

多 UCAV 协同目标分配算法研究

叶文¹, 朱爱红², 潘长鹏², 范洪达¹

(1. 海军航空工程学院兵器科学与技术系, 山东烟台 264001;

2. 海军航空工程学院训练部, 山东烟台 264001)

摘要: 针对多无人作战飞机(unmanned combat aerial vehicle, UCAV)协同目标分配问题,提出了一种基于离散粒子群算法的多 UCAV 协同目标分配方法。根据多 UCAV 协同目标分配问题的特点,建立了粒子与实际问题间的映射,设计了新的粒子群位置和速度更新公式,并且对标准粒子群算法作了改进。充分利用粒子群优化算法的全局搜索能力,有效地解决多约束条件下多 UCAV 协同目标分配问题。仿真结果表明,离散粒子群算法能够稳定快速地找到较优分配方案,并且算法简单、灵活。

关键词: 多无人作战飞机协同; 目标分配; 离散; 粒子群算法

中图分类号: V 279

文献标志码: A

Cooperation mission assignment algorithm for multi-UCAV

YE Wen¹, ZHU Ai-hong², PAN Chang-peng², FAN Hong-da¹

(1. Dept. of Ordnance Science and Technology, Naval Aeronautical and Astronautical Univ., Yantai 264001, China;

2. Dept. of Training, Naval Aeronautical and Astronautical Univ., Yantai 264001, China)

Abstract: A discrete particle swarm optimization (DPSO) algorithm is put forward for multi-UCAV (unmanned combat aerial vehicle) cooperation mission assignment problems. In the algorithm, a new code for particles and new update strategy for the position and speed of particles are applied so as to make the DPSO algorithm more suitable for multi-UCAV cooperation mission assignment problems. Aiming at the shortcoming of prematurity and poorness resulted from pure PSO, the DPSO algorithm takes full advantage of the complementary strengths of the particle swarm optimization algorithm, and the algorithm could solve multi-UCAV cooperation mission assignment effectively. The simulation results show that the DPSO algorithm could find the better assignment scheme fleetly and is simple and flexible.

Keywords: multi-UCAV cooperation; task assignment; discretization; particle swarm optimization (PSO)

0 引言

面向日趋复杂的现代战场环境,战术任务具有多重性与复杂性,单架无人作战飞机(unmanned combat aerial vehicle, UCAV)几乎无法完成指定的作战任务。多 UCAV 协同作战已经成为 UCAV 作战使用的必然选择。而多 UCAV 协同目标分配问题是多 UCAV 协同作战的保障和基础,其目的是为 UCAV 分配攻击目标,并对 UCAV 进行编队配置和设计粗略航路,使得整个 UCAV 机群的作战效能最大,作战代价最小^[1-2]。它是一个约束众多而复杂的优化问题,其解空间随目标总数的增加而呈指数级增加,使其成为一个多参数、多约束的 NP 问题。

最近发展起来的粒子群算法(particle swarm optimization, PSO)为该类问题的求解提供了一种新思路。但粒子

群算法具有连续本质,求解这类复杂离散问题面临诸多困难^[3-4]。为此,本文根据多 UCAV 协同目标分配问题的特点,设计了新的粒子群位置和速度更新公式,提出了一种基于离散粒子群算法的多 UCAV 协同目标分配方法。仿真结果表明,离散粒子群算法能够有效地解决多约束条件下多 UCAV 协同目标分配问题,并且算法简单、灵活,易于实现和扩展。

1 问题描述

设战场中有多架 UCAV $V_i (V_i \in V, i = 1, 2, \dots, N_V)$, 多个目标 $T_i (T_i \in T, i = 1, 2, \dots, N_T)$, 多个威胁或者禁飞区 $P_i (P_i \in P, i = 1, 2, \dots, N_P)$ 。设集合 $S_i = \{T_i^1, T_i^2, \dots, T_i^{\eta}\}$ 为分配给 V_i 的目标集合,则 UCAV 的目标分配问题可

