

6-28-2024

Dynamic Air Defense Resource Allocation Optimization Based on Improved Differential Evolution Algorithm

Tianyu Luo

College of Systems Engineering, National University of Defense Science and Technology, Changsha 410073, China, luotianyu951005@163.com

Lining Xing

College of Electronic Engineering, Xi'an University of Electronic Science and Technology, Xi'an 710071, China, lnxing@xidian.edu.cn

Rui Wang

College of Systems Engineering, National University of Defense Science and Technology, Changsha 410073, China

Ling Wang

Department of Automation, Tsinghua University, Beijing 100084, China

See next page for additional authors

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation. For more information, please contact xtfzxb@126.com.

Dynamic Air Defense Resource Allocation Optimization Based on Improved Differential Evolution Algorithm

Abstract

Abstract: Based on the integrated performance of weapon equipments such as radars, launchers and missiles, a mixed-integer decision model that minimizes the total target intercept value and the probability of survival based on Target-Set, Resource-Set is developed. A new improved differential evolutionary algorithm has been introduced to solve the problem, and the initial solutions is generated by using the reverse learning strategies to ensure the quality of the initial populations. An inspiration rule for the fast repair and reconstruction is designed to work at multi-stage to improve the search capability of the algorithm. The simulation experiment results show the algorithm's superiority in search time and search accuracy, which can maintain the efficient combat capabilities and decision-making under the random influence of dynamic events.

Keywords

air defense operations, dynamic air defense resource allocation, opposition-based learning, improved differential evolution algorithm

Authors

Tianyu Luo, Lining Xing, Rui Wang, Ling Wang, Jianmai Shi, and Xin Sun

Recommended Citation

Luo Tianyu, Xing Lining, Wang Rui, et al. Dynamic Air Defense Resource Allocation Optimization Based on Improved Differential Evolution Algorithm[J]. Journal of System Simulation, 2024, 36(6): 1285-1297.

基于改进差分进化算法的动态防空资源分配优化

罗天羽¹, 邢立宁^{2*}, 王锐¹, 王凌³, 石建迈¹, 孙昕¹(1. 国防科技大学 系统工程学院, 湖南 长沙 410073; 2. 西安电子科技大学 电子工程学院, 陕西 西安 710071;
3. 清华大学 自动化系, 北京 100084)

摘要: 面对动态防空资源分配问题中存在的空袭目标突现和雷达、发射车等资源受干扰现象, 在综合考虑雷达、发射车和导弹等武器装备性能的基础上, 基于目标集、资源集建立了最小化目标总拦截价值与生存概率的混合整数决策模型。提出了一种新的改进差分进化算法进行求解, 采用反向学习策略生成初始解, 确保初始种群的质量, 设计了一种快速修复与重构的启发式规则作用于多阶段, 以提升算法的搜索能力。仿真实验验证了该算法具有求解时间和求解精度上的优越性。该研究能使武器系统在动态事件的随机影响下, 保持高效的作战能力和决策效果。

关键词: 防空作战; 动态防空资源分配; 反向学习; 改进差分进化算法

中图分类号: TP391.9 文献标志码: A 文章编号: 1004-731X(2024)06-1285-13

DOI: 10.16182/j.issn1004731x.joss.24-0116

引用格式: 罗天羽, 邢立宁, 王锐, 等. 基于改进差分进化算法的动态防空资源分配优化[J]. 系统仿真学报, 2024, 36(6): 1285-1297.

Reference format: Luo Tianyu, Xing Lining, Wang Rui, et al. Dynamic Air Defense Resource Allocation Optimization Based on Improved Differential Evolution Algorithm[J]. Journal of System Simulation, 2024, 36(6): 1285-1297.

Dynamic Air Defense Resource Allocation Optimization Based on Improved Differential Evolution Algorithm

Luo Tianyu¹, Xing Lining^{2*}, Wang Rui¹, Wang Ling³, Shi Jianmai¹, Sun Xin¹

(1. College of Systems Engineering, National University of Defense Science and Technology, Changsha 410073, China; 2. College of Electronic Engineering, Xi'an University of Electronic Science and Technology, Xi'an 710071, China; 3. Department of Automation, Tsinghua University, Beijing 100084, China)

Abstract: Based on the integrated performance of weapon equipments such as radars, launchers and missiles, a mixed-integer decision model that minimizes the total target intercept value and the probability of survival based on Target-Set, Resource-Set is developed. A new improved differential evolutionary algorithm has been introduced to solve the problem, and the initial solutions is generated by using the reverse learning strategies to ensure the quality of the initial populations. An inspiration rule for the fast repair and reconstruction is designed to work at multi-stage to improve the search capability of the algorithm. The simulation experiment results show the algorithm's superiority in search time and search accuracy, which can maintain the efficient combat capabilities and decision-making under the random influence of dynamic events.

Keywords: air defense operations; dynamic air defense resource allocation; opposition-based learning; improved differential evolution algorithm

收稿日期: 2024-01-30 修回日期: 2024-04-27

基金项目: 国家自然科学基金重点项目(62036006); 陕西省重点科技创新团队项目(2023-CX-TD-07); 陕西省重点研发计划(2024GH-ZDXM-48)

第一作者: 罗天羽(1995-), 男, 博士生, 研究方向为复杂系统建模优化。E-mail: luotianyu951005@163.com

通讯作者: 邢立宁(1980-), 男, 教授, 博士, 研究方向为智能优化和资源调度。E-mail: lnxing@xidian.edu.cn

0 引言

武器-目标分配 (weapon-target assignment, WTA) 问题是军事运筹学国防应用的一个基础性课题, 其目的是为在适当的时刻向进攻目标分配指定武器, 以减少己方部队资产的预期损害^[1]。随着军事技术和武器装备水平的迅猛进步, 现代化战争中, 武器系统呈现出高度多样性, 并表现出愈加复杂的性能特征。在此背景下, 面对复杂多样的现实威胁, 传统、单一的动能拦截手段已难以有效应对^[2]。动态防空资源分配作为武器目标分配问题的重要组成部分, 正面临着新的威胁与挑战。因此, 为了更好地使武器在各类动态事件的影响下保持作战能力, 亟需一种针对典型动态防空资源分配问题的建模求解方法, 以更好地实现多种武器能力有效整合, 提高防空反导能力, 助力构建高效的战略防御体系。

文献[3]提出了 WTA 问题的概念, 并证明了该问题为 NP-Complete 问题^[4]。WTA 问题属于经典的组合优化问题, 其复杂性随着武器和目标的数量及数量呈指数级增长。目前, 常见的 WTA 问题主要分为两大类: 静态武器目标分配 (SWTA) 和动态武器目标分配 (DWTA)^[5]。SWTA 与 DWTA 问题的主要区别在于前者规定所有武器在一个阶段攻击目标, 而后者则属于一种多阶段问题, 决策者需要综合评估当前阶段的结果, 以供后续决策参考。

针对 SWTA 问题, 国内外许多学者在不同的作战场景下, 将目标拦截概率、目标价值、目标威胁值和武器成本等因素转化为目标函数, 构建了不同的 SWTA 数学模型进行求解^[3,6-13]。文献[3]综合考虑来袭目标、有限数量的武器和已知拦截概率等因素, 对 SWTA 问题进行了建模。文献[6]以拦截概率为优化目标, 提出了一种基于期望杀伤概率的武器分配优化算法。不同于以往研究将 WTA 问题作为非线性模型来处理的方法, 文献[7]将该问题作为线性模型来处理, 并提出了一种新

