

控制与决策

Control and Decision

基于改进约束差分进化算法的动态航迹规划

吴文海, 郭晓峰, 周思羽

引用本文:

吴文海, 郭晓峰, 周思羽. 基于改进约束差分进化算法的动态航迹规划[J]. *控制与决策*, 2020, 35(10): 2381–2390.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.1732>

您可能感兴趣的其他文章

Articles you may be interested in

求解电力系统经济调度问题的改进粒子群优化算法

Improved particle swarm optimization algorithm for solving power system economic dispatch problem

控制与决策. 2020, 35(8): 1813–1822 <https://doi.org/10.13195/j.kzyjc.2018.1490>

基于定向A*算法的多无人机同时集结分步策略

Serial strategy for rendezvous of multiple UAVS based on directional A*algorithm

控制与决策. 2019, 34(6): 1169–1177 <https://doi.org/10.13195/j.kzyjc.2017.1636>

基于参数动态调整的多目标差分进化算法

Adaptive multi-objective differential evolution algorithm based on the dynamic parameters adjustment

控制与决策. 2017, 32(11): 1985–1990 <https://doi.org/10.13195/j.kzyjc.2016.1250>

基于最优高斯随机游走和个体筛选策略的差分进化算法

Differential evolution based on optimal Gaussian random walk and individual selection strategies

控制与决策. 2016, 31(8): 1379–1386 <https://doi.org/10.13195/j.kzyjc.2015.0779>

改进差分进化算法在可靠性冗余分配问题中的应用

Improved differential evolution algorithm for solving reliability redundancy allocation problem

控制与决策. 2015, 30(5): 917–922 <https://doi.org/10.13195/j.kzyjc.2014.0647>

基于改进约束差分进化算法的动态航迹规划

吴文海[†], 郭晓峰, 周思羽

(海军航空大学青岛校区 航空仪电控制工程与指挥系, 山东 青岛 266041)

摘要: 为解决三维复杂环境下无人机动态航迹规划问题, 提出一种基于改进约束差分进化算法的动态航迹规划方法, 以满足对实时性及动态搜索精度的要求. 首先, 根据无人机航迹规划特点将其描述为包括飞行约束及威胁约束在内的约束优化问题, 并构造目标代价函数和约束限制函数; 其次, 将广义反向学习和自适应排序变异操作引入到约束差分进化算法中, 以提高算法的多样性、收敛速度和寻优精度; 最后, 利用自适应权衡模型对各状态下的约束限制进行处理, 充分利用“精英”个体信息, 实现对目标适应值的合理转换. 通过仿真实验以及与 3 种先进约束差分进化算法比较表明: 所提方法能够有效实现静态及动态威胁回避, 规划出安全适航的飞行路径, 实现地形跟随; 相较于其他 3 种算法, 所提方法具有寻优性能好、鲁棒性强、收敛速度快和可靠性高等优势.

关键词: 动态航迹规划; 地形跟随; 威胁回避; 约束差分进化算法

中图分类号: TP301.6

文献标志码: A

DOI: 10.13195/j.kzyjc.2018.1732

开放科学(资源服务)标识码(OSID):



引用格式: 吴文海, 郭晓峰, 周思羽. 基于改进约束差分进化算法的动态航迹规划[J]. 控制与决策, 2020, 35(10): 2381-2390.

Dynamic route planning based on improved constrained differential evolution algorithm

WU Wen-hai[†], GUO Xiao-feng, ZHOU Si-yu

(Department of Aviation Control and Command, Qingdao Branch of Naval Aeronautics University, Qingdao 266041, China)

Abstract: Aiming at the problem of three-dimensional dynamic route planning for unmanned aerial vehicles (UAV), an improved constrained differential evolution (CDE) algorithm is proposed so as to meet the requirements of instantaneity and dynamic search accuracy. Firstly, this paper formulates the route planning problem for the UAV as a constrained optimization problem and constructs the objective functions and constraint functions according to the constraints of flight and threats. Then, the diversity, convergence and accuracy of the algorithm are improved through introducing generalized opposition-based learning and adaptive ranking mutation operators into the CDE algorithm. Finally, the adaptive trade-off model is applied to handle the constraints in each state, and the information of elite individual is fully utilized to achieve a reasonable conversion of the fitness. Experiment results and comparisons with 3 state-of-the-art CDEs show that the proposed method is able to plan a safe flight route which can implement static and dynamic threat avoidance effectively and realize terrain following. Compared with other three algorithms, the presented method has the advantages of terrific optimization performance, strong robustness, good convergence and high reliability.

Keywords: dynamic route planning; terrain following; threat avoidance; constrained differential evolution

0 引言

随着任务环境日益恶劣, 如何为无人机实时规划理想的飞行航迹成为研究热点. 航迹规划的主要任务是在综合考虑地形环境、威胁因素及自身性能等约束的前提下, 从起点到目标点为无人机规划出一条最优航路, 以成功规避任务区域内静态和动态威胁,

并使能量消耗、时间及威胁代价等成本最小.

无人机任务区域地形环境通常十分复杂, 在三维环境下还需考虑其低空突防时满足地形跟随能力的要求, 对地形解算能力要求较高, 计算量较二维平面规划复杂许多; 其次, 在规划过程中应考虑无人机自身飞行性能约束的特殊性, 满足转向角、最大爬升和

收稿日期: 2018-12-19; 修回日期: 2019-04-14.

基金项目: 国家重点研发计划项目(2018YFC0806900, 2016YFC0800606, 2016YFC0800310).

责任编辑: 李少远.

[†]通讯作者. E-mail: hkdx_2017@126.com.

下降率等多种飞行限制,生成平滑适航的航迹路线;最后,还要考虑任务区域内各种静态及动态威胁,实现对静态威胁合理规避,完成动态威胁情况下的实时规划与重规划.总之,无人机航迹规划问题是一种考虑多约束条件下的复杂优化问题,同时还要防范优化运算“指数爆炸”,保证优化的高效性和实时性.

