

# 控制与决策

Control and Decision

基于D-NSGA-GKM算法的多阶段武器协同火力分配方法

于博文, 吕明

引用本文:

于博文, 吕明. 基于D-NSGA-GKM算法的多阶段武器协同火力分配方法[J]. *控制与决策*, 2022, 37(3): 605-615.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.1236>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### [一种求解约束多目标问题的协作进化算法](#)

A collaborative evolutionary algorithm for solving constrained multi-objective problems

*控制与决策*. 2021, 36(11): 2656-2664 <https://doi.org/10.13195/j.kzyjc.2020.0791>

### [带不相关并行机和有限缓冲MHFS调度的混合启发式算法](#)

Hybrid heuristic algorithm for multi-stage hybrid flow shop scheduling with unrelated parallel machines and finite buffers

*控制与决策*. 2021, 36(3): 565-576 <https://doi.org/10.13195/j.kzyjc.2019.0835>

### [基于改进多目标优化算法的分布式数据中心负载调度](#)

Multi-objective optimization of energy and performance management in distributed data centers

*控制与决策*. 2021, 36(1): 159-165 <https://doi.org/10.13195/j.kzyjc.2019.0702>

### [基于复杂昂贵仿真的体系效能多目标优化](#)

Complex and expensive simulation based multi-objective optimization to system-of-system effectiveness

*控制与决策*. 2021, 36(3): 589-598 <https://doi.org/10.13195/j.kzyjc.2019.0844>

### [基于知识粒度特征的多目标粗糙集属性约简算法](#)

Multi objective rough set attribute reduction algorithm based on characteristics of knowledge granularity

*控制与决策*. 2021, 36(1): 196-205 <https://doi.org/10.13195/j.kzyjc.2019.0490>

# 基于D-NSGA-GKM算法的多阶段武器协同火力分配方法

于博文<sup>†</sup>, 吕明

(南京理工大学 自动化学院, 南京 210094)

**摘要:** 陆战场防御作战场景中的多阶段武器协同火力分配问题是典型的约束组合优化问题,其目的是生成合理有效的火力分配方案. 为了更接近实际作战情况,引入双方对抗过程,建立包含敌方作战单元战场剩余价值、作战资源消耗、作战单元战场价值损失的武器火力分配模型. 针对多阶段武器协同火力分配问题,在非支配排序遗传算法 III(non-dominated sorting genetic algorithm III, NSGA-III) 的基础上提出一种改进的智能算法(D-NSGA-GKM). 首先,引入基于优势度矩阵的非支配排序算法,减少排序过程中的冗余操作,以提高非支配排序效率;然后,在遗传操作阶段引入修复算子,对不可行解进行修复;最后,引入遗传  $K$  均值聚类算法对初始参考点进行自动聚类,用聚类质心替代原参考点,在环境选择阶段引入基于惩罚的边界相交距离替代垂直距离,以提高算法的收敛性. 实验结果表明, D-NSGA-GKM 算法在多阶段武器协同火力分配问题上具有较好的时间性能和收敛性能.

**关键词:** 多阶段火力分配; 多目标优化; 非支配排序遗传算法 III; 优势度矩阵; 遗传  $K$  均值

中图分类号: TP273 文献标志码: A

DOI: 10.13195/j.kzyjc.2020.1236

引用格式: 于博文, 吕明. 基于 D-NSGA-GKM 算法的多阶段武器协同火力分配方法[J]. 控制与决策, 2022, 37(3): 605-615.

## Optimization method for multi-stage collaborative weapon firepower distribution based on D-NSGA-GKM algorithm

YU Bo-wen<sup>†</sup>, LV Ming

(College of Automation, Nanjing University of Science and Technology, Nanjing 210094, China)

**Abstract:** The problem of multi-stage weapon collaborative firepower distribution in land battlefield defense is a typical constrained combination optimization problem, which aims to generate a reasonable and effective firepower distribution scheme. In order to get closer to the actual operational, the confrontation game process of both sides has been introduced and a weapon firepower distribution model including the residual value of enemy combat units, combat resource consumption, and battlefield value loss of combat units is established. An improved intelligent algorithm(D-NSGA-GKM) is proposed based on the non-dominated sorting genetic algorithm III(NSGA-III) for multi-stage collaborative weapon firepower distribution. Firstly, a non-dominated sorting algorithm based on the dominance degree matrix is introduced to reduce redundant operations in the sorting process to improve the efficiency of non-dominated sorting. Then, the repair operator is introduced in the genetic operation stage to repair the infeasible solution. Finally, the genetic  $K$ -mean clustering algorithm is introduced to cluster the initial reference points automatically, the centroid of the cluster is used to replace the original reference points, and the penalization-based boundary intersection distance is introduced in the environmental selection stage to replace the vertical distance, to improve the convergence of the algorithm. The experimental results show that the D-NSGA-GKM algorithm has excellent time performance and convergence performance on the problem of multi-stage weapon cooperative fire distribution.

**Keywords:** multi-stage weapon target assignment; multi-objective optimization; non-dominated sorting genetic algorithm III; dominance degree matrix; genetic  $K$ -means

## 0 引言

协同作战是现代战争的重要组成部分,多类型武器协同火力分配是协同作战的关键技术之一,其主

要目的是在满足约束条件的情况下,根据作战任务合理地对战资源进行目标分配,从而提高整体作战效能<sup>[1]</sup>. 在陆战场防御作战中,近距离突击武器与远距

收稿日期: 2020-09-06; 录用日期: 2021-01-08.

基金项目: 江苏省自然科学基金项目(BK20180467).

责任编辑: 张文安.

<sup>†</sup>通讯作者. E-mail: yubowen\_0403@njst.edu.cn.

离支援武器协同打击拦截是最为主要的作战方式, 突击武器以单点杀伤、近距离作战为主, 支援武器以远距离火力压制为主, 在实施打击、拦截目标的过程中, 如何充分发挥武器资源的作战优势和降低武器资源作战损耗, 是能否完成打击、拦截任务的主要影响因素之一<sup>[2]</sup>. 随着先进武器和作战理论的发展, 作战指挥人员难以在复杂的作战环境下及时有效地做出作战决策, 基于此, 本文研究武器协同火力分配优化问题.

武器火力分配(WTA)问题是典型的约束组合优化问题, 它可以分为静态火力分配(SWTA)和动态火力分配(DWTA)<sup>[3-4]</sup>两类. SWTA在齐射的方式下通过调整作战单元的打击目标以最大限度地提高整体作战效能, DWTA在SWTA的基础上对多个阶段的作战资源分配进行综合调整以获得最大的作战效能. 以往对WTA的研究大多集中在SWTA上, 然而, Hosein等<sup>[3]</sup>在1990年提出DWTA这一概念之后受到越来越多的学者关注. 目前, 有很多关于DWTA多目标优化的研究. 文献[5]建立了包含多种约束的通用型DWTA模型. 文献[6]根据对战双方的作战态势, 提出了一种DWTA对抗博弈模型. 文献[7-8]建立了多阶段WTA(MWTA)问题的确定性和不确定性多目标优化模型, 通过增加自适应机制改进NSGA-II和MOEA/D两种多目标优化算法, 用来求解DWTA模型. 文献[9]使用NSGA-II求解基于空地防御的三目标DWTA优化问题. 文献[10]提出了一种改进的MOPSO算法解决DWTA问题. 比较上述研究中所提出的MOPSO, NSGA-II和MOEA/D解决DWTA问题的性能, 实验数据表明, NSGA-II在求解过程中表现最佳.

Jain等<sup>[11]</sup>在2014年提出了NSGA-III算法, 其在处理动态多目标优化问题时相对于NSGA-II具有更好的性能<sup>[12]</sup>. 近年来, 关于NSGA-III算法的研究取得了许多成果. 文献[13-14]通过引入基于惩罚的边界相交聚合函数提出了 $\theta$ -NSGA-III算法, 保证了多目标优化过程的多样性, 同时提高了算法的收敛性. 文献[15]提出了NSGA-III-OSD算法, 通过K-means聚类算法将目标空间分解为若干子空间, 在环境选择阶段引入基于惩罚的边界相交距离添加收敛信息, 增强了NSGA-III算法的收敛性.