描述为在尽量短的时间内将现有的目标分配给系统内的多架UCAV,即

$$\bigcup_{i=1}^{N_V} S_i = T \quad (1)$$

同时实现整体作战效能最大,作战代价最小,如图1所示。

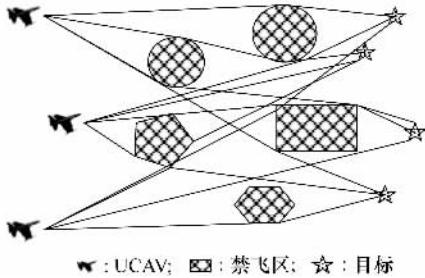


图1 UCAV作战想定

多UCAV协同目标分配是以整个UCAV编队的整体作战效能最优为目标的,而UCAV的损耗程度、目标的价值收益和飞行航程的长度是评价作战效能的主要指标^[5-6]。因此,衡量多UCAV协同目标分配方案优劣的重要性能指标主要包括以下3个部分。

(1) UCAV损耗最小指标

UCAV损耗最小指标通过最小化UCAV攻击目标的代价引导目标分配向着减小UCAV任务毁伤代价的方向进行。该指标使UCAV趋向于在安全航路飞行,使UCAV所受的威胁度最小。设第*u*架UCAV攻击目标*i*的生存概率为 PS_i ,则 $HS=1-PS_i$ 为攻击该目标*i*时的损耗。对UCAV进行分配任务,使得所有UCAV的损耗之和最小,即

$$\min \sum HS_i, i = 1, 2, \dots, m \quad (2)$$

(2) 目标价值收益最大指标

目标价值收益最大指标通过对UCAV执行任务时所获取的目标价值的评估,来引导目标分配的优化和决策向着使作战效能最大化的方向进行。该指标使UCAV趋向于攻击高价值目标。综合考虑目标的价值 v_i 、杀伤概率 PS_i ,则使用第*u*架UCAV攻击目标*i*时,收益为 $V_i = v_i \cdot PS_i$ 。为每架UCAV分配目标,使得总收益最大,即

$$\max \sum V_i, i = 1, 2, \dots, m \quad (3)$$

(3) 飞行航程最短指标

飞行航程最短指标引导目标分配首先为各个UCAV分配近距离的任务目标。设第*u*架UCAV与目标*i*间的距离为 D_i ,为每架UCAV分配目标,使得总航程最小,即

$$\min \sum D_i, i = 1, 2, \dots, m \quad (4)$$

根据目标价值收益最大化函数、UCAV损耗最小化函数和飞行航程最短函数,多UCAV协同目标分配的性能指标函数为

$$\max f = \omega_1 \sum V_i - \omega_2 \sum HS_i - \omega_3 \sum D_i, \quad i = 1, 2, \dots, m \quad (5)$$

式中, $\omega_1, \omega_2, \omega_3$ 为权系数,反映了每个子目标的重要程度。

2 离散粒子群算法

目前,对粒子群算法的研究主要集中在连续型的粒子群算法,即描述粒子状态及其运动规律的量都是连续的,而对离散粒子群优化算法的研究甚少。因为标准的粒子群算法具有连续本质,不太适宜于求解离散问题。因此,为了将其用于多UCAV协同目标分配问题的求解,本文设计了一种离散粒子群目标分配算法。

2.1 离散粒子群算法的编码方式

在粒子群算法中,每个粒子就是一个备选解,多个粒子共存、合作寻优。寻求一个合适的表达方式,使粒子和可行解对应是将算法应用于多UCAV协同目标分配中的关键。

多UCAV协同目标分配的决策关键在于确定:任务目标由哪架UCAV来执行。因此,这里采用自然数编码方式来表达,每个粒子长度等于目标总数,粒子由按目标编号顺序排列的UCAV分配编号组成,表示一种可能的分配方案。例如,UCAV数目*n*取3,目标数目*m*取4,一个粒子的表示如图2。

