

应用离散粒子群 - 郭涛算法分配多无人机协同任务*

颜 骥¹, 李相民^{1,2}, 刘 波²

(1. 海军航空工程学院 兵器科学与技术系, 山东 烟台 264001;

2. 中航工业洛阳电光设备研究所 光电控制技术重点实验室, 河南 洛阳 471023)

摘要:针对以往考虑时间窗约束的多无人机协同任务分配问题模型不能反映在有效时间窗内,任务执行时间对任务收益的影响及求解算法效率较低的问题。建立了将任务收益和任务执行时间直接联系起来的任务分配模型和可行解到粒子整数编码方式的映射,设计了混合离散粒子群 - 郭涛算法的组合优化问题求解策略。借助粒子群算法利用粒子自身信息和种群有用信息指导种群进化的本质特点,优化郭涛算法的适应性序列倒置操作;设计了可变的学习选择概率来选择个体的学习粒子,改进了序列倒置算子。仿真实验验证了该方法处理复杂任务分配问题的有效性。

关键词:离散粒子群算法;郭涛算法;任务分配;有效时间窗;多无人机

中图分类号:TP273.5 **文献标志码:**A **文章编号:**1001-2486(2015)04-165-07

Cooperative task allocation of multi-UAVs with mixed DPSO-GT algorithm

YAN Ji¹, LI Xiangmin^{1,2}, LIU Bo²

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

2. Science and Technology on Electro-optic Control Laboratory, Luoyang Institute of Electro-optical Equipment of AVIC, Luoyang 471023, China)

Abstract: A general mathematics model for cooperative task allocation of multi-UAVs with time windows constrains was proposed which incorporating task gains and execution time directly, and simplifying the model formulation and algorithm designing. By defining a suitable particle structure, an algorithm based on the principles of discrete particle swarm optimization and Guo Tao algorithm was designed. The Inver-over Operator was directed by the swarm, the local and global optimal. Variable learning selection probability is introduced into the algorithm to select the learning particles, and the Inver-Over operator was modified. Simulation verifies the proposed task planning methodology for complex missions.

Key words: discrete particle swarm optimization algorithm; Guo Tao algorithm; task allocation; time windows of validity; multi-UAVs

无人机(Unmanned Aerial Vehicle, UAV)因其作战半径大、续航能力强、速度快、高隐身、高机动、零人员伤亡等优势,将替代有人飞机在枯燥、恶劣、危险环境中执行如防空压制、大范围搜索和摧毁、纵深打击、电子攻击、情报侦察监视等各种作战任务^[1]。如此复杂的作战任务,往往需要多架飞机互相协作与配合,共同完成。将这些复杂作战任务(mission)分解成一系列 UAV 能够理解和直接执行的子任务(task)(后文统称为任务),并为每架 UAV 分配任务集合,同时确定这些任务被执行的时序是多 UAV 协同任务分配问题的关键。

多 UAV 多任务分配是一类非常复杂的组合优化问题(NP-hard)。相关文献中针对这类问题

建立的模型包括旅行商问题(Traveling Salesman Problem, TSP)和车辆路径问题(Vehicle Routing Problem, VRP)以及这些模型的变体^[1]。任务的时间窗约束是多 UAV 任务分配中很常见的一类约束,文献[2]将多 UAV 的协同侦察任务分配问题抽象为带时间窗的多旅行商(multi-TSP with time-windows, MTSPTW)问题,并利用反射禁忌搜索算法对模型求解;文献[3]建立了多无人机任务规划的考虑时间窗约束的车辆路径问题模型(VRP with time-windows, VRPTW),并采用混合整数线性规划方法求解该问题;文献[4]针对任务的时间窗约束,建立了多无人作战飞机(Unmanned Combat Aerial Vehicle,UCAV)协同任务调度模型,应用粒子群优化(Particle Swarm

* 收稿日期:2014-10-27

基金项目:航空科学基金资助项目(20135184008)

作者简介:颜骥(1984—),男,湖南湘阴人,博士研究生,E-mail:53072472@qq.com;

李相民(通信作者),男,教授,博士,博士生导师,E-mail:xiangmin_li@163.com

Optimization, PSO) 算法求解。颜骥等在上述研究的基础上,建立了考虑时间窗约束的多 UAV 任务分配模型,利用混合离散粒子群(Discrete Particle Swarm Optimization, DPSO)算法和郭涛算法的改进方法求解该问题。

1 时间窗约束的多 UAV 任务分配

假设二维战场空间内 N_u 无人机组成的战斗编队执行 N_t 任务集合,任务分配算法的目的就是找出任务到无人机无冲突的匹配使得某项全局收益最大。若每项任务不被指派给多于一架的无人机,则说分配是无冲突的。全局目标函数假设为所有无人机局部收益值之和,而单架无人机的局部收益则是指派给该无人机的任务的函数。

