, route optimization algorithms are needed to improve patrol efficiency and effectiveness². These issues often involve rational resource utilization, cost minimization, or performance maximization objectives³ to achieve effective deployment of police resources and the use of advanced technologies to improve operational efficiency.

In light of the above-mentioned needs for police patrols, many scholars have reflected on the development of the policing model of public security patrols and the optimization of patrol routes. For example: Wang et al. designed a solution for collaborative execution of area coverage tasks by multiple unmanned ships for large-scale sea patrol tasks based on the multi unmanned ship area coverage traversal algorithm⁴; Xiang et al. designed an improved Partheno-genetic Algorithm (IPGA) for quickly obtaining high-quality patrol task planning schemes⁵; Li used the Rural Postman Problem (RPP) to construct a graph theory model for traffic

¹Department of Computer and Information Security Management, Fujian Police College, Fuzhou 350007, China.

²School of Intelligence Science and Technology, University of Science and Technology Beijing, Beijing 100083, China. ³College of Information Engineering, Pingdingshan University, Pingdingshan 467002, China. ✉email: comyzhang@163.com

police patrol path optimization based on the CE2 heuristic algorithm, effectively shortening the path length and saving costs⁶; Both Zhang⁷ and Kong et al.⁸ achieved path planning by improving ant colony algorithm and using distance discriminant analysis to select multiple parameter indicators; Jie et al. proposed a new multi-agent based reinforcement learning algorithm to plan the optimal patrol path under arbitrary conditions of multiple defenders and attackers⁹; Liang adopted a combination of ant colony algorithm and genetic algorithm to improve the efficiency of sightseeing path planning in the intelligent navigation system of the park¹⁰; Shi et al. combined evolutionary algorithms with optimization experiments to establish a two-layer optimization model, and combines multi-agent genetic algorithms to solve multi-objective optimization scheduling problems in complex environments¹¹; Yan et al. combined particle swarm optimization algorithm with predictive control, overcoming the disadvantage of traditional particle swarm optimization algorithm in generating infeasible paths during particle update process¹²; Authors developed a two-layer optimization framework to solve the patrol police force allocation problem, and then used a new iterative Bender decomposition method to solve the potential optimization problem, thereby solving the spatial police patrol allocation problem with expected crime response time¹³. However, various algorithms exhibit inherent limitations.

Over the past few decades, wolf-pack optimization algorithm has attracted widespread interest from researchers due to its high parallelism, good global convergence, and computational robustness, making it particularly suitable for solving high-dimensional, multi-modal complex functions¹⁴; For instance, Tang et al. developed an automatic detection scheme for wolf pack hunting behavior in videos¹⁵, which greatly improves the accuracy of motion state recognition, reaching up to 88%; Duan et al. proposed a target allocation method based on wolf behavior mechanism¹⁶, which effectively solves the problem of collaborative target allocation among drone groups; In paper¹⁷, an entropy-based grey wolf optimizer (IEGWO) algorithm was proposed to solve global optimization problems; Chen et al. developed a Hybrid Grey Wolf Optimizer (HGWO) that combines Halton sequences and multiple strategies, which can be used to solve practical engineering problems¹⁸; Authors modelled the behavioral characteristics of wolf packs in natural environments and used it to solve the dynamic task allocation problem of drone swarms in complex scenarios¹⁹; She et al. improved the wolf-pack optimization algorithm to enhance the accuracy and efficiency of solving highly nonlinear black box function structural reliability problems²⁰; In paper²¹, researchers proposed a submarine cable positioning algorithm based on Improved Grey Wolf Optimization (IGWO), which achieves accuracy and effectiveness in submarine cable positioning.

Therefore, the effective utilization of the wolf-pack optimization algorithm mentioned above inspires us to intend to adopt an improved wolf-pack optimization algorithm to solve the optimization problem of public security patrol path planning, then this paper proposes a public security patrol path planning recommendation method based on an improved wolf-pack optimization algorithm (S3PRM-DAF-BRS-CWOA).

This paper is organized as follows. Section “[Related works](#)” provides the related existing works about this new proposed method. Section “[Improvement and design of the new proposed method](#)” elaborates on the improvement and design of the new proposed method. Section “[Performance verification experiment](#)” shows performance verification experiment and analysis of experimental results. Section “[Conclusions](#)” gives the conclusion of this article, summarizing its contributions, existing limitations, and future directions for improvement.

Related works

Swarm intelligence optimization

In nature, a variety of creatures with limited individual abilities demonstrate remarkable capabilities through group formations and inter-individual information exchange, such as ant colonies, bee colonies, and wolves. The swarm intelligent optimization algorithm is designed by simulating the characteristics of biological group behaviors in nature. It uses simple rules and interactions between many bodies to optimize the optimization problem of complex functions. The origin of the swarm intelligent optimization algorithm dates back to the 1980s. At that time, some scholars began to study the calculation model of simulating the behavior of natural biological groups. In 1986, Garnier and others proposed an Ant Random Walking Model²², which can be regarded as an early form of the swarm intelligent optimization algorithm. The swarm intelligent optimization algorithm provides new ways and shows excellent performance when solving many practical problems.

In recent years, in order to solve the limitations of traditional swarm intelligent algorithms, such as precocious convergence and the contradiction between convergence speed and accuracy, scholars have proposed many new types of swarm intelligent optimization algorithms. These new algorithms have shown stronger applicability and experimental effects in solving complex practical problems, such as Battle algorithm proposed by British scholar yang²³. These new algorithms exhibit enhanced capability in searching for optimal solutions and possess strong adaptability to various optimization problems. However, there are still many shortcomings. For example, the adjustment of bat algorithm in certain parameters may affect the performance of the algorithm. There are few cases in practical applications, and more empirical research is required to verify its effects.

Wolf-pack optimization algorithm

Among the new algorithms that have emerged in recent years, the wolf-pack optimization algorithm is a typical representative. Wolf-Pack Optimization Algorithm (for short: WPOA) was first proposed by YANG et al.²⁴ by imitating the habit of imitating the group of raw animals, the intelligence shown in the simulation of the wolf-pack foraging and hunting in nature, mainly the three intelligent behaviors of the wolf hunting behavior, calling-galloping behavior and siege are abstracted to find the optimal solution to the problem. The algorithm mainly includes the following five core processes.

Initialization The algorithm first needs to set a series of parameters such as the size of the wolves, the location of each wolf, the step length, and the number of iterations. It is an important basis for the algorithm. Appropriate

parameter settings can help algorithms to converge to optimal solutions faster, while maintaining sufficient search diversity to avoid local optimal. During the optimization process, each wolf in the pack is assigned a specific role based on its function within the algorithm. Assuming that the number of wolves is n , and the dimension of the interpretation of the space is D , then the position of each wolf is as required as a formula (1).

$$x_{id} = rang_{min} + rand(N, 1) * (rang_{max} - rang_{min}) \quad (1)$$

Migration-mechanism The process of searching the prey during the hunting of the wolf-pack in the natural environment, and searching the target separately in a certain field of a certain field of the wolf-pack members. In this process, each wolf searches the target in its nearby space according to the rules. The best Q -Po Wolf except the wolf in the solution space is regarded as the exploring wolf. The exploring wolf is centered on its own position, and the long -term searches are used as a search step to explore the prey around them. Degree value. If the adaptation value of the current wolf is greater than the point of adaptation of the position of the wolf, it indicates that the prey is relatively close to the wolf I and may capture the prey. The hunting process, that is, the current exploring wolf moves in the direction of H hunting and calculates the new location and the new adaptation value. For example, the formula (2) indicates that after moving the direction of P ($P=1,2, \dots, H, H \leq D$), the current wolf i is currently a new position in the D -dimensional space.

$$x_{id}^p = x_{id} + \sin\left(2\pi * \frac{p}{h}\right) * step_a^d \quad (2)$$

Summon-raid-mechanism After discovering the prey, the simulation wolf-pack will summon other wolves to quickly stride and participate in hunting by calling them by calling or other ways to increase the power of hunting. According to the formula (3), it is updated on the road of raid. If the new position obtained, the adaptation value is better than the current position, the wolf will move to the new position, otherwise it will stay in the current position. Among them, Z_{id} represented the position of the No. D -dimensional renewal of the No. i wolf in the wolf-pack; X_{id} is the current position of the D -dimension of the I wolf; RAND is a random number evenly distributed in the interval $[-1,1]$; Stepb is the step-size in the process of Summon-Raid; X_{id} is the position of the leader wolf in d -th dimension.

