基于改进粒子群算法的无人机三维航迹规划

方群, 徐青

(西北工业大学 航天学院, 陕西 西安 710072)

摘 要:航迹规划是无人机执行侦察和作战任务中的关键技术,规划算法的性能直接影响着航迹规划的质量。针对航迹规划最优性和实时性问题,提出一种惯性权值"阶梯式"调整策略与跳出局部最优解策略相结合的改进粒子群无人机航迹规划算法。通过引入最小威胁曲面的概念,使用特定的粒子群位置编码方式将约束条件和搜索算法相结合,缩小了搜索空间。针对无人机在线航迹重规划问题,提出定义重规划起始点权重系数来平衡重规划运算时间和航迹最优性之间矛盾的方法。由基于改进粒子群算法的离线航迹规划与在线航迹重规划仿真对比结果表明,该方法比粒子群算法典型改进方案能够搜索到更优离线航迹,且满足在线航迹重规划的实时性要求。

关 键 词:约束优化;代价函数;粒子群算法;无人机

中图分类号:V279 **文献标志码**:A **文章编号**:1000-2758(2017)01-0066-08

无人机的航迹规划是指在综合考虑无人机飞行油耗、威胁、飞行区域以及自身物理条件限制等因素的前提下,为飞行器在飞行区域内规划出从初始点到目标点最优或者满意的飞行航迹,其本质是一个多约束的目标优化问题[1]。航迹规划算法是航迹规划的核心。国内外相继开展了相关研究,提出了许多航迹规划算法,如模拟退火算法[2]、人工势场法[3]、遗传算法[4]、蚁群算法[5]等。但由于无人机面临的规划空间异常复杂、规划约束条件多且模糊性大,航迹搜索算法存在寻优能力差、计算量过大、效率不高等问题,在航迹规划的最优性和实时性方面有待进一步提高。

粒子群优化算法(particle swarm optimization, PSO)[6]是 Kennedy 和 Eberhart 于 1995 年提出的一种群体智能仿生算法,在解决一些典型的函数优化问题时,能够取得比较好的优化结果。但是 PSO 算法在求解的初期收敛速度较快,后期则容易陷入局部最优解,因而收敛效率低。针对上述缺陷,学者们提出了许多粒子群算法改进策略以提高寻优能力,但有些方法仅侧重于精度而忽略全局寻优的性能[7-9],有些则大大增加了算法本身的复杂性,降低了算法的优化速度[10-13]。本文在分析几种具有代表性的改进方案的基础上,提出了一种更加简单且有效的惯性权值调整策略,以及一种跳出局部最优解的策略。最后结合无人机航迹规划问题的特点,使用改进粒子群算法完成了无人机三维离线航迹规划与在线航迹重规划。

1 无人机航迹规划模型

1.1 航迹表示方法

一般地,无人机航迹规划的空间可以表示为某三维坐标系下所有点的集合 $\{(x,y,z) \mid x_{\min} \leqslant x \leqslant x_{\max}, y_{\min} \leqslant y \leqslant y_{\max}, z_{\min} \leqslant z \leqslant z_{\max}\}$,其中 x,y 可以表示为该节点在飞行水平面下的坐标,也可以表示为该点的经纬度,z 为高程数据或海拔高度。航迹规划的目的是获得无人机在该空间中的飞行轨迹,生成的航迹可表示为三维空间的一系列的点 $\{P_S, P_1, P_2, \cdots, P_{n-2}, P_G\}$,相邻航迹点之间用直线段连接。

1.2 航迹代价函数

在航迹规划中,常采用经过适当简化的航迹代价计算公式[14]

收稿日期:2016-05-23

作者简介:方群(1960—),女,西北工业大学教授,主要从事飞行动力学与控制研究。

$$\min J = \sum_{i=1}^{s} (k_1 L_i^2 + k_2 H_i^2 + k_3 T_i) \tag{1}$$

式中,s 表示航迹段数,L_i 表示第 i 段航迹长度,该项代表距离代价。H_i 表示第 i 段航迹的平均海拔高度,该项代表高度代价。T_i 为第 i 段航迹的威胁指数,该项代表威胁代价。k_1、k_2、k_3 分别是距离代价、高度代价和威胁代价的权重值,权重的选取与飞行任务要求相关。

2　粒子群算法及其改进

2.1　基本粒子群算法

粒子群算法初始化为一群数量为 N 的随机粒子(随机解),在 D 维空间中通过重复迭代、更新自身的位置以搜索适应度值最优解。粒子的位置代表被优化问题在搜索空间中的潜在解。在每次迭代中,粒子通过跟踪 2 个“极值” 来更新自己的速度和位置:一个是粒子自身目前所找到的最优解,即个体极值;另一个是整个粒子群目前找到的最优解,即全局极值。粒子 $i(i = 1,2,\cdots,N)$ 在第 $j(j = 1,2,\cdots,D)$ 维的速度 v_{ij} 和位置 x_{ij} 按如下格式更新:

$$\begin{cases} v_{ij}(t+1) = \omega \cdot v_{ij}(t) + c_1 r_{1j}(t)(p_{ij} - x_{ij}(t)) + \\ \qquad c_2 r_{2j}(t)(p_{gj} - x_{ij}(t)) \\ x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \end{cases} \tag{2}$$