1.1 时间窗约束下的任务收益函数

UAV 每完成一项任务就会得到收益,记为 v_{ij} 。收益值的大小体现了任务的重要程度和飞机执行该任务的能力, $v_{ij} = P_{ij} \times V_j$, 其中 i 为飞机序号, j 为任务序号, $P_{ij} \in [0, 1]$ 为 UAV $_i$ 执行任务 j 的成功概率,与飞机及其挂载的传感器、武器性能和目标相关; V_1, V_2, \dots, V_{N_t} 为各项任务的价值,其与任务的基准价值和被执行时间相关。为将时间窗约束合并到收益函数中,文献[5]给出了任务有效时间窗的相关定义:

1) 评分基线($s_j(t)$): $s_j(t)$ 表示智能体在时刻 t 到达任务 j 时,能从该任务获得的收益,其与任务的基准价值 R_j 及伴随的时间惩罚相关。令 $s_j(t) = e^{-\lambda_j(t-t_j^{\text{start}})} R_j$, 其中 $t - t_j^{\text{start}}$ 是任务期望开始时刻和智能体到达时刻之差, $\lambda_j > 0$ 是对迟到惩罚的折扣因子,若不考虑时间折扣,则评分基线为 $s_j(t) = R_j$ 。

2) 有效时间窗($u_j(t)$): 任务的有效时间窗指任务允许开始的时间段。任务 j 的有效时间窗定义为:

$$u_j(t) = \begin{cases} 1, & t_j^{\text{start}} \leq t \leq t_j^{\text{end}} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

其中 $[t_j^{\text{start}}, t_j^{\text{end}}]$ 表示任务 j 可以被执行和期望被完成的时间限制,若任务开始时间不在此时间段内,则任务收益为 0。时间窗的使用为对任务执行时间过早或过晚提供了惩罚框架。

基于上述定义,定义无人机 i 执行任务 j 的收益函数为

$$\begin{cases} v_{ij} = P_{ij} \times V_j \\ V_j = s_j(\tau_{ij}) \times u_j(\tau_{ij}) \end{cases} \quad (2)$$

无人机 i 执行任务 j 的时间 τ_{ij} 是无人机到达

任务 j 之前任务执行路径 p_i 的函数,而无人机的任务执行路径又由其任务集合 \mathbf{x}_i 唯一确定。给定任务集 $\mathbf{x}_i \in \{\emptyset \cup \mathcal{J}\}^{L_i}$ 和任务执行路径(任务执行时序) $\mathbf{p}_i \in \{\emptyset \cup \mathcal{J}\}^{L_i}$, τ_{ij} 可计算为

$$\tau_{ip_i(n)} = \begin{cases} \max\{t_{p_i(n)}^{\text{start}}, \tau_{i0} + \Delta(\text{tloc}_{i0}, \text{tloc}_{p_i(n)})\}, & n = 1 \\ \max\{t_{p_i(n)}^{\text{start}}, \tau_{ip_i(n-1)} + d_{p_i(n-1)} + \Delta(\text{tloc}_{p_i(n-1)}, \text{tloc}_{p_i(n)})\}, & n > 1 \end{cases} \quad (3)$$

其中, $\mathcal{J} \triangleq \{1, \dots, N_t\}$ 为任务下标集合; $\mathbf{p}_i(n)$ 表示任务执行路径上的第 n 项任务; d_j 表示任务 j 的持续时间; $\Delta(a, b)$ 表示无人机由任务位置 a 移动到任务位置 b 所耗费的时间; tloc_i 表示任务 i 所处的位置; τ_{i0} 表示无人机 i 开始执行任务的当前时间。

1.2 UAV 任务执行代价函数

UAV 执行任务过程中,会消耗武器、燃油,还会受到来自敌防御系统的反击,造成各种资源的损耗,统一称为 UAV 执行任务时付出的代价。代价的计算不仅与飞机和任务本身相关,还与飞机执行任务的路径有关,如飞行路径越长,则燃油消耗越大;在敌方雷达探测区域内暴露的时间越长,则被敌人发现和摧毁的可能性就越大^[4]。因此,单独计算无人机 i 执行任务 j 付出的代价 c_{ij} 是不切实际的,只能在给定的任务执行序列下计算无人机总的任务执行代价。为模型表述上的方便,相关文献一般采用启发式方法计算 c_{ij} ,具体计算方法可借鉴文献[6-8]。