传统方法在航迹规划方面取得了许多成果, Franco等^[1]受到松弛技术的启发,提出了一种新的路径规划方法,利用二次规划对松弛进行局部运动,使得在短时间内生成具有较低约束违反度的路径. Matiussi等^[2]利用一致理论结合模型预测控制框架并将顺序凸规划嵌入模型中,解决了非凸约束的调和问题,实现了无碰撞轨迹优化. Zhang等^[3]提出了一种具有收敛性分析的顺序凸规划算法以解决无人机航迹规划问题,将航迹规划建模为非线性最优控制问题,通过凹凸过程逼近顺序凸规划,最终实现稳定求解.虽然传统方法能够实现静态威胁环境的航迹规划,但随着任务区域扩大、环境复杂程度增加、威胁因素变多,求解最优航迹的计算量呈指数增长,难以实现实时精确规划^[4].智能算法在处理优化问题时具有结构简单、搜索效率高、鲁棒性强的优点^[5],因此被广泛应用于解决复杂环境下的航迹规划问题^[6].其中, Chen等^[7]将粒子群算法(PSO)与遗传算法(GA)的变异思想应用于中心引力算法(CFO)中,实现对短距垂直起降旋翼飞机的三维航迹规划. Roberge等^[8]将PSO算法与GA算法性能在复杂三维环境下进行比较,并采用多目标代价函数对最优航迹评估,实现航迹的实时规划. Zhang等^[9]提出了一种改进的蚁群算法,抽象出地形和障碍物为优化提供数据基础,并采用蚁群算法的自然并发性加速计算. Pandey等^[10]提出了一种基于离散萤火虫群优化算法的新算法,生成从原点到目标点的可行路径,并与沿途障碍物保持安全距离.此外,还有人工蜂群算法(ABC)^[11]、灰狼优化算法(GWO)^[12]等.

差分进化算法(DE)结构简单、寻优性能优、鲁棒性强、易于工程实现,已广泛应用于各类实际优化问题中^[13].DE算法采用“贪婪竞争”的种群全局寻优策略,基于差分变异和一对一竞争生存,有效削弱遗传操作复杂性.同时,DE算法能够根据当前寻优状态和特有的“记忆”能力,利用“精英”个体信息,动态调整搜索策略,降低算法评价次数,从而大大提高算法寻优效率和精确性.其次,DE算法结构简单、参量少,特别适用于航迹规划此类复杂性、特殊性以及实

时性强的优化问题,具有较强的可行性.国内外大量学者对基于DE算法的航迹规划问题进行研究: Zhou等^[14]利用混沌理论提出一种改进DE算法,在同时考虑最短飞行路径和最大安全范围的基础上,规划出三维航迹; Li等^[15]将多种群DE算法应用到无人机航迹规划问题中,采用两种新型操作方法实现种群变异,提高了搜索效率; Zhang等^[16]提出了一种新型变异策略以权衡DE算法航迹规划的开发与探索能力,并采用B样曲线对航迹进行平滑处理.尽管上述算法都能够实现较优的航迹规划任务,却无法飞行过程中对地形跟随及动态威胁回避实时性和高效性要求.

针对上述不足,本文提出一种基于广义反向学习自适应约束差分进化算法(GOBL-ACDE)的三维航迹动态规划方法.算法引入广义反向学习机制,在开始阶段执行基于广义反向学习的种群初始化操作,每代进化结束后广义采用基于反向学习“代跳”操作,生成反向种群.依概率论可知,反向种群等概率优于原种群适应值^[17],因此将GOBL框架与DE算法相结合,在不增加算法复杂度以及保持原有搜索空间的基础上,综合利用反向与原始两个种群信息,有效地提高算法的寻优性能,降低搜索难度,提高搜索效率,增加算法寻得最优解的概率.自适应排序变异操作适用于算法变异和交叉阶段,依据排序值差异,采用余弦和反余弦模型解算个体选择概率,充分利用“精英”个体支配优势和种群多样性信息;考虑到其他算法搜索策略单一不变的不足,按照可行比率自适应采取“improved rand-to-best and current/2”变异策略并实时更新变异因子和交叉因子,实现种群进化,提高算法寻优效率及实时性.自适应权衡模型应用于算法的选择阶段,在约束优化问题中不能直接将目标函数值等价于最终的适应值,需充分考虑约束条件对目标函数值的影响,为了使算法能够更加合理,实现对静态威胁和动态威胁回避,只有对飞行约束和威胁约束进行处理,获得在不同约束影响下相应的适应值,才能获得最准确的寻优结果,以此提高算法寻优精度,加速算法寻优效率.因此,广义反向学习机制、自适应排序变异操作和自适应权衡模型对航迹规划任务具有极强的必要性和可行性.

本文将GOBL-ACDE算法与3种先进约束差分进化算法进行比较,验证了GOBL-ACDE算法在航迹规划问题中良好的寻优性能及地形跟随/威胁回避能力,并设置具有两处动态威胁的任务区域,验证GOBL-ACDE算法动态规划的实时性和可行性.

1 航迹规划建模

通常,三维动态航迹规划是指在任务区域 $M = \{m = (x, y, z)\}$ 内,依据载机任务综合信息,在满足飞行约束和威胁约束的前提下,计算获得下一最优航路点 $p_i = (x_i, y_i, z_i)$,最终使整个飞行航路代价值最小,其优化问题可描述为

$$\begin{aligned} \min J &= J_L + J_H. \\ \text{s.t. } g_i &\leq 0, \quad i = 1, 2, \dots, m; \\ h_j &= 0, \quad j = m + 1, \dots, n. \end{aligned} \quad (1)$$

其中: J_L 、 J_H 分别为计算候选航路距离和高度的代价函数; g_i 、 h_j 分别为飞行过程中各等式约束和不等式约束条件,包括飞行约束条件及威胁约束条件.

1.1 目标代价函数

1.1.1 距离代价函数

距离代价函数 J_L 用来最小化当前位置与目标点的距离,规划出最短飞行路径,可表示为

$$J_L = \sum_{i=0}^{N-1} l_i. \quad (2)$$

其中: l_i 为第 i 段航迹的距离,有

$$l_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}, \quad (3)$$

i 为第 i 个航路点 p_i , x_i 、 y_i 、 z_i 分别为对应侧向位置坐标和飞行高度.

1.1.2 高度代价函数

飞机处于较低飞行高度时,可充分利用地形优势隐蔽飞行以此规避未知雷达等威胁因素.高度代价函数 J_H 整合航路高度,规划出较低高度飞行路径,可表示为

$$J_H = \int_p H_p dl. \quad (4)$$

其中 H_p 为

$$H_p = \begin{cases} 0, & z_p < 0; \\ z_p, & \text{otherwise.} \end{cases} \quad (5)$$

1.2 飞行约束条件

1.2.1 转向角约束

为了保证航迹平滑可行,在每一航路点的最大转向角^[18]可表示为

$$\varphi_{\max} = \frac{n_{\max} g}{V} \cdot \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}. \quad (6)$$

