

# 基于入侵杂草蝙蝠双子群优化的 装备保障编组协同任务规划

王坚浩<sup>†</sup>, 张 亮, 史 超, 车 飞, 张鹏涛

(空军工程大学 装备管理与无人机工程学院, 西安 710051)

**摘 要:** 针对装备保障编组协同任务规划问题, 构建以时效优先为目标, 考虑保障任务时序逻辑关系、任务执行质量和保障编组占用冲突等复杂约束以及保障编组能力更新机制的数学模型, 提出一种基于入侵杂草蝙蝠混合算法的双子群任务规划方法. 首先, 采用佳点集初始化方法, 在解空间生成具有均匀分布特征的种群; 其次, 设计具有修复操作的解编码和任务优先排序, 实现任务-编组-时间的匹配和冲突消解; 再次, 划分双子群, 利用入侵杂草优化算法和 Fuch 混沌蝙蝠优化算法协同进化; 最后, 应用重组算子引导种群进化, 均衡算法全局探索和局部搜索能力. 仿真算例表明, 所提方法可对大规模复杂任务分配方案进行精确高效的求解.

**关键词:** 装备保障编组; 任务规划; 双子群; 佳点集; 入侵杂草优化算法; 蝙蝠优化算法

中图分类号: TP18; TP301

文献标志码: A

## Cooperative task scheduling for equipment support groups using invasive weed bat dual-subpopulation optimization algorithm

WANG Jian-hao<sup>†</sup>, ZHANG Liang, SHI Chao, CHE Fei, ZHANG Peng-tao

(Equipment Management and Unmanned Aerial Vehicles Engineering College, Air Force Engineering University, Xi'an 710051, China)

**Abstract:** This paper proposes a hybrid task scheduling method based on the invasive weed bat algorithm upon dual-subpopulation to achieve objective of minimal task implementation time for equipment support groups cooperative task scheduling. The mathematical model is constructed considering complex constrains such as task sequential logical relationship, implementation quality, occupancy conflicts and capability renewal mechanism of support groups. Firstly, the good-point set theory is used to generate initial population with uniform distribution in the solution space. Then, the strategies of repair operator encoding and task priority ordering are designed to achieve tasks-groups-time sequential matching and conflict resolution. Moreover, all individuals are divided into dual-subpopulation by cooperative evolution using the invasive weed optimization algorithm and the Fuch chaotic bat optimization algorithm. Finally, the recombination operator for guiding population evolution is given to balance the exploration and exploitation ability. The simulation example is given to illustrate that the proposed method has better robustness and solving precision for the complicated task allocation scheme.

**Keywords:** equipment support groups; task scheduling; dual-subpopulation; good-point set; invasive weed optimization algorithm; bat optimization algorithm

## 0 引 言

信息化作战样式催生装备保障模式发生质的变革, 首要目标是达成保障资源和保障力量的精确化和协同化<sup>[1]</sup>, 即对多种保障要素, 如各类设备、器材、备件等保障资源和各类专业人员等保障力量进行整合和编组, 以保障编组为保障力量的基本单位, 寻求发挥一加一大于二的装备保障整体效能, 解决长期以来

多种保障要素独立行动、不协调、不同步, 保障能力不均衡的问题. 装备保障任务通常可以分解为性质不相一致的多项复杂序列任务, 且序列任务间存在时序逻辑关系约束、保障编组执行保障任务时的能力约束和保障编组占用冲突. 因此, 装备保障编组协同任务规划作为装备保障资源和力量编成及部署的重要内容, 其实质便是将保障编组配置到存在时序逻辑关

收稿日期: 2017-12-19; 修回日期: 2018-05-10.

基金项目: 国家自然科学基金项目(61503409).

责任编辑: 李德荣.

<sup>†</sup>通讯作者. E-mail: hamilton\_wang@sina.com.

系约束和能力约束的保障任务中,使整个保障编组又好又快地完成装备保障任务,最终实现“任务-编组-时间”三者之间的最佳匹配<sup>[2-3]</sup>。

装备保障编组协同任务规划研究主要包括任务规划建模与求解算法两个方面,现有大多数装备保障编组任务规划建模普遍存在以下不足:首先是缺乏对协同保障模式的支持,通常只允许一个保障编组执行一项保障任务<sup>[4-5]</sup>;其次是约束条件较为简单,一般只考虑保障任务执行时序逻辑关系约束和执行时间约束<sup>[6-7]</sup>,而实际上对于不同的保障任务需求和保障编组配置,保障能力需求和实际保障能力满足程度通常存在不匹配的问题。此外,在任务规划求解算法方面,传统的诸如整数规划等精确求解方法耗时长,不适用于装备保障编组协同任务规划问题;启发式方法如规划列表算法和启发式智能优化算法求解效率较高,但传统规划列表算法的局部搜索策略都是基于贪心策略的,文献[8]在此基础上针对资源调度问题设计了3种不同的贪心策略,并提出了混合贪心求解算法,但无法保证以稳定的高概率得到最优方案。针对传统规划列表算法存在的不足,启发式智能优化算法受到了广大学者的关注和研究,文献[9-10]基于动态列表规划(DLS)方法,结合遗传算法(GA)和量子遗传算法(QGA)提出了混合任务分配方法;文献[4]针对装备保障任务规划问题,提出了基于优先排序和粒子群优化(PSO)的混合任务规划方法;此外,模拟退火算法(SA)<sup>[11]</sup>和蝙蝠优化算法(BOA)<sup>[5]</sup>等启发式智能优化算法也应用于此类任务规划问题,但是上述这些典型算法与其他启发式智能优化算法一样普遍存在全局搜索能力与局部搜索能力难以协调等问题。因此,如何均衡算法的全局搜索能力和局部搜索能力是求解此类协同任务规划优化问题的关键。

为了均衡算法全局和局部搜索能力,近年来有学者提出了不同类型的双子群优化算法,例如文献[12]提出了一种双子群蝙蝠优化算法,外部子种群和内部子种群分别采用动态惯性权重模型和莱维飞行模型更新蝙蝠速度,从而提高了全局和局部搜索能力;另一类双子群优化方法是采用两类不同的子种群或优化策略的混合算法,文献[13]将入侵杂草优化算法(IWO)生长繁殖、空间扩散和竞争生存机制融入BOA,提出了一种入侵杂草蝙蝠优化算法(IWBOA),但IWBOA算法是在BOA主循环后使用入侵杂草算子,即对蝙蝠种群当前全局最优位置执行入侵杂草算子操作,是一种串行进化算法;文献[14]和文献[15]分别结合人工蜂群(ABC)与PSO和ABC与差分进化

算法(DE),提出了双子群人工蜂群粒子群混合算法(HTCAP)和双子群差分蜂群混合算法(BDABC),该类算法通过划分双子群,不同子种群采用不同的优化策略进行协同寻优,是一种并行进化算法。

基于以上分析,本文围绕装备保障编组协同任务规划问题,建立以时效优先为目标,考虑保障任务时序逻辑关系、任务执行质量和保障编组占用冲突等复杂约束以及保障编组能力更新机制的优化模型,受并行进化型双子群优化算法启发,提出一种基于入侵杂草蝙蝠混合算法的双子群求解算法。首先,采用佳点集初始化方法,在解空间生成具有均匀分布特征的种群;其次,通过设计具有修复操作的解编码和任务优先排序实现任务-编组-时间的匹配和冲突消解;再次,划分双子群,利用IWO算法和Fuch混沌BOA算法协同进化,并应用重组算子引导种群进化,均衡算法全局探索和局部搜索能力;最后,通过案例仿真验证方法的有效性和优越性。