$$z_{id} = x_{id} + Rand \times stepb \times (x_{id} - x_{id}) \quad (3)$$

Siege-mechanism During the capture process, the wolf-pack appeared around the leader wolf to adopt a small step to siege prey. That is, first generate a random number of RM in $[0,1]$. If RM is smaller than θ (θ is a pre-set threshold), then the I wolf does not move. The wolf surrounded the prey with the leader wolf. The updated wolf position x_i^{t+1} is shown in the formula (4), $Stepc$ means the siege length; X_j is the position of the head wolf.

$$X_i^{t+1} = \begin{cases} X_i^t r_m < \theta \\ X_l + Rand \times stepc r_m > \theta \end{cases} \quad (4)$$

Regeneration-mechanism Simulation of the natural evolution of wolves is a key link in the iteration of the algorithm. Reserve strong performance in the wolves, eliminate wolves with poor performance, and then randomly supplement a certain number of artificial wolves. Update the population of the wolf by evaluating the adaptation value in the population, and the suspension conditions during iteration can be set. It ensures that the wolf clusters can adapt to evolution and eventually converge to the optimal solution, which improves the efficiency and effect of algorithms when solving complex optimization problems.

S3P-function

In the real social life, the task of public security patrols requires that the patrol paths of police officers must cover important nodes, and the number of important nodes is often large, so this results in multiple actual patrol paths with varying distances and disorganization, as shown in Fig. 1a. As a positive outcome, police officers often have rich patrol experience and have their own “optimal patrol path” based on this kind of experience, as shown in Fig. 1b. However, the “optimal patrol path” based on police officers’ experience shown in Fig. 1b is not truly optimal, especially in the case of a large number of important patrol nodes resulting to huge number of patrol path options, it is unlikely to find the globally optimal one due to limitations in human mental arithmetic ability, and Fig. 1c gives the true global optimal patrol path compared to Fig. 1b.

It is based on the above problem that this article abstracts a problem model serving the planning of public security patrol paths. Without loss of generality, assuming that the security patrol task is carried out in a plain area and ignoring the influence of altitude, that is, the Euclidean distance is used to calculate the distance between all key points; thereby, this paper constructs an objective function of public security patrol path planning (S3P-Function), the calculation formula is as Equation 5 (This paper takes a 2000 km by 2000 km area and 30 security patrol locations generated randomly).

$$Distance_{total} = \sqrt{x_{p_n} - x_{p_1}^2 + y_{p_n} - y_{p_1}^2} + \sum_{n=2}^N \sqrt{x_{p_n} - x_{p_{n-1}}^2 + y_{p_n} - y_{p_{n-1}}^2} \quad (5)$$

where, $Distance_{total}$ means the whole distance of some a public security patrol path; p_n means the n -th patrol point along some a security patrol path while p_1 means the first one; x_{p_n} and y_{p_n} present the horizontal and vertical coordinates respectively; N means the total number of patrol points along some a security patrol path.

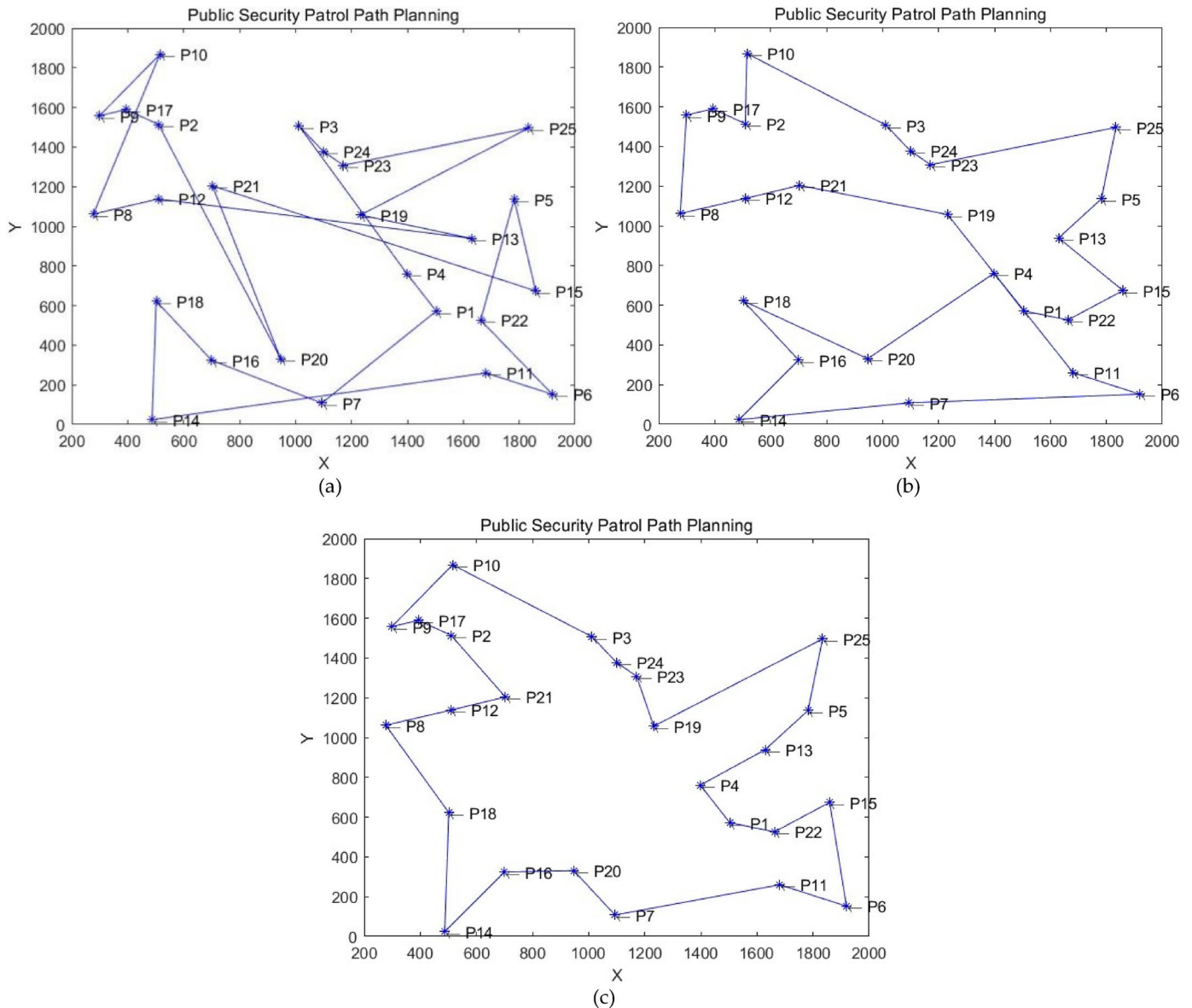


Fig. 1. (a) Randomly arranged public security patrol path planning; (b) Preliminary sorting of public security patrol path planning; (c) Global optimal public security patrol path planning.

Obviously, it is completely that different from classical test functions in 2.3 that the above description is a discrete problem. To solve this problem, this paper made a discrete algorithm based on the idea of WPOA to solve the discrete problem.

Improvement and design of the new proposed method

In order to solve the problems established above, compared with the comparative algorithm in ²⁵, the article titled “an adaptive distribution size (ADS) wolf-pack optimization algorithm using the strategy of jumping for raid (for short, WDX-WPOA)” demonstrates significant advantages in optimization capabilities, including higher global search accuracy, stronger robustness, and faster convergence speed. However, WDX-WPOA also has certain shortcomings, such as significantly longer iteration times in some test function applications shown in ²⁶; so this paper designed two following strategies to improve the optimization performance in response to this issue, and then the new proposed algorithm will be utilized to achieve a more efficient solution for the security patrol path planning problem.