的精确算法进行求解。

在实际作战中, 目标的连续出现、武器的损坏/故障等现象往往呈现多阶段性, 因此, 研究 DWTA 问题更符合现实作战需求。文献[14]将目标函数定义为武器系统接战顺序的不泄密概率最大化和稳定性最大化, 构建了一种新的双目标模型。文献[15]考虑了多个阶段出现的目标造成的威胁排序顺序不同, 提出了一种时间采样动态武器分配模型。考虑到作战行动的有效性和结果在很大程度上取决于对现有传感器和武器的合理利用, 文献[16]考虑了武器与传感器之间的相互依赖关系、资源约束、能力约束和策略约束, 构建了传感器与武器协同配置的数学模型。文献[17]研究了 DWTA 问题中存在的确定性因素, 将其描述为一个非线性整数规划模型, 并提出了一种两阶段混合启发式搜索算法。文献[18]为了加快求解 DWTA 问题的速度, 提出了一种改进的简化群优化算法。防空资源分配问题作为一种任务分配问题, 已有学者针对更加复杂的场景提出了诸如遗传算法、模拟退火算法和粒子群算法等有效的方法^[19-25]。

现阶段关于 DWTA 问题的研究还较少, 且主要集中在对复杂动态作战环境下雷达信号资源、传感器等资源的分配, 而缺乏对高烈度对抗下防空资源分配方案的快速修复和重构。本文研究了一种新的动态防空资源分配问题, 旨在将目标突现、武器资源故障等典型动态因素进行综合考虑, 实现在各类动态事件的随机影响下保持高效作战能力和决策效果的目的。

1 问题描述及数学模型

1.1 问题描述

带武器类型的武器目标分配问题是一种典型的传统 WTA 问题^[2], 在传统 WTA 问题的基础上考虑武器类型时, 相同属性的武器不需单独进行处理。该问题在形式上可定义为在指定区域内, 假设配备 m 种武器, $w_i (i = 1, 2, \dots, m)$ 为每种武器的具

体数量; 存在 n 个来袭目标, 且每个目标 j 具有一定的价值 $V_j (j=1, 2, \dots, n)$; p_{ij} 为目标 j 分配武器 i 时的拦截概率; x_{ij} 为分配给目标 j 的 i 类武器数量; 优化目标为最小化价值加权总生存概率。这类问题也被证明为经典的整数非线性规划问题^[2], 其目标函数及约束为 $\min \sum_{j=1}^n v_j \left[\prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \right]$ 。

分配给所有目标的每种武器数量不能超过其上限:

$$\sum_{j=1}^n x_{ij} \leq w_i, \forall i=1, 2, \dots, m \quad (1)$$

分配给每一目标的武器数量不能为负数:

$$0 \leq x_{ij} \leq \alpha, \forall i=1, 2, \dots, m, j=1, 2, \dots, n \quad (2)$$

式中: α 为正整数。

由于在作战过程中存在如防御策略、目标和武器的特性、实际的战斗情况等多种因素的影响, 导致 DWTA 的模型构建需要综合分析不同防御场景存在的多种特性^[26]。

面对动态防空资源分配问题的现实状况, 通过对拦截目标的实际需求进行分析和总结, 本文明确了动态防空资源分配的具体研究问题: 在一定时间与区域范围内, 防御方发现 T 个来袭目标, 且目标在不同时刻从不同的位置与角度出现。在来袭目标对防御方造成破坏之前, 防御方可以使用区域范围内的防空资源(雷达和发射车)来对每一目标进行拦截。在来袭目标飞行的过程中, 防御方的防空资源也会受到攻击方的干扰, 导致设备在一定时间内无法工作。雷达和发射车的拦截示意图如图1所示, 圆圈为雷达, 扇形为雷达探测范围; 方块为发射车, 虚线圆为拦截范围。以某个目标为例, 假设其直线飞行, 其预测轨迹为红色虚线。

本文研究的目的是基于当前雷达探测信息和轨迹预测结果, 对每个被雷达探测到的来袭目标, 在规定的时间内结合己方防空资源信息来制定出一套高效的拦截方案。方案由对每一来袭目标使用何种型号的发射车在何时发射导弹, 以及对该

目标采用哪部雷达提供跟踪制导信息所组成。雷达探测的时间范围为指定发射车朝指定目标发射导弹开始至命中目标截止。

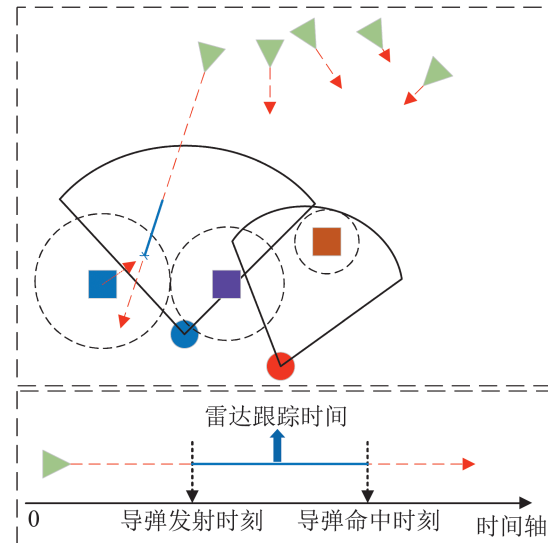


图1 防空资源分配示意图

Fig. 1 Air defense resource allocation diagram

此外, 鉴于该问题为一种在连续时间区间内分配可行方案的连续优化问题, 为优化计算资源, 本文通过设定固定的时间片段将这一连续问题转化为离散问题, 来实现对不同时间片段进行资源分配的目的。基于此, 对动态防空资源分配问题做出如下假设:

- (1) 来袭目标飞行的角度固定, 即目标飞行轨迹为直线;
- (2) 来袭目标和发射车发射的导弹飞行速度恒定, 且匀速飞行;
- (3) 每一发射车只配备同种类型的导弹;
- (4) 每枚导弹在飞行中的每个时间段, 仅有1部雷达在跟踪该导弹拦截的目标。

1.2 数学模型

在实际作战中, 受到来袭目标飞行时间及目标飞行目的地等因素的影响, 其所构建的数学模型需要将上述因素转换为目标的拦截价值, 以进行优化。因此, 该问题的优化目标是在考虑目标拦截价值的前提下最大化总拦截概率, 即最小化

来袭目标总生存概率,以合理运用防空资源,目标函数及约束为 $\min \sum_{i=1}^{n_1} \omega_i \prod_{l=1}^l \left[\prod_{t=1}^{n_2} (1 - x_{i,l}^t \times p_{i,l}^t) \right]$, 其中, ω_i 为目标 i 的拦截价值; $x_{i,l}^t$ 为决策变量, 为1表示目标 i 被发射车 l 发射的导弹在 t 时刻拦截, 否则为0; $p_{i,l}^t$ 为目标 i 在时刻 t 被发射车 l 拦截的概率。