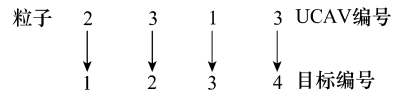


图2 一个粒子的表示方式

图中第2架UCAV攻击第1个目标,第3架UCAV攻击第2个目标,第1架UCAV攻击第3个目标,第3架UCAV攻击第4个目标。保证约束条件中对每一个目标必须分配一个UCAV的限制,单位粒子取值范围为 $[1, n]$ 。

2.2 离散粒子群算法的位置和速度更新公式

粒子群算法的实质在于粒子根据自己和同伴的飞行经验不断调整位置和速度,从而向最优位置飞行。粒子的新位置是粒子的速度、个体极值和全局极值相互作用的结果。根据多UCAV协同目标分配问题的实际特点,对粒子群算法的位置和速度更新公式进行重新定义^[7-8],即

$$X_i(t+1) = c_2 \otimes F_3(c_1 \otimes F_2(\omega \otimes F_1(X_i(t)), p_i(t)), p_g(t)) \quad (6)$$

式中, $X_i = (x_{i1}, x_{i2}, \dots, x_{im})$ 为粒子*i*在迭代中的位置; $p_i = (p_{i1}, p_{i2}, \dots, p_{im})$ 为粒子*i*的个体极值; $p_g = (p_{g1}, p_{g2}, \dots, p_{gm})$ 为全局极值; ω 为惯性权重; c_1 是认知系数,调节向 p_i 的飞行步长; c_2 是社会系数,调节向 p_g 的飞行步长。 $F_1(X_i(t))$ 是关于粒子 $X_i(t)$ 的函数,其作用是考虑粒子本身速度对其位置变化的影响; $F_2(X_i(t), p_i(t))$ 为 $X_i(t)$ 对 $p_i(t)$ 的学习操作; $F_3(X_i(t), p_g(t))$ 为 $X_i(t)$ 对 $p_g(t)$ 的学习操作。

位置更新公式由三部分构成,设 Ψ_i, Φ_i 为临时变量:

$$(1) \quad \Psi_i(t) = \omega \otimes F_1(X_i(t)) = \begin{cases} F_1(X_i(t)), & rand() < \omega \\ X_i(t), & rand() \geq \omega \end{cases}$$

这是粒子的“惯性”部分,表示粒子对自身飞行速度的思考。其中, $\omega \otimes F_1(X_i(t))$ 表示粒子的速度,即为一个概率为 ω 的目标置换操作; $rand()$ 为区间 $[0, 1]$ 上的随机数。它的实现方法为:由 $rand()$ 产生一个区间 $[0, 1]$ 上的随机数 r , 如果 $r < \omega$, 将对粒子进行置换操作 $\Psi_i(t) = F_1(X_i(t))$, 即产生两个在 $[1, m]$ 之间不同的随机数 a 和 b , 然后将粒子位置矢量的第 a 个数值与第 b 个数值互换, 也就是将攻击第 a 个目标的UCAV与攻击第 b 个目标的UCAV进行互换, 如图3所示; 如果 $r \geq \omega$, 则 $\Psi_i(t) = X_i(t)$ 。

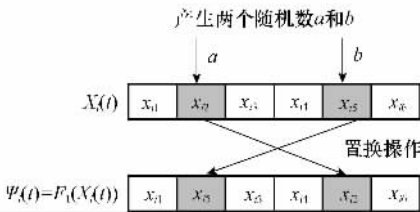


图3 目标置换操作

$$(2) \quad \Phi_i(t) = c_1 \otimes F_2(\Psi_i(t), p_i(t)) = \begin{cases} F_2(\Psi_i(t), p_i(t)), & rand() < c_1 \\ \Psi_i(t), & rand() \geq c_1 \end{cases}$$