1.3 时间窗约束下的问题模型

综上所述,任务分配问题可用如下整数规划模型(一般为非线性)描述:

$$\begin{aligned} \max \quad & \sum_{i=1}^{N_u} \left(\sum_{j=1}^{N_t} (\alpha v_{ij} - \beta c_{ij}) (\tau_{ij}(\mathbf{p}_i(\mathbf{x}_i))) x_{ij} \right) \\ \text{subject to:} \quad & \sum_{j=1}^{N_t} x_{ij} \leq L_i, \forall i \in \mathcal{U} \\ & \sum_{i=1}^{N_u} x_{ij} \leq 1, \forall j \in \mathcal{J} \\ & \sum_{n=1}^{|\mathbf{p}_i|} \Delta(\text{tloc}_{p_i(n-1)}, \text{tloc}_{p_i(n)}) \leq M_{p_i} \\ & x_{ij} \in \{0, 1\}, \forall (i, j) \in \mathcal{U} \times \mathcal{J} \end{aligned} \quad (4)$$

其中,二值决策变量 x_{ij} 为 1 则无人机 i 被指派给任务 j , 否则为 0; $\mathcal{U} \triangleq \{1, \dots, N_u\}$, $\mathcal{J} \triangleq \{1, \dots, N_t\}$ 分别为无人机和任务的下标集。约束 1 为无人机组

荷约束, L_i 表示无人机能完成的任务数上限; 约束 2 表示各项任务仅能被一架无人机执行; 约束 3 表示无人机的任务执行路径长度不能超过无人机的最大航程约束 M_{pi} ; 约束 4 表示各项任务至多能被执行一次。

若仅从燃油损耗来计算无人机 i 执行整个分配任务的代价 C_i , 则

$$C_i = \gamma_i \times \sum_{n=1}^{|p_i|} \Delta(tloc_{p_i(n-1)}, tloc_{p_i(n)}) \quad (5)$$

其中, γ_i 为燃油消耗系数, $p_i(0)$ 表示无人机的初始位置, $|\cdot|$ 表示集合 (\cdot) 中包含的元素个数。假设 $\sum N_i \times L_i \geq N_i, i = 1, \dots, N_u$ 。

由于任务的时间窗约束很大程度上决定了任务的收益, 在任务分配之初, 决策者只能通过无人机的有效载荷、最大航程约束和各任务的执行时间, 初步确定参加分配的任务和执行任务的无人机。因此, 并非所有参与分配的任务必须被执行, 而是根据约束情况来执行任务, 使得任务分配的收益最大化。

2 DPSO 和郭涛算法

2.1 DPSO 算法

PSO 算法^[9]中每个粒子对应一个可行解, 具有位置和速度两个属性, 分别表示当前粒子在解空间的位置和移动速度, 并以粒子位置向量对应的适应度函数值确定粒子的“优劣”程度。每个粒子通过下列公式更新自身状态:

$$\begin{cases} \mathbf{V}_i^{t+1} = w\mathbf{V}_i^t + c_1 \text{rand}_1(\mathbf{X}_i^t - \mathbf{P}_i^t) + c_2 \text{rand}_2(\mathbf{X}_i^t - \mathbf{P}_g^t) \\ \mathbf{X}_i^{t+1} = \mathbf{X}_i^t + \mathbf{V}_i^{t+1} \end{cases} \quad (6)$$

其中, \mathbf{V}_i^t 和 \mathbf{X}_i^t 分别表示粒子 i 在时刻 t 的速度和位置, \mathbf{P}_i^t 表示粒子 i 在飞行过程中经历的最好位置, \mathbf{P}_g^t 则表示整个群体目前找到的最好位置。 w 为惯性权, c_1, c_2 为学习因子, rand_1 和 rand_2 是 $[0, 1]$ 上的随机数。 PSO 的本质是利用粒子自身信息、个体极值信息和全局极值信息指导粒子向问题空间中更好位置“飞行”, 进而搜索到最优解。

目前 PSO 的研究大多局限在连续领域, 为求解组合优化问题, 一般采用 3 种方式对其进行离散化处理, 称为离散粒子群算法(DPSO), 各求解方法采取的策略和优缺点见文献[10], 此处不再赘述。

2.2 郭涛算法

郭涛算法^[11]是武汉大学郭涛提出的一种基于序列倒置(Inver-over)算子的优化算法(简称

GT 算法)。 GT 算法通过对种群内每个个体的子序列, 按照一定的选择概率进行自适应的倒置, 以此来提高算法的收敛速度。因郭涛算法兼具一元算子(以较小的概率 p 执行同一父体内的随机倒序)和二元算子(以较大的概率 $1-p$ 执行不同父体间的导向性倒序)的成分, 从而使其成为既有强的选择压力又有适应性算子的演化算法^[12]。