对目标 i 发射导弹数量约束:

$$0 \leq \sum_{l=1}^{n_2} x_{i,l}^t \leq 1, \forall i \in I, t \in \phi_i \quad (3)$$

式中: I 为目标集合, $I = \{1, 2, \dots, n_1\}$; ϕ_i 为目标 i 的可拦截时间节点集合, $\phi_i = \{1, 2, \dots, t_i\}, t_i = \lceil T_i/\mu \rceil$; T_i 为目标 i 的最大飞行时长; μ 为时间离散化标准。

发射车 l 发射的导弹数量不能超过上限:

$$\sum_{i=1}^{n_1} \sum_{t=1}^{t_i} x_{i,l}^t \leq \theta_l, \forall l \in L \quad (4)$$

式中: L 为发射车集合, $L = \{1, 2, \dots, n_2\}$; θ_l 为发射车 l 配备的导弹数量。

雷达 r 在任一时刻 t 所使用的通道数不超过上限, 其中雷达通道数上限表示来袭目标在某一时刻被一部雷达跟踪所占用的通道资源, $t_{i,p}^c$ 为 $t_{i,p}^s$ 加上导弹 p 飞行的总时间:

$$0 \leq c_{i,r} \sum_{p=1}^{n_1} y_{i,r}^t \leq C_r, \forall r \in R, t \in \left[\left\lceil \frac{t_{i,p}^s}{\mu} \right\rceil, \left\lceil \frac{t_{i,p}^c}{\mu} \right\rceil \right] \quad (5)$$

式中: $t_{i,p}^s$ 为第 p 枚导弹朝目标 i 发射的时刻; $t_{i,p}^c$ 为第 p 枚导弹命中目标 i 的时刻; $c_{i,r}$ 为目标 i 占用雷达 r 的通道数; C_r 为雷达 r 的最大通道数。

朝目标 i 发射导弹至命中这个过程中只允许一部雷达进行跟踪:

$$y_{i,r}^t = y_{i,r}^{t+1}, \forall t \in \left[\left\lceil \frac{t_{i,p}^s}{\mu} \right\rceil, \left\lceil \frac{t_{i,p}^c}{\mu} \right\rceil \right], r \in R, i \in I \quad (6)$$

式中: R 为雷达集合, $R = \{1, 2, \dots, n_3\}$; $y_{i,r}^t$ 为决策变量, 为1表示目标 i 在时刻 t 被雷达 r 跟踪, 否则为0;

$$x_{i,l}^t \in \{0, 1\}, \forall i \in I, l \in L, t \in \phi_i \quad (7)$$

$$y_{i,r}^t \in \{0, 1\}, \forall i \in I, r \in R, t \in \phi_i \quad (8)$$

2 算法设计

通过对连续时间进行固定时间片段的划分, 将动态防空资源分配问题转化为离散问题, 即组合优化问题。在面对复杂的组合优化问题时, 设计高效的优化算法一直是学术界关注的热点问题。目前, 解决带有复杂约束的组合优化问题的主要方法包括精确算法、启发式算法和元启发式算法。由于动态防空资源分配问题存在强约束、多目标、离散变量共存等复杂性, 精确算法和启发式算法难以在短时间有效求解这些问题。

差分进化算法(differential evolution, DE)作为一种经典的元启发式算法, 因其优秀的可靠性和鲁棒性被广泛用于求解各类复杂组合优化问题^[17,27], 并在部分资源分配问题中取得了较好成果^[12,28-29]。因此, 本文基于DE算法, 结合反向学习策略和启发式规则, 提出了一种改进的差分进化算法(IDE)来求解动态防空资源分配问题。

为避免算法过早陷入局部最优, 本文在产生初始种群和迭代过程中, 引入了反向学习的机制来提升种群的多样性。在算法迭代过程中, 设计了一种局部搜索算子提升算法的局部搜索能力。在突发状况发生后, 通过构造一种快速修复与重构的规则作用于已分配方案, 来减少突发状况对已分配方案中需要再次分配的目标数量, 降低解空间, 如图2所示。

2.1 初始种群构造

在DE算法中, 初始种群的产生主要为随机原则, 而由于随机原则的不确定性, 其产生种群多样性的好坏会直接导致求解效率的差异。为了降低随机原则产生初始种群对算法求解效率的影响, 本文在产生初始种群的过程中引入反向学习策略, 通过分析候选解在解空间中的镜像位置来增加种群多样性, 提升初始种群的质量和算法的收敛速度^[30]。

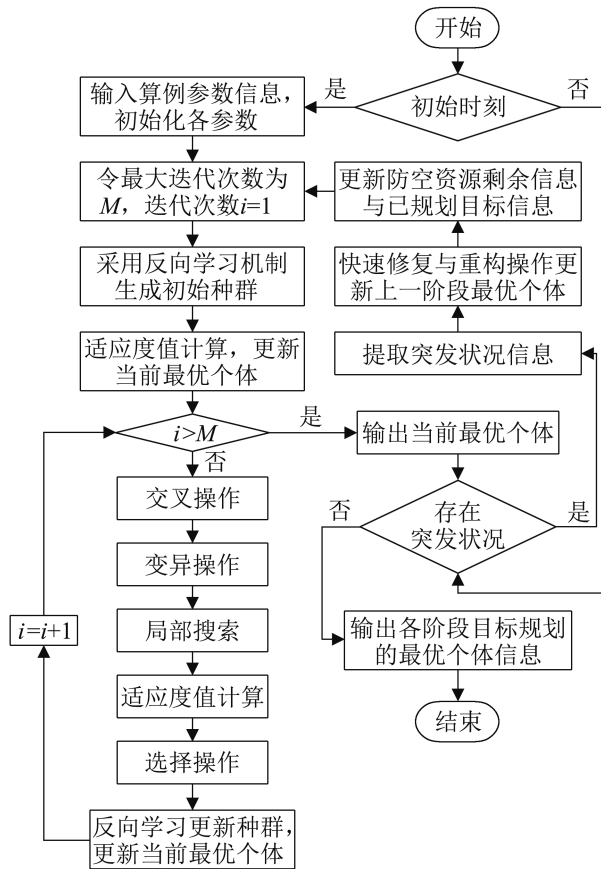


图2 算法流程图
Fig. 2 Algorithm flow chart

此外, 考虑到动态防空资源分配问题需要的方案信息包括对每一来袭目标分配的雷达、发射车和导弹发射时间, 本文构造初始种群如算法 1 所示。

算法 1: 种群初始化

输入: 算例参数 Parameter; 种群大小 PopSize

输出: 父代种群 Population

Scheme=Parameter.data; //提取拦截方案集合信息

NTarget=numel(Parameter.Target); //获取目标个数

for i=1: PopSize

for j=1: NTarget

pick=TournamentSelection(); //锦标赛选择

decs(j)= pick;

decs1= Scheme(pick);

```

end for
Population(i).decs=decs;
end for
//反向学习产生新个体
TmpPop=RevLearning();
//根据染色体信息, 判断约束, 并计算适应度值
Population=CalObj();
return Population
    
```