这是粒子的“认知”部分,表示粒子根据个体极值 p_i 调整自身的位置。其中, $c_1 \otimes F_2(\Psi_i(t), p_i(t))$ 为一个概率为 c_1 的交叉操作。它的实现方法如下:由 $rand()$ 产生一个区间 $[0, 1]$ 上的随机数 r , 如果 $r < c_1$, 将对粒子进行交叉操作 $\Phi_i(t) = F_2(\Psi_i(t), p_i(t))$, 即产生两个在 $[1, m]$ 之间不同的随机数 a 和 b , 然后将第一部分互换操作后得到的粒子位置矢量的第 a 个与第 b 个之间的数值与个体 i 的个体极值 p_i 矢量的第 a 个与第 b 个之间的数值进行交叉操作, 也就是将攻击第 a 个目标到第 b 个目标的UCAV全部用个体 i 的个体极值 p_i 中攻击第 a 个目标到第 b 个目标的对应UCAV进行替换, 即进行交叉操作, 如图4所示; 如果 $r \geq c_1$, 则 $\Phi_i(t) = \Psi_i(t)$ 。

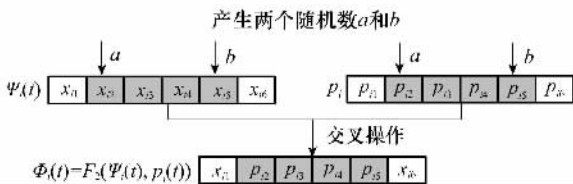


图4 与个体极值的交叉操作

$$(3) \quad X_i(t+1) = c_2 \otimes F_3(\Phi_i(t), p_g(t)) = \begin{cases} F_3(\Phi_i(t), p_g(t)), & rand() < c_2 \\ \Phi_i(t), & rand() \geq c_2 \end{cases}$$

这是粒子的“社会”部分,表示粒子根据全体极值 p_g 调整自身的位置。其中 $c_2 \otimes F_3(\Phi_i(t), p_g(t))$ 为一个概率为

c_2 的交叉操作。它的实现方法如下:由 $rand()$ 产生一个区间 $[0, 1]$ 上的随机数 r , 如果 $r < c_2$, 将对粒子进行交叉操作 $X_i(t+1) = F_3(\Phi_i(t), p_g(t))$, 即产生两个在 $[1, m]$ 之间不同的随机数 a 和 b , 然后将第二部分互换操作后得到的粒子位置矢量的第 a 个与第 b 个之间的数值与全体极值 p_g 矢量的第 a 个与第 b 个之间的数值进行交叉操作, 也就是将攻击第 a 个目标到第 b 个目标的UCAV全部用全体极值 p_g 矢量中攻击第 a 个目标到第 b 个目标的对应UCAV进行替换, 即进行交叉操作, 如图5所示; 如果 $r \geq c_2$, 则 $X_i(t+1) = \Phi_i(t)$ 。