由于战场态势具有高动态性, 武器火力分配问题对优化算法的时间性能和收敛性能具有较高的要求, 在上述研究的基础上, 本文基于陆战场防御作战场景制定一个介于SWTA与DWTA之间的多阶段武器

协同火力分配问题(MWTA), 并提出一种高效的D-NSGA-GKM多目标优化算法以解决MWTA问题. 首先, 考虑到实际对战是一个对抗博弈过程<sup>[16]</sup>, 以最小敌方作战单元战场剩余价值、最小作战资源消耗、最小作战单元战场价值损失为优化目标构建多阶段武器协同火力分配模型; 然后, 针对多阶段武器协同火力分配优化问题, 基于NSGA-III算法提出D-NSGA-GKM多目标优化算法; 最后, 通过实验结果表明, D-NSGA-GKM算法具有更好的时效性、收敛性, 在求解多阶段武器协同火力分配问题时更加有效.

## 1 多阶段武器协同火力分配模型

本文所描述火力分配问题如下: 我方拥有近距离突击武器 $W = (w_1, w_2, \dots, w_m)$ 和远距离支援武器 $Q = (q_1, q_2, \dots, q_n)$ 用来防御某重要资源. 敌方作战单位 $E = (e_1, e_2, \dots, e_k)$ 准备对我方重要资源进行攻击, 在敌方进攻前我方对其进行打击拦截, 将打击拦截作战分为 $S$ 个作战阶段. 图1为突击武器和支援武器执行拦截作战方案的过程, 在 $s$ 阶段与 $s+1$ 阶段之间对敌方作战单元进行打击拦截, 完成打击拦截任务后, 我方指挥人员根据当前作战态势信息进行下一阶段的武器火力分配方案.

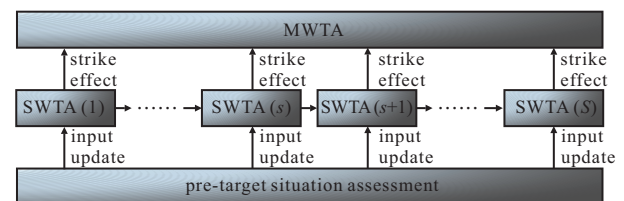


图1 多阶段武器协同火力分配过程

### 1.1 确定性决策模型

定义近距离突击武器和远距离支援武器的打击变量 $X = [x_{sik}]_{S \times W \times E}$ 和 $Y = [y_{sjk}]_{S \times Q \times E}$ , 其中 $x_{sik} = 1$ 表示第 $i$ 个突击武器在 $s$ 阶段分配给了第 $k$ 个敌方作战单元, 否则 $x_{sik} = 0$ ;  $y_{sjk} = 1$ 表示第 $j$ 个支援武器在 $s$ 阶段分配给了第 $k$ 个敌方作战单元, 否则 $y_{sjk} = 0$ . 定义突击武器、支援武器和敌方目标直接打击命中概率为 $f(l_s)$ , 有

$$f(l_s) = r_1 l_s^k + r_2 l_s^{k-1} + \dots + r_k l_s + r_{k+1}. \quad (1)$$

其中: $l_s$ 为第 $s$ 阶段各突击武器、支援武器与敌方作战单元之间的距离(本文所有位置信息仅考虑二维平面下的情况),  $r_m$ 为 $f(l_s)$ 的各项系数.  $p_{ik}(l_{sik}) = f(l_{sik})u_i$ 和 $q_{jk}(l_{sjk}) = f(l_{sjk})u_j$ 分别代表第 $i$ 个突击武器和第 $j$ 个支援武器在 $s$ 阶段对第 $k$ 个敌方作战单元的毁伤概率, 其中 $u_i$ 和 $u_j$ 分别为突击武器和支援武器的毁伤效能系数. 在 $s$ 阶段, 近距离突击武器 $W$

和远距离支援武器 $Q$ 对第 $k$ 个敌方作战单元的毁伤概率分别为 $P_{ks}$ 和 $Q_{ks}$ ,如下所示:

$$P_{ks} = 1 - \prod_{i=1}^W (1 - p_{ik}(l_{sik}))^{x_{sik}}, \quad (2)$$

$$Q_{ks} = 1 - \prod_{j=1}^Q (1 - q_{jk}(l_{sjk}))^{y_{sjk}}. \quad (3)$$

为了提高我方整体作战能力,突击武器和支援武器协同打击敌方作战单元,在 $s$ 阶段第 $k$ 个敌方作战单元的毁伤概率为 $P_{ks} \times Q_{ks}$ . 敌方作战单元的战场价值为 $VE = (ve_1, \dots, ve_k)$ ,由于敌方战场价值总值固定,可通过火力打击降低其战场价值,我方毁伤效能越高敌方战场价值越低,通过剩余战场价值衡量我方火力分配的火力打击毁伤能力. 因此,设置最小敌方作战单元战场剩余价值为优化目标,整个防御作战阶段敌方作战单元战场剩余价值为

$$F_1 = \sum_{k=1}^E ve_k \left( \prod_{s=1}^S (1 - P_{ks} Q_{ks}) \right). \quad (4)$$

假设第 $i$ 个突击武器和第 $j$ 个支援武器在一个作战阶段中的消耗系数为 $cw_i$ 和 $cq_j$ ,由于战斗物资受限,期望在战时以最小消耗物资获得最大的作战效能,因此设置最小作战资源消耗为优化目标,整个防御作战阶段作战资源消耗的总量为

$$F_2 = \sum_{s=1}^S \sum_{k=1}^E \sum_{i=1}^W cw_i x_{sik} + \sum_{s=1}^S \sum_{k=1}^E \sum_{j=1}^Q cq_j y_{sjk}. \quad (5)$$

在实际作战过程中,敌、我双方都在尽力保全自己,同时摧毁对方. 考虑到敌方作战单位的反击,我方作战单元可能会受到攻击,火力分配方案生成后,若我方战斗损失惨重同样也是不可取的,因此设置最小作战单元战场价值损失为优化目标. 假设敌方作战单元具有同时攻击多个目标的能力,整个防御作战阶段我方作战单元战场价值损失为

$$F_3 = \sum_{s=1}^S \sum_{i=1}^W \sum_{k=1}^T vw_i x_{sik} u_k f(l_{sik}) + \sum_{s=1}^S \sum_{j=1}^Q \sum_{k=1}^T vq_j y_{sjk} u_k f(l_{sjk}). \quad (6)$$

其中: $vw_i$ 为第 $i$ 个突击武器的战场价值, $vq_j$ 为第 $j$ 个支援武器的战场价值, $u_k$ 为第 $k$ 个敌方作战单位的反击效能.

这3个优化目标在某种意义上是相互冲突的,如果分配较多的战斗资源给敌方目标,则预期的伤害效能会更高,但这会导致更高的弹药浪费;如果分配较少的作战单元给敌方目标,则预期的弹药消耗会减少,但会导致我方战斗效能降低;若火力分配结果具

有较好的伤害效能和消耗成本,但是我方战斗损失惨重,则同样也是不可取的. 因此,目标是在毁伤效能、消耗成本与战斗损失之间寻找一种平衡. 以敌方作战单元战场剩余价值最小、作战资源消耗最小、作战单元战场价值损失最小为多阶段武器协同火力打击分配模型的优化目标,如下式所示:

$$\min F_1, \min F_2, \min F_3. \quad (7)$$

$$\text{s.t. } x_{sik} \leq f_{sik}, \forall s \in R_S, i \in R_W, k \in R_E; \quad (8)$$

$$y_{sjk} \leq f_{sjk}, \forall s \in R_S, j \in R_Q, k \in R_E; \quad (9)$$

$$\sum_{i=1}^W x_{sik} \leq wn_k, \forall s \in R_S, k \in R_E; \quad (10)$$

$$\sum_{j=1}^Q y_{sjk} \leq qm_k, \forall s \in R_S, k \in R_E; \quad (11)$$

$$\sum_{s=1}^S \sum_{k=1}^E cw_{ik} x_{sik} \leq F_i, \forall i \in R_W, s \in R_S; \quad (12)$$

$$\sum_{s=1}^S \sum_{k=1}^E cq_{jk} y_{sjk} \leq F_j, \forall j \in R_Q, s \in R_S; \quad (13)$$

$$R_W = \{1, 2, \dots, m\}, R_Q = \{1, 2, \dots, n\},$$

$$R_E = \{1, 2, \dots, k\}, R_S = \{1, 2, \dots, s\}. \quad (14)$$

约束条件(8)和(9)表示突击武器 $w_i$ 和支援武器 $q_j$ 火力打击分配的可行性. 如果在 $s$ 阶段可以将突击武器 $w_i$ 分配给敌方作战单元 $e_k$ ,则 $f_{sik} = 1$ ,否则 $f_{sik} = 0$ ;如果在 $s$ 阶段可以将支援武器 $q_j$ 分配给敌方作战单元 $e_k$ ,则 $f_{sjk} = 1$ ,否则 $f_{sjk} = 0$ . 约束条件(10)和(11)用来限制在 $s$ 阶段突击武器 $W$ 和支援武器 $Q$ 可分配给敌方作战单元 $e_k$ 的最大数目,根据实际情况,设置 $wn_k = 2, qm_k = 1$ . 约束条件(12)和(13)中 $F_i$ 和 $F_j$ 表示所持有弹药的总量,保证各参战单元在作战阶段拥有充足的弹药.