根据算法 1, 对初始种群的构造过程进行描述: 通过输入的雷达信息(坐标、类型、探测范围与探测角度)、发射车信息(坐标、类型、拦截范围与导弹飞行速度)来分析对当前区域的可探测范围与拦截范围信息。根据此类信息, 结合时间离散化结果, 针对每一来袭目标, 按照导弹发射时间的先后顺序可得出其可被拦截的方案信息集合, 并以此作为初始种群构造的输入参数 Parameter。其中, 每一可拦截方案中包括发射车编号 l 、雷达编号 r 、导弹发射时刻 $t_{i,p}^s$ 、导弹命中的时刻 $t_{i,p}^e$ 与拦截概率 $p_{i,l}^t$ 。个体编码示意图如图 3 所示。

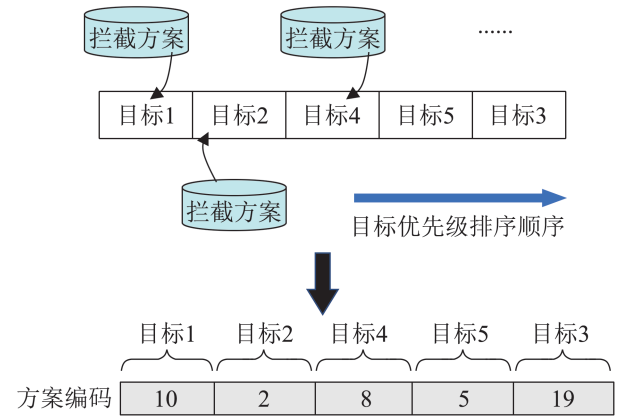


图3 个体编码示意图
Fig. 3 Individual coding schematic diagram

不同于以往在 DE 算法中采用随机向量生成初始种群的方法, 本文使用锦标赛选择的方法, 将每一目标的备选方案集合作为输入参数, 按照拦截概率的大小从备选方案集中为每一目标选择一种拦截方案。

在初始种群产生的过程中引入了反向学习的思想^[31]，将原始初始种群作为原候选解集，用式(11)的方法产生反向解集。

$$y_i = a_i + b_i - x_i \quad (9)$$

式中： y_i 为反向数； $a_i=1$ 为个体 i 的编码取值下界； b_i 为个体 i 的编码取值上界； x_i 为对目标 i 分配的方案编号， $x_i \in [a_i, b_i]$ 。

在生成原候选解集和反向解集后采用反向学习方法来挑选新的解集进行迭代计算，反向学习方法为

$$x_i = \begin{cases} x_i, f(x_i) > f(y_i) \\ y_i, \text{其他} \end{cases} \quad (10)$$

式中： f 为适应度函数值，表示在考虑目标优先级的前提下最小化来袭目标总生存概率。

2.2 交叉变异策略

交叉变异操作的作用在于通过可行操作，最大程度地生成更优个体，以推动种群的进化过程。在DE算法中，变异操作是其中一个关键步骤，本文采用DE/rand-to-best/1的变异操作来生成新的个体。该变异操作为

$$y_i = \left\lceil \left\lfloor x_i + F[(x_{\text{best}} - x_i) + (x_{r_1} - x_{r_2})] \right\rfloor \right\rceil \quad (11)$$

式中： x_i 为当前选中的个体； F 为比例系数； x_{best} 为当前种群中最优的个体； x_{r_1} 和 x_{r_2} 为从Population中随机选取的个体； $r_1 \neq r_2 \neq i$ 。

由于编码的特性，需对产生的个体 y_i 进行范围判断。如果个体 $y_i > b_i$ ，则需进行修复操作：

$$y_i' = \begin{cases} b_i, y_i/b_i = \lfloor y_i/b_i \rfloor \\ y_i - \lfloor y_i/b_i \rfloor b_i, \text{其他} \end{cases} \quad (12)$$

在进行变异操作之后需进行交叉操作：

$$u_{i,j} = \begin{cases} y_{i,j}', \xi \leq P_c \text{ 或 } j = \tau \\ x_{i,j}, \text{其他} \end{cases} \quad (13)$$

式中： $u_{i,j}$ 为交叉向量； $x_{i,j}$ 为原种群中第 i 个个体的第 j 维数据； τ 为区间 $[1, D]$ 中的一个随机整数； ξ 为区间 $[0, 1]$ 中的一个随机数； D 为染色体的维度，表示当前阶段待分配方案的目标数量； P_c 为交叉

概率。

在交叉操作完成后，需将该步骤生成的所有新个体与当前父代种群进行比较，以筛选出较为优秀个体，作为下一轮迭代的父代种群：

$$x_i = \begin{cases} u_i, f(u_i) > f(x_i) \\ x_i, \text{其他} \end{cases} \quad (14)$$

2.3 启发式规则设计

为了提升算法的局部搜索能力，本文在传统DE算法中设计了一种基于启发式规则的局部搜索算子，伪代码如算法2所示。

算法2：局部搜索算子

输入：算例参数 Parameter；个体信息

Solution

输出：个体信息 Solution

//提取相关信息

Launch_missile=Solution.missile; //发射车剩余
弹量

NTarget=Parameter.NTarget; //目标总数

Scheme=Parameter.data; //拦截方案集合信息

if sum(Launch_missile)>0

for $i=1:NTarget$

if Solution.cons(i)~=0

//找到当前目标拦截最高概率

maxP=max(Scheme.p(i));

curP= Solution.cons(i);

if curP<maxP

//释放当前方案占用雷达与发射车资源

信息

ReleaseResource;

else

Continue

end if

end if

CurScheme= Scheme(i);

CurScheme=sort(CurScheme);

for $j=1:\text{numel}(\text{CurScheme})$


```

ConstraintJudge(); //检查约束
RadarFill(); //按规则填充雷达编号
//更新当前个体方案信息
NewSolution=UpdateScheme();
    end for
end for
end if
if NewSolution.objs>Solution.objs
    Solution=NewSolution;
end if
return Solution

```

如算法 2 所示, 在进行交叉操作后, 本文采用启发式规则对个体进行优化。主要操作为结合剩余防空资源, 在满足所有约束的前提下, 从方案集合中挑选拦截概率最高的方案进行分配。

为了应对突发状况对已有分配方案造成的影响, 设计了一种启发式规则来进行方案的快速修复与重构, 伪代码如算法 3 所示。

算法 3: 快速修复与重构

输入: 算例参数 **Parameter**; 最优个体信息 **BestSol**

输出: 最优个体 **BestSol**; 待分配个体 **NedTarget**

```

EmergSit=Parameter.Sit; //提取突发状况信息
CurTim= Parameter.Tim; //提取当前时刻
//提取已出现但还未拦截的目标
RestTarget=SelectRest(CurTim);
if ~isempty(EmergSit.NewTarget)
//当前时刻出现了新的拦截目标
//根据拦截价值高低进行排序,并从RestTarget
选出拦截价值低于NewTarget中最高价值的所有
个体
[TmpTarget,RestTarget]=Sort();
//释放 TmpTarget 占用的雷达、发射车资源,
更新剩余防空资源信息
RestResource=RelsResource();

```

```

//更新待分配个体信息
NewTarget=UpdateTarget();
end if
//当前时刻出现了雷达、发射车故障
if ~isempty(EmergSit.Resource)
//更新雷达、发射车剩余资源信息
RestResource=UpdateResource();
//找出RestTarget中受到影响的所有目标,采用
规则的方法对原有方案进行修复与重插入
NedTarget=SelectTarget();
for i=1:numel(NedTarget)
    curTarget=NedTarget(i);
//从备选方案中找出满足约束条件且拦截
概率最高的方案进行修复与重插入
    BestSol=RepairSol();
end for
end if
return BestSol NedTarget