式中,ω 为非负数,称为惯性权值(惯性因子),描述了粒子对之前速度的“继承”,即体现出粒子的“惯性”;c_1 和 c_2 为非负常数,称为学习因子(加速因子),体现了粒子的社会性,即粒子向全局最优粒子学习的特性;r_1 和 r_2 为(0,1) 之间的随机数;$p_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$ 表示粒子 i 的个体极值所在位置;$p_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$ 表示所有粒子的全局极值所在位置。

速度更新公式的第一项,反映粒子当前速度的影响,每一个粒子按照惯性权值的比重沿着自身速度的方向搜索,起到了平衡全局的作用,同时避免算法陷入局部最优;第二项体现了个体最优值对粒子速度的影响,即粒子本身的记忆和认识,使得粒子具有全局搜索能力。第三项则反映群体对个体的影响,即群体间的信息共享起到加速收敛的作用。

2.2　对基本粒子群算法的改进

虽然粒子群算法具有容易实现、计算简单、收敛快、鲁棒性好等许多优点,但是后期容易陷入局部最优解,收敛效率低[15]。针对该缺陷,学者们对其做了各种改进。惯性权值是粒子群算法中的一个重要参数,它表示粒子对其原始速度的继承状况。Shi 和 Eberhart 提出,较大的惯性权值有助于粒子跳出局部极小点以便于全局搜索,而较小的惯性权值有助于粒子对当前的搜索区域进行精细搜索以便于算法收敛。因此在算法进行过程中,有必要通过一些方法和手段来调整惯性权值,使算法在全局搜索和精细搜索之间达到平衡。

文献[16]给出了一种惯性权值 ω 线性递减的调整策略,即:

$$\omega_1 = \omega_{\max} - (\omega_{\max} - \omega_{\min})(t/T_{\max}) \tag{3}$$

式中,t 为迭代所进行的次数;$T_{\max}$ 为最大迭代次数;$\omega_{\max}$ 和 $\omega_{\min}$ 分别是惯性权值的上限值和下限值。线性递减策略简单、直观,具有较好的寻优性能,相比于惯性权值一直取固定值(ω = const),线性递减调整策略提高了算法的性能,但却不能很好地适应复杂的优化问题。学者们受线性递减策略的启发,提出了许多非线性递减函数的调整策略,旨在进一步提高算法的优化性能,典型的有二次函数和余弦函数等[17-18],函数表达式为:

$$\omega_2 = \omega_{\max} - (\omega_{\max} - \omega_{\min})(t/T_{\max})^2 \tag{4}$$

$$\omega_3 = 0.5(\omega_{\max} + \omega_{\min}) + 0.5(\omega_{\max} - \omega_{\min})\cos[\pi \cdot (t/T_{\max})] \tag{5}$$

此外,还有更高次的调整函数,以及各种指数和对数形式的复杂函数。

不难看出,此前关于惯性权值改进策略所构造的减函数,基本上只是单纯地在整个区间上使用一次函数、二次函数甚至更高次函数,以及正(余)弦函数等三角函数、各种指数和对数形式的函数,有些学者甚至为不同而不同,构造出非常复杂的减函数。如此构造出复杂的函数,一方面无疑会增加计算机运算量,降低改进算法的优化速度;另一方面脱离粒子群算法本身的特点盲目地改进,对惯性权值减函数的研究流于表面而不够深入,缺乏统一性和指导性。

结合粒子群算法在短暂的迭代前期收敛速度极快,而在漫长的迭代后期则基本陷于停滞状态的特点,本文提出如下惯性权值改进策略:惯性权值在迭代前期取恒定的较大值,在迭代后期取恒定的较小值,本文将其称为“阶梯式”调整策略。具体的调整公式为

$$\omega = \begin{cases} \omega_{\max}, & t/T_{\max} \leqslant \lambda \\ \omega_{\min}, & t/T_{\max} > \lambda \end{cases} \tag{6}$$

式中，$\lambda \in (0,1)$ 表示迭代前期占整个迭代周期的比例。适宜的 λ 值并非固定不变，它是由待优化问题的复杂程度、要求优化精度以及 $T_{\max}$ 的取值综合权衡确定的。但我们通过两三次尝试就能找到比较适宜的 λ 值，如尝试 $\lambda = 0.1$、0.3 或 0.7。一般情况下不推荐取 $\lambda > 0.5$，这是因为 $\lambda > 0.5$ 意味着迭代后期所占的比例较小，而迭代后期主要是精细搜索，如果迭代后期太短就不利于保证解的精度。如果发现比较适宜的 $\lambda > 0.5$，此时应调大 $T_{\max}$ 的值，如此适宜的 λ 值自然就会变小。当然，一味地调大 $T_{\max}$ 的值会引起迭代周期变长，所以在对结果精度要求不高、而对运算时间要求较高时，可取 $\lambda > 0.5$。