## 1 装备保障编组协同任务规划模型

### 1.1 问题描述

保障任务(Task,  $T$ ):记包含  $N$  项任务的保障任务集为  $T = \{T_1, T_2, \dots, T_N\}$ ,对于  $\forall T_i \in T$ ,其属性包括:任务地理坐标位置  $LT_i = (XT_i, YT_i)$ ;任务开始时间  $ST_i$ ;任务持续时间  $CT_i$ ;任务能力需求向量  $DT_i = \{DT_{i1}, DT_{i2}, \dots, DT_{iL}\}$ ,  $DT_{il} (1 \leq l \leq L)$  为成功执行任务  $T_i$  所需的第  $l$  项能力值,若  $DT_{il} = 0$ ,则表示执行任务  $T_i$  并不需要第  $l$  项能力;各任务间的时序约束关系  $C^{T-T} = (c_{ij}^{T-T})_{N \times N}$ ,若任务  $T_i$  是任务  $T_j$  的前导任务,则  $c_{ij}^{T-T} = 1$ ,否则  $c_{ij}^{T-T} = 0$ 。保障任务集  $T$  中各任务之间具有顺序依赖和执行约束关系,如图1所示。

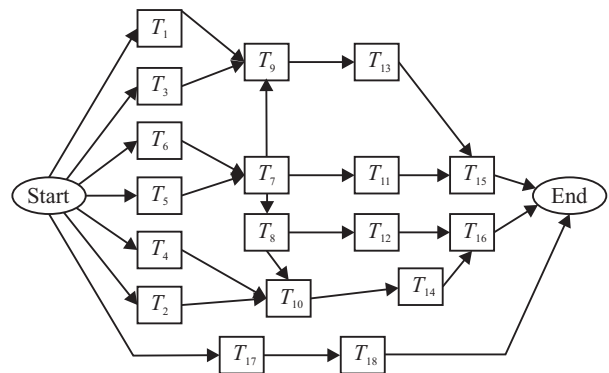


图1 保障任务间的时序约束关系

保障编组(Group,  $G$ ):拥有各类装备设施、器材、备件等保障资源和专业人员保障力量的编组,是保障任务的直接执行者。记包含  $M$  个编组的保障编组集为  $G = \{G_1, G_2, \dots, G_M\}$ ,  $\forall G_j \in G$ ,其属性包

括: 保障编组的初始地理坐标位置  $LG_j = (XG_j, YG_j)$ ; 保障编组平均移动速度  $VG_j$ ; 保障编组具备的初始保障能力向量  $CG_j = \{CG_{j1}, CG_{j2}, \dots, CG_{jL}\}$ ,  $CG_{jl} (1 \leq l \leq L)$  为保障编组  $G_j$  第  $l$  项保障能力值, 若  $CG_{jl} = 0$ , 则表示保障编组  $G_j$  不具备第  $l$  项保障能力。

装备保障编组协同任务规划问题是保障任务、保障编组和保障任务时序三者之间的匹配问题, 需要将存在时序逻辑关系约束和能力约束的保障任务分配给保障编组, 其分配方案可由矩阵  $D^{T-G} = (d_{ij}^{T-G})_{N \times M}$  表示, 若任务  $T_i$  分配给保障编组  $G_j$  执行, 则  $d_{ij}^{T-G} = 1$ , 否则  $d_{ij}^{T-G} = 0$ 。当一个保障编组需要执行多项任务时, 该保障编组将按照任务的优先级大小依次执行各项任务; 当一项任务由于不满足条件而无法开始执行时, 已经到达该任务位置的保障编组将在此等待直到该任务执行完成。

### 1.2 保障编组能力损耗及更新模型

保障过程中, 保障编组执行任务会造成其保障能力的损耗, 因此在保障编组执行完某一任务后, 应对其保障能力进行如下更新:

$$CG_{jl}^{\text{renewed}} = CG_{jl} \left( 1 - \omega_l \frac{CG_{jl}^{\text{real}}}{CG_{jl}} \right), \quad l \in [1, L]. \quad (1)$$

其中:  $\omega_l$  为第  $l$  项保障能力的损耗系数, 设定  $\omega_l$  为第  $l$  项保障能力类型的一种自身属性, 与保障编组本身和执行的任务无关;  $CG_{jl}$  为保障编组  $G_j$  在执行某一任务前所具备的第  $l$  项保障能力值;  $CG_{jl}^{\text{renewed}}$  为其执行该任务后的第  $l$  项保障能力值;  $CG_{jl}^{\text{real}}$  为保障编组  $G_j$  在执行任务过程中的第  $l$  项保障能力的实际消耗值。

保障编组  $G_j$  在执行任务  $T_i$  过程中保障能力的实际消耗值可表示如下:

$$CG_{jl}^{\text{real}} = \begin{cases} 0, & DT_{il} = 0; \\ DT_{il}, & DT_{il} \leq CG_{jl}; \\ CG_{jl}, & DT_{il} > CG_{jl}. \end{cases} \quad (2)$$

任务完成后, 对保障编组的保障能力进行更新, 更新后的保障编组能力值为

$$CG_j^{\text{renewed}} = \{CG_{j1}^{\text{renewed}}, CG_{j2}^{\text{renewed}}, \dots, CG_{jL}^{\text{renewed}}\}. \quad (3)$$

### 1.3 协同任务规划目标函数

对于  $\forall T_i \in T$ , 其结束时间记为  $ET_i = ST_i + CT_i$ , 其前导任务构成的集合记为  $S_{T_i}^T = \{T_k | c_{ki}^{T-T} = 1, 1 \leq k \leq N\}$ , 执行任务  $T_i$  的保障编组构成的集合记为  $S_{T_i}^G = \{G_k | d_{ki}^{T-G} = 1, 1 \leq k \leq M\}$ , 执行任务  $T_i$  时需要的能力类型构成的集合记为  $R_{T_i} = \{l | DT_{il} >$

$0, 1 \leq l \leq L\}$ .