```

如算法 3 所示, 考虑到可能发生的突发状况(新目标的出现、雷达和发射车故障等)对已规划方案造成的影响, 本文在不同阶段之间设计了一种启发式规则对已有方案进行快速修复与重构。其具体修复与重构步骤如下:

(1) 已发生的突发状况判断是否出现了新的来袭目标。如果存在新的来袭目标, 则需要将已分配但还未拦截的目标与来袭目标进行综合判断, 从中选出拦截价值低于新的来袭目标中拦截价值最高的所有个体, 并更新已分配目标与方案信息。

(2) 判断此刻是否出现雷达、发射车故障的突发状况。如果存在, 则需要逐一对已分配方案但还未拦截的目标进行筛选, 从中选出需要进行修复与重构的个体。

(3) 对需要修复的个体, 优先选择满足雷达通道约束与发射车剩余弹量约束的资源进行修复与重构。

3 实验与结果分析

3.1 测试用例构造

为了检验IDE算法在动态防空资源分配问题下的有效性,分别在3种不同规模的算例下构造了不同突发状况来进行仿真实验分析。此外,考虑动态防空资源分配问题的复杂程度也与防空资源数量(雷达数量和通道数上限、发射车数量与配备导弹数量)相关,本文根据不同规模算例构造了资源不足和资源充足2类场景:资源不足表示所有雷达通道数上限为1,发射车配备的总导弹数量小于目标总数;资源充足表示所有雷达通道数上限大于1,发射车配备的总导弹数量大于目标总数。相关算例信息如表1所示。算例S5表示目标在0、10、20 s分别出现15、5、4个目标;1辆发射车在10 s发生故障;1部雷达在15 s发生故障。

3.2 实验环境与参数设置

为评估改进差分进化算法的性能,从元启发式算法中挑选DE、SA、GA和改进遗传算法(IGA)^[32]与IDE进行对比分析。本文的实验均在配置为AMD Ryzen 3700x 4.2GHz(16 GB RAM),操作系统为Win10的环境下进行,上述所有算法均在MATLAB2021b上实现。此外,DE、SA、GA和IGA算法不进行算法3的操作,且在不同时刻只进行重复调用算法操作。

各算法的参数设置:DE和IDE中的缩放因子 F 为0.8;交叉因子 P_c 为0.8;种群大小为30;迭代次数为100。SA和SA/wr中的初始温度为80;降温速率为0.999 5;终止温度为10。GA、IGA和GA/wr中的种群大小为30;迭代次数为100;交叉概率为0.8;变异概率为0.2。此外,模型中的时间离散化标准 μ 为5。

表1 算例信息
Table 1 Example information

算例	目标信息		发射车			雷达		
	数量	出现时间/s	数量	导弹数量	故障数量/时间(s)	数量	通道上限	故障数量/时间(s)
S0	24	0	6	24	/	6	1	/
S1	15/5/4	0/10/20	4	20	/	4	1	/
S2	15/5/4	0/10/20	4	40	/	4	2	/
小规模 S3	24	0	6	24	1/10	5	2	1/15
S4	24	0	6	48	1/10	5	2	1/15
S5	15/5/4	0/10/20	8	32	1/10	5	3	1/15
S6	15/5/4	0/10/20	8	64	1/5	5	3	1/15
M1	40/5/15	0/10/20	8	56	/	7	1	/
M2	40/5/15	0/10/20	8	112	/	7	1	/
中规模 M3	60	0	10	60	1/10	8	2	1/15
M4	60	0	10	120	1/10	8	2	1/15
M5	40/5/15	0/10/20	12	72	1/10	8	3	1/15
M6	40/5/15	0/10/20	12	132	1/5	8	3	1/15
L1	60/20/20	0/10/20	14	98	/	14	1	/
L2	60/20/20	0/10/20	14	168	/	14	1	/
大规模 L3	100	0	16	112	1/10	16	2	1/15
L4	100	0	16	208	1/10	16	2	1/15
L5	60/20/20	0/10/20	18	144	1/10	18	3	1/15
L6	60/20/20	0/10/20	18	198	1/5	18	3	1/15

注: / 代表无值。

3.3 实验结果与分析

3.3.1 算法有效性分析

为了验证IDE求解动态防空资源问题的有效性,基于表1中的算例,将IDE与其他算法进行具体分析。为了进一步验证启发式规则的作用,本文也设计了带有启发式规则的SA(SA/wr)与带有启发式规则的GA(GA/wr)进行了对比分析。其中,取所有算法重复运行10次的平均目标值 C_{avg} 、最优目标值 C_{best} 和平均求解时间 T 进行对比分析。求解结果如表2~4所示。此外,本文还引入了GAP值来分析不同算法求解结果之间的差距,其计算公式为

$$GAP = \frac{C_0 - C_1}{\max\{C_0, C_1\}} \quad (17)$$

式中: C_1 为IDE的最优目标值; C_0 为对比算法的最优目标值。

由表2~4可知,IDE相较于其他算法的优势主

要体现在:

(1)在求解质量方面,本文通过对不同规模下的算例采用不同算法进行求解,结果表明,IDE在所有规模的算例中均表现出显著优势。其中,在3种规模的算例下,IDE相较于DE,其平均提升率分别达到了9.68%、5.82%和11.3%。此外,加入启发式规则后的SA/wr和GA/wr对比SA和GA虽然能输出更优解,但相较IDE还存在一定差距。由此可知,通过在不同阶段加入快速修复与重构机制的启发式规则,能够较好地对不同阶段出现的各类不同价值的目标进行合理分配,并能有效结合防空资源信息对目标实现精准分配,最小化来袭目标生存概率。从IDE在求解质量上的提升也能看出,本文将反向学习机制融入该算法中,能够有效避免算法过早陷入局部最优,提升算法对复杂解空间的搜索能力。