## 1.2 不确定性决策模型

为了保证突击武器与支援武器的协同作战效能,协同任务分配模型需要有精确的任务参数. 然而,现实世界中受到各种随机和非随机因素的影响,建模和任务参数的获取无法避免地存在误差,确定性假设模型不再成立. 因此,针对这些不可避免的影响因素,提出一种不确定性多阶段武器协同火力分配模型. 受自然因素、作战单元由于长时间机动造成的作战效能退化、不同敌方目标具有不同的作战性能等不确定因素的影响,火力执行单元作战效能存在理论上的偏差. 这里对突击武器、支援武器进行不确定性假设,有

$$p_{ik}^R(l_{sik}) \in [(1 - \vartheta_{iks})p_{ik}(l_{sik}), p_{ik}(l_{sik})], \quad (15)$$

$$q_{jk}^R(l_{sjk}) \in [(1 - \vartheta_{jks})q_{jk}(l_{sjk}), q_{jk}(l_{sjk})], \quad (16)$$

其中  $\vartheta_{iks}$ 、 $\vartheta_{jks}$  分别为突击武器、支援武器在  $s$  阶段的不确定度, 可通过历史数据确定. 但是, 该不确定模型是基于场景的, 为了确定一个具有较为优秀性能的不确定性效能模型需要进行大量的场景模拟, 无形中增加了计算量. 根据文献[17]的结论, 提出一种更为灵活的模型, 改进后得到新的敌方作战单元作战剩余价值和作战单元战场价值损失目标函数, 分别为

$$P_{ks}^R = 1 - \prod_{i=1}^W (1 - (1 - \delta\vartheta_{iks})p_{ik}(l_{sik}))^{x_{sik}}, \quad (17)$$

$$Q_{ks}^R = 1 - \prod_{j=1}^Q (1 - (1 - \delta\vartheta_{jks})q_{jk}(l_{sjk}))^{y_{sjk}}, \quad (18)$$

$$F_1^R = \sum_{k=1}^E \text{ve}_k \left( \prod_{s=1}^S (1 - P_{ks}^R Q_{ks}^R) \right), \quad (19)$$

$$F_3^R = \sum_{s=1}^S \sum_{i=1}^W \sum_{k=1}^T v w_i x_{sik} (1 - \delta\vartheta_{ks}) u_k f(l_{sik}) + \sum_{s=1}^S \sum_{j=1}^Q \sum_{k=1}^T v q_j y_{sjk} (1 - \delta\vartheta_{ks}) u_k f(l_{sjk}). \quad (20)$$

其中:  $\vartheta_{ks}$  为敌方作战单元的不确定度,  $\delta \in [0, 1]$  为不确定模型的调节因子. 由于作战单元的不确定度是通过历史数据预先设置的, 指战员可根据当前战场作战态势对作战单元的作战能力进行状态预估, 通过不确定调节因子调整作战单元的作战效能. 为了更贴近实际情况, 采用不确定性决策模型作为火力分配决

策模型, 其优化目标函数为

$$\begin{cases} \min F_1^R, \\ \min F_2, \\ \min F_3^R. \end{cases} \quad (21)$$

## 2 多阶段武器协同打击火力分配求解算法

根据上节所提出的优化目标可以看出, 多阶段防御作战场景下突击武器、支援武器协同打击火力分配是一个多目标优化问题, 因此, 本文采用 NSGA-III 算法对多阶段武器协同火力分配进行优化. 根据过往的研究, NSGA-III 算法具有较好的多样性, 但是其非支配排序过程具有较高的计算复杂度<sup>[18]</sup> 和较弱的收敛性<sup>[15]</sup>. 根据上述描述对 NSGA-III 算法进行改进, 用来求解多阶段武器协同打击火力分配问题.

### 2.1 染色体编码

定义染色体  $\text{ind}_i$  编码如下所示:

$$\text{ind}_i = [x_{1,1}, x_{1,2}, \dots, x_{1,m}, y_{1,1}, y_{1,2}, \dots, y_{1,n}, \dots, x_{s,1}, x_{s,2}, \dots, x_{s,m}, y_{s,1}, y_{s,2}, \dots, y_{s,n}]. \quad (22)$$

图2为染色体编码示意图,  $\text{ind}_i$  的编码长度为  $S(M + N)$ , 每个染色体由  $S$  个阶段组成, 各阶段包含两部分: 第1部分为突击武器的打击变量,  $x_{s,m}$  表示在  $s$  阶段第  $m$  个突击武器要攻击的敌方目标编号; 第2部分为支援武器的打击变量,  $y_{s,n}$  表示在  $s$  阶段第  $n$  个支援武器要攻击的敌方目标编号.