其中: n_{\max} 为最大横向过载, g 为重力加速度, V 为飞行速度,转向角约束为

$$g_1 = \max(\varphi_i - \varphi_{\max}) \leq 0. \quad (7)$$

1.2.2 爬升/下降约束

航路点 p_i 处的斜率 s_i 由最大爬升斜率 α_i 和最小下降斜率 β_i 约束^[19],可表示为

$$s_i = \frac{z_{i+1} - z_i}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}, \quad (8)$$

$$\alpha_i = -1.5377 \times 10^{-10} z_i^2 - 2.6997 \times 10^{-5} z_i + 0.4211, \quad (9)$$

$$\beta_i = -2.5063 \times 10^{-9} z_i^2 - 6.3014 \times 10^{-6} z_i - 0.3257. \quad (10)$$

爬升和下降约束分别为

$$g_2 = \max(s_i - \alpha_i) \leq 0, \quad (11)$$

$$g_3 = \max(\beta_i - s_i) \leq 0. \quad (12)$$

1.2.3 地形约束

为了保证飞行安全,最低飞行高度应高于预设飞行安全高度,地形约束可表示为

$$g_4 = H_{\text{safe}} - \min(z_i - H_{\text{ter}}(x_i, y_i)) \leq 0. \quad (13)$$

其中: H_{safe} 为预设飞行安全高度, $H_{\text{ter}}(x_i, y_i)$ 为 (x_i, y_i) 处地形高度.

1.3 威胁约束条件

1.3.1 防空威胁约束

防空威胁约束是为了避免在飞行过程中进入到地面防空雷达和防空武器的威胁范围内,其约束代价由每段航迹的长度和威胁概率决定,可表示为

$$h_1 = \sum_{i=1}^{N-1} \left(l_i \sum_{j=1}^{n_T} P_{j,i} \right) = 0, \quad (14)$$

其中 $P_{j,i}$ 为第 j 个威胁在第 i 航迹段中点处的威胁概率,本文包括防空雷达、防空导弹和防空火炮3类威胁,为了简化计算忽略高度影响.

防空雷达的发现概率 P_R 由飞机雷达横截面 RCS 及其与雷达之间的距离 d 决定^[19],可表示为

$$P_R = \begin{cases} 0, & d > R_{R\max}; \\ \frac{1}{1 + \zeta_2 (d^4 / \text{RCS})^{\zeta_1}}, & \text{otherwise.} \end{cases} \quad (15)$$

其中: ζ_1 、 ζ_2 为雷达固有参数, $R_{R\max}$ 为雷达最大探测距离.

防空导弹的简化毁伤概率 P_M ^[18]可表示为

$$P_M = \begin{cases} 0, & d > R_{M\max}; \\ \frac{R_{M\max}^4}{d^4 + R_{M\max}^4}, & \text{otherwise.} \end{cases} \quad (16)$$

其中 $R_{M\max}$ 为在确定高度下的最大毁伤距离.

防空火炮简化毁伤概率 P_G ^[18]可表示为

$$P_G = \begin{cases} 1, & d < R_G; \\ \frac{1}{d}, & R_G \leq d < R_{M\max}; \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

1.3.2 禁飞区约束

真实环境中由于各种因素存在禁止飞行区域,例如高威胁地区、未知区域以及气候恶劣区域等^[20],本文设置相应禁飞区,以立方体模型表示,禁飞区约束可表示为

$$h_2 = L_{\text{InNFZ}} = 0, \quad (18)$$

其中 L_{InNFZ} 为航迹穿越禁飞区的长度.

2 基于GOBL-ACDE算法的三维航迹规划

2.1 标准差分进化算法

DE通过变异、交叉、选择等过程产生较优后代,实现进化^[13].

变异过程:父辈个体 $x_{i,t}$ 经变异生成新的变异向量 $v_{i,t}$,变异策略包括:

1) “DE/rand/1”:

$$v_{i,t} = x_{r1,t} + F(x_{r2,t} - x_{r3,t}), \quad (19)$$

2) “DE/best/1”:

$$v_{i,t} = x_{\text{best},t} + F(x_{r2,t} - x_{r3,t}). \quad (20)$$

其中: $x_{r1,t}$ 、 $x_{r2,t}$ 、 $x_{r3,t}$ 为互异随机个体, $x_{\text{best},t}$ 为最优个体,变异因子 $F \in (0, 1)$.

交叉过程:父辈个体 $x_{i,t}$ 与变异向量 $v_{i,t}$ 经交叉生成实验向量 $u_{i,t}$,具体如下:

$$u_{ji,t} = \begin{cases} v_{ji,t}, & \text{rand}_j \leq \text{CR or } j = j_{\text{rand}}; \\ x_{ji,t}, & \text{otherwise.} \end{cases} \quad (21)$$

其中: j_{rand} 为控制参数以使实验向量 $u_{i,t}$ 异于父辈个体 $x_{i,t}$,交叉因子 $\text{CR} \in (0, 1)$.

选择过程:将父辈个体 $x_{i,t}$ 与实验向量 $u_{i,t}$ 适应值对比,保留较优结果到下一代进化中,有

$$x_{i,t+1} = \begin{cases} u_{i,t}, & f(u_{i,t}) \leq f(x_{i,t}); \\ x_{i,t}, & \text{otherwise.} \end{cases} \quad (22)$$

2.2 GOBL-ACDE算法

2.2.1 约束优化问题

约束优化问题一般可定义为

$$\begin{aligned} \min & f(x), x \in D. \\ \text{s.t.} & \begin{cases} g_j(x) \leq 0, & j = 1, 2, \dots, m; \\ h_j(x) = 0, & j = m + 1, \dots, n. \end{cases} \end{aligned} \quad (23)$$

其中: $x = (x_1, x_2, \dots, x_n)$ 为决策向量, $f(x)$ 为目标函数, $g_j(x)$ 、 $h_j(x)$ 分别为第 j 个不等式和等式约束.

处理约束优化问题时,通常将等式替换为不等式约束,有

$$|h_j(x)| - \delta \leq 0, \quad j = m + 1, \dots, n, \quad (24)$$

其中 δ 为正容忍值,大小为 0.000 1. 解 x 与第 j 个约束距离由下式确定:

$$G_j(x) = \begin{cases} \max\{0, g_j(x)\}, & 1 \leq j \leq l; \\ \max\{0, |h_j(x)| - \delta\}, & l + 1 \leq j \leq m. \end{cases} \quad (25)$$