由于 GT 算法是随机选择种群中的个体进行学习的, 当前个体并非每次都能学到有用经验, 其学习的盲目性, 影响了算法的收敛速度^[10]。并且, 随着问题规模的扩大, GT 算法解的质量下降很快^[13]。

3 混合 DPSO-GT 算法的多 UAV 任务分配

多 UAV 多任务分配是复杂的组合优化问题, 受 DPSO 算法和 GT 算法思想的启发, 采用混合 DPSO-GT 算法求解该问题。

3.1 基本思路与粒子编码方式

混合算法的基本思路是: 在 DPSO 算法中引入序列倒置算子, 该算子在学习时不是随机选择学习对象, 而是按照一定规律, 从具有全局极值的粒子、具有局部极值的粒子子群以及种群中随机选择的粒子; 保留 GT 算法中粒子自身的变异操作(随机倒序), 保证种群的多样性; 改进 GT 算法的序列倒置算子, 提高算法的收敛速度和精度; 单粒子序列倒置操作后立即对粒子评价, 若新粒子优于旧粒子, 更新旧粒子。

采用正整数向量编码的方式。对每个可能的解, 编码应能表示 N_t 个任务在 N_u 架无人机之间的分配。设 3 架无人机执行 6 项任务某个可能的解可用路径序列^[14]表示为:

$$5-0-4-3-6-0-2-1$$

这表示无人机 1 执行任务 5, 无人机 2 依次执行任务 4, 3 和 6, 无人机 3 则依次执行任务 2 和 1。若路径序列中存在相邻的两个 0, 如

$$5-2-1-0-0-4-3-6$$

则表示有一架无人机(无人机 2)未被分配任务。

因此, 多无人机多任务分配问题解的编码可用 $N_t + N_u - 1$ 维的向量表示。向量每一维对应的正整数表示任务或任务分割节点, 该正整数在向量中所处的维数则表示其在任务路径序列中的位置。不至于引起后文倒置算子的混淆, 用大于 N_t 的整数表示任务分割节点 0。如上例 3 架无人机 6 项任务的任务路径序列

$$5-0-4-3-6-0-2-1$$

可用编码表示为:

$$5-7-4-3-6-8-2-1$$

3.2 序列倒置算子的改进

假设当前粒子 s 为 5-7-4-3-6-8-2-1, 图 1 为基本序列倒置操作的单次迭代过程:

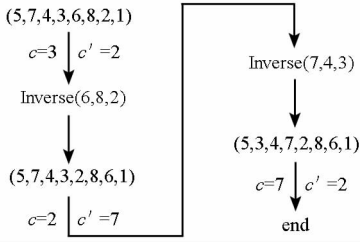


图 1 基本序列倒置操作单次迭代过程
Fig. 1 Single iteration of basic inver-over operator

由图 1 可知,第一次倒置操作为当前解引入了新的边(3,2),第二次倒置操作在引入新边(7,2)的同时却将上次倒置引入的新边(3,2)从解中移除。

事实上,基于当前种群中各个体所包含的有用信息来指导倒序是郭涛算法成功的关键^[12],为尽量保留这些有用的信息,即倒置操作引入的新边,采用带方向的循环序列倒置^[13,15]方法,将粒子的编码看成一个有方向的虚拟圆环,编码中所有的倒置操作均发生在从 c 的下一位基因到基因 c' 的有向子序列中,如图 2 所示。

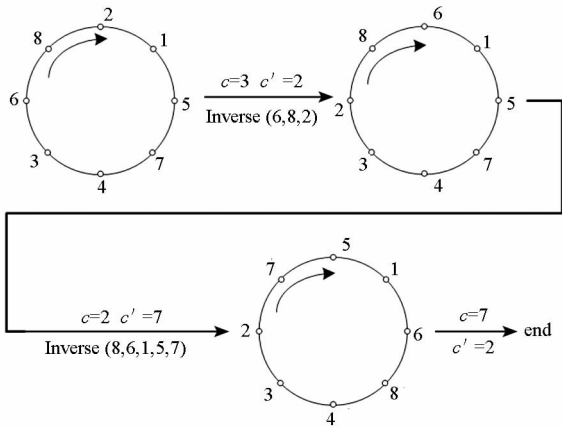


图 2 带方向的循环序列倒置操作单次迭代过程
Fig. 2 Single iteration of inver-over operator considering the direction of the tour

3.3 混合 DPSO-GT 算法参数设置

根据 3.1 节的思路,设计的混合算法引入 3 个学习选择概率参数和 1 个局部最优子群比参数: p_1, p_2, p_3 和 r_s 。局部最优子群^[10]指由种群中适应度较高的部分粒子组成的子群,若种群粒子规模 $M = 100, r_s = 0.2$,则局部最优子群由适应度较高的前 20 个粒子组成。