Dynamic-adjustment-factor

During the raid process, in order to find the global optimal solution around the leader wolf, wolf-pack will be hoped to approach the leader wolf as soon as possible. However, the distance between wolves except the leader wolf should be precisely adjusted, that is, the distance should not be too small to avoid the wolf pack falling into

local optimum, nor too large to prevent slow convergence. By setting reasonable distances, the efficiency of the optimization process can be improved while ensuring global search capability.

According to existing methods, the number of airdropped wolves in the ASGS-CWOA and WDX_WPOA does not change during the process of Summon-Raid. Obviously, this approach is insufficient to accurately reflect the dynamic changes during the raid process and may lead to a decrease in optimization performance, and to overcome this deficiency, Dynamic-Adjustment-Factor (DAF) was proposed to dynamically adjust the number of airdropped wolves based on real-time changes during the process of Summon-Raid, thereby improving the overall performance of the algorithm. The specific dynamic adjustment mechanism is shown in Eq. (6) following.

$$\begin{cases} factor = (t/T) / 2 + 1/2 \\ num_{raid} = floor(numW * factor) \\ new_{raid-wolf-index} = index_tem(1 : num_{raid}) \\ old_{raid-wolf-index} = index_tem((num_{raid} + 1) : numW) \end{cases} \quad (6)$$

where, *factor* depends the number of wolves following the old strategy of Summon-Raid as in ASGS-CWOA and the number of other wolves following the new proposed strategy of Summon-Raid ; *t* means the number of current iteration; *T* means the number of the maximum iteration; num_{raid} means the number of wolves to be replaced with new wolves; *floor* means a function that returns the calculated result rounded down to ensure that num_{raid} is an integer; *numW* means the number of wolf-pack that is 50 in this paper; $new_{raid-wolf-index}$ means an index array that means the position of the wolf to be replaced with the new wolf while $old_{raid-wolf-index}$ is the one in its original state; *index_tem* is an index array obtained by sorting the current fitness, from the worst fitness to the best fitness.

From the Eq. (6), it's observed that the factor will vary between 0.5 and 1, and it will increase the number of $new_{raid-wolf-index}$ as the number of iterations increases, thereby allowing more wolves to participate in the raid process. This mechanism allows the algorithm to introduce fewer new wolves in the early stages to reduce computational complexity; In the later stage, most of the wolves at this stage have completed global exploration, and more wolves are needed to search for the best of the locals. This strategy of Eq. 6 achieves the goal of focusing on global search in the early stage of the algorithm and strengthening local search in the later stage of the algorithm, so as to achieve the goal of accelerating convergence without losing the global performance. At the same time, this strategy is also more in line with the habit of wolves to discover prey and unite to prey after exploring.

From Fig. 2, it was seen that the convergence of the wolf-pack during the siege phase as the number of iterations increases. Obviously, the curve in the Fig. 2c is smoother than the others, and the convergence speed of the wolf-pack is also faster. This indicates that introducing the DAF to dynamically adjust the number of airdropped wolves and wolves running according to the original strategy can not only improve the convergence speed of the wolf-pack and shorten the overall running time of the algorithm, but also not affect the algorithm's ability to globally find the optimal solution. In other words, as the number of iterations increases, the wolf-pack will get closer to the optimal solution.

Balanced-raid-strategy

During the process of Summon-Raid, for every wolf, 2^D locations generated by the coordinate arrangement combination of current location and the middle one between the current wolf and the best wolf are checked up as well as the reverse location of the current wolf regarding the best wolf, shown in Fig. 3a about ASGS-CWOA. Accordingly, in WDX-WPOA, half of the wolf-pack follows the Summon-Raid strategy in ASGS-CWOA, while the other half is directly airdropped to the surrounding of current optimal wolf, which means that only $2^{(D-1)} + 1$ locations will be checked up per wolf on average and accelerates the gathering and siege rhythm of the wolf-pack, and accelerates the convergence speed of the algorithm in another word, shown in Fig. 3b.

Furthermore, Balanced-Raid-Strategy (BRS) was proposed in the new proposed algorithm to improve the existing original strategy of Summon-Raid by not only examining the reverse location of the current wolf, but also the ones generated according to certain rules between the reverse position of the current wolf regarding the current optimal wolf for half of wolf-pack. This greedy search rule increases the search space in the current stage, and it is more likely to find the optimal value in the current space, so as to achieve the purpose of accelerating convergence and finding the optimal fitness value more quickly, shown in Fig. 3c.

$$\begin{cases} wolf_{i-opposite} = 2 * bestwolf - pop(i, :) \\ locations_{opposite} = original_Summon_Raid_Mesh_Process(wolf_{i-opposite}, bestwolf) \\ ft_{opposite-locations} = FitnessFunction(locations_{opposite}) \\ [ft_{opposite-locations-best}, NO_{opposite-locations-best}] = \min(ft_{opposite-locations}) \\ wolf_{opposite-locations-best} = locations_{opposite}(NO_{opposite-locations-best}, :) \end{cases} \quad (7)$$

where, $wolf_{i-opposite}$ means the opposite location of the current wolf while *bestwolf* means the current best wolf and $pop(i, :)$ means the current wolf-*ith*; $locations_{opposite}$ represents the locations between $wolf_{i-opposite}$ and *bestwolf* according to the Summon-Raid rules in WDX-WPOA; *original_Summon_Raid_Mesh_Process* is a function that returns some related locations determined by Summon-Raid rules in WDX-WPOA while *FitnessFunction* is another one that returns the fitness of the given wolves or locations; $ft_{opposite-locations}$ means the fitness regarding $locations_{opposite}$; $ft_{opposite-locations-best}$ means the best one of $ft_{opposite-locations}$ while $NO_{opposite-locations-best}$ is the serial number of $ft_{opposite-locations-best}$ in $ft_{opposite-locations}$, and $wolf_{opposite-locations-best}$ is the best one of $locations_{opposite}$ around $wolf_{i-opposite}$.