当保障编组按照分配方案执行各项任务时, 所有任务的最终结束时间为

$$TIT = \max_{T_i \in T} (ET_i). \quad (4)$$

装备保障编组协同任务规划以最小化任务执行时间为目标, 则目标函数为

$$\min TIT = \min(\max_{T_i \in T} (ET_i)). \quad (5)$$

### 1.4 协同任务规划约束分析

装备保障协同任务规划问题的约束条件包括以下3类: 任务-编组分配约束; 任务-时间分配约束, 即任务时序逻辑约束; 任务执行质量约束。

1) 任务-编组分配约束. 每一项任务成功执行至少需要分配一个保障编组, 即满足

$$\sum_{j=1}^M d_{ij}^{T-G} \geq 1, \quad d_{ij}^{T-G} = \{0, 1\}. \quad (6)$$

2) 任务-时间分配约束. 对于  $\forall G_j \in S_{T_i}^G$ , 若保障编组  $G_j$  在执行  $T_i$  之前没有执行其他任务, 则其到达任务  $T_i$  所在位置的时间为

$$TIT_{ij} = \frac{\sqrt{(XT_i - XG_j)^2 + (YT_i - YG_j)^2}}{VG_j}. \quad (7)$$

若保障编组  $G_j$  在执行  $T_i$  之前执行的最后一个任务为  $T_j$ , 则其到达任务  $T_i$  所在位置的时间为

$$TIT_{ij} = \frac{\sqrt{(XT_i - XT_j)^2 + (YT_i - YT_j)^2}}{VG_i} + ET_j. \quad (8)$$

由于设定任务只有在所有先导任务都已完成, 且分配给其的所有保障编组都到达任务所在位置方能开始, 任务  $T_i$  的开始时间  $ST_i$  不小于先导任务  $T_k$  结束时间  $ET_k$  和分配给其的所有保障编组的到达时间  $TIT_{ij}$  之间的最大值, 因此有

$$ST_i \geq \max(\max_{T_k \in S_{T_i}^T} (ET_k), \max_{G_j \in S_{T_i}^G} (TIT_{ij})). \quad (9)$$

3) 任务执行质量约束. 保障编组执行任务  $T_i$  的质量定义为

$$QT_i = \left( \prod_{l \in R_{T_i}} CR_{il} \right)^{1/\|R_{T_i}\|}. \quad (10)$$

其中:  $\|R_{T_i}\|$  表示任务  $T_i$  能力需求类型的数量;  $CR_{il}$  表示任务  $T_i$  的第  $l$  项能力需求满足度, 可以利用其获得的保障能力  $CG_{jl}$  与能力需求  $DT_{jl}$  的比值表示, 即

$$CR_{il} = \begin{cases} 1, & \sum_{G_j \in S_{T_i}^G} CG_{jl} \geq DT_{il}; \\ \sum_{G_j \in S_{T_i}^G} \frac{CG_{jl}}{DT_{il}}, & \sum_{G_j \in S_{T_i}^G} CG_{jl} < DT_{il}. \end{cases} \quad (11)$$

保障编组集执行所有任务的平均质量定义为

$$Q_{\text{avg}} = \left( \prod_{i=1}^N Q_{T_i} \right)^{1/N}, \quad (12)$$

其中  $Q_{\text{avg}} \in [0, 1]$ ,  $Q_{\text{avg}}$  越大, 保障编组集执行所有任务的质量越好.

保障编组集执行所有任务的平均质量  $Q_{\text{avg}}$  必须高于某一下限阈值  $\varphi$ , 否则将导致任务执行效果远远低于期望效果, 即满足

$$Q_{\text{avg}} \geq \varphi. \quad (13)$$

### 1.5 协同任务规划数学模型

以任务执行时间最小化为目标, 综合考虑任务-编组-时间分配约束和任务执行质量约束, 因此装备保障编组协同任务规划问题的数学模型描述如下:

$$\begin{aligned} \min \text{TIT} &= \min(\max_{T_i \in T} (\text{ET}_i)). \\ \text{s.t.} \quad &\sum_{j=1}^M d_{ij}^{T-G} \geq 1, \quad d_{ij}^{T-G} = \{0, 1\}; \\ &\text{ST}_i \geq (\max_{T_i \in S_{T_i}^T} (\text{ET}_k), \max_{G_j \in S_{T_i}^G} (\text{TIT}_{ij})); \\ &Q_{\text{avg}} \geq \varphi; \\ &i = 1, 2, \dots, N, j = 1, 2, \dots, M. \end{aligned} \quad (14)$$

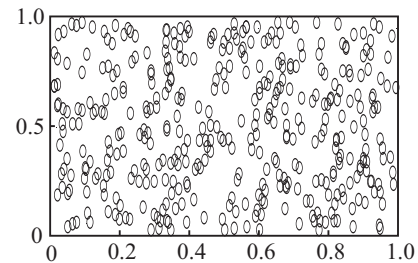
## 2 入侵杂草蝙蝠双子群任务规划算法

研究表明, BOA 算法<sup>[16-18]</sup>是一种全局型智能优化算法, 具有全局寻优能力强等特点, 但存在进化后期收敛速度慢、精度低等问题; 而 IWO 算法<sup>[19]</sup>具有很强的局部搜索能力, 但杂草种群易陷入局部极值, 而且不能很好地跳出局部极值. 本文针对 BOA 和 IWO 算法各自的优势和缺陷, 受并行进化型双子群优化算法启发, 针对装备保障编组协同任务规划问题(14), 设计一种入侵杂草蝙蝠双子群算法(IWBDA).

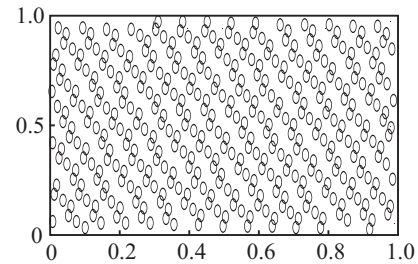
IWBDA 算法利用佳点集理论初始化种群, 使种群均匀分布在解空间, 避免其陷入局部最优; 通过设计具有修复操作的解编码, 基于优先排序的冲突消解和适应度函数设计与约束惩罚处理等可行解初始化方式, 保证算法的可达空间与模型的可行域相对应; 引入混沌搜索思想, 利用混沌变量的随机性、遍历性和规律性特征进行优化搜索; 采用 T 型逻辑函数进行离散化映射操作, 实现算法由连续域向二元离散域拓展; 应用重组算子选择最优个体作为双子群的全局极值, 从而实现算法间实时信息交流以达到协同进化, 使算法具有全局并行搜索和局部串行搜索的能力. 下面从以上 5 个方面介绍 IWBDA 算法.

### 2.1 基于佳点集的种群初始化

在种群初始化问题中, 通常采用随机初始化方法, 这是因为随机初始化方法简化了启发式智能算法的复杂度, 但该方法可能使得种群初始分布不均, 易导致算法陷入局部最优, 影响算法的搜索效率. 佳点集理论是由华罗庚和王元提出, 具有优越的特性, 佳点集的构造与空间维数无关, 为高维近似计算提供了一种非常好的方法. 有关佳点集的基本定义和性质可参考文献[20-22]. 研究表明, 采用佳点集选取的点比随机选取的偏差要小得多, 例如对于均匀地选取  $n$  个点, 采用随机方法的偏差为  $O(n^{-1/2})$ , 采用佳点集方法的偏差为  $O(n^{-1+\epsilon})$ ,  $\epsilon$  为无穷小量. 在相同取点个数的情况下, 相比于随机方法, 采用佳点集方法选取的点更均匀. 以取点数 400 为例, 采用随机方法和佳点集方法的二维空间取点效果对比如图 2 所示.



(a) 随机方法



(b) 佳点集方法

图 2 二维空间随机方法与佳点集方法取点效果对比

为了增强种群的多样性, 提升算法性能, 本文采用佳点集初始化方法, 在解空间中生成具有均匀分布特征的种群. 假设种群规模为  $\text{Pop}$ , 在  $D = N \times M$  维欧氏空间中生成  $\text{Pop}$  个点的佳点集为

$$P_{\text{Pop}}(i) = \{(\{r_1 \times i\}, \{r_2 \times i\}, \dots, \{r_D \times i\}), \\ i = 1, 2, \dots, \text{Pop}\}. \quad (15)$$

其中:  $r_k = \{e^k, k = 1, 2, \dots, D\}$ ,  $\{x\}$  表示  $x$  的小数部分.