如图 3 所示,概率 p_1 用于决定当前粒子是从

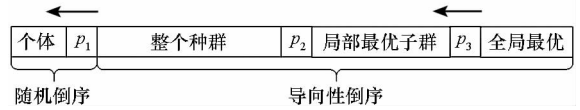


图 3 粒子学习来源及选择概率区分
Fig. 3 Learning resource of particles and selection probability differentiating

自身挑选一个基因进行随机倒置操作还是从其他粒子中挑选基因进行导向性倒序操作; p_2 和 p_3 则用于区分当前粒子向种群中其他粒子学习时,粒子的来源。类似 PSO 算法中的可变惯性权重 w , 概率值 p_1 和 p_3 随迭代的进行逐步减小,如图 3 中箭头所示,其值按式(7)变化

$$\begin{cases} p_1 = 0.1 \times \exp\left(\frac{g}{N_g} \times \lg(0.2/0.5)\right) \\ p_3 = 0.7 \times \exp\left(\frac{g}{N_g} \times \lg(0.2/0.5)\right) \end{cases} \quad (7)$$

其中, g 为当前运行的代数, N_g 为算法设定的运行代数。在算法运行初期, p_1 和 p_3 较大,类似文献^[10]的方法,粒子随机倒置操作和向种群中全局最优以外的其他粒子学习的导向性倒置操作较多,有利于算法跳出局部极小点,便于全局搜索;而到了算法运行后期,则算法类似文献^[12]的快速倒序算法,粒子向种群中其他粒子学习的概率和向全局最优值学习的概率相近,在保持种群多样性的同时,利用当前最优解指导倒序,加快算法收敛。概率值 p_2 不变,设定为 0.5。

3.4 适应度函数

算法的适应度函数即为式(4),通过对粒子的编码可实现约束 2 和约束 4,对于无人机任务能力约束和最大航程约束,采用如下方法:

当分配方案(粒子)中某架无人机的任务数超出其能完成最大任务数时,算法不再计算方案中后续任务给该架无人机带来的收益和付出的代价。

当分配方案中某架无人机的任务数在其能完成最大任务数之内时,判断无人机的当前航程是否超出其最大航程,若是,则将任务列表中的当前及其后续任务去除。

3.5 混合 DPSO-GT 算法描述

混合算法可描述如下:

步骤 1 设置种群大小 M 、运行总迭代数 N_g 、学习选择概率 $p_1 \sim p_3$ 、局部最优子群比 r_s 。

步骤 2 根据种群大小,按 3.1 节的编码方式随机产生粒子并计算其适应度值。求每个粒子的 P_i , 并求出种群的 P_g 。

步骤 3 按式(7)计算 p_1 和 p_3 , 根据 r_s 值确

定局部最优子群 M' 。

步骤4 对种群中的每个粒子 S , 随机选择一个基因 c 。

步骤5 产生一个 $(0,1)$ 随机数 r_d , 若 $r_d \leq p_1$, 则从当前粒子 S 中随机选出第二个基因 c' ; 若 $p_1 < r_d \leq p_2$, 则从种群 M 中随机选择一个粒子 S' 作为学习对象, 即, 令 S' 中 c 的下一个基因为 c' ; 若 $p_2 < r_d \leq p_3$, 则随机从局部最优子群 M' 中选择一个粒子作为学习对象; 若 $r_d > p_3$, 则选择全局最优粒子 P'_g 作为学习对象。

步骤6 判断 S 的基因 c 和 c' 是否相邻。若相邻, 判断是否种群中所有粒子都已完成学习, 如学习未完成, 转入步骤4; 若已完成, 转入步骤9。若两基因不相邻, 则转入步骤7。

步骤7 按3.2节提供的方法进行子序列倒置。单次倒序操作后, 按照式(4)、式(5), 立即计算该粒子的适应度值, 若该值大于序列倒置前适应度值, 则更新当前粒子; 否则, 不更新当前粒子。若更新后该粒子的适应度值大于其 P_i 值, 则更新该粒子及其 P_i 值, 否则不更新。同样, 若 P_i 值大于 P_g 值, 则更新相应的 P_g 值及其对应的粒子。