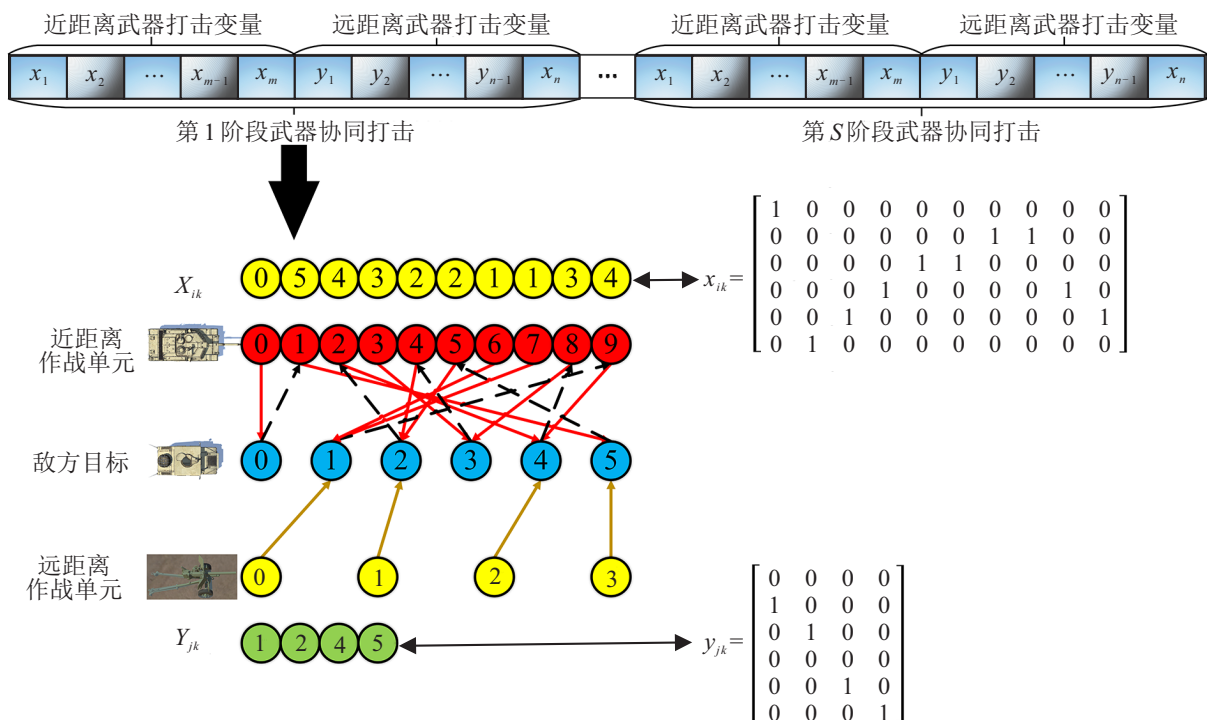


图2 染色体编码示意图

## 2.2 基于优势度矩阵的非支配排序

NSGA-III算法具有较高的计算复杂度,主要体现在非支配排序的过程中存在较多冗余比较,向量的排序比较相对于标量的排序比较更难处理.应用较为有效的非支配排序方法是提高算法效率的有效途径.文献[19]提出一种新的数据结构优势度矩阵,并利用优势度矩阵设计了一种基于优势度矩阵的非支配排序算法(DDA-NS).NSGA-III中的快速非支配排序算法<sup>[11,20]</sup>在进行目标向量非支配排序的计算复杂度为 $O(MN^2)$ ,而DDA-NS的计算复杂度仅为 $O(MN \log N)$ ,可以看出,DDA-NS可以有效提高非支配排序的效率.基于此,本文利用基于优势度矩阵的非支配排序算法替换NSGA-III中的快速非支配排序算法.

定义两个目标向量

$$S_1 = [S_{11}, S_{12}, \dots, S_{1m}], \quad (23)$$

$$S_2 = [S_{21}, S_{22}, \dots, S_{2m}]. \quad (24)$$

若 $S_{1i} \leq S_{2i} (\forall i \in \{1, 2, \dots, m\})$ ,则称 $S_1$ 支配 $S_2$ .若 $S_1$ 没有被其他向量所支配,则称 $S_1$ 为Pareto解.

1) 优势度.

通过研究 $S_1$ 、 $S_2$ 内包含的元素关系确定支配关系,定义 $S_1$ 相对于 $S_2$ 的优势度为满足 $S_{1i} \leq S_{2i}$ 元素对 $(S_{1i}, S_{2i})$ 的数量,定义优势度为

$$d(S_1, S_2) = |\{i | i \in \{1, 2, \dots, m\}, S_{1i} \leq S_{2i}\}|. \quad (25)$$

可以看出 $0 \leq d(S_1, S_2) \leq m$ .当 $S_1$ 可以支配 $S_2$ 时, $d(S_1, S_2) = m$ .当 $d(S_1, S_2) = d(S_2, S_1) = m$ 时, $S_1 = S_2$ .

定义 $R = \{A^1, A^2, \dots, A^N\}$ 为一个包含 $N$ 个目标向量的集合,其中 $A^i = [a_{1i}, a_{2i}, \dots, a_{mi}]^T \in R^m, i \in \{1, 2, \dots, N\}$ .由此可以定义基于集合 $R$ 上的优势度矩阵 $D$ ,集合 $R$ 中的支配关系由 $d_{ij}$ 表示,有

$$D = (d_{ij})_{N \times N}, \quad (26)$$

$$d_{ij} = d(A^i, A^j), i, j \in \{1, 2, \dots, N\}. \quad (27)$$

2) 计算优势度矩阵.

使用快速排序算法对每个目标函数集合 $R$ 中的目标向量进行排序,排序完成后利用比较矩阵构造优势度矩阵.构造行向量 $W = (w_1, w_2, \dots, w_N) \in R^N$ ,定义比较矩阵 $C_W = (cw_{ij})_{N \times N}$ ,有

$$cw_{ij} = \begin{cases} 1, & w_i \leq w_j; \\ 0, & \text{otherwise.} \end{cases} \quad (28)$$

显然,一组向量 $R$ 中包含 $m$ 个目标向量,通过对所有目标向量 $W_i$ 的比较矩阵求和,可以得到优势度矩

阵, $D = D + C_{W_i}$ .

3) 基于优势度矩阵的非支配排序.

以生成第 $t$ 代种群为例, $R_t = P_t \cup Q_t$ ,其染色体长度为 $2N$ .根据上述内容可知,优势度矩阵的大小为 $D_{2N \times 2N}$ .根据非支配解的概念,具有相同描述向量的解集会被归入相同的非支配层级.为了消除这些相同元素,将优势度矩阵 $D$ 中的相应元素设置为0.定义一个大小为 $2N$ 的行向量 $\max(D)$ ,行向量用来记录优势度矩阵 $D$ 中每一列的最大值.根据优势度定义可知,若 $\max(D)$ 中某个元素小于 $m$ ,则 $R_t$ 中对应的解集为非支配解集,可以构成第1个非支配层级 $F_1$ ,并将其添加到 $S_t$ 中.从优势度矩阵 $D$ 中删除与非支配层级 $F_1$ 相关的所有行、列向量,将得到的优势度矩阵命名为 $D_1$ .重复上述过程获得 $S_t$ 和非支配层级 $F_2$ 、 $F_3$ 等,直到 $|S_t|$ 首次大于等于 $N$ ,完成对 $R_t$ 中所需要解的非支配排序.

## 2.3 基于Genetic K-means的参考点生成

本文基于文献[15]的思想对解集 $S_t/F_j$ 进行聚类分组,在该阶段引入genetic K-means算法,实现参考点自动聚类分组,消除人工选取的 $k$ 值对聚类分组的影响,以聚类分组质心为参考点选取进入下一代种群的染色体,参考点生成的主要步骤如下.

step 1: 染色体初始化,随机选取 $k_{ini}(k_{ini} = 3)$ 个火力分配方案 $Z_j(j \in \{1, \dots, k_{ini}\})$ ,计算其他火力分配方案与 $Z_j$ 之间的距离,按照距离最小的原则进行聚类分组,生成 $k_{ini}$ 个分组 $C_j(j \in \{1, 2, \dots, k_{ini}\})$ .计算聚类分组中心 $Z_j^*$ ,有

$$Z_j^* = \frac{1}{|C_j|} \sum_{Z_{rp_{ij}} \in C_j} Z_{rp_{ij}}, \quad (29)$$

其中 $Z_{rp_{ij}}$ 为第 $j$ 个分组里第 $i$ 个火力分配方案.若 $Z_j^*$ 满足 $\|Z_{rp_{ij}} - Z_j^*\|^2 < \|Z_{rp_{ij}} - Z_j\|^2$ ,则将 $Z_j^*$ 作为染色体上的基因编码(实数编码),染色体长度为 $k_{ini}$ ,如图3所示.



图3 GKM算法中的染色体结构

step 2: 适应度函数计算.定义组内距离IND,有

$$IND = \sum_{j=1}^k \sum_{Z_{rp_{ij}} \in C_j} \|Z_{rp_{ij}} - Z_j^*\|^2. \quad (30)$$

其中: $k$ 为聚类分组数目, $C_j$ 为第 $j$ 个分组, $Z_j^*$ 为第 $j$ 个聚类分组中的聚类质心.定义组间距离ITD为

$$ITD = \max_{i,j=1,2,\dots,k} \|Z_i - Z_j\|^2, \quad (31)$$

其中 $Z_i$ 、 $Z_j$ 为第 $i$ 、 $j$ 个聚类分组的质心.根据式(30)

和(31)定义适应度函数为

$$\text{fitness}(k) = \frac{1}{k(k-1)} \times \frac{\text{ITD}}{\text{IND}}. \quad (32)$$

由式(32)可以看出,组内距离越小,组间距离越大,聚类分组的效果越好.

**step3:** 遗传进化. 在以锦标赛选择法和单点交叉为基础的进化过程中,为了获得最好的聚类分组效果,在初始种群中选取适应度函数最大的染色体作为学习范例,其他染色体以范例为标准进行学习. 若在变异过程中染色体的长度具有上升趋势,则在下一次变异操作时通过增加或删除染色体上的基因提高适应度函数的数值. 当需要增加染色体长度时,选取与当前染色体中心距离最远的点进行添加;当需要减少染色体长度时,则移除染色体中最接近当前染色体中心的点,从而完成自动聚类分组.