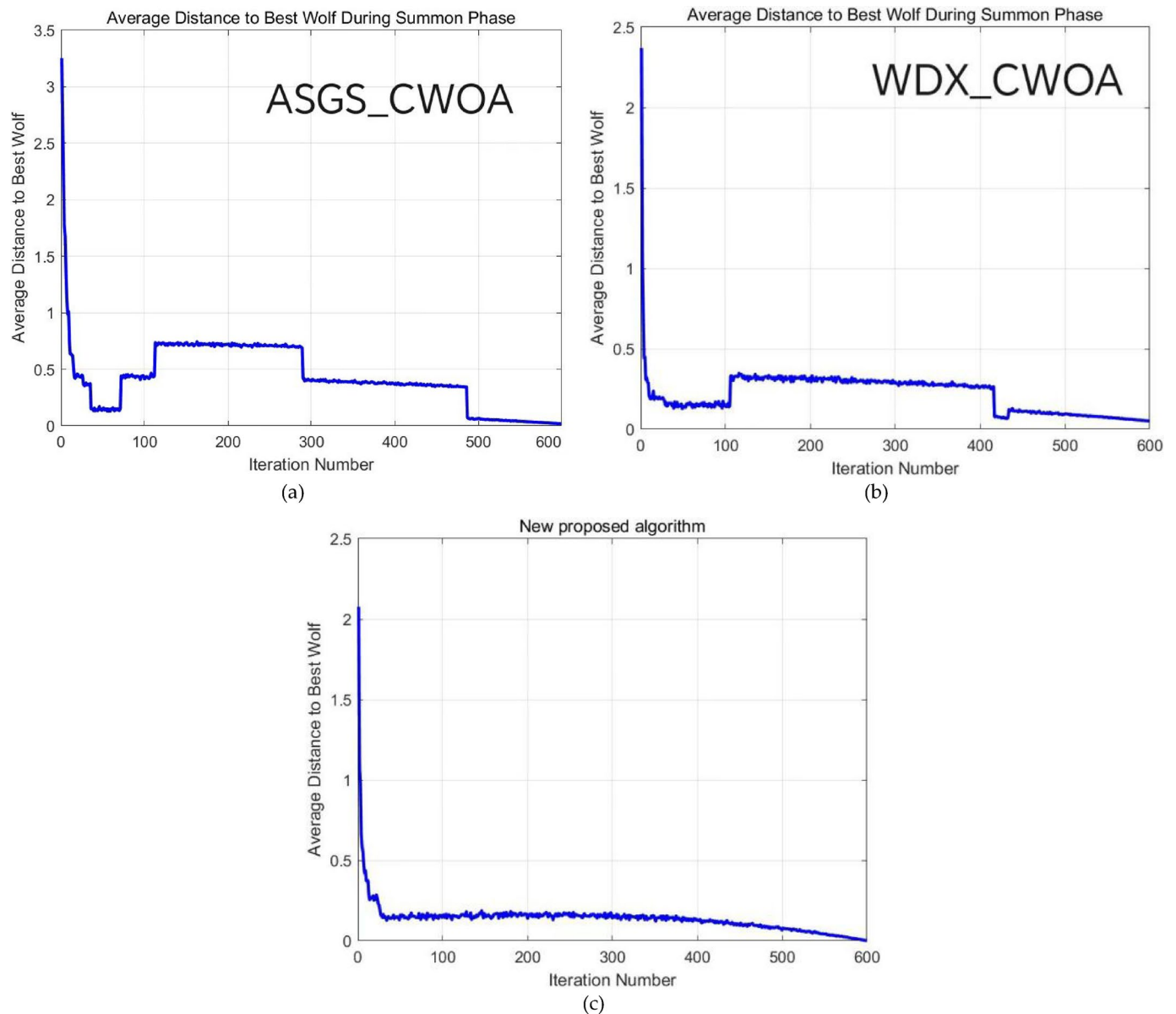


Fig. 2. (a) the original raiding strategy in ASGS; (b) the old raiding strategy in WDX-WPOA; (c) the new adaptive raiding strategy in DAF-BRS-CWOA.

The Eq. (7) gives the key rules regarding BRS, the symmetrical coordinates of the current wolf relative to the optimal wolf are found according to the first equation. The wolf pack then follows the second equation to find the midpoint between the current wolf, the symmetrical wolf, and the best wolf, and arrange them by dimension. The point coordinates generated by the formula are shown in Fig. 3c with W3, W4 and O3, O4 and the surrounding points. This strategy fills the search gap in the region opposite to best wolf, increases the probability of finding the global optimal solution. Moreover, other equations are used to select the best one in the above points to replace the current wolf, so as to reflect the survival rule of the wolf pack to eliminate the weak.

Steps Of DAF-BRS-CWOA

On the basis of the strategy of dynamically increasing the number of airdrop wolves and increasing the number of examined locations, WCOA was improved and DAF-BRS-CWOA was proposed. Here are the steps on how to implement the new improved algorithm, as shown in Fig. 4a.

All steps are consistent with WDX-WPOA except DAF and BRS are used in the Summon-Raid process, detailed in "Dynamic-adjustment-factor" and "Balanced-raid-strategy" respectively. At the same time, we made it as a discrete -type algorithm according to the idea of DAF-BRS-CWOA to solve the S3P-Function. The general process is as shown in Fig. 4b.

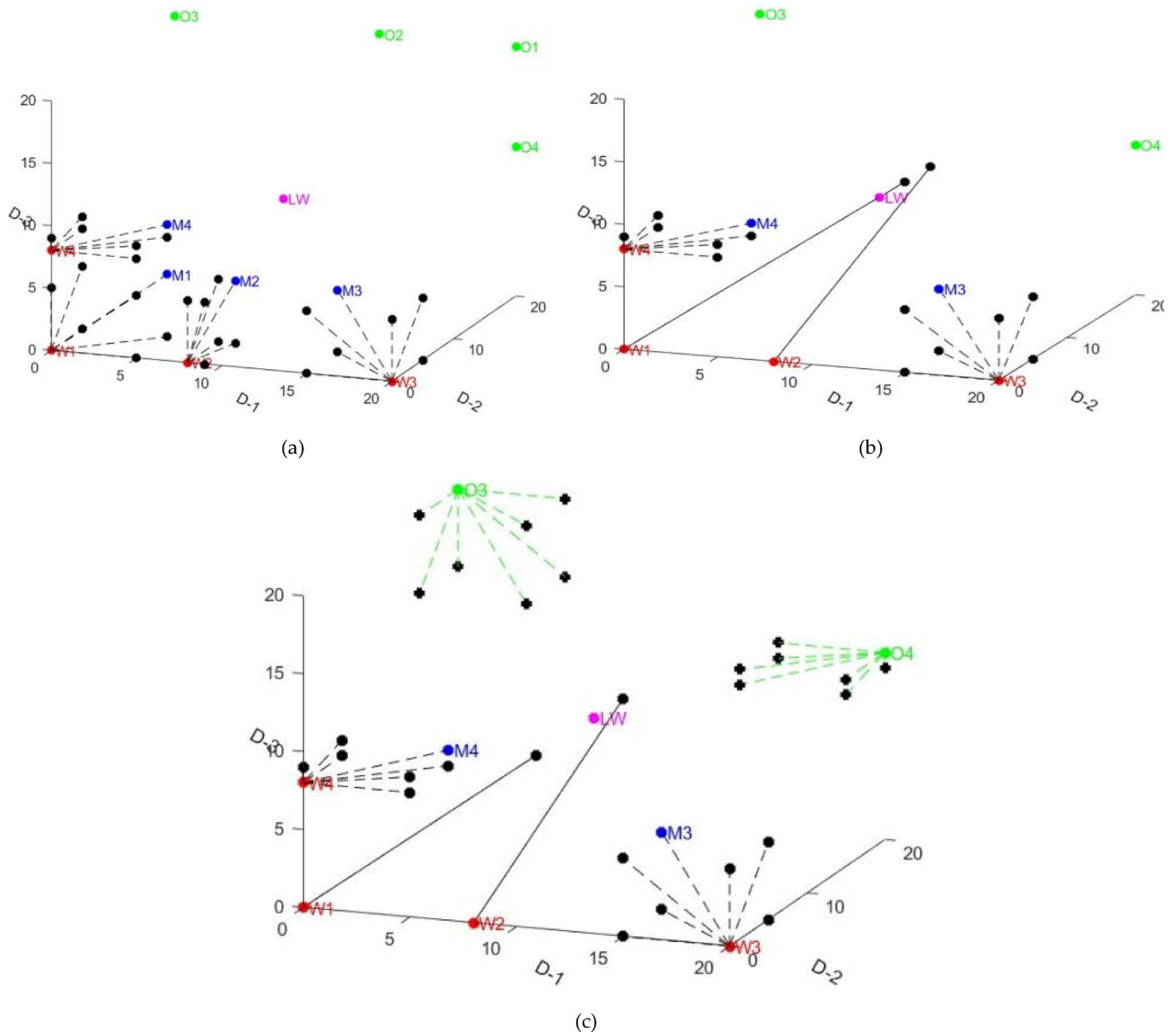


Fig. 3. (a) about ASGS-CWOA; (b) WDX-WPOA; (c) New proposed algorithm.

Performance verification experiment

Experimental designment

In order to test the optimization capabilities of DAF-BRS-CWOA, the classic genetic algorithm (GA), particle swarm algorithm (PSO) and WDX-WPOA three optimization algorithms shown in Table 1 are used to perform experiments. And 20 test functions such as 6 multi -model methods and the distance test function of the patrol path, the theoretical optimal of order, function, expression, dimension, characteristics, scope, and each test function is given. The numerical experiment results on the 21st test function for the next test can be analyzed and studied for the next step. Details are shown in tables following.

All numerical experiments are implemented on a computer equipped with Windows 11 Home 22H2 operating system, Intel(R) Core (TM) i7-12700H processor and 16G memory while the integrated development environment is MATLAB-2024a. For genetic algorithm, toolbox in Matlab 2017a is utilized for GA experiments; the PSO experiments were implemented by a “PSOT” toolbox for Matlab; Experiments on WDX-WPOA are done according to the steps from reference ²⁶; and the new algorithm DAF-BRS-CWOA is implemented by utilizing MATLAB-2024a with M programming language. The specific configuration of the above algorithms is shown in Table 2. In order to verify the excellent performance of the new algorithm, like the other three algorithms, it runs 30 optimization calculations on each test function. Then, all algorithms are evaluated from the aspects of the best value, worst value, average, standard deviation, the number of average iterations, and the average time of the global optimal.