采用佳点集法初始化种群个体如下:

$$x_{ik} = \begin{cases} 0, & \{r_k \times i\} < 0.5; \\ 1, & \text{otherwise.} \end{cases} \quad (16)$$

其中  $x_{ik}$  为第  $i$  个种群个体的第  $k$  维码值.

2.2 可行解初始化

2.2.1 具有修复操作的解编码

任务分配方案由矩阵  $D^{T-G} = (d_{ij}^{T-G})_{N \times M}$  表示, 直接对  $D^{T-G}$  进行编码, 此时解编码的冗余度较高. 但由于任务-编组分配约束, 其中产生的大多数解是不可行的. 为了进一步减小最优解搜索范围以提高运算效率, 本文设计强制修正机制对解进行修复. 若初始解不满足任务-编组分配约束, 即  $\sum_{j=1}^M d_{ij}^{T-G} = 0$ , 则强制给该任务随机分配一个编组. 为了更好地理解, 下面给出具体的伪代码.

begin

按照 2.1 节所述的基于佳点集的种群初始化方法产生初始化种群  $\mathbf{x} = \{x_i, i = 1, 2, \dots, D\}$

$R = \text{reshape}(\mathbf{x}, M, N)$

$d = \text{sum}(R)$

$R = R^T$

for  $i = 1 : N$

if  $d(i) == 0$

$r = \text{unidrnd}(M)$ ;

$R(i, r) = 1$ ; /\* 强制随机分配一个编组 \*/

end if

end for

end

其中:  $R = \text{reshape}(\mathbf{x}, M, N)$  为将种群个体  $\mathbf{x}$  转换为矩阵  $R$ ;  $r = \text{unidrnd}(M)$  为随机产生  $1 \sim M$  之间的整数.

2.2.2 基于优先排序的冲突消解

一个保障编组需要执行多项任务时会产生编组争用问题, 或造成任务的执行时序违背保障任务顺序依赖和执行约束关系, 因此需要对生成的解进行冲突消解. 编组产生争用冲突时, 按照任务优先权系数进行排序, 依次消解冲突, 首先满足优先级较高的任务. 任务优先权系数的影响因素主要包括任务持续时间、直接后续任务数量及其优先权系数, 任务优先权系数越大, 表示任务优先级越高. 任务优先权系数定义为

$$PT_i = CT_i + \frac{\sum_{j \in \text{OUT}(i)} PT_j}{\max_{j \in \text{OUT}(i)} PT_j} \quad (17)$$

其中:  $PT_i$  为任务  $T_i$  的优先权系数;  $\text{OUT}(i)$  为任务  $T_i$  的直接后续任务集.

由保障任务间的时序约束关系图 1 和任务优先权系数定义表达式 (17), 计算得到任务优先权系数如表 1 所示, 其中终止任务优先权系数一般设置为 0.

表 1 任务优先权系数

任务	优先权系数	任务	优先权系数
$T_1$	78	$T_{10}$	47
$T_2$	78	$T_{11}$	26
$T_3$	58	$T_{12}$	26
$T_4$	58	$T_{13}$	36
$T_5$	81.8	$T_{14}$	36
$T_6$	81.8	$T_{15}$	15
$T_7$	70.8	$T_{16}$	15
$T_8$	58.5	$T_{17}$	31
$T_9$	47	$T_{18}$	20

2.2.3 适应度函数设计与约束惩罚处理

按上述解编码方式, 虽然解决了任务-编组-时间约束问题, 但仍然会产生一些不可行解, 即不满足任务执行质量约束. 因此, 在计算各个解的适应度函数值之前, 需要对每个解是否满足任务执行质量约束进行判断, 当不满足任务执行质量约束时, 采用较为常用的罚函数处理方法, 通过构建相应的惩罚函数, 对该解的适应度函数值进行惩罚. 适应度函数设计为

$$\text{fitness} = \begin{cases} \max_{T_i \in T} (ET_i), \left( \prod_{i=1}^N QT_i \right)^{1/N} < \varphi; \\ \lambda, \text{ otherwise,} \end{cases} \quad (18)$$

其中  $\lambda$  为惩罚系数.

2.3 Fuch混沌搜索

混沌是非线性系统所特有的一种非周期运动现象, 具有随机性、遍历性和规律性等特征, 利用混沌变量的随机性、遍历性等内在特性进行优化搜索可使搜索过程避免陷入局部极值, 但不同的混沌映射对混沌优化过程的有较大影响. 目前常用的混沌映射为 Logistic 和 Tent, 但这两类混沌映射对搜索初始值较为敏感, 且搜索效率较低, 有可能导致算法效率降低. 傅文渊等<sup>[23]</sup>指出 Fuch 映射比 Logistic 和 Tent 映射具有更高的混沌动态优化性能和搜索效率, 且不依赖于初始值位置, Fuch 映射表达式为

$$y_i^t = \cos[1/(y_i^{t-1})^2]. \quad (19)$$

其中:  $y_i^0 \neq 0, t \in Z^+$ .

本文通过混沌遍历蝙蝠脉冲频率、响度和发生率变化区间, 使得蝙蝠脉冲频率、响度和发生率得到充分变化, 具体表达式为

$$f_i = f_{\min} + (f_{\max} - f_{\min}) \cdot |y_i^t|, \quad (20)$$

$$A_i^{t+1} = \left[ A_{\text{initial}} + (A_{\text{final}} - A_{\text{initial}}) \frac{t}{t_{\max}} \right] \cdot |y_i^t|, \quad (21)$$

$$r_i^{t+1} = \left[ r_{\text{initial}} + (r_{\text{final}} - r_{\text{initial}}) \frac{t}{t_{\max}} \right] \cdot |y_i^t|. \quad (22)$$

其中:  $f_{\min}$  和  $f_{\max}$  分别为脉冲频率的最小值和最大值,  $A_{\text{initial}}$  和  $A_{\text{final}}$  分别为脉冲响度的初值和终值,  $r_{\text{initial}}$  和  $r_{\text{final}}$  分别为脉冲发生率的初值和终值,  $t$  为

迭代次数.

### 2.4 离散化映射操作

编组协同任务规划是求解任务-编组-时间3者之间匹配的二元离散型问题. 然而,传统IWO算法和BOA算法采用浮点数编码,主要用于求解连续空间优化问题,对于此类二元离散型空间优化问题的求解优势并不明显. 目前大多数二元离散型优化算法采用就近取整或四舍五入的离散化方式,往往存在稳定性差、成功率低等问题. 本文提出采用T型逻辑函数映射方法将浮点数映射到0或1,从而实现二元离散化映射操作. 假设当前蝙蝠位置和杂草产生种子的位置分别为  $x_i^t = \{x_{i1}^t, x_{i2}^t, \dots, x_{ik}^t, \dots, x_{iPop}^t\}$  和  $x_j^t = \{x_{j1}^t, x_{j2}^t, \dots, x_{jk}^t, \dots, x_{jPop}^t\}$ , 则映射到二元离散型空间的方式为

$$x_{ik}^t \text{ or } x_{jk}^t = \begin{cases} 1, & \text{rand} < T(x_{ik}^t \text{ or } x_{jk}^t); \\ 0, & \text{otherwise.} \end{cases} \quad (23)$$

其中:rand为[0,1]范围内均匀分布产生的随机数,  $T(x_{id}^t) = \frac{e^{-\tau|x_{id}^t|} - 1}{e^{-\tau|x_{id}^t|} + 1}$  为T型逻辑函数,  $\tau$ 为设计参数.