**step4:** 重复 step2 和 step3, 直到达到最大迭代次数,生成聚类分组的质心  $Z = (Z_1, Z_2, \dots, Z_k)$ .

## 2.4 修正算子

由于在遗传算法的交叉和变异操作过程中无法进行约束处理,生成的部分染色体可能无法满足火力分配模型的约束条件,因此,需要对不满足约束条件的染色体进行修复操作. 图4所示为染色体的修复过程,根据多阶段武器协同火力分配模型的约束条件对染色体进行修复操作. 通过修复操作可以有效保障遗传操作生成染色体的合法性.

以第  $t$  阶段需要修复的染色体突击武器部分为例(支援武器修复过程与此类似),设需要修复的染色体为  $\text{ind} = [c_1, c_2, \dots, c_M]$ ,主要修复步骤如下.

**step1:** 对要进行修复的染色体进行整数修复,设置染色体编码交叉、变异的范围  $(0, R_E)$ , 编码向下取整  $c_i = \text{math.floor}(c_i) (c_i \in \text{ind})$ .

**step2:** 计算  $W$  打击  $E$  所消耗的资源  $M_{W \times E}$ , 查找需要修复的  $W_R = \{w_r | F_r < m_{rk}, w_r \in W, m_{rk} \in M_{W \times E}\}$  和可供分配的  $E_A = \{e_a | F_r > m_{ra}, e_a \in E, m_{ra} \in M_{W \times E}\}$ . 随机从  $W_R$  中选择一个作战单元记为  $w_b$ , 计算  $w_b$  打击  $E_A$  中目标毁伤效能  $\text{ve}_{ra} = \text{ve}_a \times (1 - p_{ra})$ , 从中选取  $\text{ve}_{ra}$  最大的目标作为  $w_b$  的分配目标,  $\text{cw}_r = e_a$ .

**step3:** 根据  $t$  阶段的可行性变量, 查找需要修复的  $W_R = \{w_r | f_{trk} = 0, w_r \in W\}$  和可供分配的  $E_A = \{e_a | f_{tra} = 1, e_a \in E\}$ , 随机从  $W_R$  中选择一个作战单元记为  $w_b$ , 计算  $w_b$  打击  $E_A$  中目标毁伤效能  $\text{ve}_{ra} = \text{ve}_a \times (1 - p_{ra})$ , 从中选取  $\text{ve}_{ra}$  最大的目标作为  $w_b$  的分配目标,  $\text{cw}_r = e_a$ .

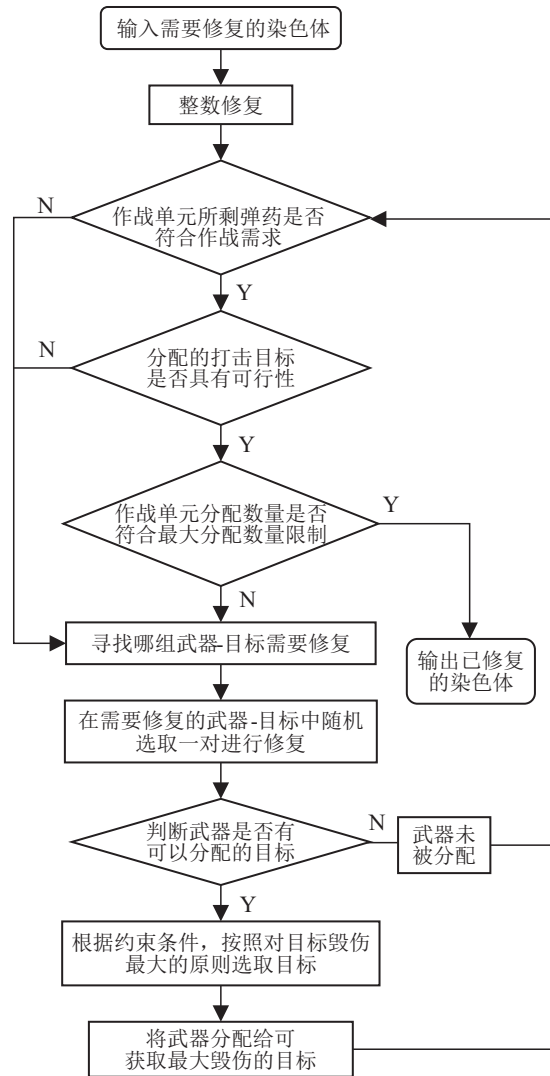


图4 染色体修复过程

**step4:** 根据约束条件(10)查找火力资源分配过多的目标  $E_A = \{e_a | \sum_i^W x_{tia} - \text{wn}_a > 0, e_a \in E\}$ , 计算与  $e_a$  相关联的武器资源  $W_R$  的毁伤效能, 将毁伤效能最小的  $|W_R| - \text{wn}_a$  个武器资源依次分配给  $E/E_A$  中毁伤效能最高的目标.

**step5:** 重复上述步骤直至所有染色体满足约束条件(8)、(10)和(12).

## 2.5 基于D-NSGA-GKM算法的多阶段武器协同火力分配求解步骤

针对NSGA-III收敛性较弱和非支配排序较为复杂的问题进行如下改进: 在非支配排序阶段采用基于优势度矩阵的排序算法, 以提高非支配排序的效率; 在环境选择阶段引入 genetic  $K$ -means 算法对优秀解集进行自动聚类, 以聚类分组的质心为选择参考点, 以基于惩罚的边界交叉距离数值作为评价解集优先级的标准, 选取进入下一代种群的染色体. 具体求解步骤如下.

step 1: 初始化染色体  $\text{ind}_i$  的编码, 根据约束条件 (8)~(13) 对打击变量进行初始化, 保证打击变量的可行性和火力分配的合理性.

step 2: 生成初始种群  $P_t$  (大小为  $L$ ), 根据染色体初始化各优化目标的数值. 通过模拟交叉算子、多项式变异算子和修复算子生成  $Q_t$ , 合并种群  $P_t$  和  $Q_t$  生成  $R_t$  (大小为  $2L$ ). 通过基于优势度矩阵的非支配排序方法将  $R_t$  分为多个非支配层级 ( $F_1, F_2, \dots$ ), 按照非支配层级关系依次选取优秀的染色体进入  $S_t$ , 直至  $|S_t|$  的大小首次大于等于  $L$ .

1) 当  $|S_t| = L$  时, 直接输出  $S_t$  作为第  $t + 1$  代种群.

2) 当  $|S_t| > L$  时, 此时非支配层级记为  $F_J$ , 将  $S_t/F_J$  中的解集计入到  $P_{t+1}$  中, 舍去  $F_J$  及其后的所有解集, 从  $F_J$  中选取  $L - |P_{t+1}|$  个染色体, 通过后续步骤加入到  $P_{t+1}$  中.

step 3: 按照文献 [11] 的归一化方法对目标空间中的解集进行归一化操作.

step 4: 通过 genetic  $K$ -means 算法对解集  $S_t/F_J$  进行聚类分组, 聚类分组生成的质心作为环境选择阶段的参考点  $Z^* = (Z_1^*, Z_2^*, \dots, Z_k^*)$ .

step 5: 在小生境选择操作<sup>[21]</sup> 的基础上, 以基于惩罚的边界交叉距离数值最小为原则从  $F_J$  中选择  $L - |P_{t+1}|$  个优秀染色体进入下一代种群  $P_{t+1}$  中, 基

于惩罚的边界交叉距离计算公式如下所示:

$$d(\text{ind}_i) = d_1(\text{ind}_i) + \theta d_2(\text{ind}_i), \quad (33)$$

$$d_1(\text{ind}_i) = \frac{\|(f(\text{ind}_i) - Z^*)^T \omega^*\|}{\|\omega^*\|}, \quad (34)$$

$$d_2(\text{ind}_i) = \|f(\text{ind}_i) - (Z^* - d_1(\text{ind}_i)\omega^*)\|. \quad (35)$$

其中:  $\text{ind}_i$  为多阶段武器协同火力分配方案,  $\omega^*$  为  $f(\text{ind}_i)$  到参考点  $Z^*$  的方向向量,  $d_1(\text{ind}_i)$  和  $d_2(\text{ind}_i)$  分别为  $f(\text{ind}_i)$  到  $\omega^*$  的投影距离和垂直距离,  $\theta$  为惩罚系数.

step 6:  $t = t + 1$ , 循环执行 step 2~step 5, 直至达到算法终止条件 (最大迭代次数).