表2 IDE与DE和SA的求解结果对比
Table 2 Solution results of IDE compared with DE and SA

算例	DE				SA				IDE		
	C_{best}	C_{avg}	T/s	GAP/%	C_{best}	C_{avg}	T/s	GAP/%	C_{best}	C_{avg}	T/s
S0	23.186	23.540	1.999	8.530	29.448	31.878	3.147	27.980	21.208	22.179	1.939
S1	30.594	31.233	1.422	2.920	32.192	33.285	2.253	7.740	29.700	30.312	0.782
S2	19.572	20.511	1.896	12.090	27.512	28.934	2.328	37.460	17.206	17.577	0.780
S3	24.436	24.818	1.754	11.240	28.124	29.560	2.081	22.880	21.690	22.787	1.089
S4	18.750	19.851	1.938	10.500	26.016	27.218	2.269	35.490	16.782	16.966	1.239
S5	19.596	20.622	1.867	11.930	24.198	24.981	2.237	28.680	17.258	17.401	1.323
S6	19.440	20.548	1.847	10.580	25.128	26.882	2.104	30.820	17.384	17.482	1.177
M1	130.732	135.295	3.180	8.810	140.204	147.610	4.719	14.980	119.208	126.814	3.210
M2	104.040	109.532	3.243	10.530	120.638	128.076	4.793	22.840	93.088	99.257	3.131
M3	120.594	126.194	3.802	6.090	138.024	140.968	5.922	17.950	113.248	120.483	4.253
M4	94.402	99.800	4.270	5.360	134.086	139.738	6.118	33.370	89.338	94.270	4.268
M5	105.384	111.063	3.391	1.110	133.684	138.927	4.264	22.050	104.212	107.245	3.289
M6	100.322	106.566	3.422	2.990	137.224	143.133	4.033	29.080	97.318	103.013	3.419
L1	278.544	281.555	5.662	7.620	279.014	290.922	10.012	7.770	257.332	263.581	5.941
L2	250.356	253.103	5.810	15.490	268.276	275.675	10.144	21.140	211.572	230.211	6.230
L3	248.100	250.223	7.716	7.140	259.234	269.988	12.895	11.130	230.394	236.956	7.899
L4	225.410	227.661	8.439	9.860	258.346	264.050	13.482	21.360	203.174	211.253	8.146
L5	223.144	224.941	7.208	13.060	257.184	264.182	8.974	24.560	194.010	206.494	6.619
L6	223.878	228.582	7.261	14.630	252.328	262.629	8.869	24.250	191.130	196.640	6.630

表3 IDE与GA和IGA的求解结果对比
Table 3 Solution results of IDE compared with GA and IGA

算例	GA				IGA				IDE		
	C_{best}	C_{avg}	T/s	GAP/%	C_{best}	C_{avg}	T/s	GAP/%	C_{best}	C_{avg}	T/s
S0	23.684	24.351	2.005	10.450	22.486	23.475	2.663	5.680	21.208	22.179	1.939
S1	33.678	33.873	0.823	11.810	30.782	31.045	1.025	3.520	29.700	30.312	0.782
S2	22.688	23.650	1.890	24.160	18.798	19.427	1.065	8.470	17.206	17.577	0.780
S3	27.416	27.801	2.020	20.890	24.712	25.114	1.562	12.230	21.690	22.787	1.089
S4	22.058	23.746	1.821	23.920	19.214	19.806	1.771	12.660	16.782	16.966	1.239
S5	23.138	24.300	1.817	25.410	20.278	20.841	1.758	14.890	17.258	17.401	1.323
S6	23.282	24.116	1.829	25.330	19.792	20.551	1.532	12.170	17.384	17.482	1.177
M1	146.094	149.081	3.587	18.400	129.400	133.362	4.265	7.880	119.208	126.814	3.210
M2	122.730	129.209	3.511	24.150	106.064	110.423	4.226	12.230	93.088	99.257	3.131
M3	133.604	135.879	4.168	15.240	129.538	131.975	4.952	12.580	113.248	120.483	4.253
M4	124.300	126.213	4.584	28.130	113.840	117.509	5.449	21.520	89.338	94.270	4.268
M5	120.808	127.354	3.537	13.740	118.502	123.771	3.946	12.060	104.212	107.245	3.289
M6	124.156	128.557	3.393	21.620	120.212	125.497	3.997	19.040	97.318	103.013	3.419
L1	295.112	300.402	6.292	12.800	262.988	273.304	7.573	2.150	257.332	263.581	5.941
L2	275.074	283.269	7.203	23.090	240.190	244.523	7.573	11.910	211.572	230.211	6.230
L3	264.708	273.724	9.271	12.960	240.290	252.052	10.153	4.120	230.394	236.956	7.899
L4	234.522	245.642	9.925	13.370	226.090	233.112	11.196	10.140	203.174	211.253	8.146
L5	226.352	238.521	7.666	14.290	228.340	236.458	8.446	15.030	194.010	206.494	6.619
L6	229.078	232.032	7.623	16.570	220.300	231.101	8.124	13.240	191.130	196.640	6.630

表4 IDE与SA/wr和GA/wr的求解结果对比
Table 4 Solution results of IDE compared with SA/wr and GA/wr

算例	SA/wr				GA/wr				IDE		
	C_{best}	C_{avg}	T/s	GAP/%	C_{best}	C_{avg}	T/s	GAP/%	C_{best}	C_{avg}	T/s
S0	22.744	24.267	3.597	6.750	22.100	22.509	2.530	4.040	21.208	22.179	1.939
S1	30.594	31.276	3.649	2.800	30.262	30.835	1.900	1.730	29.700	30.312	0.782
S2	19.624	20.415	3.601	12.320	17.920	18.530	3.804	3.980	17.206	17.577	0.780
S3	22.242	24.128	3.118	0.450	23.580	24.607	3.726	6.100	21.690	22.787	1.089
S4	18.750	19.836	3.942	10.500	18.286	19.327	3.379	8.220	16.782	16.966	1.239
S5	19.690	20.499	3.666	12.350	18.942	20.013	3.131	8.890	17.258	17.401	1.323
S6	19.440	20.405	3.555	10.580	18.850	19.694	2.995	7.780	17.384	17.482	1.177
M1	132.124	135.744	6.329	9.780	133.334	136.122	5.210	10.590	119.208	126.814	3.210
M2	107.552	110.099	5.852	13.260	107.680	110.728	5.498	13.370	93.088	99.257	3.131
M3	127.606	131.423	7.238	11.250	125.230	127.266	5.338	9.570	113.248	120.483	4.253
M4	98.314	100.516	7.891	9.130	98.314	100.516	5.842	9.130	89.338	94.270	4.268
M5	120.734	124.774	5.960	13.680	108.574	112.425	4.933	4.020	104.212	107.245	3.289
M6	119.654	125.907	5.159	17.730	102.052	108.754	4.467	3.540	97.318	103.013	3.419
L1	269.386	277.018	11.143	4.470	271.560	277.173	7.976	5.240	257.332	263.581	5.941
L2	243.190	250.130	11.237	13.000	242.016	246.666	8.605	12.580	211.572	230.211	6.230
L3	252.132	256.430	13.903	8.620	242.080	250.143	11.254	4.830	230.394	236.956	7.899
L4	243.410	245.558	14.905	15.790	233.554	236.306	11.328	12.230	203.174	211.253	8.146
L5	219.578	232.414	10.630	11.640	230.346	235.586	9.287	15.770	194.010	206.494	6.619
L6	217.082	225.181	10.592	11.950	218.050	222.727	8.778	12.350	191.130	196.640	6.630

(2) 在求解时间方面, 从表2~4中的求解结果可以看出, IDE在所有算例下的求解时间都优于其他算法。其中, IDE在所有算例下的平均求解时间仅为3.756 s, 而DE、SA、GA和IGA方法的求解时间分别为4.007、5.823、4.367、4.804 s。由此可以验证本文将启发式规则的运算效率及定制化优势与DE算法的广域搜索优势相结合, 能够有效避免无效搜索, 在相同时间内能获得更高质量的解。

为了进一步验证IDE算法的收敛效果, 本文构造了算例S0, 即所有目标在同一时刻出现, 且在目标飞行的过程中无其他突发状况发生, 以此来对比7种算法在求解过程中的收敛速度, 如图4所示。