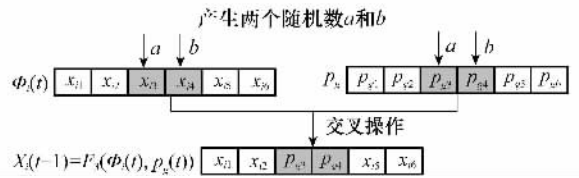


图5 与全体极值的交叉操作

迭代过程中, $X_i(t)$, $p_i(t)$ 和 $p_g(t)$ 不断更新, 最后输出的 $p_g(t)$ 就是全局最优解。粒子的位置矢量具体更新过程如图6所示。假设粒子的置换和交叉概率都为1, 且粒子 i 的位置矢量为 $X_i(t) = (2, 3, 1, 3, 4, 5)$, 粒子 i 的个体极值为 $p_i(t) = (2, 1, 3, 2, 5, 4)$, 粒子的全体极值为 $p_g(t) = (1, 2, 3, 2, 4, 5)$ 。如图6所示, 首先进行置换操作, 将粒子位置矢量中第一个目标对应的UCAV与第三个目标对应的UCAV进行置换, 得到临时变量 $\Psi_i(t)$; 接着 $\Psi_i(t)$ 与 $p_i(t)$ 进行交叉操作, 将 $\Psi_i(t)$ 中第一个和第二个目标对应的UCAV分别用 $p_i(t)$ 中第一个和第二个目标对应的UCAV的替换, 得到临时变量 $\Phi_i(t)$; 最后 $\Phi_i(t)$ 与 $p_g(t)$ 进行交叉操作, 将 $\Phi_i(t)$ 中第三个和第四个目标对应的UCAV分别用 $p_g(t)$ 中第三个和第四个目标对应的UCAV的替换, 得到下一时刻粒子的位置矢量 $X_i(t+1)$, 更新过程结束。



图6 粒子更新实例

2.3 离散粒子群算法的算法流程

离散粒子群算法的具体步骤为:

步骤 1 初始化 ω, c_1, c_2 和 λ (λ 为惯性权重下降速率)。

步骤 2 随机初始化粒子群中粒子的位置,并评价目标分配方案的性能指标,以此作为粒子的适应度。

步骤 3 令 $p_i(t) = X_i(t), p_g(t)$ 设置为初始群体中最佳粒子的位置。

步骤 4 判断算法收敛准则是否满足,如满足,则输出 $p_g(t)$,并由 $p_g(t)$ 得到最佳目标分配方案,算法结束,否则执行步骤 5。

步骤 5 对群体中的所有粒子进行如下操作:

步骤 5.1 按式(3)~式(6)更新粒子位置,并计算适应度;

步骤 5.2 如果粒子适应度优于 $p_i(t)$,令 $p_i(t) = X_i(t)$;