### 2.6 D-NSGA-GKM算法复杂度分析

以 D-NSGA-GKM 算法生成一代种群的计算过程为例对其计算复杂度进行分析. D-NSGA-GKM 算法的计算复杂度主要集中在遗传操作、环境选择这两个部分. 假设种群规模为  $2N$ , 具有  $M$  维目标向量. 遗传操作中模拟二进制交叉和多项式变异的计算复杂度为  $O(VN)$  ( $V$  为决策变量的数量), 修复算子的计算复杂度为  $O(VN \log N)$ . 环境选择主要包括以下步骤: 种群非支配排序、生成参考点、分配算子、基于惩罚的边界相交距离的计算、小生境选择. 非支配排序的计算复杂度为  $O(MN \log N)$ . 基于 genetic  $K$ -means 算法生成参考点的计算复杂度为  $O(tKN)$ . 其中:  $t$  为迭代次数,  $K$  为聚类分组的数目. 分配算子的计算复杂度为  $O(MN^2)$ . 计算基于惩罚的边界相交距离的计算复杂度为  $O(MN)$ . 假设  $L = |F_J|$ , 则小生境选择操作的计算复杂度为  $O(L^2)$  或者  $O(LN)$ . 在假设中,  $N > M, N > tK$ , 因此, 环境选择阶段的计算复杂度为  $O(MN^2)$ . 综上所述, D-NSGA-GKM 算法的计算复杂度为  $O(MN^2)$ , 整个优化求解过程可以满足实际作战需求.

## 3 实验分析

### 3.1 实验环境描述

本文算法采用 python 语言编程, 环境为 Anaconda3 (Python3.6), 编译器为 PyCharm, 使用一台 CPU 主频为 2.3 GHz、内存为 16 GB, 运行 Windows 10 操作系统的笔记本作为运行环境. 由于真实战场数据的特殊性, 根据作战想定设计测试算例生成程序以生成不同的测试算例, 用来比较不同算法的性能. 假设有 10 个近距离突击武器, 6 个敌方目标, 4 个远距离支援武器, 将整个防御作战分为 5 个阶段. 实验中近距离突击武器  $W$ 、远距离支援武器  $Q$ 、敌方目标  $T$  的属性参数配置如表 1 所示.

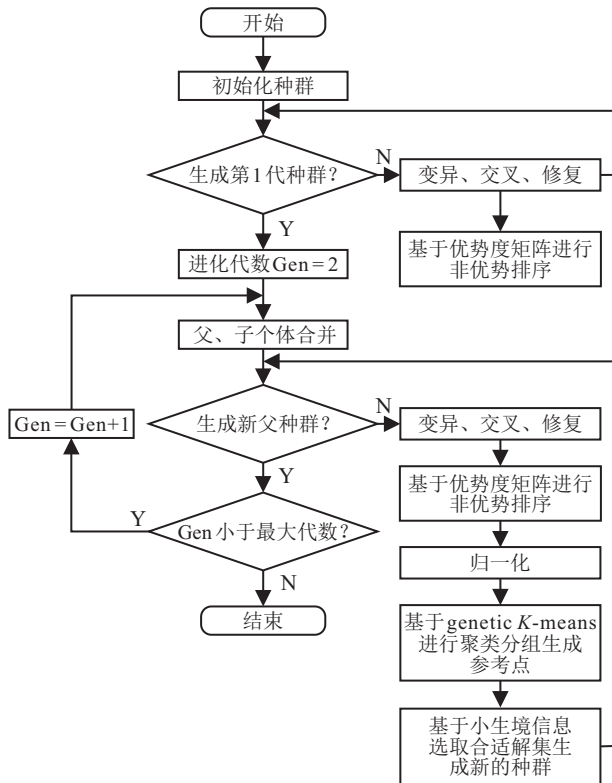


图 5 D-NSGA-GKM 流程

表1 算例生成参数设置

变量	值	
$POS_Q$	$(\text{rand}(100, 1\ 600), \text{rand}(100, 2\ 500))$	
$vw_i, vq_j$	$1 + \text{rand}(1, 9)$	
$vt_k$	$1 + \text{rand}(1, 9)$	
ini stage	$cw_{ik}$	$\text{rand}(5, 10)$
	$cq_{jk}$	$\text{rand}(10, 15)$
$u_i, u_j, u_k$	$\text{rand}(0.4, 0.8)$	
$F_i$	$\text{rand}(50, 80)$	
$F_i$	$\text{rand}(80, 100)$	
every stage	$POS_W$	$(\text{rand}(100, 1\ 600), \text{rand}(1\ 000, 1\ 600))$
	$POS_E$	$(\text{rand}(400, 1\ 300), \text{rand}(3\ 200, 4\ 000))$
	$f_{iks}$	$(\text{sgn}(\text{rand}(0, 1) - 0.5) + 1)/2$
	$f_{jks}$	$(\text{sgn}(\text{rand}(0, 1) - 0.5) + 1)/2$

突击武器、支援武器与敌方作战单元的打击命中概率系数分别为

$$\begin{aligned} r_{mW} &= [1.575 \times 10^{-11}, -1.408 \times 10^{-7}, \\ &\quad 1.195 \times 10^{-4}, 9.455 \times 10^{-1}], \\ r_{mQ} &= [1.555 \times 10^{-11}, -1.335 \times 10^{-7}, \\ &\quad 8.625 \times 10^{-5}, 9.645 \times 10^{-1}], \\ r_{mE} &= [1.565 \times 10^{-11}, -1.535 \times 10^{-7}, \\ &\quad 9.625 \times 10^{-5}, 9.645 \times 10^{-1}]. \end{aligned}$$

选取 NSGA-III、NSGA-III-OSD、ANSGA-II、AMOEAD 作为对比算法, D-NSGA-GKM 与对比算法的种群大小设置为 150, 最大迭代次数为 150, 其他实验参数按照文献 [8, 11, 15] 进行设置. 其中: D-NSGA-GKM 的初始  $k_{\text{ini}}$  值为 3, 惩罚系数  $\theta$  为 0.5, 不确定度  $\vartheta_{iks}$ 、 $\vartheta_{jks}$ 、 $\vartheta_{ks}$  均取 0.2. 经过多次实验, 在同等资源消耗的情况下, 调节因子越小, 获得的作战效能越大, 为了方便计算调节因子  $\delta$  取 0.2.

### 3.2 结果分析

独立进行 4 组实验, 每组实验进行 20 次, 实验结果如表 2 所示, 分别为 D-NSGA-GKM (算法 1)、NSGA-III-OSD (算法 2)、NSGA-III (算法 3)、ANSGA-II (算法 4)、AMOEAD (算法 5) 优化算法对协同火力分配模型求解的最优值 (敌方作战单元战场剩余价值  $J_1$ 、作战资源消耗  $J_2$ 、作战单元战场价值损失  $J_3$ ). 由表 2 可见, 4 组实验中, 第 2、第 3 组中 D-NSGA-GKM 算法与 NSGA-III-OSD 算法结果相同; 第 4 组中 D-NSGA-GKM 算法与 NSGA-III 算法结果接近, 其他 3 组实验中 D-NSGA-GKM 算法的结果相对于 NSGA-III 算法均具有一定优势.

表2 各算法优化结果对比

组号	目标	算法1	算法2	算法3	算法4	算法5
第 1 组	$J_1$	2.829 6	2.831 9	2.842 3	2.842 6	2.845 3
	$J_2$	501.000 0	502.000 0	505.000 0	508.000 0	510.000 0
	$J_3$	44.161 7	44.205 9	44.215 7	44.221 3	44.226 7
第 2 组	$J_1$	2.830 1	2.830 1	2.842 8	2.843 6	2.845 8
	$J_2$	501.000 0	501.000 0	506.000 0	509.000 0	511.000 0
	$J_3$	44.146 1	44.146 1	44.186 1	44.224 5	44.227 5
第 3 组	$J_1$	2.833 3	2.833 3	2.846 3	2.844 2	2.851 2
	$J_2$	500.000 0	500.000 0	505.000 0	508.000 0	510.000 0
	$J_3$	44.096 2	44.096 2	44.199 3	44.200 9	44.205 6
第 4 组	$J_1$	2.829 5	2.829 9	2.830 1	2.842 3	2.843 3
	$J_2$	501.000 0	501.000 0	502.000 0	509.000 0	511.000 0
	$J_3$	44.153 9	44.158 9	44.160 2	44.223 1	44.225 1

第 1 组实验各算法解集分布对比图如图 6 所示. 由图 6(b) 可见: 作战单元战场价值损失固定时, 作战资源消耗越多, 敌方作战单元战场剩余价值越小; 敌方作战单元战场剩余价值固定时, 作战资源消耗越多, 作战单元战场价值损失越大; 作战资源消耗固定时, 敌方作战单元战场剩余价值越小, 作战单元战场价值损失越小, 符合实际作战情况. 因此, 最终的协同火力分配方案可以按照实际战场作战态势在近似 Pareto 前沿中折中选择作战方案, 使得整体作战效能、资源消耗、作战损失三者之间得到平衡. 从这 4 组实验结果可以看出, D-NSGA-GKM 可以对多阶段武器协同火力分配进行有效优化, 具有可行性.