则其约束违反程度为

$$G(x) = \sum_{j=1}^p G_j(x). \quad (26)$$

2.2.2 广义反向学习

广义反向学习(GOBL)定义^[21]如下: n 维空间中,若 $\check{x}_i = (\check{x}_{i,1}, \check{x}_{i,2}, \dots, \check{x}_{i,n})$ 为 $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$ 的反向个体,则 $\check{x}_i = (\check{x}_{i,1}, \check{x}_{i,2}, \dots, \check{x}_{i,n})$ 可表示为

$$\check{x}_{i,j} = k(a_j + b_j) - x_{i,j}. \quad (27)$$

若反向个体 $\check{x}_{i,j}$ 跳出动态搜索范围,即 $\check{x}_{i,j} \notin [a_j, b_j]$,则在 $[a_j, b_j]$ 内随机生成反向解 $\check{x}_{i,j}$,有

$$\check{x}_{i,j} = \text{rand}(a_j, b_j). \quad (28)$$

2.2.3 自适应权衡模型

自适应权衡模型^[22]根据可行个体比率将种群分为3种不同状态,通过对应处理机制获得约束条件下的变换适应值.不可行状态下,种群仅存在不可行个体,约束处理机制仅考虑约束违反程度以代替适应值,有

$$f_{\text{fitness}}(x_i) = G(x_i). \quad (29)$$

半可行状态下,约束处理机制的主要任务是根据种群中可行个体所占比率,在目标函数值与约束违反程度之间寻求平衡^[23].

约束处理机制半可行状态下的个体分为 Z_1 和 Z_2 两个集合,有

$$\begin{aligned} Z_1 &= \{i | G(x_i) = 0, i = 1, 2, \dots, \text{NP}\}, \\ Z_2 &= \{i | G(x_i) > 0, i = 1, 2, \dots, \text{NP}\}, \end{aligned} \quad (30)$$

其中 NP 为种群规模.不同个体 x_i 的目标函数值 $f(x_i)$ 为

$$f'(x_i) = \begin{cases} f(x_i), & i \in Z_1; \\ \max\{\varphi f(x_{\text{best}}) + (1 - \varphi)f(x_{\text{worst}}) f(x_i)\}, & i \in Z_2. \end{cases} \quad (31)$$

其中: φ 为可行个体比率, x_{best} 和 x_{worst} 为 Z_1 中最优和最差个体.

对式(31)进行标准化转换,有

$$f_{\text{nor}}(x_i) = \frac{f'(x_i) - \min_{j \in Z_1 \cup Z_2} f'(x_j)}{\max_{j \in Z_1 \cup Z_2} f'(x_j) - \min_{j \in Z_1 \cup Z_2} f'(x_j)}. \quad (32)$$

对违约值进行相同标准化转换,有

$$G_{\text{nor}}(x_i) = \begin{cases} 0, & i \in Z_1; \\ \frac{G(x_i) - \min_{j \in Z_2} G(x_j)}{\max_{j \in Z_2} G(x_j) - \min_{j \in Z_2} G(x_j)}, & i \in Z_2. \end{cases} \quad (33)$$

因此,半可行状态下转换适应值为

$$f_{\text{fitness}}(x_i) = f_{\text{nor}}(x_i) + G_{\text{nor}}(x_i). \quad (34)$$

可行状态下,种群仅存在可行个体,因此个体适应值即为目标函数值,有

$$f_{\text{fitness}}(x_i) = f(x_i). \quad (35)$$

2.2.4 自适应权衡模型

改进自适应排序变异操作依据个体适应值大小对种群进行排序^[24],排序值可表示为

$$R_i = NP + 1 - i, \quad i = 1, 2, \dots, NP, \quad (36)$$

其中 i 为该个体在排序中的编号.

根据种群所处的不同状态,个体选择概率应分别进行计算.不可行状态选择概率为

$$p_i = 0.5 \cdot \left[1 - \cos \left(\frac{R_i}{NP} \cdot \pi \right) \right]. \quad (37)$$

半可行状态选择概率为

$$p_i = 0.5 \cdot \left[1 - \cos \left(\frac{R_i}{NP} \cdot \pi \right) \right]. \quad (38)$$

可行状态选择概率为

$$p_i = \frac{\arccos \left(1 - 2 \cdot \frac{R_i}{NP} \right)}{\pi}. \quad (39)$$

采用余弦模型增加“精英”个体被选择的概率,加速种群寻得可行域,采用反余弦模型以提高种群多样性,避免算法早熟.变异与交叉操作过程中,分别采用“rand-to-best and current/2”和“rand-to-current/2”两种变异策略,并选取自适应变异因子与交叉因子^[25],具体计算如下:

$$F_Z = F_{r1,G} + N(0, 0.5)(F_{r2,G} - F_{r3,G}), \quad (40)$$

$$\text{Cr}_Z = \text{Cr}_{r1,G} + N(0, 0.5)(\text{Cr}_{r2,G} - \text{Cr}_{r3,G}), \quad (41)$$

其中 $F_{i,G}$ 和 $\text{Cr}_{i,G}$ 分别为服从 $N(0, 0.15)$ 的随机数.

2.3 GOBL-ACDE 算法

基于 GOBL-ACDE 算法的航迹规划系统结构如图 1 所示.航迹规划具体步骤如下.

step 1: 加载任务参数和环境信息,设置种群规模,最大进化代数,根据式(27)获得广义反向学习初始化种群 P .

step 2: 由式(25)和(26)计算个体 x_i 约束违反程度和可行个体所占比率 φ ,判断种群所处状态.

step 3: 由式(40)和(41)更新变异因子 F_Z 和交叉因子 Cr_Z ,采用改进自适应排序变异操作对每一个体 x_i 生成变异向量 v_i ,根据交叉因子生成实验向量 u_i .

step 4: 由式(29)、(34)和(35)计算个体 x_i 与实验向量 u_i 适应值,选择较优个体保留至下一代进化,更新最优个体 x_{best} 和最小代价值 $f(x_{\text{best}})$.

step 5: 判断是否达到最大进化代数,若达到,则输出最优结果,若未达到,则返回 step 2.

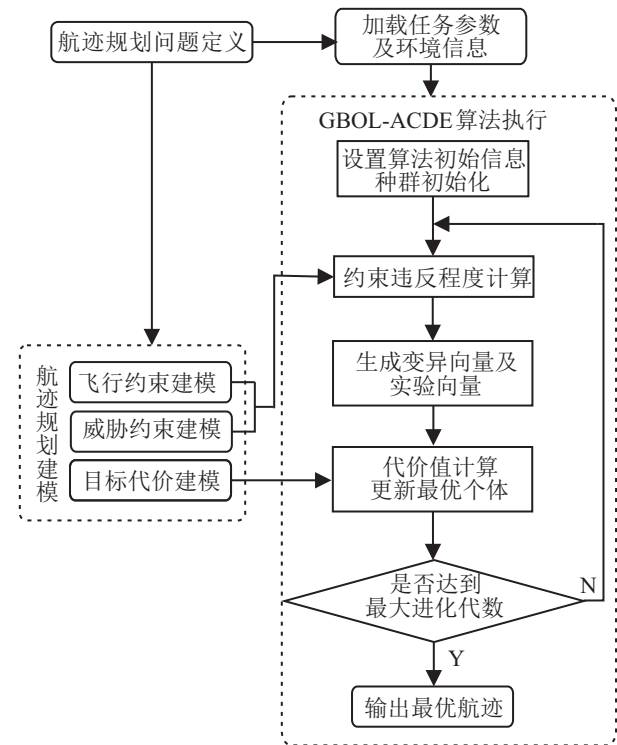


图 1 基于 GOBL-ACDE 算法的航迹规划系统结构

2.4 基于 GOBL-ACDE 算法的航迹规划伪代码

算法 1 基于 GOBL-ACDE 算法的航迹规划.