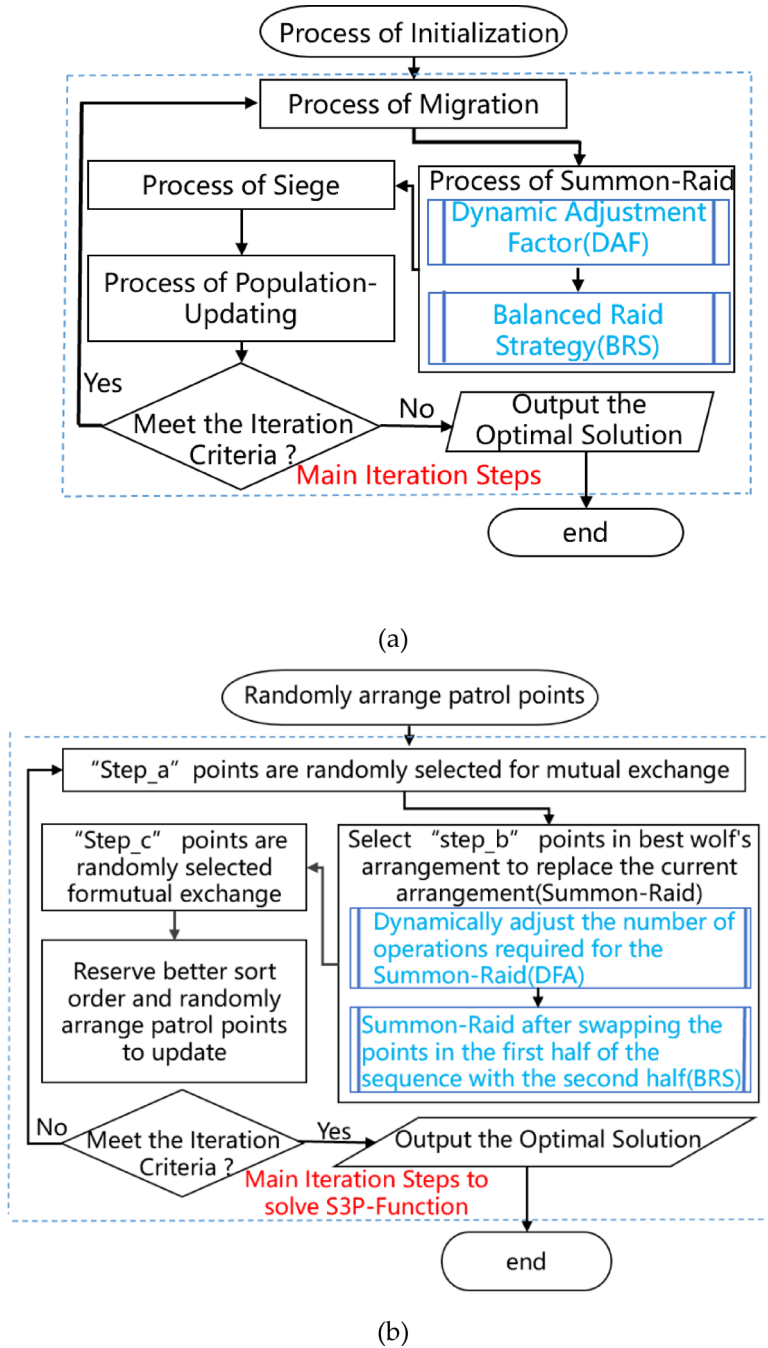


Fig. 4. Flowchart about DAF-BRS-CWOA.

Order	Algorithm name	Configuration
1	GA	Crossover probability is 0.8, the mutation probability is 0.01, the max iteration T = 600
2	PSO	Inertia weight is 0.5, the Cognitive coefficient is 1.5, the social coefficient is 1.5, the max iteration T = 600
3	WDX-WPOA	Initial value of search step size $step_{a_0} = 1.5$; the initial max value of siege step size $step_{c_{max}} = 1e6$ and the minimum value of siege step size $step_{c_{min}} = 1e-40$; the max iteration T = 600; the amount of the wolf population N = 50
4	DAF-BRS-CWOA	Initial value of search step size $step_{a0} = 1.5$; the initial max value of siege step size $step_{c_{max}} = 1e6$ and the minimum value of siege step size $step_{c_{min}} = 1e-40$; the max iteration T = 600; the amount of the wolf population N = 50

Table 1. Configuration of algorithms.

Order	Function	Expression	Dimension	Range	Optimum
1	Ackley	$F1 = -20 \exp \left(-0.2 \sqrt{\frac{1}{2} \sum_{i=1}^2 x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i) \right) + 20 + \exp(1)$	2	[-32.768,32.768]	Min f=0
2	Bukin6	$F2 = 100 \sqrt{ x_2 - 0.01x_1^2 } + 0.01 x_1 + 10 $	2	[-15,3]	Min f=0
3	Drop-Wave	$F3 = -\frac{1 + \cos \left(12 \sqrt{\frac{x_1^2 + x_2^2}{2}} \right)}{0.5(x_1^2 + x_2^2) + 2}$	2	[-5.12,5.12]	Min f=-1
4	Eggholder	$F4 = -(x_2 + 47) \sin \left(\sqrt{\left x_2 + \frac{x_1}{2} + 47 \right } \right) - x_1 \sin \left(\sqrt{ x_1 - (x_2 + 47) } \right)$	2	[-512,512]	Min f=-959.6407
5	Griewank	$F5 = \sum_{i=1}^2 \frac{x_i^2}{4000} - \prod_{i=1}^2 \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	2	[-600,600]	Min f=0
6	Levy	$F6 = \sin^2(\pi w_1) + \sum_{i=1}^{2-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_2 - 1)^2 [1 + \sin^2(2\pi w_2)],$ <p>where $w_i = 1 + \frac{x_i - 1}{4}$, for all $i = 1, 2$</p>	2	[-10,10]	Min f=0
7	Levy13	$F7 = \sin^2(3\pi x_1) + (x_1 - 1)^2 [1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2 [1 + \sin^2(2\pi x_2)]$	2	[-10,10]	Min f=0
8	Rastrigin	$F8 = 10d + \sum_{i=1}^2 [x_i^2 - 10 \cos(2\pi x_i)]$	2	[-5.12,5.12]	Min f=0
9	Schaffer2	$F9 = 0.5 + \frac{\sin^2 \left(\frac{x_1^2 - x_2^2}{2} \right) - 0.5}{\left[1 + 0.001(x_1^2 + x_2^2) \right]^2}$	2	[-100,100]	Min f=0
10	Bohachevsky1	$F10 = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$	2	[-100,100]	Min f=0
11	Perm0-d-β	$F11 = \sum_{i=1}^2 \left(\sum_{j=1}^2 (j + \beta) \left(x_j^i - \frac{1}{j^i} \right) \right)^2$	2	[-2,2]	Min f=0
12	Rotated Hyper-Ellipsoid	$F12 = \sum_{i=1}^2 \sum_{j=1}^i x_j^2$	2	[-65.536,65.536]	Min f=0
13	Sum Squares	$F13 = \sum_{i=1}^2 i x_i^2$	2	[-10,10]	Min f=0
14	Trid	$F14 = \sum_{i=1}^2 (x_i - 1)^2 - \sum_{i=2}^2 x_i x_{i-1}$	2	[-4,4]	Min f=2
15	Booth	$F15 = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	2	[-10,10]	Min f=0
16	Matyas	$F16 = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	2	[-10,10]	Min f=0
17	Zakharov	$F17 = \sum_{i=1}^2 x_i^2 + \left(\sum_{i=1}^2 0.5ix_i \right)^2 + \left(\sum_{i=1}^2 0.5ix_i \right)^4$	2	[-5,10]	Min f=0
18	Easom	$F18 = -\cos(x_1) \cos(x_2) \exp \left[-(x_1 - \pi)^2 - (x_2 - \pi)^2 \right]$	2	[-4,4]	Min f=-1
19	Eggcrate	$F19 = x_1^2 + x_2^2 + 25 \left(\sin^2 x_1 + \sin^2 x_2 \right)$	2	[-π, π]	Min f=0
20	Bohachevsky3	$F20 = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1 + 4\pi x_2) + 0.3$	2	[-100,100]	Min f=0

Table 2. Datasets for performance validation.