步骤8 令 $c = c'$, 转入步骤5。

步骤9 记录该次迭代的最大值 P'_g , 并判断是否达到最大迭代次数, 若是, 算法结束; 否则, 转入步骤3。

算法的具体流程如图4所示。

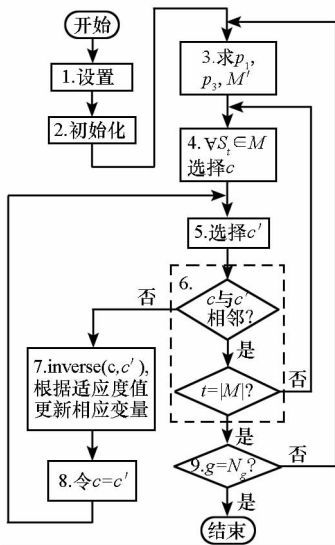


图4 混合 DPSO-GT 算法流程框图

Fig.4 Flow chart of mixed DPSO-GT algorithm

4 仿真算例

4.1 任务想定

假设经前期作战, 我方由3架具备侦察、监视

能力的无人机组成的编队, 欲实施对敌10个目标的侦察、监视任务。本节仿真分析运用混合算法, 完成3架无人机对上述10个目标进行任务分配的效果。

仿真场景中, 3架无人机的初始位置, 10个任务的期望开始时间和位置坐标在横坐标为 $(-2\text{km}, 2.5\text{km})$, 纵坐标为 $(-1.5\text{km}, 5.5\text{km})$ 的二维平面内随机生成; 设侦察监视任务的持续时间均为10s; 采用式(1)~(5)的无人机任务分配模型。

表1和表2中的数据为某次仿真实验中的无人机信息和目标信息。假设目标和无人机在二维平面内运动, 便于说明问题, 假设任意两目标位置距离远大于无人机最小转弯半径, 无人机在不同位置间的航行距离为该两点间的 Euclidean 距离。

表1 无人机信息表

Tab.1 Information of UAVs

标志	初始位置/km	速度/(km/s)	燃油成本
A1	(-0.883, 4.018)	0.2	1
A2	(-0.885, 0.563)	0.2	1
A3	(-1.531, -0.878)	0.2	1

表2 目标信息表

Tab.2 Information of targets

目标标志	目标坐标/km	最早执行时间/s	最晚执行时间/s
T1	(1.404, 4.491)	4.675	14.675
T2	(2.159, -1.271)	21.169	31.169
T3	(-0.757, 1.376)	23.176	33.176
T4	(0.878, 5.448)	37.595	47.595
T5	(0.616, 0.278)	93.679	103.679
T6	(-1.448, 4.479)	93.795	103.795
T7	(1.915, 2.699)	6.425	16.425
T8	(-1.857, 0.269)	43.279	53.279
T9	(-0.874, 4.666)	94.652	104.652
T10	(1.378, -0.683)	63.745	73.745

4.2 任务分配结果

基于4.1节中的想定, 设各架无人机最大任务数为4, 使用混合 DPSO-GT 算法得到任务分配结果如图5、图6所示。图5表示各无人机的任务分配和任务执行路径(任务执行顺序), 图6表示各无人机的任务计划表, 即任务的执行时间, 图

6 中粗实线表示任务实际执行时间,细虚线表示任务期望执行时间。

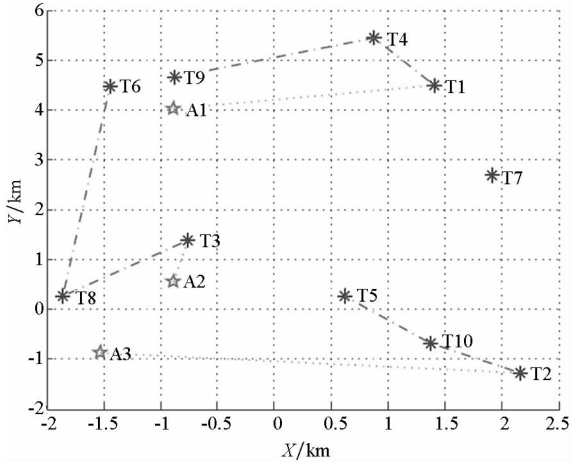


图 5 时间窗约束下的无人机任务路径
Fig. 5 UAV paths with time windows

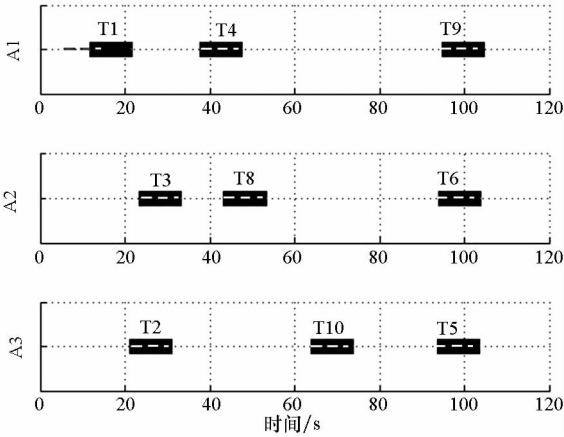


图 6 无人机计划时刻表
Fig. 6 UAV schedules

由图 6 可知,运用本算法生成的无人机任务计划表,任务 1 比预计时间推迟 7 秒执行,任务 7 因各架无人机距其较远,无法在其有效时间内到达而无法执行,其他任务都按计划执行,取得了很好的规划结果。设各架无人机最大任务数为 2,则规划结果如图 7 所示。