### 2.5 重组算子

在双子群协同进化过程中,每一次迭代时均利用重组算子使蝙蝠子群和杂草子群互相跟踪对方的全局最优解  $B_{best}$  和  $W_{best}$ ,并通过设计选择概率  $P_{best}$  选取对应算法的最优解Best作为下一代杂草子群和蝙蝠子群的全局最优解. 选择概率  $P_{best}$  和最优解Best表达式分别为

$$P_{best} = \frac{f_{B_{best}}}{f_{B_{best}} + f_{W_{best}}}; \quad (24)$$

$$Best_d = \begin{cases} B_{best}^d, & \text{rand} < P_{best}; \\ W_{best}^d, & \text{otherwise.} \end{cases} \quad (25)$$

其中:选择概率  $P_{best}$  为选择BOA算法获取最优解  $B_{best}$  的概率;  $f_{B_{best}}$  为BOA算法全局最优解  $B_{best}$  的适应度函数值;  $f_{W_{best}}$  为IWO算法全局最优解  $W_{best}$  的适应度函数值;  $Best_d$  为最优解Best的第  $d$  维最优解,  $d \in [1, D]$ ;  $B_{best}^d$  和  $W_{best}^d$  分别为BOA和IWO算法所获取的第  $d$  维最优解.

### 2.6 算法步骤

综上所述,本文提出的IWBDA算法流程如图3所示.

IWBDA算法主要步骤如下:

Step 1: 参数设置. 包括种群规模Pop,最大迭代次数  $t_{max}$ ,脉冲频率最小值  $f_{min}$  和最大值  $f_{max}$ ,脉冲响应初值  $A_{initial}$  和终值  $A_{final}$ ,脉冲发生率初值  $r_{initial}$  和

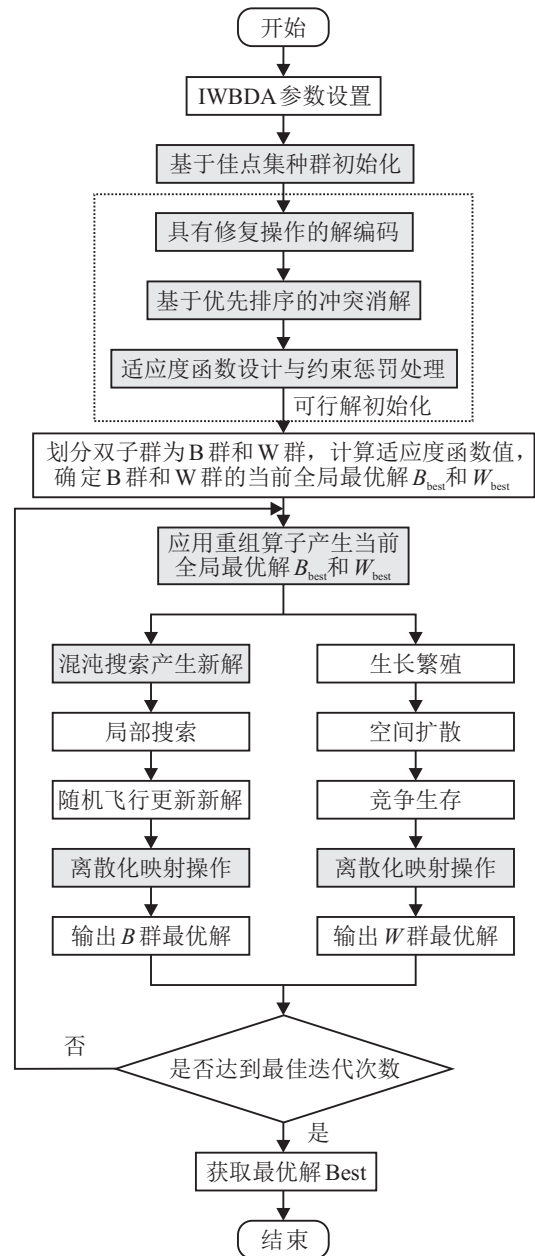


图3 IWBDA算法流程

终值  $r_{final}$ , 最小繁殖个数  $\psi_{min}$  和最大繁殖个数  $\psi_{max}$ , 标准差初值  $\sigma_{initial}$  和终值  $\sigma_{final}$ , 非线性调合指数  $k$ .

Step 2: 种群初始化. 基于佳点集的种群初始化, 使种群均匀分布在整个解空间.

Step 3: 可行解初始化. 包括具有修复操作的解编码、基于优先排序的冲突消解和适应度函数设计与约束惩罚处理.

Step 4: 划分双子群. 计算种群个体的适应度函数值, 并将种群等分为双子群  $B$  群和  $W$  群.

Step 5: 蝙蝠和入侵杂草并行优化. 利用BOA和IWO算法对  $B$  群和  $W$  群中所有个体优化.

Step 6: 确定  $B$  群和  $W$  群最优个体, 并计算相应适应度函数值.

Step 7: 利用重组算子获取最优解Best, 并将最优

解Best作为下一代B群和W群的全局最优解。

Step 8: 记录当前最优解。

Step 9: 判断算法终止条件. 如果达到给定的最大迭代次数, 则终止算法, 否则转入Step 5.

### 2.7 算法时间复杂度分析

根据2.6节算法步骤分析算法时间复杂度。

Step 1: 初始化种群规模Pop和其他参数, 时间复杂度为 $O(\text{Pop})$ ;

Step 2: 对初始解是否满足任务-编组分配需要判断Pop次, 判断编组争用问题需要进行1次任务优先级排序, 对不满足任务执行质量约束的解需要进行Pop次约束惩罚处理, 时间复杂度最多为 $O(\text{Pop} + 1)$ ;

Step 3: 将种群划分为双子群B群和W群, 计算B群和W群的最优值和全局最优值, 时间复杂度为 $O(\text{Pop} + 4)$ ;

Step 4: 蝙蝠和入侵杂草并行优化, 时间复杂度为

$2 \times O(\text{Pop}/2)$ ;

Step 5: 利用重组算子产生当前全局最优解, 时间复杂度为 $O(1)$ .