### 3.3 算法性能分析

下面通过 HV (hype volume) 超体积指标<sup>[22]</sup>、IGD (inverted generational distance) 逆世代距离指标<sup>[23]</sup>、SC (set coverage) 覆盖率指标<sup>[8]</sup>、消耗时间<sup>[19]</sup>这 4 个指标对算法的性能进行比较. 通过枚举法产生实验算例的部分真实 Pareto 前沿, 以计算后续指标.

HV 超体积指标可以用来综合评价算法的多样性和收敛性, HV 指标数值越大, 算法的综合性能越好, 根据文献 [22] 的结论, 通过大量实验选取 1 000 个参考点计算 HV 指标, 数值越大算法总体性能越好, 计算公式如下所示:

$$V_{\text{HV}}(A^*, B^*) = \bigcup_{f \in A^*} [f_1, B_1^*] \times \dots \times [f_D, B_D^*]. \quad (36)$$

通过 IGD 逆世代距离同样可以对算法的多样性、收敛性进行综合评价, 数值越小, 解集  $A^*$  中的元素越接近真实的 Pareto 前沿, 分布越均匀. 因此, 解集的 IGD 数值越小, 算法性能越高, 收敛性越好, 计算如下:

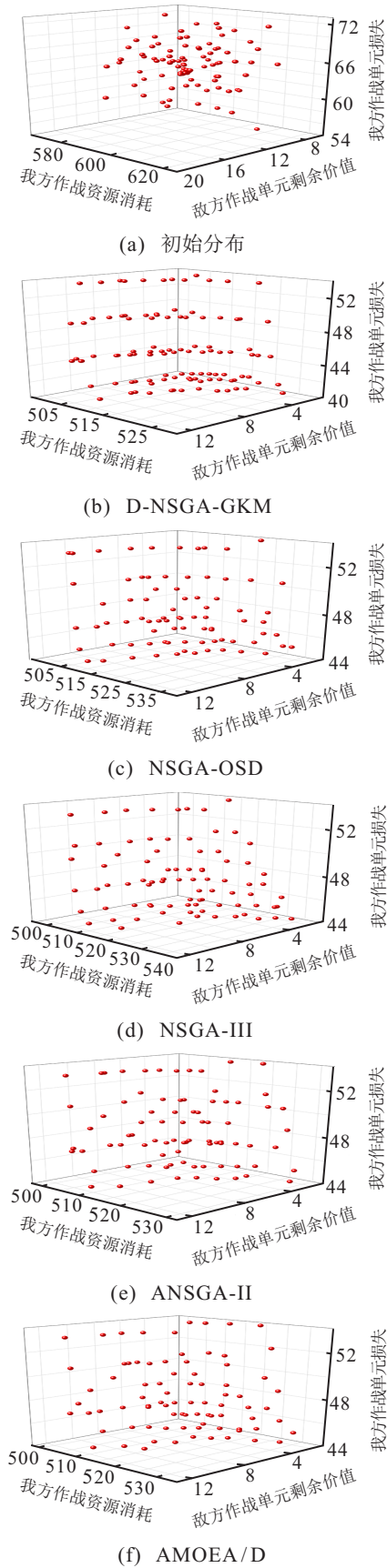


图6 在目标空间内各算法解集分布对比

$$IGD(A^*, P^*) = \frac{1}{|P^*|} \sum_{x \in P^*} \min_{y \in A^*} D^*(x, y). \quad (37)$$

其中: 解集  $A^*$  为 Pareto 前沿的近似集合,  $B^*$  为空间参考点(其可被  $A^*$  中的任意解支配),  $P^*$  集合为沿着真

实 Pareto 前沿均匀分布的点.

通过 SC 解集覆盖率指标比较本文算法与另外几种多目标优化算法产生的 Pareto 解集之间的支配关系, 计算如下:

$$C(A^*, B^*) = \frac{1}{|B^*|} |\{x \in B^* | \exists y \in A^* : y \prec x\}|. \quad (38)$$

其中:  $A^*$  和  $B^*$  为两个 Pareto 解集,  $C(A^*, B^*)$  为解集  $B^*$  中被解集  $A^*$  中的解所支配或与其相等的解在解集  $B^*$  中所占的比例. 如果  $C(A^*, B^*)$  的值大于  $C(B^*, A^*)$  的值, 则表明解集  $A^*$  比解集  $B^*$  优秀.

上述4组实验中, 各算法 HV 指标的统计结果如表3所示. 表3中: B 代表  $HV_{best}$ , M 代表  $HV_{mid}$ , W 代表  $HV_{worst}$ . IGD 指标综合取平均值, 如图7所示; SC 解集覆盖率指标如表4所示. 可以看出, D-NSGA-GKM 的 IGD 指标、HV 指标的最优值  $HV_{best}$ 、中值  $HV_{mid}$  和解集支配性均优于其他几种多目标优化算法, 表明 D-NSGA-GKM 算法具有较好的多样性和收敛性, 解集质量较高.

表3 各算法 HV 指标统计量

	算法1	算法2	算法3	算法4	算法5
第 B	4.322e-01	4.318e-01	3.922e-01	3.913e-01	3.783e-01
1 M	2.117e-01	2.106e-01	1.979e-01	1.952e-01	1.872e-01
组 W	1.285e-01	1.258e-01	1.092e-01	1.078e-01	9.981e-02
第 B	4.252e-01	4.212e-01	3.996e-01	3.982e-01	3.825e-01
2 M	2.183e-01	2.055e-01	1.956e-01	1.948e-01	1.731e-01
组 W	1.305e-01	1.298e-01	1.146e-01	1.106e-01	9.847e-02
第 B	4.436e-01	4.431e-01	4.011e-01	3.965e-01	3.824e-01
3 M	2.226e-01	2.098e-01	1.932e-01	1.912e-01	1.746e-01
组 W	1.279e-01	1.267e-01	9.962e-02	1.081e-01	1.111e-01
第 B	4.282e-01	4.322e-01	3.994e-01	3.943e-01	3.863e-01
4 M	2.196e-01	2.101e-01	1.946e-01	1.892e-01	1.762e-01
组 W	1.298e-01	1.311e-01	1.088e-01	1.092e-01	9.981e-02

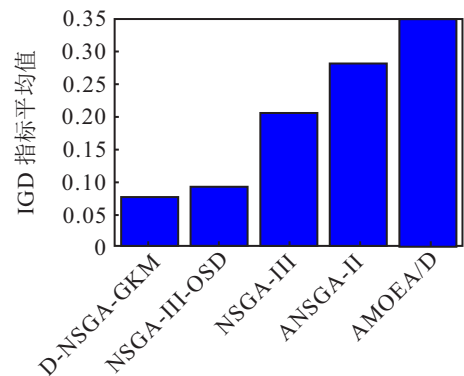


图7 各算法 IGD 指标的平均值(归一化结果)

表4 各算法解集覆盖率对比

解集相互覆盖度	第1组	第2组	第3组	第4组
$C$ (算法1,算法3)	1.000	0.991	0.995	0.998
$C$ (算法3,算法1)	0.955	0.949	0.959	0.962
$C$ (算法1,算法2)	1.000	0.987	1.000	0.982
$C$ (算法2,算法1)	0.992	1.000	0.996	1.000
$C$ (算法1,算法4)	1.000	0.997	0.998	1.000
$C$ (算法4,算法1)	0.937	0.926	0.914	0.913
$C$ (算法1,算法5)	1.000	1.000	1.000	1.000
$C$ (算法5,算法1)	0.891	0.883	0.856	0.877

考虑到实际战场中需要在短时间内做出打击决策,因此,算法的时间消耗也是评价其性能的重要指标之一.上述实验时间消耗的统计信息如图8和表5所示.其中D-NSGA-GKM平均消耗92.5253s、NSGA-III-OSD平均消耗103.2094s、NSGA-III平均消耗98.8493s、ANSGA-II平均消耗100.5834s、AMOEAD平均消耗106.1362s.由图8可见,D-NSGA-GKM耗时最少,AMOEAD耗时相对较多.