设各架无人机最大任务数为 4,单架无人机最大航程不超过 6km,则规划结果如图 8 所示。

4.3 与其他方法的比较

为验证混合 DPSO-GT 算法(后文简称之为 MIDPSO 算法)的有效性,将该方法和文献[4]的 DPSO 算法、文献[12]的快速倒序算子(Faster Inversion Operator, FIO)、文献[10]的反序 DPSO(Inver-over DPSO, IDPSO)算法对上述想定条件下进行 100 次蒙特卡洛仿真结果比对,其中种群

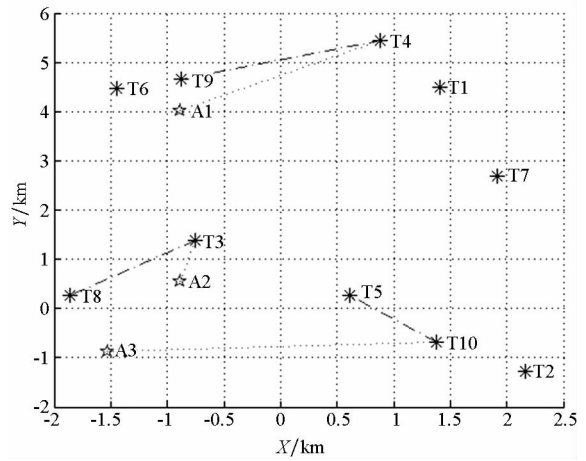


图 7 最大任务数约束下的无人机任务路径
Fig. 7 UAV paths with maximum workload constraints

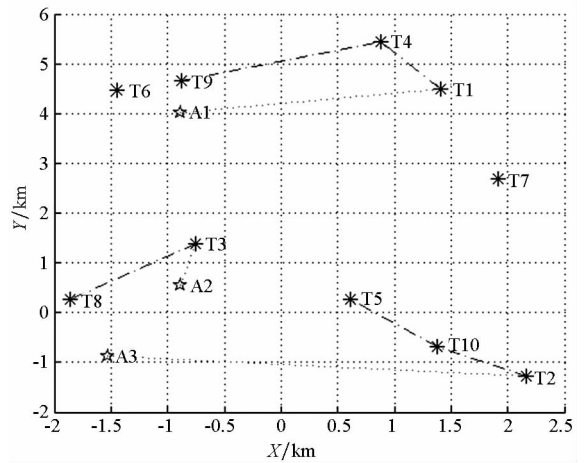


图 8 最大航程约束下的无人机任务路径
Fig. 8 UAV paths with maximum range constraints

规模为 40,迭代次数为 100,其他参数设置见 3.3 节及相关文献。

表 3 为 DPSO 算法, IDPSO 算法, FIO 和 MIDPSO 算法的运行结果。

表 4 为其他 3 种算法获得与 MIDPSO 算法同等求解质量时,种群大小、迭代次数、平均运行时间的 100 次蒙特卡洛仿真统计值。

表 3 4 种算法同等运行条件下的求解质量
Tab. 3 Solution quality of 4 algorithms under the same running conditions

	DPSO	FIO	IDPSO	MIDPSO
最优值	4	89	98	100
95% 次数	1	56	68	89
平均时间/s	0.294 8	1.690 7	1.702 2	1.915 0

表4 4种算法同等求解质量下的运行条件

Tab.4 Running conditions of 4 algorithms under the same solution quality

	DPSO	FIO	IDPSO	MIDPSO
种群规模	500	70	60	40
迭代次数	500	130	120	100
平均时间/s	17.985	3.172	2.937	1.915

由表3、表4结果可知,在同等运行条件下,和同等的求解质量要求下,MIDPSO算法相比其他算法,在时效性和求解质量方面都具有显著优势。

仿真发现,MIDPSO算法的运行效率受学习选择概率参数 $p_1 \sim p_3$ 的影响较大,3.3节对3个参数的设置是经过多次蒙特卡洛仿真实验得出的结果,限于篇幅,不做进一步说明。