综上所述, IWBD A算法经过 $t_{\max}$ 迭代, 整个算法的时间复杂度为 $O(t_{\max} \times (\text{Pop} + 6))$ . 与标准BOA和IWO算法相比, IWBD A算法时间复杂度虽有所增加, 但仍处于同一数量级; 此外, 与文献[13]对于IWBOA算法时间复杂度分析相比较而言, IWBD A算法时间复杂度的优势较为明显。

## 3 仿真实验及比较

### 3.1 仿真算例及结果

以文献[24]中的想定为例, 保障任务间时序逻辑约束关系如图1所示, 保障任务集平均执行质量下限阈值 $\varphi = 0.8$ , 保障能力的损耗系数 $\omega_l$ 均设置为0.1, 保障任务和保障编组属性如表2和表3所示。

表2 保障任务属性

任务	能力需求向量								持续时间	位置	
	DT <sub>1</sub>	DT <sub>2</sub>	DT <sub>3</sub>	DT <sub>4</sub>	DT <sub>5</sub>	DT <sub>6</sub>	DT <sub>7</sub>	DT <sub>8</sub>	CT	XT	YT
T <sub>1</sub>	5	3	10	0	0	8	0	6	30	70	15
T <sub>2</sub>	5	3	10	0	0	8	0	6	30	64	75
T <sub>3</sub>	0	3	0	0	0	0	0	0	10	15	40
T <sub>4</sub>	0	3	0	0	0	0	0	0	10	30	95
T <sub>5</sub>	0	3	0	0	0	0	10	0	10	28	73
T <sub>6</sub>	0	0	0	10	14	12	0	0	10	24	60
T <sub>7</sub>	0	0	0	10	14	12	0	0	10	28	73
T <sub>8</sub>	0	0	0	10	14	12	0	0	10	28	83
T <sub>9</sub>	5	0	0	0	0	5	0	0	10	28	73
T <sub>10</sub>	5	0	0	0	0	5	0	0	10	28	83
T <sub>11</sub>	0	0	0	0	0	10	5	0	10	25	45
T <sub>12</sub>	0	0	0	0	0	10	5	0	10	5	95
T <sub>13</sub>	0	0	0	0	0	8	0	6	20	25	45
T <sub>14</sub>	0	0	0	0	0	8	0	6	20	5	95
T <sub>15</sub>	0	0	0	20	10	4	0	0	15	25	45
T <sub>16</sub>	0	0	0	20	10	4	0	0	15	5	95
T <sub>17</sub>	0	0	0	0	0	8	0	4	10	5	60
T <sub>18</sub>	0	0	0	8	6	0	4	10	20	5	60

表3 保障编组属性

保障编组	初始保障能力向量								速度	位置	
	CG <sub>1</sub>	CG <sub>2</sub>	CG <sub>3</sub>	CG <sub>4</sub>	CG <sub>5</sub>	CG <sub>6</sub>	CG <sub>7</sub>	CG <sub>8</sub>	VG	XG	YG
G <sub>1</sub>	10	10	1	0	9	5	0	0	2	85	40
G <sub>2</sub>	1	4	10	0	4	3	0	0	2	85	40
G <sub>3</sub>	10	10	1	0	9	2	0	0	2	85	40
G <sub>4</sub>	0	0	0	2	0	0	5	0	2	85	40
G <sub>5</sub>	1	0	0	10	2	2	1	0	1.35	85	40
G <sub>6</sub>	5	0	0	0	0	0	0	0	4	85	40
G <sub>7</sub>	3	4	0	0	6	10	1	0	4	85	40
G <sub>8</sub> ~ G <sub>10</sub>	1	3	0	0	10	8	1	0	4	85	40
G <sub>11</sub> ~ G <sub>13</sub>	6	1	0	0	1	1	0	0	4.5	85	40
G <sub>14</sub>	0	0	0	0	0	0	10	0	2	85	40
G <sub>15</sub> ~ G <sub>16</sub>	0	0	0	0	0	0	0	6	5	85	40
G <sub>17</sub>	0	0	0	6	6	0	1	10	2.5	85	40
G <sub>18</sub> ~ G <sub>20</sub>	1	0	0	10	2	2	1	0	1.35	85	40

本文仿真实验环境所涉及的代码均使用Matlab R2011a软件编写, 编译运行的PC机参数为: 32位Windows7操作系统, Intel(R) Core(TM)i-7 4790 3.60 GHz CPU, 4.00 GB内存。

为了验证IWBD A算法的有效性, 分别与DLS-GA<sup>[9]</sup>、DLS-QGA<sup>[10]</sup>和IWBOA<sup>[13]</sup>算法进行比较, 所有算法的共同参数设置为: 种群规模Pop = 100, 最大迭代次数 $t_{\max} = 300$ . 经过反复多次实验, 确定

本文所提IWBD A算法的其他最优参数设置为:脉冲频率最小值 $f_{\min} = 0$ ,最大值 $f_{\max} = 2$ ;脉冲响度初值 $A_{\text{initial}} = 1$ ,终值 $A_{\text{final}} = 0.1$ ;脉冲发生率初值 $r_{\text{initial}} = 0.1$ ,终值 $r_{\text{final}} = 0.8$ ;最小繁殖个数 $\psi_{\min} = 0$ ,最大繁殖个数 $\psi_{\max} = 5$ ;标准差初值 $\sigma_{\text{initial}} = 0.5$ ,终值 $\sigma_{\text{final}} = 0.001$ ;非线性调合指数 $k = 3$ ;T型逻辑函数设计参数 $\tau = 2$ .DLS-GA和DLS-QGA算法实验参数按照对应参考文献设置,IWBOA算法实验参数参考IWBD A算法.

为了体现仿真实验的有效性,通过对4种算法分别运行30次仿真计算实验所得到目标函数值(任务执行时间TIT)的最优值(Best)、最差值(Worst)、均值(Mean)、标准方差(Std)和平均运行时间(Avgtime)来考察算法性能,4种算法性能比较如表4所示.其中IWBD A算法最优解对应的任务-编组分配方案如表5所示.

表4 4种算法优化性能比较

性能	IWBD A	IWBOA	DLS-QGA	DLS-GA
Best	151.685 5	151.868	169.296 9	232.727 4
Worst	187.385 9	207.087 9	222.921 5	338.233 2
Mean	167.834	174.821 1	190.292 9	284.163 5
Std	10.020 3	13.760 8	15.241 7	21.883 3
Avgtime	36.230 5	40.503 5	43.601 2	45.936 2

表5 最优任务-编组分配方案

任务	分配编组
$T_1$	$G_1$
$T_2$	$G_6$
$T_3$	$G_4, G_5, G_9, G_{10}, G_{12}, G_{13}, G_{15}, G_{17}, G_{20}$
$T_4$	$G_4, G_6, G_7, G_{10}, G_{11}, G_{12}, G_{13}, G_{16}, G_{18}$
$T_5$	$G_1, G_3, G_7, G_{13}$
$T_6$	$G_1, G_2, G_7, G_8, G_9, G_{10}, G_{11}, G_{16}, G_{17}$
$T_7$	$G_8, G_9, G_{11}, G_{12}, G_{16}, G_{18}$
$T_8$	$G_1, G_3, G_6, G_7, G_{11}, G_{15}, G_{16}$
$T_9$	$G_3, G_8, G_{11}, G_{13}, G_{15}, G_{17}$
$T_{10}$	$G_4, G_{10}, G_{15}, G_{16}$
$T_{11}$	$G_6, G_7, G_8, G_9, G_{12}, G_{14}, G_{15}, G_{16}$
$T_{12}$	$G_2, G_8, G_9, G_{10}, G_{19}$
$T_{13}$	$G_2, G_7, G_8, G_9, G_{12}$
$T_{14}$	$G_5, G_{11}$
$T_{15}$	$G_3, G_6, G_7, G_{17}$
$T_{16}$	$G_2, G_4, G_5, G_9, G_{11}, G_{12}, G_{13}, G_{14}, G_{15}, G_{20}$
$T_{17}$	$G_{10}, G_{14}, G_{15}, G_{16}, G_{18}, G_{19}$
$T_{18}$	$G_1, G_3, G_4, G_{12}, G_{13}, G_{14}, G_{15}, G_{16}, G_{20}$

由仿真结果可知,基于IWBD A算法的协同任务规划方法具有较好的最优值、均值和方差,而且最优值和最差值变化区间较小,算法平均耗时稍短.这同时也验证了算法时间复杂度的分析结论,因此相比于IWBOA、DLS-QGA和DLS-GA,基于IWBD A算法的协同任务分配方案更为紧凑,时效性更优.