任务信息: 任务空间 R , 起点 x_s , 目标点 (x_T, y_T) , 威胁类型, 威胁中心 $(x_{\text{threat}_j}, y_{\text{threat}_j})$, 威胁半径 R_j , 飞机性能参数;

输入: 目标函数 f , 约束函数 G , 最大进化代数 G_{max} , 种群规模 NP ;

输出: 最优解和最优航迹.

$t = 1$;

随机化初始种群 $P_0 = (x_1, x_2, \dots, x_{NP})$

for $i = 1$ to NP do

for $j = 1$ to n do

$\text{GOP}_{ji,0} = k(a_j + b_j) - P_{ji,0}$;

```

if  $GOP_{ji,0} < a_j \parallel GOP_{ji,0} > b_j$  then
     $GOP_{ji,0} = \text{rand}(a_j, b_j)$ ;
end if
end for
end for
选择  $GOP_0 \cup P_0$  中 NP 个“精英”个体组成  $P$ 
while  $t < G_{\max}$  do
    for  $i = 1$  to NP do
        计算选择概率  $p_i$ 
        for  $j = 1$  to  $n$  do
            if  $\text{rand}(0, 1) < \varphi$  then
                 $v_{i,t} = x_{1,t} + Fz((x_{r2,t} - x_{i,t}) + (x_{r3,t} - x_{r4,t}))$ 
            else
                 $v_{i,t} = x_{1,t} + Fz((x_{r\text{best},t} - x_{r2,t}) + (x_{r3,t} - x_{i,t}))$ 
            end if
            if  $\text{rand}(0, 1) < Cr_{i,j}$  or  $j = j_{\text{rand}}$  then
                 $u_{i,j} = v_{i,j}$ 
            else
                 $u_{i,j} = x_{i,j}$ 
            end if
        end for
    end for
    for  $i = 1$  to NP do
        if  $f(u_i) \leq f(x_i)$  then
             $x_{i,t+1} = u_i$ 
        else
             $x_{i,t+1} = x_{i,t}$ 
        end if
    end for
     $t = t + 1$ 
end while

```

3 基于GOBL-ACDE算法的三维航迹规划

为了评估GOBL-ACDE算法对求解约束条件下航迹规划问题的有效性,设置两组仿真实验.在静态威胁情况下,将GOBL-ACDE算法与CDE^[26]、DDE^[27]和 ϵ DE^[28]3种算法性能进行比较;在动态威胁情况下,对GOBL-ACDE算法动态寻优性能进行评估.实验在Inter Core i5-7500 CPU(3.4 GHz)、8 GB内存个人计算机上进行,采用Matlab 2016a编写.

3.1 静态威胁

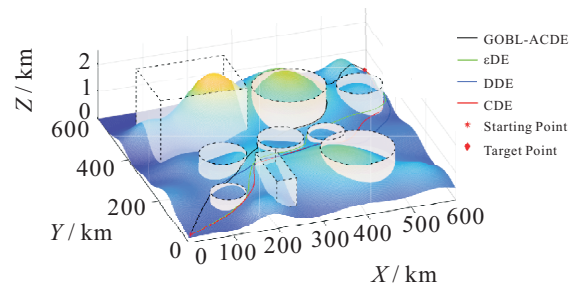
设置任务区域 $M = [0, 600] \times [0, 600] \times [0, 2.5]$ km,其中存在9处已知静态威胁,各威胁中心点坐标及最大威胁半径如表1所示.起点和目标点的X轴和Y轴坐标分别为(5,5)和(595,595),最小飞行高度

$H_{\text{Lsafe}} = 0.05$ km,最大飞行高度 $H_{\text{Usafe}} = 2.5$ km,飞行速度 200 m/s,最大横向过载 $n_{\max} = 5g$.4种算法设置相同种群规模 $NP = 70$ 和最大进化代数 $G_{\max} = 200$,分别独立运行 50 次,其余参数设置与原文献相同.

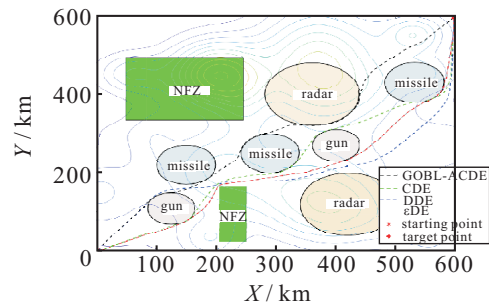
表1 威胁参数

威胁类型	坐标/km	最大威胁半径/km
防空火炮1	(125, 110)	40
防空火炮2	(400, 270)	40
防空导弹1	(290, 250)	50
防空导弹2	(150, 220)	50
防空导弹3	(530, 430)	50
防空雷达1	(360, 400)	80
防空雷达2	(420, 120)	80
禁飞区1	(50, 335)	$L = 195, W = 160$
禁飞区2	(205, 25)	$L = 45, W = 140$

在50次独立实验中,所有4种算法均能规划出可行飞行航线.图2为4种约束差分算法在50次仿真实验中最优航迹的三维规划图和二维规划图,图2(a)中粉色圆柱体代表防空火炮,蓝色圆柱体代表防空导弹,黄色圆柱体代表防空雷达,白色立方体代表禁飞区.



(a) 最优航迹三维规划图



(b) 最优航迹二维规划图

图2 静态威胁规划航迹

由图2可见,4种算法的最优航迹均能够有效规避飞行区域中的静态威胁,获得可行航迹.由图2(b)可知,GOBL-ACDE算法获得的航迹在飞行距离上最小,具有最小的航迹综合代价,仅为2008.6,远远小于其余3种算法规划的航迹, ϵ DE算法次之,DDE算法表现最差,为了回避威胁而选择牺牲航迹综合代价,

选择较远的路线.