Data-sets

To systematically evaluate the optimization performance of DAF-BRS-CWOA, we conducted comparative experiments using a curated suite of benchmark functions and datasets (Table 1) that collectively probe three critical dimensions: global search robustness, convergence precision, and dynamic adaptation. Multimodal functions with multi-peak characteristics and high noise interference were selected to simulate complex non-convex search spaces, rigorously testing the algorithm's ability to escape local optima traps while maintaining stability in deceptive fitness landscapes. Smooth unimodal functions with bowl-shaped convex structures were employed to quantify convergence dynamics, particularly the efficiency of gradient-driven optimization and numerical accuracy in attaining near-theoretical optima. Additionally, gently sloped functions with gradual fitness variations were integrated to mimic real-world scenarios involving time-varying objectives, thereby assessing the algorithm's adaptability to slow environmental shifts and its capacity for persistent exploration in near-flat solution regions. This multi-faceted validation strategy aligns with the core challenges of security patrol path optimization, where balancing exploration–exploitation tradeoffs, rapid convergence under constraints, and responsiveness to dynamic risks are paramount, which is also consistent with the problems encountered in the planning of public safety patrol paths in the real world.

Experimental results and analysis

Firstly, as shown in Table 3, seen from the optimal value, the DAF-BRS-CWOA can find the theoretical optimal values of all the test functions as well as WDX-WPOA, but no one else can do it; furthermore, seen from the worst value and average value, DAF-BRS-CWOA has the best performance in all 4 algorithms (PSO, GA, WDX-WPOA and DAF-BRS-CWOA) that its worst and average values both reach the theoretical optimal values while the other algorithms have no such ability. Therefore, DAF-BRS-CWOA has better optimization accuracy.

In addition, in 30 tests, the new algorithm had zero standard deviation on almost all test functions except function 2 “Bukin6”, function 4 “Eggholder”, function 6 “Levy” and function 21 “3P-Function”. Even so, in function 2 “Bukin6” and function 4 “Eggholder”, the standard deviation of DAF-BRS-CWOA is better than the standard deviation of other algorithms. Although the standard deviation of DAF-BRS-CWOA in function 6 is larger than that of GA and PSO, GA and PSO cannot find the optimal value in 30 tests, while DAF-BRS-CWOA can, as show the standard deviation in Table 3. Therefore, DAF-BRS-CWOA has good stability in general.

Moreover, in term of average number of iterations, DAF-BRS-CWOA has smallest iterations on all test functions except function 2 “Bukin6”, function 4 “Eggholder”, function 21 “S3P-Function”, and on function 2, 4 and 21 the number of iterations is 600 which is same to the values on other algorithms, shown in Table 3 Average iteration. Hence, in general, DAF-BRS-CWOA has better advantage in terms of iteration.

Finally, as shown in Table 3 Average Time, the average time spent on Functions 14, 15, 18, and 19 is minimal for DAF-BRS-CWOA. Moreover, the four algorithms spend about the same amount of time on test functions 1, 9, and 20, and DAF-BRS-CWOA runs longer than PSO or GA on the rest of the test functions, but the performance of finding the optimal value is better than that of other algorithms (PSO and GA). Nonetheless, it is a consolation that the proposed algorithm outperforms WDX-WPOA in terms of running time without losing optimization accuracy. As shown in Table 4 and Fig. 5, In all test functions, the DAF-BRS-CWOA is improved compared to WDX-WPOA. Therefore, the DAF-BRS-CWOA has a good convergence speed.

In fact, on the premise of guaranteeing the absolute global optimization ability, the conclusion shows that DAF-BRS-CWOA spent less time seen from Table 4 and Fig. 3, it is seen that all the improvement rates are positive and that means DAF-BRS-CWOA spent less time than WDX-WPOA on all the 21 algorithms involved in this paper. Especially, it is worth mentioning that the improvement rates on function “Ackley” and “Eggholder” are both more than 30% and the value on function “Bohachevsky3” is more than 20% as well as most of the functions are both more than 10%. On the other functions, all the improvement rate are more than 5% and the worst improvement rate reaches about 5.17%.

Especially, it is in the optimization of the function 21 that DAF-BRS-CWOA possess the shortest path length comparing PSO, GA and WDX-WPOA during 30 independent experiments, and DAF-BRS-CWOA takes a certain improvement in time-spent compared to WDX-WPOA, as shown in Table 3. Therefore, DAF-BRS-CWOA has excellent performance in public security patrol path planning.

In a word, DAF-BRS-CWOA possesses excellent optimization accuracy, good stability, less time-spent and quick peed of convergence. At the same time, the test results of function 21 “S3P-Function” also fully show that the algorithm proposed in this paper uses less computing resources while ensuring better optimization effect, which is an excellent algorithm suitable for solving public safety patrol path planning.

Conclusions

The public security patrol path planning is very important in public security work, however, the existing public security patrol path planning has some shortcomings to varying degrees. To address these shortcomings, this paper proposes a public security patrol path planning recommendation method based on an improved wolf-pack optimization algorithm, which mainly focuses on three aspects. Firstly, Dynamic-Adjustment-Factor was introduced in WDX-WPOA to dynamically adjust the number of airdropped wolves; Moreover, Balanced-Raid-Strategy with symmetric property was proposed to improve the existing original strategy of Summon-Raid by not only examining the reverse position of the current wolf, but also the ones generated according to certain rules between the reverse position of the current wolf and the current optimal wolf; Finally, an optimization objective function regarding the public security patrol path planning was abstracted and counted to obtain the optimal recommendation for the public security patrol path planning. Comparative and numerical experiments with four similar swarm intelligence optimization algorithms (PSO, GA, WDX-WPOA and DAF-BRS-CWOA) on 20 public datasets and the proposed objective function RPB Function were conducted and the results demonstrated that the DAF-BRS-CWOA has superior performance.