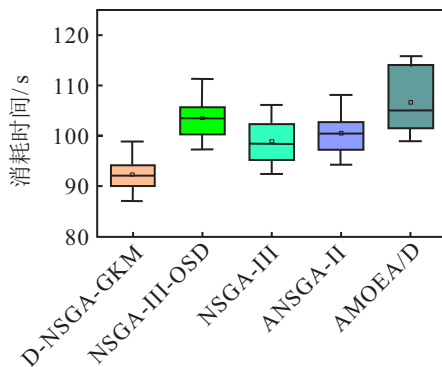


图8 各类算法消耗时间对比

表5 NSGA-III和D-NSGA-GKM的非支配排序和遗传操作消耗时间对比 单位:s

	NSGA-III	D-NSGA-GKM
总计	98.8493	92.5253
非支配排序(每次)	57.3326	49.8671
非支配排序(总计)	0.3822	0.3324
遗传选择(每次)	41.5167	42.6582
遗传选择(总计)	0.2768	0.2844

由表5可见,在相同初始参数设置的情况下,D-NSGA-GKM算法相对于NSGA-III算法在非支配排序阶段耗时减少了13.02%,由于在环境选择阶段加入了修正算子、genetic  $K$ -means算法和基于惩罚的边界相交距离的计算,计算量略微增加,使得算法总体耗时减少了6.40%.显然,通过基于优势度矩阵的非支配排序算法有效地提高了算法运算效率.

通过上述4个指标可以表明,所提出的D-NSGA-GKM算法具有较好的收敛性和多样性,在处理多阶段武器协同火力分配问题上具有较好的时间性能.

## 4 结论

针对陆战场防御作战场景中的多阶段武器协同火力分配问题,提出了一种基于D-NSGA-GKM算法的多阶段武器协同火力分配方法.首先,建立了以最小敌方作战单元战场剩余价值、最小我方作战资源消耗、最小我方作战资源战斗损失为优化目标的多阶段武器协同火力分配模型;然后,在NSGA-III算法的基础上通过引入基于优势度矩阵的非支配排序算法、修复算子和genetic  $K$ -means均值聚类算法对其做出相应改进,提出了D-NSGA-GKM算法;最后,通过实验验证在相同初始参数的条件下,D-NSGA-GKM算法相对于NSGA-III等主流多目标优化算法具有较好的收敛性和时间性能,可以有效处理多阶段武器协同火力分配问题.所提出的优化方法可为战时作战指挥员提供作战辅助决策,同时,也为解决多阶段武器协同火力分配问题提供了方法参考.后续将对更高维度、更大尺度、约束条件更为复杂的火力分配问题进行深入研究.

## 参考文献(References)

- [1] Rai R N, Bolia N. Optimal decision support for air power potential[J]. IEEE Transactions on Engineering Management, 2014, 61(2): 310-322.
- [2] 孔德鹏, 常天庆, 郝娜, 等. 基于对抗的突击武器与支援武器协同火力打击决策方法[J]. 兵工学报, 2019, 40(3): 629-640.  
(Kong D P, Chang T Q, Hao N, et al. Confrontation-based cooperative fire strike decision-making method of assault weapons and support weapons[J]. Acta Armamentarii, 2019, 40(3): 629-640.)
- [3] Hosein P A, Athans M. Preferential defense strategies — Part I: The static case[R]. Massachusetts: MIT Laboratory Information Decision System, 1990.
- [4] Hosein P A, Athans M. Preferential defense strategies — Part II: The dynamic case[R]. Massachusetts: MIT Laboratory Information Decision System, 1990.
- [5] Wang Y, Li J, Huang W L, et al. Dynamic weapon target assignment based on intuitionistic fuzzy entropy of discrete particle swarm[J]. China Communications, 2017, 14(1): 169-179.
- [6] Wang Y, Zhang W G, Li Y. An efficient clonal selection algorithm to solve dynamic weapon-target assignment game model in UAV cooperative aerial combat[C].

- The 35th Chinese Control Conference (CCC). Chengdu: IEEE, 2016: 9578-9581.
- [7] Li J, Chen J, Xin B, et al. Solving the uncertain multi-objective multi-stage weapon target assignment problem via MOEA/D-AWA[C]. 2016 IEEE Congress on Evolutionary Computation (CEC). Vancouver, 2016: 4934-4941.
- [8] Li J, Chen J, Xin B, et al. Solving multi-objective multistage weapon target assignment problem via adaptive NSGA-II and adaptive MOEA/D: A comparison study[C]. 2015 IEEE Congress on Evolutionary Computation (CEC). Sendai: IEEE, 2015: 3132-3139.
- [9] Lötter D P, Vuuren V J H. A tri-objective, dynamic weapon assignment model for surface-based air defence[J]. ORiON, 2016, 32(1): 1-22.
- [10] Peng G, Fang Y W, Chen S H, et al. A hybrid multi-objective discrete particle swarm optimization algorithm for cooperative air combat DWTA[C]. International Conference on Bio-Inspired Computing: Theories and Applications. Xi'an, 2016: 114-119.
- [11] Jain H, Deb K. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach—Part I: Solving problems with box constraints[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(4): 577-601.
- [12] Wu X L, Li J, Shen X L, et al. NSGA-III for solving dynamic flexible job shop scheduling problem considering deterioration effect[J]. IET Collaborative Intelligent Manufacturing, 2020, 2(1): 22-33.
- [13] Yuan Y, Xu H, Wang B. An improved NSGA-III procedure for evolutionary many-objective optimization[C]. Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. Vancouver: ACM, 2014: 661-668.
- [14] Yuan Y, Xu H, Wang B, et al. A new dominance relation-based evolutionary algorithm for many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2015, 20(1): 16-37.
- [15] Bi X J, Wang C. An improved NSGA-III algorithm based on objective space decomposition for many-objective optimization[J]. Soft Computing, 2017, 21(15): 4269-4296.
- [16] Faied M, Girard A. Game formulation of multiteam target assignment and suppression mission[J]. IEEE Transactions on Aerospace and Electronic Systems, 2014, 50(2): 1234-1248.
- [17] Bertuccelli L F, Alighanbari M, How J P. Robust planning for coupled cooperative UAV missions[C]. The 43rd IEEE Conference on Decision and Control. Nassau, 2004: 2917-2922.
- [18] Pierro D F, Khu S T, Savic D A. An investigation on preference order ranking scheme for multiobjective evolutionary optimization[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(1): 17-45.
- [19] Zhou Y R, Chen Z F, Zhang J. Ranking vectors by means of the dominance degree matrix[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(1): 34-51.
- [20] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [21] Bi X J, Wang C. A niche-elimination operation based NSGA-III algorithm for many-objective optimization[J]. Applied Intelligence, 2018, 48(1): 118-141.
- [22] Zitzler E, Brockhoff D, Thiele L. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration[C]. Lecture Notes in Computer Science. Berlin: Springer, 2007: 862-876.
- [23] Ishibuchi H, Imada R, Setoguchi Y, et al. Reference point specification in inverted generational distance for triangular linear pareto front[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(6): 961-975.

### 作者简介

于博文(1988—), 男, 博士生, 从事智能火力与指挥控制系统、智慧互联与智能控制的研究, E-mail: yubowen\_0403@njust.edu.cn;

吕明(1980—), 女, 副研究员, 博士, 从事智能火力指挥与控制系统等研究, E-mail: 18761654456@163.com.

(责任编辑: 郑晓蕾)