步骤 5.3 如果粒子适应度优于 $p_g(t)$,令 $p_g(t) = X_i(t)$ 。

步骤 6 令 $\omega = \lambda\omega$,转步骤 4。

离散粒子群算法的流程图如图 7 所示。

3 仿真实验

假设有 8 架UCAV和 15 个地面目标。为便于与文献[9]中的模型和算法相比较,目标的价值、UCAV对目标的杀伤概率均采用文献[9]中的数据,见表 1。

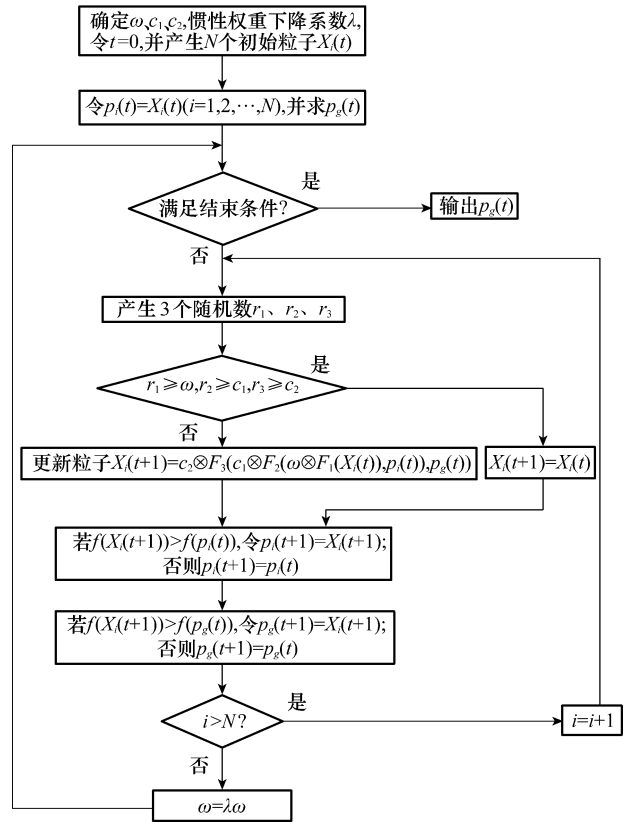


图 7 离散粒子群算法流程图

表 1 目标价值和UCAV对目标的杀伤概率

	V	UCAV1	UCAV2	UCAV3	UCAV4	UCAV5	UCAV6	UCAV7	UCAV8
目标 1	0.47	0.87	0.87	0.87	0.87	0.87	0.87	0.62	0.48
目标 2	0.97	0.52	0.52	0.52	0.52	0.52	0.52	0.87	0.20
目标 3	0.76	0.11	0.11	0.11	0.11	0.11	0.11	0.70	0.42
目标 4	0.62	0.78	0.78	0.78	0.78	0.78	0.78	0.22	0.16
目标 5	0.48	0.72	0.72	0.72	0.72	0.72	0.72	0.80	0.43
目标 6	0.77	0.69	0.69	0.69	0.69	0.69	0.69	0.42	0.58
目标 7	0.33	0.94	0.94	0.94	0.94	0.94	0.94	0.43	0.69
目标 8	0.74	0.72	0.72	0.72	0.72	0.72	0.72	0.90	0.03
目标 9	0.54	0.36	0.36	0.36	0.36	0.36	0.36	0.13	0.34
目标 10	0.65	0.28	0.28	0.28	0.28	0.28	0.28	0.95	0.72
目标 11	0.43	0.27	0.27	0.27	0.27	0.27	0.27	0.18	0.15
目标 12	0.35	0.74	0.74	0.74	0.74	0.74	0.74	0.19	0.24
目标 13	0.63	0.24	0.24	0.24	0.24	0.24	0.24	0.12	0.29
目标 14	0.66	0.78	0.78	0.78	0.78	0.78	0.78	0.61	0.30
目标 15	0.57	0.45	0.45	0.45	0.45	0.45	0.45	0.35	0.75