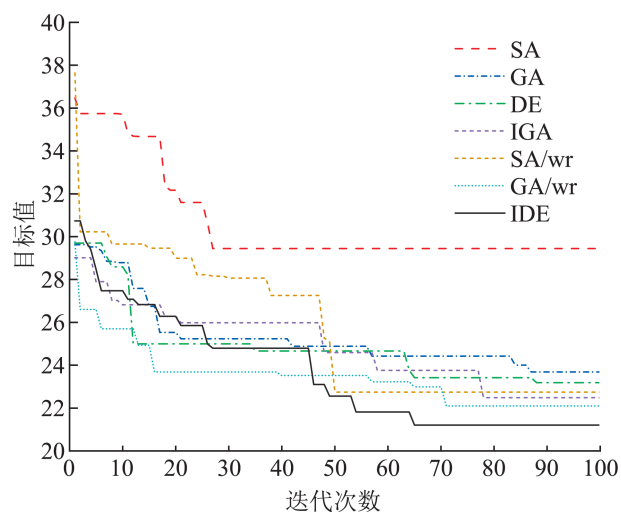


图4 算法迭代图

Fig. 4 Algorithm iteration diagram

根据图4可知, 相比其他算法, IDE表现出较好的收敛速度, 且对复杂解空间表现出更强的搜索能力。其中, 通过对比DE与IDE的收敛图可知, 在反向学习与局部搜索算子的作用下, IDE能在复杂的解空间以更高效的搜索能力寻到较优解。其主要原因在于IDE将启发式规则与演化算法相结合, 实现了两者之间的优势互补。在算法迭代过程中, IDE采用了规则的方法对部分个体进行资源释放与再分配, 提高了其对局部空间的

搜索能力。突发状况发生后, IDE采用快速修复与重构的策略, 避免算法对个体进行重复搜索, 降低了解空间复杂度。

3.3.2 算法稳定性分析

为了进一步验证IDE的稳定性, 本文在所有算例下采用IDE重复求解10次, 并计算各算例下的平均目标值、标准差和平均目标值与最优目标值之间的相对差, 结果如表5所示。

表5 稳定性实验结果
Table 5 Experimental results of stability

算例	平均值	最优解	相对差	标准差
S0	22.179	21.208	0.971	0.534
S1	30.312	29.700	0.612	0.462
S2	17.577	17.206	0.371	0.476
S3	22.787	21.690	1.097	0.567
S4	16.966	16.782	0.184	0.137
S5	17.401	17.258	0.143	0.143
S6	17.482	17.384	0.098	0.119
M1	126.814	119.208	7.606	4.590
M2	99.257	93.088	6.169	4.168
M3	120.483	113.248	7.235	2.830
M4	94.270	89.338	4.932	2.645
M5	107.245	104.212	3.033	1.760
M6	103.013	97.318	5.695	2.965
L1	263.581	257.332	6.249	5.675
L2	230.211	211.572	18.639	11.240
L3	236.956	230.394	6.562	4.736
L4	211.253	203.174	8.079	3.885
L5	206.494	194.010	12.484	6.201
L6	196.640	191.130	5.510	2.649

由表5可知, IDE在进行10次重复求解时, 其平均结果与最优解之间的差距较小。IDE在3种规模算例下的平均相对差分别为0.497、5.778和9.587, 最大相对差仅为L2算例下的18.639。此外, 从标准差一栏可知, 相较于平均值, 各个算例下的标准差值较小, 这进一步证实了在多次运算中, IDE所得到的最优解的离散程度相对较低。因此, 对于高强度的动态防空资源分配问题而言, IDE展现出较强的稳定性。

4 结论

本文针对一种高强度环境下的动态防空资源分配问题进行了以下研究。

(1) 研究了一种动态防空资源分配问题, 在此问题中综合考虑多阶段持续出现目标、武器发生故障等突发状况。

(2) 在考虑雷达通道上限、雷达/发射车种类, 以及雷达/发射车数量等多重约束条件的基础上, 构建了动态防空资源分配的数学优化模型。

(3) 提出了一种融合反向学习和启发式规则的IDE算法。通过在初始种群生成过程中引入反向学习机制, 以提升种群多样性; 在算法迭代过程中, 设计定制化的启发式规则提升算法的局部搜索能力; 不同阶段采用了一套快速修复与重构的启发式规则, 以减少算法的无效搜索, 并最大化资源的利用效率。

在3种不同规模和多种突发状况下进行仿真实验, 相较于DE、SA、GA、IGA、SA/wr和GA/wr算法, IDE的优化效果平均提升了8.973%、23.238%、18.754%、11.133%、10.318%和8.103%, 且求解时间明显优于其他算法, 进一步验证了IDE的有效性。

此外, IDE算法虽然在动态防空资源分配优化问题上取得了显著成果, 但仍有进一步研究的空间。在实际作战中, 突发事件不仅包括新目标的出现和资源出故障, 同时也会存在目标的消失和资源受干扰等。因此, 针对更加复杂的动态防空资源分配问题, 如何设计更为高效的算法来实现及时的动态响应, 值得进一步研究。

参考文献:

- [1] 李梦杰, 常雪凝, 石建迈, 等. 武器目标分配问题研究进展:模型、算法与应用[J]. 系统工程与电子技术, 2023, 45(4): 1049-1071.
Li Mengjie, Chang Xuening, Shi Jianmai, et al. Developments of Weapon Target Assignment: Models, Algorithms, and Applications[J]. Systems Engineering and Electronics, 2023, 45(4): 1049-1071.
- [2] Kline A, Ahner D, Hill R. The Weapon-target Assignment Problem[J]. Computers & Operations Research, 2019, 105: 226-236.
- [3] Manne A S. A Target-assignment Problem[J]. Operations Research, 1958, 6(3): 346-351.
- [4] Lloyd S P, Witsenhausen H S. Weapons Allocation Is NP-complete[J]. 1986 Summer Computer Simulation Conference, 1986: 1054-1058.
- [5] 陈英武, 蔡怀平, 邢立宁. 动态武器目标分配问题中策略优化的改进算法[J]. 系统工程理论与实践, 2007, 27(7): 160-165.
Chen Yingwu, Cai Huaiping, Xing Lining. An Improved Algorithm of Policies Optimization of Dynamic Weapon Target Assignment Problem[J]. Systems Engineering-Theory & Practice, 2007, 27(7): 160-165.
- [6] Bogdanowicz Z R, Tolano A, Patel K, et al. Optimization of Weapon-target Pairings Based on Kill Probabilities[J]. IEEE Transactions on Cybernetics, 2013, 43(6): 1835-1844.
- [7] Lu Yiping, Chen D Z. A New Exact Algorithm for the Weapon-target Assignment Problem[J]. Omega, 2021, 98: 102138.
- [8] Ji Eun Kim, Chang Hun Lee, Mun Yong Yi. New Weapon Target Assignment Algorithms for Multiple Targets Using a Rotational Strategy and Clustering Approach[J]. IEEE Access, 2022, 10: 43738-43750.
- [9] Chang Xuening, Shi Jianmai, Luo Zhihao, et al. Adaptive Large Neighborhood Search Algorithm for Multi-stage Weapon Target Assignment Problem[J]. Computers & Industrial Engineering, 2023, 181: 109303.
- [10] 宋贵宝, 强裕功, 刘铁, 等. 动态武器目标分配问题的现状与进展[J]. 兵器装备工程学报, 2022, 43(12): 83-88.
Song Guibao, Qiang Yugong, Liu Tie, et al. The Present Situation and Progress of Dynamic Weapon Target Assignment[J]. Journal of Ordnance Equipment Engineering, 2022, 43(12): 83-88.
- [11] 朱晓雯, 范成礼, 卢盈齐, 等. 基于改进BBO算法和模糊期望效果的反导武器目标分配建模与实现[J]. 系统工程与电子技术, 2023, 45(11): 3544-3554.
Zhu Xiaowen, Fan Chengli, Lu Yingqi, et al. Anti-missile Weapon Target Allocation Modeling and Implementation Based on Improved BBO Algorithm and Fuzzy Expectation Effect[J]. Systems Engineering and Electronics, 2023, 45(11): 3544-3554.
- [12] 吴文海, 郭晓峰, 周思羽, 等. 改进差分进化算法求解武器目标分配问题[J]. 系统工程与电子技术, 2021, 43(4): 1012-1021.
Wu Wenhai, Guo Xiaofeng, Zhou Siyu, et al. Improved Differential Evolution Algorithm for Solving Weapon-target Assignment Problem[J]. Systems Engineering and