Function	Algorithm	Optimal value	Worst value	Average value	Standard deviation	Average iteration	Average time
1 Ackley min $f=0$	GA	7.92E-06	0.00012803	0.000051735	6.95E-10	176.7	0.23365
	PSO	1.71E-05	0.00057189	0.00011362	6.89E-09	600	0.046053
	WDX-WPOA	0	0	0	0	37.6333	0.053561
	DAF-BRS-CWOA	0	0	0	0	22.2667	0.033704
2 Bukin6 min $f=0$	GA	0.6912	11.2453	3.7961	2.6443	600	0.013
	PSO	0.0012517	0.13	0.058051	0.041452	600	0.013559
	WDX-WPOA	0.0023801	0.082569	0.029872	0.022073	600	0.80598
	DAF-BRS-CWOA	0.075576	0.50493	0.28726	0.12965	600	0.64729
3 Drop-Wave min $f=-1$	GA	-0.99992	-0.78573	-0.93986	0.04808	600	0.012028
	PSO	-1	-0.93625	-0.98512	0.026965	218.6333	0.004777
	WDX-WPOA	-1	-1	-1	0	14.4333	0.018366
	DAF-BRS-CWOA	-1	-1	-1	0	12.1333	0.016619
4 Eggholder min $f=-959.6407$	GA	-959.6387	-629.6112	-876.5954	78.8508	600	0.012111
	PSO	-959.6407	-718.1675	-926.7076	53.5701	600	0.012546
	WDX-WPOA	-959.6407	-935.338	-947.2171	11.904	600	1.0656
	DAF-BRS-CWOA	-959.6404	-935.3379	-948.744	11.7553	600	0.72768
5 Griewank min $f=0$	GA	0.004788	0.31789	0.075813	0.063402	600	0.013661
	PSO	0	0.019719	0.0026303	0.0045421	339.3	0.0085504
	WDX-WPOA	0	0	0	0	17.5667	0.025562
	DAF-BRS-CWOA	0	0	0	0	13.5	0.020979
6 Levy min $f=0$	GA	0.00024335	1.1263	0.12324	0.21281	600	0.038618
	PSO	1.50E-32	1.50E-32	1.50E-32	1.09E-47	600	0.039204
	WDX-WPOA	0	5.98E-01	0.019935	0.10735	549.9	0.84128
	DAF-BRS-CWOA	0	0.39478	0.013159	0.070865	479.1333	0.7371
7 Levy13 min $f=0$	GA	0.011247	2.2797	0.22303	0.76184	600	0.01238
	PSO	0.00010961	-0.97283	-0.97283	3.33E-16	600	0.012789
	WDX-WPOA	0	0	0	0	25.1333	0.038007
	DAF-BRS-CWOA	0	0	0	0	21.7	0.03573
8 Rastrigin min $f=0$	GA	0.013678	6.3489	2.2711	1.7587	600	0.0119
	PSO	0	0.99496	0.066331	0.24819	110.4333	0.0023772
	WDX-WPOA	0	0	0	0	12.3	0.018345
	DAF-BRS-CWOA	0	0	0	0	10.9	0.016472
9 Schaffer2 min $f=0$	GA	1.03E-06	0.042464	0.010477	0.0093243	600	0.01321
	PSO	0	0	0	0	66.9667	0.0019346
	WDX-WPOA	0	0	0	0	11.7333	0.016255
	DAF-BRS-CWOA	0	0	0	0	9.7	0.01489
10 Bohachevsky1 min $f=0$	GA	0.011268	0.91934	0.48134	0.25563	600	0.011984
	PSO	0	0	0	0	78.1667	0.0017231
	WDX-WPOA	0	0	0	0	14.5	0.020572
	DAF-BRS-CWOA	0	0	0	0	12.3	0.018081
11 Perm0-d- β min $f=0$	GA	0.011057	388.7314	26.2837	71.8592	600	0.01213
	PSO	0	0	0	0	175.3667	0.0052171
	WDX-WPOA	0	0	0	0	25.8	0.030897
	DAF-BRS-CWOA	0	0	0	0	21.7667	0.026103
12 Rotated Hyper-Ellipsoi min $f=0$	GA	0.00039244	0.12985	0.034819	0.03493	600	0.012687
	PSO	1.96E-134	1.02E-129	8.56E-131	2.28E-130	600	0.013828
	WDX-WPOA	0	0	0	0	25.9667	0.029679
	DAF-BRS-CWOA	0	0	0	0	22.7	0.027767
13 Sum Squares min $f=0$	GA	2.42E-06	0.0025094	0.00051005	0.00048937	600	0.012084
	PSO	5.91E-137	2.18E-132	3.76E-133	5.82E-133	600	0.013233
	WDX-WPOA	0	0	0	0	24.3667	0.029123
	DAF-BRS-CWOA	0	0	0	0	21.1667	0.027215
14 Trid min $f=-2$	GA	-0.037736	-1.9991	-1.8925	0.11219	600	0.01241
	PSO	-2	-2	-2	0	600	0.014356
	WDX-WPOA	-2	-2	-2	0	10.6667	0.013702
	DAF-BRS-CWOA	-2	-2	-2	0	9.1667	0.011635
Continued							

Function	Algorithm	Optimal value	Worst value	Average value	Standard deviation	Average iteration	Average time
15 Booth min $f=0$	GA	4.93E-12	4.92E-09	8.73E-10	1.11E-18	74.79	0.088336
	PSO	5.62E-23	8.78E-17	5.13E-18	2.36E-34	600	0.02997
	WDX-WPOA	0	0	0	0	24.0667	0.027151
	DAF-BRS-CWOA	0	0	0	0	20.4	0.025748
16 Matyas min $f=0$	GA	9.11E-06	0.042161	0.010059	0.010711	600	0.01241
	PSO	1.76E-120	2.71E-116	2.87E-117	5.48E-117	600	0.013081
	WDX-WPOA	0	0	0	0	24.5	0.030189
	DAF-BRS-CWOA	0	0	0	0	20.3333	0.025239
17 Zakharov min $f=0$	GA	2.32E-06	0.0016735	0.00069625	0.00057972	600	0.015955
	PSO	3.10E-137	7.27E-131	3.57E-132	1.30E-131	600	0.016975
	WDX-WPOA	0	0	0	0	24.8	0.053188
	DAF-BRS-CWOA	0	0	0	0	21.1	0.045726
18 Easom min $f=-1$	GA	-1	0	-0.75001	0.18749	72.91	0.084762
	PSO	-1	-6.30E-61	-0.90001	0.089988	593.02	0.033852
	WDX-WPOA	-1	-1	-1	0	13.6667	0.018649
	DAF-BRS-CWOA	-1	-1	-1	0	11.7333	0.016503
19 Eggcrate min $f=0$	GA	2.01E-02	6.72E-01	2.91E-01	1.61E-01	600	0.024449
	PSO	6.23E-24	1.42E-08	1.42E-10	1.99E-18	597.56	0.030566
	WDX-WPOA	0	0	0	0	14.9667	0.020518
	DAF-BRS-CWOA	0	0	0	0	12.9	0.018693
20 Bohachevsky3 min $f=0$	GA	0.032611	0.7588	0.26482	0.18579	600	0.024356
	PSO	0	0	0	0	87.4667	0.016874
	WDX-WPOA	0	0	0	0	14.2333	0.023575
	DAF-BRS-CWOA	0	0	0	0	11.9	0.017347
21 3P-Function	GA	15442.6497	24126.8603	19416.3543	1816.0666	600	0.79052
	PSO	9629.7786	13812.6688	11390.4309	844.0894	600	5.0392
	WDX-WPOA	8586.7186	11586.8103	10318.2428	762.6918	600	79.7046
	DAF-BRS-CWOA	8586.7186	11387.3388	9997.6471	710.5523	600	73.1483

Table 3. Raw data of experiments.

Although the algorithm proposed in this paper is superior, there are still some shortcomings. Such as shown in Table 3, DAF-BRS-CWOA spent more time than GA and PSO in 14 test functions. And it is in function-21 that DAF-BRS-CWOA spent a considerable amount of time as well as the results of each operation are not so stable due to the higher complexity and it still requires more computing resources for the planning of public safety patrol paths. It confirms a principle: no perfect standard that applies universally. In view of the above problems, future work will focus on exploring new siege methods to reduce the use of computing resources and improve stability, and also apply them to more fields.

Algorithm			
Function	WDX-WPOA	DAF-BRS-CWOA	Improvement-rate (%)
F1	0.053561	0.033704	37.07
F2	0.80598	0.64729	19.69
F3	0.018366	0.016619	9.51
F4	1.0656	0.72768	31.71
F5	0.025562	0.020979	17.93
F6	0.84128	0.7371	12.38
F7	0.038007	0.03573	5.99
F8	0.018345	0.016472	10.21
F9	0.016255	0.01489	8.40
F10	0.020572	0.018081	12.11
F11	0.030897	0.026103	15.52
F12	0.029679	0.027767	6.44
F13	0.029123	0.027215	6.55
F14	0.013702	0.011635	15.09
F15	0.027151	0.025748	5.17
F16	0.030189	0.025239	16.40
F17	0.053188	0.045726	14.03
F18	0.018649	0.016503	11.51
F19	0.032477	0.028214	13.13
F20	0.023575	0.017347	26.42
F21	79.7046	73.1483	8.23

Table 4. Time-spent comparison by DAF-BRS-CWOA and WDX-WPOA (The following sequence numbers correspond to the sequence numbers and details of the functions in Table 2).