表 2 为 GOBL-ACDE、CDE、DDE 和 ϵ DE 四种算法在 50 次独立实验中成功获得可行航迹时, 航迹综合代价统计结果比较, 包括最优值、最差值、平均值、中位值、标准差以及成功率, 黑体表示较优结果.

表 2 不同算法航迹综合代价统计结果比较

算法	最优值	最差值	平均值	中位值	标准差	成功率/%
GOBL-ACDE	2008.6	2073.2	2042.6	2032.5	16.021 1	100
CDE	2308.3	2493.6	2403.9	2405.6	57.267 8	94
DDE	2398.4	2615.2	2410.2	2446.5	59.974 7	88
ϵ DE	2082.9	2176.3	2131.6	2135.5	28.978 4	100

由表 2 可见, GOBL-ACDE 算法在航迹代价的最优值、最差值、平均值和中位值均为最优, 其余 3 种算法的最优值甚至劣于 GOBL-ACDE 算法的最差值, 表明 GOBL-ACDE 算法具有更好的优化性能. 其次, GOBL-ACDE 算法的标准差仅为 16.021 1, 优于其他算法, 表明 GOBL-ACDE 算法具有更强的鲁棒性. 在成功率方面, GOBL-ACDE 与 ϵ DE 算法同为 100%, 而 CDE 和 DDE 算法分别仅为 94% 和 88%, 表明 GOBL-ACDE 算法在约束条件下的全局寻优能力具有更高的可靠性.

图 3 为 4 种算法 50 次独立实验最优航迹代价平均收敛曲线. 由图 3 可见, GOBL-ACDE 算法执行广义反向学习种群初始化, 充分利用原始和反向两个种群信息, 在进化初期即获得较小代价值. 其次, GOBL-ACDE 算法充分利用“精英”个体排序信息, 自适应选择变异策略, 提高了算法的收敛速度. 相较于其他 3 种算法, GOBL-ACDE 算法具有更优的最佳航迹综合代价, 搜索到全局最优所需进化代数更小, 体现了其寻优的高效性.

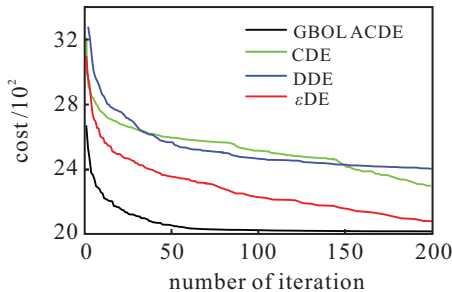
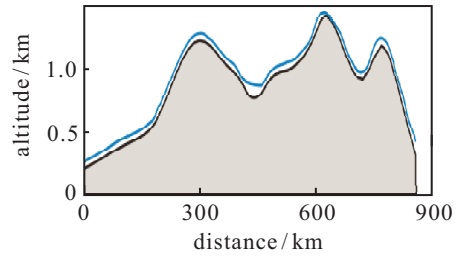


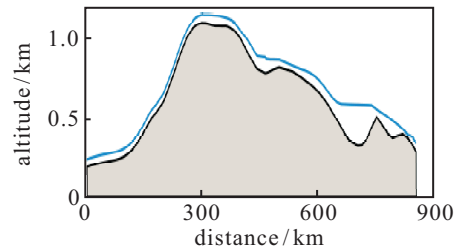
图 3 不同算法最佳航迹综合代价收敛曲线

图 4 为 4 种算法最优航迹的垂直剖面图, X 轴为距离, Y 轴为高度, 灰色部分代表地形剖面, 蓝色曲线代表垂直剖面飞行轨迹. 对比图 4 不同算法垂直剖面图可以得出: GOBL-ACDE 算法规划的航迹平滑, 具有更好的地形跟随能力, 能够充分利用地形特点进

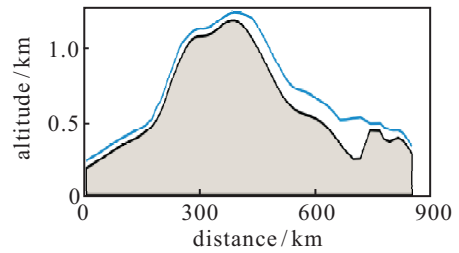
行隐蔽飞行, 在满足飞行约束的前提下, 通过减小飞行高度来提高飞行安全品质; 相较于 GOBL-ACDE 算法, CDE 与 DDE 算法所规划的航迹距地面高度较大, 未能较好地利用地形特点实现地形跟随.



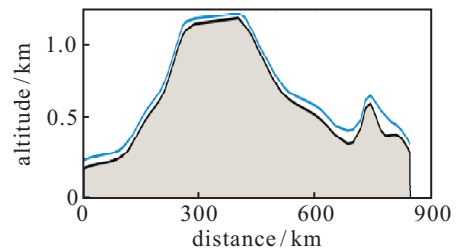
(a) GOBL-ACDE 算法垂直剖面图



(b) CDE 算法垂直剖面图



(c) DDE 算法垂直剖面图



(d) ϵ DE 算法垂直剖面图

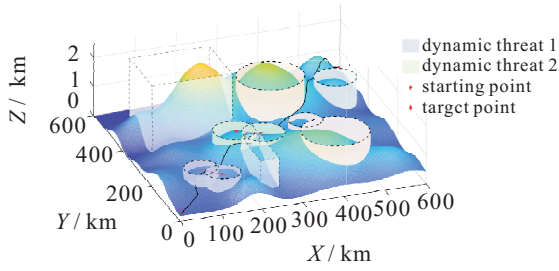
图 4 不同算法航迹垂直剖面图

3.2 动态威胁

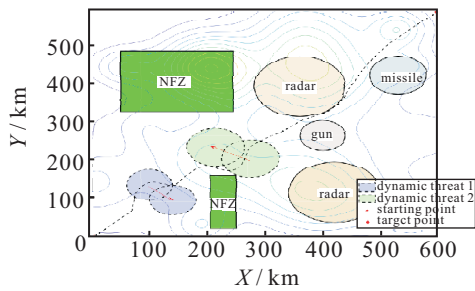
为了验证 GOBL-ACDE 算法在动态威胁环境下航迹规划效果, 在上述任务区域内设置 5 处静态威胁 (与静态威胁环境下的位置相同) 和 2 处动态威胁. 为简化实验, 假设动态威胁为圆柱体禁飞区, 第 1 处动态威胁开始点中心坐标为 (100, 140), 威胁半径为 40 km, 第 2 处动态威胁开始点中心坐标为 (275, 210), 威胁半径为 50 km. GOBL-ACDE 算法设置种群规模 $NP = 70$, 最大进化代数 $G_{max} = 200$, 独立运行 50 次.