针对这一想定,我们分别采用遗传算法、离散粒子群算法对上述问题进行仿真,并将各种算法的仿真结果进行对

比。图 8 和图 9 分别为遗传算法和离散粒子群算法在 60 次迭代过程中最好解、平均解和最差解的变化曲线,算法的

比较结果如图 10 和表 2 所示。

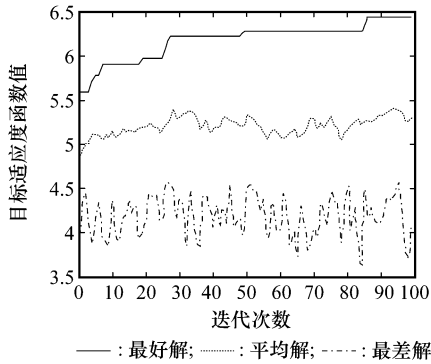


图 8 遗传算法的收敛曲线

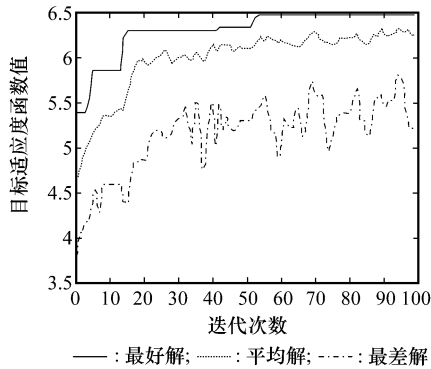


图 9 离散粒子群算法的收敛曲线

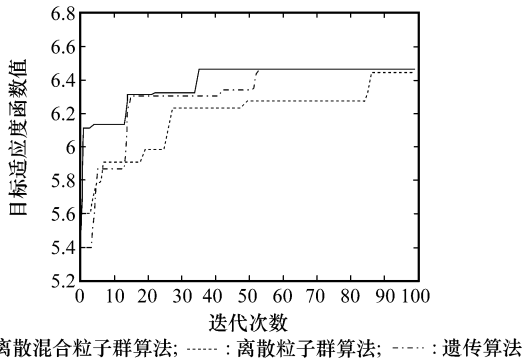


图 10 算法性能比较

表 2 算法性能比较

比较指标	遗传算法	离散粒子群算法
迭代 60 次后算法的最好解	6.440 4	6.471 9
迭代 60 次后算法的平均解	5.305 1	6.270 1
迭代 60 次后算法的最差解	4.122 1	5.219 3
最好解首次出现的代数	87	54
搜索到的最优分配结果	(5 7 7 4 7 2 6 7 6 7 2 2 5 3 8)	(2 7 7 1 7 2 3 7 4 7 2 2 8 6 8)

从图 10 的各种算法收敛速度和解的性能看,离散粒子群算法比遗传算法有较好的收敛特性,能够比较快地找到问题的最优解。从离散粒子群算法的解群随迭代次数的变化情况看,离散粒子群算法在较少的进化代数内,其解群就

向最优解的方向收敛,说明离散粒子群算法的优化效率较高。从离散粒子群算法平均解的变化曲线也可以看出,即使在迭代的后期,也不会出现所有粒子趋同的现象,使算法继续保持寻优能力。

从离散粒子群算法的原理看来,算法每代解群中所有解的“信息”的共享性和各个解的“自我素质”的提高,使得每代种群中的解具有“自我”学习提高和向“他人”学习的双重优点,这样,使得其下一代解有针对性地“从先辈”那里继承更多的信息。在这种“自我提高”和“取人之长,补己所短”的前提下,很快就达到了群体最优,从而能在较少的代数内找到最优解。

4 结 论

根据多 UCAV 协同目标分配问题的特点,本文建立了粒子与实际问题的映射,设计了新的粒子群位置和速度更新公式,并且对标准粒子群算法作了改进,充分利用粒子群优化算法的全局搜索能力,有效地解决多约束条件下多 UCAV 协同目标分配问题。仿真结果表明,离散粒子群算法具有高约束满足率的优良性能,算法能够稳定快速地找到较优分配方案,并且算法简单、灵活,易于实现和扩展,为该多约束多目标整数规划问题的求解提供了新的思路和方法。

参 考 文 献:

- [1] 叶媛媛. 多 UCAV 协同任务规划方法研究[D]. 长沙:国防科学技术大学, 2005.
- [2] 龙涛. 多 UCAV 协同任务控制中分布式任务分配与任务协调技术研究[D]. 长沙:国防科学技术大学, 2006.
- [3] Rasmussen S, Chandler P R, Mitchell J W, et al. Optimal vs. heuristic assignment of cooperative autonomous unmanned air vehicles[C]// *AIAA Guidance, Navigation, and Control Conference*, 2003:679-708.
- [4] Curz J B, Jr C, Chen G. Particle swarm optimization for resource allocation in UAV cooperative control[C]// *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004:16-19.
- [5] Fan Chunxia, Wan Youhong. An adaptive simple particle swarm optimization algorithm[C]// *Control and Decision Conference*, 2008:3067-3072.
- [6] Pan Q K, Tasgetiren M F, Liang Y C. A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem with makespan criterion[C]// *Proc. of the Int Workshop on UK Planning and Scheduling Special Interest Group*, 2005:31-41.
- [7] Pan Q K, Tasgetiren M F, Liang Y C. Minimizing total earliness and tardiness penalties with a common due date on a single-machine using a discrete particle swarm optimization algorithm[J]. *Lecture Notes in Computer Science*, 2006,4150: 460-467.
- [8] 潘全科,王文宏,朱剑英. 解决无等待流水车间调度问题的离散粒子群优化算法[J]. *计算机集成制造系统*, 2007,13(6):1127-1132.
- [9] 雷英杰, 张善文, 李续武. 遗传算法工具箱及应用[M]. 西安:西安电子科技大学出版社, 2005:128-130.