5 结论

讨论了时间窗约束下的多UAV协同多任务分配问题。时间窗的约束,任务必须在指定的时间范围内完成,建立了时间窗约束下的任务分配模型;设计了简洁的任务分配编码方式;提出的混合DPSO-GT算法综合了离散粒子群算法和郭涛算法的优势,同时又较好地克服它们的缺点。以多无人机编队执行对敌监视侦察任务为例,说明了该算法的有效性。考虑无人机航路可飞性约束以及任务的时序性约束将是课题进一步讨论的内容。

参考文献 (References)

- [1] 沈林成,牛轶峰,朱华勇.多无人机自主协同控制理论与方法[M].北京:国防工业出版社,2013:1-20.
SHEN Lincheng, NIU Yifeng, ZHU Huayong. Theories and methods of autonomous cooperative control for multiple UAVs[M]. Beijing: National Defense Industry Press, 2013: 1-20. (in Chinese)
- [2] Ryan J L, Bailey T G, Moore J T, et al. Reactive tabu search in unmanned aerial reconnaissance simulations [C]//Proceedings of Winter Simulation Conference, 1998, 1: 873-879.
- [3] Weinstein A L, Schumacher C. UAV scheduling via the vehicle routing problem with time windows (Preprint)[R]. AFRL-VA-WP-2007-306, 2007.
- [4] 霍霄华,沈林成.多UCAV协同控制中的任务调度问题研究[J].系统仿真学报,2007,19(16):3623-3626.
HUO Xiaohua, SHEN Lincheng. Task scheduling in multi-UCAV cooperative control [J]. Journal of System Simulation, 2007, 19(16): 3623-3626. (in Chinese)
- [5] Ponda S, Redding J, Choi H L, et al. Decentralized planning for complex missions with dynamic communication constraints[C]//Proceedings of 2010 American Control Conference, Baltimore, MD: AACC, 2010:3998-4003.
- [6] 杜继永,张凤鸣,杨骥,等.多UCAV协同任务分配模型及粒子群算法求解[J].控制与决策,2012,27(11):1751-1755.
DU Jiyong, ZHANG Fengming, YANG Ji, et al. Cooperative task assignment for multiple UCAV using particle swarm optimization [J]. Control and Decision, 2012, 27(11): 1751-1755. (in Chinese)
- [7] 郇斌,周锐,丁全心.多无人机分布式协同异构任务分配[J].控制与决策,2013,28(2):274-278.
DI Bin, ZHOU Rui, DING Quanxin. Distributed coordinated heterogeneous task allocation for unmanned aerial vehicles[J]. Control and Decision, 2013, 28(2): 274-278. (in Chinese)
- [8] 王婷,符小卫,高晓光.基于改进遗传算法的异构多无人机任务分配[J].火力与指挥控制,2013,38(5):37-41.
WANG Ting, FU Xiaowei, GAO Xiaoguang. Cooperative task assignment for improved genetic algorithm [J]. Fire Control & Command Control, 2013, 38(5): 37-41. (in Chinese)
- [9] Kennedy J, Eberhartr C. Particle swarm optimization [C]//Proceedings of IEEE International Conference on Neural Network, USA: IEEE, 1995:1942-1948.
- [10] 郑东亮,薛云灿,杨启文,等.基于Inver-Over算子的改进离散粒子群优化算法[J].模式识别与人工智能,2010,23(1):97-102.
ZHENG Dongliang, XUE Yuncan, YANG Qiwen, et al. Modified discrete particle swarm optimization algorithm based on Inver-Over operator[J]. Pattern Recognition and Artificial Intelligence, 2010, 23(1): 97-102. (in Chinese)
- [11] Tao G, Michalewicz Z. Inver-over operator for the TSP[C]//Proceedings of the 5th International Conference on Parallel Problem Solving from Nature, 1998:803-812.
- [12] 闭应洲,丁立新,杨小雄.快速倒序算子的研究[J].计算机工程与应用,2009,45(4):45-47.
BI Yingzhou, DING Lixin, YANG Xiaoxiong. On faster inversion operator [J]. Computer Engineering and Applications, 2009, 45(4):45-47. (in Chinese)
- [13] Wang Y T, Sun J, Li J Q, et al. A modified Inver-Over operator for the traveling salesman problem[C]//Proceedings of 7th International Conference on Intelligent Computing, 2012:17-23.
- [14] Peng Y, Zhu H Y. Research on vehicle routing problem with stochastic demand and PSO-DP algorithm with Inver-Over operator [J]. Systems Engineering-Theory & Practice, 2008, 28(10): 76-81.
- [15] 安昌,徐森.一种结合粒子群优化理论改进的郭涛算法及其应用[J].计算机应用与软件,2014,31(2):296-299,320.
AN Jin, XU Seng. A PSO theory integrated improved GUO TAO algorithm and its application [J]. Computer Applications and Software, 2014, 31(2): 296-299, 320. (in Chinese)