图 5 为 GOBL-ACDE 算法在 50 次仿真实验中最

优航迹的三维规划图和二维规划图,紫色圆柱代表动态威胁1,绿色圆柱代表动态威胁2,红色箭头表示动态威胁移动方向。



(a) 最优航迹三维规划图



(b) 最优航迹二维规划图

图5 动态威胁规划航迹

由图5可见,GOBL-ACDE算法在具有2处动态威胁情况下能够规划出安全且合理的平滑航迹,最优航迹的飞行代价仅为2063.7。

图6为动态威胁下最优航迹的垂直剖面图.由图6可见,在威胁移动的情况下,GOBL-ACDE算法规划的航迹仍能较好地实现地形跟随功能,以较低高度飞行。

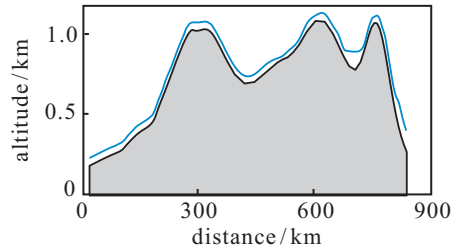
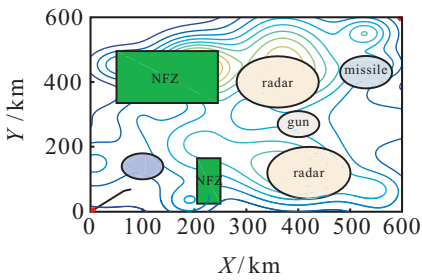
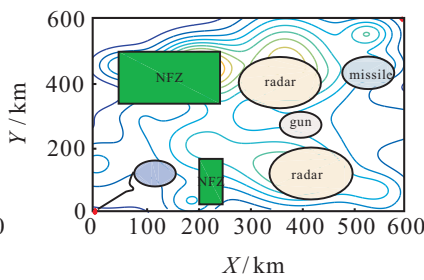


图6 动态威胁航迹垂直剖面图

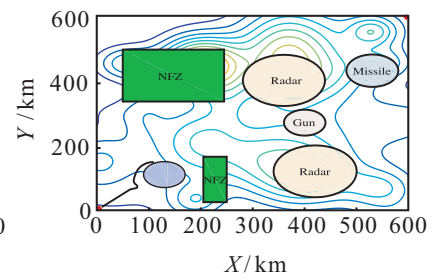
图7为动态威胁情况下分段规划航迹,其中图7(a)~图7(d)为动态威胁1移动过程中,4处不同航路点处动态威胁1位置与规划航迹的平面图,图7(e)~图7(h)为动态威胁2移动过程中,4处不同航路点处动态威胁2位置与规划航迹的平面图。



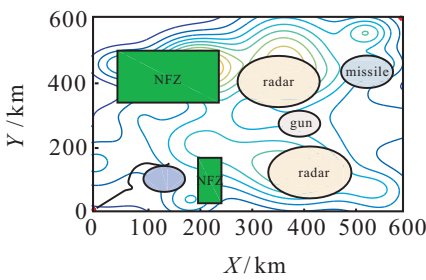
(a) 航路点1



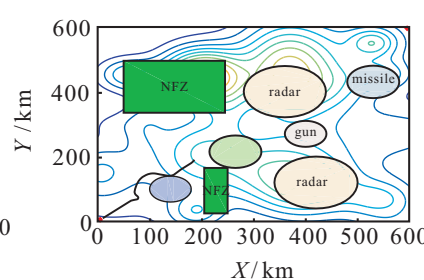
(b) 航路点2



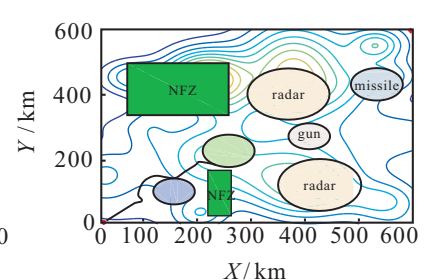
(c) 航路点3



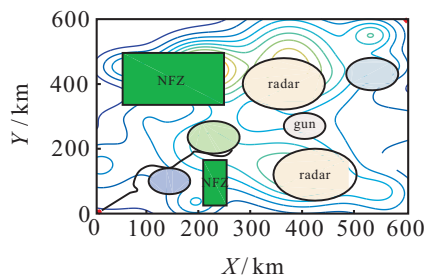
(d) 航路点4



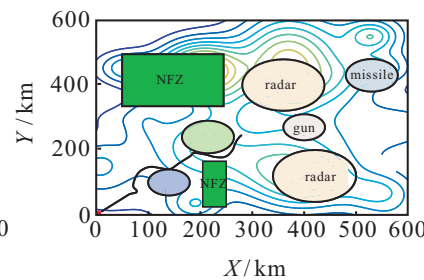
(e) 航路点5



(f) 航路点6



(g) 航路点7



(h) 航路点8

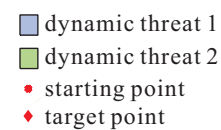


图7 动态威胁分段规划航迹

由图7(a)和图7(b)可见,在动态威胁出现初期,GOBL-ACDE算法以航迹综合代价最小为目标最小化飞行距离,在图7(a)处朝动态威胁1东南方向规划航迹,但由于动态威胁1朝东南方向移动,其移动过程中,GOBL-ACDE算法规划的航迹与其方向相反,在保证规避威胁的前提下实现较小代价,获得图7(b)处航迹。随着动态威胁1朝东南方向移动,GOBL-ACDE算法不断朝向与目标点相距最小位置进行规划,在图7(d)处到达与目标点相距最小位置,并以此朝目标点继续规划。动态威胁2出现,移动过程中与态威胁1相同,GOBL-ACDE算法在图7(h)处获得与目标点相距最小航路点,并以此继续规划。通过图7可以得出,GOBL-ACDE算法在动态威胁情况下具有较强的实时规划能力。

4 结论

本文提出的GOBL-ACDE算法通过广义反向学习和自适应排序操作提高算法寻优精度与收敛速度,避免算法早熟、陷入局部最优,采用自适应权衡模型处理约束限制带来的适应值变化。应用GOBL-ACDE算法实现三维环境下动态航迹实时规划,在充分考虑飞行约束及威胁约束条件的基础上,通过仿真实验表明,GOBL-ACDE算法在每次运行中均获得了安全、平滑、适航的理想航迹。通过与CDE、DDE和 ϵ DE算法在静态威胁实验环境下的比较表明,GOBL-ACDE算法规划的综合航迹代价最小,在寻优性能、鲁棒性、收敛性、可靠性和地形跟随能力等方面均表现出明显的优势。在动态威胁实验环境下,GOBL-ACDE算法对突发动态威胁具有较好的反应能力,能够根据威胁的动态性实时规划出地形跟随强、综合航迹代价小的理想航迹。