- Electronics, 2021, 43(4): 1012-1021.
- [13] 陆一平, 李慧慧. 静态武器目标分配问题的攻击界整数规划求解方法[J]. 系统工程理论与实践, 2019, 39(3): 783-789.
- Lu Yiping, Li Huihui. An Attack-number Bounded Integer Programming Method for the Static WTA Problem[J]. Systems Engineering-Theory & Practice, 2019, 39(3): 783-789.
- [14] Ahmet Silav, Esra Karasakal, Orhan Karasakal. Bi-objective Dynamic Weapon-target Assignment Problem with Stability Measure[J]. Annals of Operations Research, 2022, 311(2): 1229-1247.
- [15] Liu Chang, Li Jiang, Wang Ye, et al. A Time-driven Dynamic Weapon Target Assignment Method[J]. IEEE Access, 2023, 11: 129623-129639.
- [16] Xin Bin, Wang Yipeng, Chen Jie. An Efficient Marginal-return-based Constructive Heuristic to Solve the Sensor-weapon-target Assignment Problem[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, 49(12): 2536-2547.
- [17] Ma Yingying, Wang Guoqiang, Hu Xiaoxuan, et al. Two-stage Hybrid Heuristic Search Algorithm for Novel Weapon Target Assignment Problems[J]. Computers & Industrial Engineering, 2021, 162: 107717.
- [18] Lai C M, Wu T H. Simplified Swarm Optimization with Initialization Scheme for Dynamic Weapon-target Assignment Problem[J]. Applied Soft Computing, 2019, 82: 105542.
- [19] 王正元, 严小琴. 基于仿真的武器-目标分配问题求解方法[J]. 系统仿真学报, 2009, 21(20): 6597-6599.
- Wang Zhengyuan, Yan Xiaoqin. Simulation-based Solution to Weapon-target Assignment[J]. Journal of System Simulation, 2009, 21(20): 6597-6599.
- [20] Yao Feng, Du Yonghao, Li Lei, et al. General Modeling and Optimization Technique for Real-world Earth Observation Satellite Scheduling[J]. Frontiers of Engineering Management, 2023, 10(4): 695-709.
- [21] Li Wei, Jing Jianghui, Chen Yangtao, et al. Evolutionary Experience-driven Particle Swarm Optimization with Dynamic Searching[J]. Complex System Modeling and Simulation, 2023, 3(4): 307-326.
- [22] Du Yonghao, Wang Tao, Xin Bin, et al. A Data-driven Parallel Scheduling Approach for Multiple Agile Earth Observation Satellites[J]. IEEE Transactions on Evolutionary Computation, 2020, 24(4): 679-693.
- [23] Qiao Kangjia, Liang Jing, Yu Kunjie, et al. Self-adaptive Resources Allocation-based Differential Evolution for Constrained Evolutionary Optimization[J]. Knowledge-Based Systems, 2022, 235: 107653.
- [24] 王艺鹏, 辛斌, 陈杰. 多阶段传感器-武器-目标分配问题的建模与优化求解[J]. 控制理论与应用, 2019, 36(11): 1886-1895.
- Wang Yipeng, Xin Bin, Chen Jie. Modeling and Optimization of Multi-stage Sensor-weapon-target Assignment[J]. Control Theory & Applications, 2019, 36(11): 1886-1895.
- [25] 陈晖, 马亚平. 基于多级协同MOEA/D的联合火力打击目标分配方法[J]. 系统仿真学报, 2018, 30(8): 2942-2949.
- Chen Hui, Ma Yaping. Model of Target Assignment in Joint Fire Strike Operations[J]. Journal of System Simulation, 2018, 30(8): 2942-2949.
- [26] Xin Bin, Chen Jie, Peng Zhihong, et al. An Efficient Rule-based Constructive Heuristic to Solve Dynamic Weapon-target Assignment Problem[J]. IEEE Transactions on Systems Man and Cybernetics Part A-Systems and Humans, 2011, 41(3): 598-606.
- [27] 周鹤翔, 徐扬, 罗德林. 针对动态目标的多无人机协同组合差分进化搜索方法[J]. 控制与决策, 2023, 38(11): 3128-3136.
- Zhou Hexiang, Xu Yang, Luo Delin. A Composite Differential Evolution Algorithm for Multi-UAV Cooperative Dynamic Target Search[J]. Control and Decision, 2023, 38(11): 3128-3136.
- [28] Li Jianyu, Du Kejing, Zhan Zhihui, et al. Distributed Differential Evolution with Adaptive Resource Allocation[J]. IEEE Transactions on Cybernetics, 2023, 53(5): 2791-2804.
- [29] 隆雨佟, 陈爱国, 史红权, 等. 基于改进差分进化算法的跨平台武器目标分配方法[J]. 系统工程与电子技术, 2024, 46(3): 953-962.
- Long Yutong, Chen Aiguo, Shi Hongquan, et al. Cross Platform Weapontarget Allocation Method Based on Improved Differential Evolution Algorithm[J]. Systems Engineering and Electronics, 2024, 46(3): 953-962.
- [30] Kiran Ilyas, Irfan Younas. Enhancing Dynamic Multi-objective Optimization Using Opposition-based Learning and Simulated Annealing[J]. International Journal on Artificial Intelligence Tools, 2023, 32(4): 2350037.
- [31] 夏学文, 刘经南, 高柯夫, 等. 具备反向学习和局部学习能力的粒子群算法[J]. 计算机学报, 2015, 38(7): 1397-1407.
- Xia Xuewen, Liu Jingnan, Gao Kefu, et al. Particle Swarm Optimization Algorithm with Reverse-learning and Local-learning Behavior[J]. Chinese Journal of Computers, 2015, 38(7): 1397-1407.
- [32] Ren Teng, Xu Hongbo, Jin Kangning, et al. Optimisation of Takeaway Delivery Routes Considering the Mutual Satisfactions of Merchants and Customers[J]. Computers & Industrial Engineering, 2021, 162: 107728.