从统计学角度对4种算法优化性能进行比较,4种算法独立运行30次的实验盒须图如图4所示.

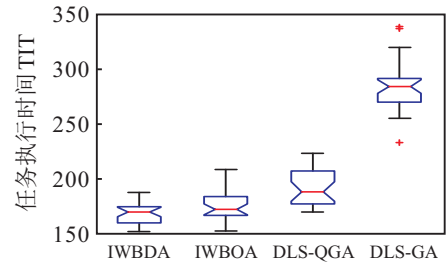


图4 4种算法任务执行时间统计分布

图4中:盒代表指标的四分位差,盒越短,数据越集中;盒中的横线代表数据的中位数,对于任务执行时间指标,中位数越小越好;盒上端的上须代表数据的最大值,盒下端的下须代表数据最小值,超出上下须的数据用“+”表示,代表该数据为异常值.由图4可知,基于IWBD A算法的协同任务规划方法能够在统计意义上得到全局最优和任务执行时间最为紧凑的方案,有效节省了全部任务集处理时间,提高了整个任务集的执行质量和编组资源利用率.

### 3.2 参数 $\varphi$ 对算法结果的影响分析

装备保障编组协同任务规划希望整个编组又好又快地完成所有任务,为了分析任务执行质量与任务执行时间的影响关系,通过对平均执行质量下限阈值 $\varphi$ 选取多个不同的数值进行仿真实验,分析参数 $\varphi$ 对任务执行质量的影响情况.

为了体现仿真实验的有效性,通过对IWBD A算法在不同平均执行质量下限阈值,即 $\varphi$ 分别为0.75、0.8、0.85和0.9的情况下分别运行30次仿真计算实验所得到目标函数值(任务执行时间TIT)的最优值(Best)、最差值(Worst)、均值(Mean)和标准方差(Std)来考察参数 $\varphi$ 对算法结果的影响情况,不同 $\varphi$ 值对算法结果比较如表6所示.

表6 不同 $\varphi$ 值对算法结果比较

性能	$\varphi = 0.75$	$\varphi = 0.8$	$\varphi = 0.85$	$\varphi = 0.9$
Best	142.551 7	151.685 5	149.376 4	155.913 3
Worst	176.427 3	187.385 9	189.995 4	205.897 7
Mean	157.984 6	167.834	170.683 3	177.076 5
Std	9.106 2	10.020 3	10.189 6	11.816 1

由仿真实验结果可知,平均执行质量下限阈值 $\varphi$ 对算法结果具有一定的影响,随着平均执行质量下限阈值 $\varphi$ 的不断提高,任务执行时间随之增大.总体而言,任务执行质量和任务执行时间是两个对立的指标.因此,在装备保障编组协同任务规划时,需要根据任务执行时间要求合理选择平均执行质量下限阈值这一任务执行质量指标.

## 4 结论

本文针对装备保障编组协同任务规划问题,建立了以时效优先为目标,考虑保障编组能力更新机制的

优化模型,并提出了一种IWBD A求解算法.通过运用佳点集理论初始化种群,同时引入Fuch混沌搜索机制,降低了算法陷入局部最优的可能性,增加了种群多样性;设计具有修复操作的解编码方式、基于优先排序的冲突消解和适应度函数设计与约束惩罚处理,保证了算法的可达空间与模型的可行域相对应;采用T型逻辑函数进行离散化映射操作,实现了算法由连续域向二元离散域拓展,并采用BOA和IWO算法协同进化,应用重组算子引导种群进化,均衡了算法的全局探索和局部搜索能力.装备保障编组协同任务规划案例的仿真结果表明,基于IWBD A算法的任务规划方法可以有效应对存在保障任务时序逻辑关系、任务执行质量和保障编组占用冲突等复杂约束的任务规划问题,能够在统计意义上获得更优更稳定的任务分配方案.

#### 参考文献(References)

- [1] 徐航,陈春良.装备精确保障概论[M].北京:国防工业出版社,2012:1-8.  
(Xu H, Cheng C L. Equipment efficient support generality[M]. Beijing: National Defense Industry Press, 2012: 1-8.)
- [2] Han X, Bui H, Mandal S. Optimization-based decision support software for a team-in-the-loop experiment: Asset package selection and planning[J]. IEEE Trans on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2013, 43(2): 237-251.
- [3] Han X, Mandal S, Pattipati K R, et al. An optimization-based distributed planning algorithm: A blackboard-based collaborative framework[J]. IEEE Trans on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2014, 44(6): 673-686.
- [4] 彭鹏菲,于钱,李启元.基于优先排序与粒子群优化的装备保障任务规划方法[J].兵工学报,2016,37(6): 1082-1088.  
(Peng P F, Yu Q, Li Q Y. A planning method of equipment support task based on priority ordering and particle swarm optimization algorithm[J]. Acta Armamentarii, 2016, 37(6): 1082-1088.)
- [5] 王坚浩,张亮,史超,等.装备精确保障任务规划建模与混沌蝙蝠算法求解[J].控制与决策,2018,33(9): 1625-1630.  
(Wang J H, Zhang L, Shi C, et al. Task scheduling modeling and chaotic bat algorithm solving method of equipment efficient support[J]. Control and Decision, 2018, 33(9): 1625-1630.)
- [6] 曾斌,姚路,胡炜,等.考虑不确定因素影响的保障任务调度算法[J].系统工程与电子技术,2016,38(3): 595-601.  
(Zeng B, Yao L, Hu W, et al. Scheduling algorithm for maintenance tasks under uncertainty[J]. Systems Engineering and Electronics, 2016, 38(3): 595-601.)
- [7] 彭鹏菲,于钱,李启元.基于改进粒子群优化的多目标装备保障任务规划方法[J].系统工程与电子技术,2017,39(3): 562-568.  
(Peng P F, Yu Q, Li Q Y. Method of multi-object equipment support task planning based on improved particle swarm optimization[J]. Systems Engineering and Electronics, 2017, 39(3): 562-568.)
- [8] 孙昱,姚佩阳,张少华,等.含区间参数的战场资源动态调度模型及算法[J].系统工程理论与实践,2017,37(4): 1080-1088.  
(Sun Y, Yao P Y, Zhang S H, et al. Dynamic battlefield resource scheduling model and algorithm with interval parameters[J]. Systems Engineering — Theory & Practice, 2017, 37(4): 1080-1088.)
- [9] 张杰勇,姚佩阳,周翔翔,等.基于DLS和GA的作战任务-平台资源匹配方法[J].系统工程与电子技术,2012,34(5): 947-954.  
(Zhang J Y, Yao P Y, Zhou X X, et al. Approach to operation task and platform resource matching based on DLS and GA[J]. Systems Engineering and Electronics, 2012, 34(5): 947-954.)
- [10] 万路军,姚佩阳,周翔翔,等.多编组协同任务分配及DLS-QGA算法求解[J].控制与决策,2014,29(9): 1562-1568.  
(Wan L J, Yao P Y, Zhou X X, et al. Cooperative task allocation methods in multiple groups using DLS-QGA[J]. Control and Decision, 2014, 29(9): 1562-1568.)
- [11] 姚佩阳,万路军,孙鹏,等.基于RHP-IVFSA的多智能体编组任务分配动态优化[J].系统工程与电子技术,2014,36(7): 1309-1319.  
(Yao P Y, Wan L J, Sun P, et al. Dynamic task allocation in multiple agent groups based on RHP-IVFSA[J]. Systems Engineering and Electronics, 2014, 36(7): 1309-1319.)
- [12] Luo J, Liu L, Wu X. A double-subpopulation variant of the bat algorithm[J]. Applied Mathematics & Computation, 2015, 263(7): 361-377.
- [13] 屈迟文,傅彦铭,侯勇顺.融合入侵杂草算子的蝙蝠算法[J].计算机应用与软件,2015,32(4): 243-246.  
(Qu C W, Fu Y M, Hou Y S. Bat algorithm fused with invasive weed operator[J]. Computer Applications and Software, 2015, 32(4): 243-246.)
- [14] 匡芳君,金忠,徐蔚鸿,等.Tent混沌人工蜂群与粒子群混合算法[J].控制与决策,2015,30(5): 839-847.  
(Kuang F J, Jin Z, Xu W H, et al. Hybridization algorithm of Tent chaos artificial bee colony and particle swarm optimization[J]. Control and Decision, 2015, 30(5): 839-847.)
- [15] 暴勋,曾建潮.一种双种群差分蜂群算法[J].控制理论与应用,2011,28(2): 266-272.  
(Bao L, Zeng J C. A bi-group differential artificial bee colony algorithm[J]. Control Theory & Applications, 2011, 28(2): 266-272.)
- [16] Yang X S. A new metaheuristic bat-inspired algorithm[C]. Nature Inspired Cooperative Strategies for