下一步的工作将重点研究符合现实情况的目标代价函数及约束因素模型,使问题求解更加具有现实意义。其次,本文所设均为独立威胁,今后将进一步研究在威胁联网条件下的航迹规划问题。

参考文献(References)

- [1] Franco F, Olivier K, Philippe M. Improving relaxation-based constrained path planning via quadratic programming[C]. International Conference on Intelligent Autonomous Systems. Singapore: Springer, 2018: 15-26.
- [2] Matiussi Ramalho G, Carvalho S R, Finardi E C, et al. Trajectory optimization using sequential convex programming with collision avoidance[J]. Journal of Control, Automation and Electrical Systems, 2018, 29(3): 318-327.
- [3] Zhang Z, Li J X, Wang J. Sequential convex programming for nonlinear optimal control problem in UAV path planning[J]. Aerospace Science & Technology, 2018, 76: 280-290.
- [4] Willms A R, Yang S X. An efficient dynamic system for real-time robot-path planning[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2006, 36(4): 755-766.
- [5] 李宝磊, 施心陵, 李敬敬, 等. 基于改进多元优化算法的动态路径规划[J]. 航空学报, 2015, 36(7): 2319-2328.
(Li B L, Shi X L, Li J J, et al. Dynamic path planning based on improved multivariate optimization algorithm[J]. Acta Aeronautica et Astronautica Sinica, 2015, 36(7): 2319-2328.)
- [6] Zhao Y J, Zheng Z, Yang L. Survey on computational-intelligence-based UAV path planning[J]. Knowledge-Based Systems, 2018, 158: 54-64.
- [7] Chen Y B, Yu J Q, Mei Y S, et al. Modified central force optimization (MCFO) algorithm for 3D UAV path planning[J]. Neurocomputing, 2016, 171: 878-888.
- [8] Roberge V, Tarbouchi M, Labonte G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning[J]. IEEE Transactions on Industrial Informatics, 2013, 9(1): 132-141.
- [9] Zhang M, Jiang Z, Wang L, et al. Research on parallel ant colony algorithm for 3D terrain path planning[C]. Asian Simulation Conference. Singapore: Springer, 2017: 74-82.
- [10] Pandey P, Shukla A, Tiwari R. Three-dimensional path planning for unmanned aerial vehicles using glowworm swarm optimization algorithm[J]. International Journal of System Assurance Engineering and Management, 2018, 9(4): 836-852.
- [11] Xu C F, Duan H B, Liu F. Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning[J]. Aerospace & Technology, 2010, 14(8): 535-541.
- [12] Mittal N, Singh U, Sohi B S. Modified grey wolf optimizer for global engineering optimization[J]. Applied Computational Intelligence and Soft Computing, 2016, 2016: 1-16.
- [13] Storn R, Price K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces[J]. J of Global Optimization, 1997, 11(4): 341-359.
- [14] Zhou Z, Duan H, Li P, et al. Chaotic differential evolution approach for 3D trajectory planning of unmanned aerial

- vehicle[C]. IEEE International Conference on Control & Automation. Piscataway: IEEE, 2013: 368-372.
- [15] Li Z, Jia J, Cheng M, et al. Solving path planning of UAV based on modified multi-population differential evolution algorithm[C]. International Symposium on Neural Networks. Singapore: Springer, 2014: 602-610.
- [16] Zhang X, Chen J, Xin B, et al. Online path planning for UAV using an improved differential evolution algorithm[J]. IFAC Proceedings Volumes, 2011, 44(1): 6349-6354.
- [17] Rahnamayan S, Tizhoosh H R, Salama M M A. Opposition versus randomness in soft computing techniques[J]. Applied Soft Computing, 2008, 8(2): 906-918.
- [18] Zhang X Y, Duan H B. An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning[J]. Applied Soft Computing Journal, 2015, 26: 270-284.
- [19] Besada-Portas E, Torre L D L, Cruz J M D L, et al. Evolutionary trajectory planner for multiple UAVs in realistic scenarios[J]. IEEE Transactions on Robotics, 2010, 26(4): 619-634.
- [20] Duan H B, Luo Q N, Yu Y X. Trophallaxis network control approach to formation flight of multiple unmanned aerial vehicles[J]. Science China Technological Sciences, 2013, 56(5): 1066-1074.
- [21] Wang H, Wu Z J, Rahnamayan S, et al. Enhancing particle swarm optimization using generalized opposition-based learning[J]. Information Sciences, 2011, 181(20): 4699-4714.
- [22] Wang Y, Cai Z X, Zhou Y R, et al. An adaptive trade-off model for constrained evolutionary optimization[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 80-92.
- [23] Wang Y, Cai Z. Constrained evolutionary optimization by means of $(\mu + \lambda)$ -differential evolution and improved adaptive trade-off model[J]. Evolutionary Computation, 2014, 19(2): 249-285.
- [24] Gong W Y, Cai Z H, Liang D W. Adaptive ranking mutation operator based differential evolution for constrained optimization[J]. IEEE Transactions on Cybernetics, 2015, 45(4): 716-727.
- [25] Elsayed S M, Sarker R A, Essam D L. Multi-operator based evolutionary algorithms for solving constrained optimization problems[J]. Computers and Operations Research, 2011, 38(12): 1877-1896.
- [26] Becerra R L, Coello C A C. Cultured differential evolution for constrained optimization[J]. Computer Methods in Applied Mechanics & Engineering, 2006, 195(33): 4303-4322.
- [27] Efrén Mezura Montes, Jesús Velázquez-Reyes, Coello C A C. Promising infeasibility and multiple offspring incorporated to differential evolution for constrained optimization[C]. Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation. New York: ACM, 2005: 225-232.
- [28] 郑建国, 王翔, 刘荣辉, 等. 求解约束优化问题的 ε DE 算法[J]. 软件学报, 2012, 23(9): 2374-2387. (Zheng J G, Wang X, Liu R H, et al. ε -differential evolution algorithm for constrained optimization problems[J]. Journal of Software, 2012, 23(9): 2374-2387.)

作者简介

吴文海(1962—), 男, 教授, 博士生导师, 从事精确制导与控制等研究, E-mail: hkdx_2017@126.com;

郭晓峰(1992—), 男, 博士生, 从事智能算法与航迹规划的研究, E-mail: gxf123@126.com;

周思羽(1983—), 男, 副教授, 博士, 从事精确制导与控制等研究, E-mail: ezhousiyu@aliyun.com.

(责任编辑: 郑晓蕾)