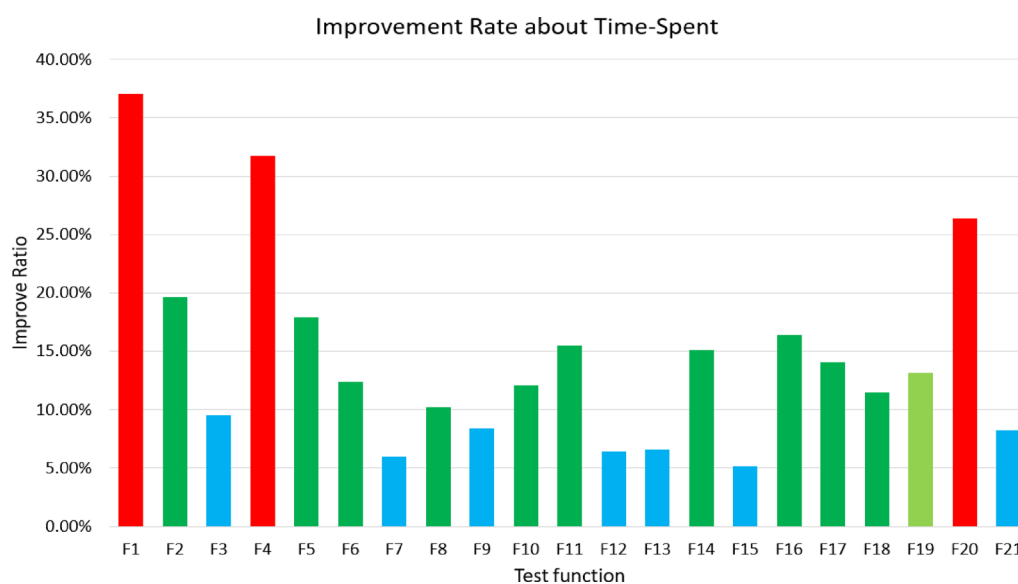


Fig. 5. Histograms of improvement rate about time-spent by comparing DAF-BRS-CWOA and WDX-WPOA on all 21 benchmark functions in this paper. (F1:Ackley F2:Bukin6 F3:Drop-Wave F4:Eggholder F5:Griewank F6:Levy F7:Levy13 F8:Rastrigin F9:Schaffer2 F10:Bohachevsky1 F11:Perm0-d- β F12:Rotated Hyper-Ellipsoid F13:Sum Squares F14:Trid F15:Booth F16:Matyas F17:Zakharov F18:Easom F19:Bohachevsky2 F20:Bohachevsky3 F21:S3P-Function).

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 5 February 2025; Accepted: 12 May 2025

Published online: 05 June 2025

References

- Dewinter, M., Vandeviver, C., Vander Beken, T. & Witlox, F. Analysing the police patrol routing problem: A review. *ISPRS Int. J. Geo Inf.* **9**(3), 157. <https://doi.org/10.3390/ijgi9030157> (2020).
- Lee, J., Lee, J. H., Im, J. & Jang, J. Enhancing urban safety: Optimal patrol route strategies for volunteer security squads based on integrated BIM-GIS data. *Buildings* **14**(10), 3074. <https://doi.org/10.3390/buildings14103074> (2024).
- Tian, A.-Q., Liu, F.-F. & Lv, H.-X. Snow geese algorithm: A novel migration-inspired meta-heuristic algorithm for constrained engineering optimization problems. *Appl. Math. Model.* **126**, 327–347. <https://doi.org/10.1016/j.apm.2023.10.045> (2024).
- Wang, C. et al. Multi unmanned ship path planning method and simulation for maritime patrol. *Chin. J. Image Gr.* **28**(8), 2536–2548 (2023).
- Xiang, R., Zhu, M. & Xu, L. Planning method for multi UAV highway patrol tasks. *Radio Eng.* **007**, 052 (2022).
- Li, L. Research and application development of traffic police patrol route optimization based on open-source GIS and rural postman problem (Doctoral dissertation, East China Normal University) (2020).
- Zhang, L. & Fan, X. Key patrol route planning under ant colony algorithm. *Pract. Underst. Math.* **50**(16), 9 (2020).
- Kong, L., Zhang, G., Su, Z. & Jiang, J. Ambulance emergency rescue path planning based on improved ant colony algorithm. *Comput. Eng. Appl.* (2018)
- Jie, Yi. & Gu, Yi. Multi agent reinforcement learning police patrol path planning based on Stackelberg strategy. *J. Beijing Inst. Technol.* **37**(1), 7 (2017).
- Liang, J. Optimization of tourist park sightseeing path planning based on improved ant colony algorithm. *Control Inf. Technol.* (2024).
- Shi, W., Fu, Y., Zhao, X., Liu, H. & Li, P. Research on patrol task allocation and path optimization method based on multi agent. *J. Chin. Acad. Electron. Sci.* **16**(7), 6 (2021).
- Yan, Z., He, Li. & Li, J. Multi domain limited multi AUV patrol route planning method. *J. Underw. Unmanned Syst.* **25**(4), 6 (2017).
- Mukhopadhyay, A. et al. Optimal allocation of police patrol resources using a continuous-time crime model. In *Decision and Game Theory for Security: 7th International Conference, GameSec 2016, New York, NY, USA, November 2–4, 2016, Proceedings* 7 139–158. (Springer, 2016).
- Zhang, L., Gao, T., Cai, G. & Hai, K. L. Research on electric vehicle charging safety warning model based on back propagation neural network optimized by improved gray wolf algorithm. *J. Energy Storage* **49**, 104092 (2022).
- Tang, D. et al. Behavior automatic analysis for wolf pack hunting: Making fast behavior analysis of massive data possible. *Behav. Ecol. Sociobiol.* **77**(6), 67 (2023).
- Duan, H. et al. Unmanned aerial systems coordinate target allocation based on wolf behaviors. *Sci. China. Inf. Sci.* **62**(1), 14201 (2019).
- Yao, K. et al. An information entropy-based grey wolf optimizer. *Soft Comput.* **27**(8), 4669–4684 (2023).
- Chen, S. & Zheng, J. A hybrid grey wolf optimizer for engineering design problems. *J. Comb. Optim.* **47**(5), 86 (2024).
- Wang, Z. & Zhang, J. A task allocation algorithm for a swarm of unmanned aerial vehicles based on bionic wolf pack method. *Knowl.-Based Syst.* **250**, 109072 (2022).
- She, A., Wang, L., Peng, Y., & Li, J. Structural reliability analysis based on improved wolf pack algorithm AK-SS. In *Structures* vol. 57 105289. (Elsevier, 2023).
- H. W., Wen, J., & Xu, Z. Underwater robot submarine cable positioning algorithm based on improved grey wolf swarm optimization algorithm. *Control Decis.-Mak.* 1–7 (2024).
- Cortes, C. & Vapnik, V. Support-vector networks. *Mach. Learn.* **20**(3), 273–297. <https://doi.org/10.1023/A:1022627411411> (1995).
- Yang, X. S. A new metaheuristic bat-inspired algorithm. *Comput. Knowl. Technol.* **284**, 65–74 (2010).
- Yang, C., Tu, X. & Chen, J. Algorithm of marriage in honey bees optimization based on the wolf-pack search. In *International Conference on Intelligent Pervasive Computing* 11–13 (IEEE Computer Society, 2007).
- Wang, D., Qian, X., Liu, K., & Ban, X. (2018). An Adaptive Shrinking Grid Search Chaos Wolf Optimization Algorithm with Adaptive Standard-Deviation Updating Amount.
- Wang, D., Qian, X., Liu, K., Ban, X. & Guan, X. An adaptive distributed size wolf-pack optimization algorithm using strategy of jumping for raid. *IEEE Access* 1–1 (2018).

Acknowledgements

The authors are grateful for peer experts for full support of this paper and thank University of Science and Technology Beijing for supporting necessary scientific environment. This study was funded by Henan Key Research Projects of Higher Education Institutions (NO.24B510010), Science and Technology Research Project of Henan Provincial Department of Science and Technology (NO.242102321103), Fujian Province's 2024 Science and Technology Driven Police Research Special Project (No. 2024Y0065) and National Natural Science Foundation of China (NO.62332017).

Author contributions

Dongxing Wang and Gaomin Zhang designed the experiments and gave the innovation points. Fangbo Wu and Zhiyang Huang prepared the main manuscript text in chapter 3 and chapter 4. Yiping Chen prepared the main manuscript text in Chapter 1 while Dongxing Wang and Gaomin Zhang wrote other manuscript text. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to G.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025