- Optimization. Berlin: Springer, 2010: 65-74.
- [17] Pravesjit S. A hybrid bat algorithm with natural-inspired algorithm for continuous optimization problem[J]. *Artificial Life and Robotics*, 2016, 21(1): 112-119.
- [18] 李煜, 裴宇航, 刘景森. 融合均匀变异与高斯变异的蝙蝠优化算法[J]. *控制与决策*, 2017, 32(10): 1775-1781. (Li Y, Pei Y H, Liu J S. Bat optimal algorithm combined uniform mutation with gaussian mutation[J]. *Control and Decision*, 2017, 32(10): 1775-1781.)
- [19] Mehrabian A R, Lucas C. A novel numerical optimization algorithm inspired from weed colonization[J]. *Ecological Informatics*, 2006, 1(4): 355-366.
- [20] 张铃, 张钊. 佳点集遗传算法[J]. *计算机学报*, 2001, 24(9): 917-922. (Zhang L, Zhang B. Good point set based genetic algorithm[J]. *Chinese J of Computers*, 2001, 24(9): 917-922.)
- [21] Yan H W, Cao Y L, Yang J X. Statistical tolerance analysis based on good point set and homogeneous transform matrix[J]. *Procedia CIRP*, 2016, 43: 178-183.
- [22] 朱旭辉, 倪志伟, 程美英, 等. 融合协同进化离散型人工鱼群算法和多重分形的雾霾预测方法[J]. *系统工程理论与实践*, 2017, 37(4): 999-1010. (Zhu X H, Ni Z W, Cheng M Y, et al. Haze prediction method based on multi-fractal dimension and co-evolution discrete artificial fish swarm algorithm[J]. *Systems Engineering — Theory & Practice*, 2017, 37(4): 999-1010.)
- [23] 傅文渊, 凌朝东. 自适应折叠混沌优化[J]. *西安交通大学学报*, 2013, 47(2): 33-38. (Fu W Y, Ling Z D. An adaptive iterative chaos optimization method[J]. *J of Xi'an Jiaotong University*, 2013, 47(2): 33-38.)
- [24] Yu F L, Tu F, Pattipati K R. Integration of a holonic organizational control architecture and multiobjective evolutionary algorithm for flexible distribute scheduling[J]. *IEEE Trans on Systems, Man, and Cybernetic, Part A: Systems and Humans*, 2008, 38(5): 1001-1017.

### 作者简介

王坚浩(1982—), 男, 讲师, 博士, 从事装备保障任务规划与资源调度, 智能控制、计算与优化等研究, E-mail: hamilton\_wang@sina.com;

张亮(1982—), 男, 副教授, 博士, 从事装备综合保障工程的研究, E-mail: 179108235@qq.com;

史超(1978—), 男, 副教授, 博士, 从事装备综合保障工程的研究, E-mail: sc78115@sohu.com;

车飞(1983—), 男, 讲师, 博士, 从事装备综合保障工程的研究, E-mail: cfei\_1983@163.com;

张鹏涛(1982—), 男, 副教授, 硕士, 从事装备综合保障工程的研究, E-mail: zptkgy@163.com.

(责任编辑: 齐 霖)

## 第32届中国控制与决策会议(2020 CCDC)征文通知

第32届中国控制与决策会议(2020 CCDC)将于2020年5月23日~25日在中国合肥举行. 中国控制与决策会议是当前控制、系统、决策理论与相关技术领域的大型国际学术会议. 会议旨在为从事相关领域教育和研究的国内外专家、学者及工程技术人员提供一个学术交流平台. 会议交流形式有: 大会报告、杰出讲座、会前研讨会、专题研讨会、专题辅导、教育论坛、分组报告和张贴论文等. 会议指定交流语言为英文. 本届会议由东北大学和中国自动化学会信息物理系统控制与决策专业委员会主办, 安徽大学承办. 会议英文论文将提交 IEEE Xplore Data Base, 被 EI 检索.

合肥古称庐州、庐阳,“一带一路”和长江经济带战略双节点城市,具有国际影响力的创新之都,也是一座具有2000多年历史的古城,素有“三国故地,包拯家乡”之称,是中国优秀旅游城市. 本届会议场地设在合肥世纪金源大饭店,2020 CCDC将会与您一同交流学术思想,感受古镇文化.

### 会议须知

1、统一网上投稿,并设立“张嗣瀛(CCDC)优秀青年论

文奖”,具体投稿程序、评奖申报资格和要求等事宜请登录大会官网进行了解: <http://www.ccdc.neu.edu.cn>.

2、热情欢迎业内专家以会议为平台,针对国际控制与决策领域前沿热点方向、科研学者普遍关心的话题,组织“邀请分会”、“特别专题”、“专家论坛”、“高峰论坛”等学术研讨活动,会议将提供优质的平台服务. 拟组织者请于2019年10月31日前与大会秘书处(secretary\_ccdc@ise.neu.edu.cn)联系并提交组织建议书.

3、经过专家评审,高质量的会议论文将被推荐提交到 TAYLOR & FRANCIS GROUP 出版的《Journal of Control and Decision》(EI Compendex, Scopus 检索)发表.

### 重要日期

初稿提交截止日期: 2019年10月31日;

录用通知发布日期: 2020年02月10日;

终稿提交截止日期: 2020年03月10日;

作者注册截止日期: 2020年03月10日.

中国控制与决策会议秘书处