本文提出的惯性权值“阶梯式”调整策略，具有如下优点：①“阶梯式”调整函数相比于其他学者提出的减函数，形式远远更加简单，有效地避免了各种函数繁杂的运算，有利于减少算法优化时间；②结合了粒子群算法本身的寻优特点，在不同优化阶段选取不同的惯性权值，能够真正有效地兼顾算法全局搜索程度和精细搜索程度，相比其他调整策略，两者兼顾的比例还具有非常直观的可调性；③针对不同优化问题的复杂程度、不同优化精度要求、不同最大迭代次数设置，存在一个可调参数 λ，避免了“一刀切”的调整策略，通用性更强。

仅靠对惯性权值的改进，并不能完全解决粒子群算法容易陷入局部最优的问题。所以，必须在改进惯性权值的基础上对算法作进一步的改进。学者们提出引进粒子群浓度参数、调整选择概率、拓扑结构等改进思想，也有的学者将粒子群与遗传算法的思想相结合等等，使用各种不同的思想方法对粒子群参数进行调整。不难看出，这些改进的策略虽然可能使得寻优性能更佳，但是也引入了其他算法的概念和思想，使得粒子群算法不再纯粹，失去了它自身概念简单、形式简洁的魅力。

本文基于不引入其他算法概念和思想的理念，进一步提出如下改进策略：在迭代过程中，当相邻 2 次的最优适应度值之差小于某个值 ε 时，就认定粒子群可能已经陷入局部最优，此时保留当前获得的全局最优粒子位置信息，重新随机初始化粒子群的位置。经过一定的迭代次数后，相邻 2 次的最优适应度值之差会再次小于 ε 值，此时再次保留当前粒子群的全局最优位置，然后重新初始化粒子群的位置，反复如此，直到迭代结束，计算机伪代码如下

```
if | fitness(p_{g,t+1}) − fitness(p_{g,t}) | < ε
    x_ij = − X_max + 2X_max rand( ),
        i = 1,2,…,N; j = 1,2,…,D
end
```

上面所提出的改进策略旨在使迭代跳出局部最优解，最终可以改善粒子群算法的全局搜索能力。它具有如下优点：①避免引入其他算法概念和思想的麻烦，完全保留了粒子群算法自身概念简单的优点，在每次算法可能陷入局部最优时都对粒子群的位置重新初始化，这个过程只需设定一个 ε 值，相比其他复杂的改进策略，操作更加简单快捷，保证了粒子群算法的简洁性；②可以使算法有效跳出局部最优解，理论上通过如此反复，经过足够大的迭代次数就可以求得全局最优解；最优解的精度直接依赖于最大迭代次数，因此可以通过调整最大迭代次数来调节结果向最优解逼近的程度。

本文将惯性权值“阶梯式”调整策略与上述跳出局部最优解策略相结合来改进粒子群算法，记为 IMPSO-1，将惯性权值线性递减策略对应的改进粒子群算法记为 IMPSO-2。经过 Sphere 函数、Schaffer 函数、Rastrigrin 函数 3 个经典测试函数的仿真对比，相比 IMPSO-2，IMPSO-1 无论是在优化精度还是优化稳定性（方差）上均有明显提升。

3 粒子群算法在无人机三维航迹规划中的应用

3.1 最小威胁曲面

为了降低规划的难度，本文引入“最小威胁曲面”，将三维航迹规划问题转换为二维航迹规划问题，所谓最小威胁曲面，是指在建立数字地图时，将火力威胁等效成为山峰威胁，考虑到威胁对于数字地图的抬高后，由距离抬高地形高程 h_e 的点所构成的曲面[19]。这个曲面具有这样一个性质：飞行器在这个曲面上进行飞行时受到的地形威胁最小，即飞行器的最小危险航迹一定位于该曲面上。

3.2 粒子群位置信息的编码

在给定了航迹规划的起始点和目标点后，最小威胁曲面上连接飞行起始点和目标点的任一条曲线都可以看作是一条待选航迹，注意到每条航迹在水平面内的投影也是一条曲线，根据一一对应的关系，

通过对待选航迹在二维平面投影的寻优,可以找到最小威胁曲面在投影面上的二维航迹。很显然,在投影面中的二维航迹和最小威胁曲面中的三维航迹有一一对应的关系。如果在投影面内找到了最优的二维航迹,再将该最优二维航迹反向投影到最小威胁曲面上,即可得到在最小威胁曲面上的最优三维航迹。也就是说,想要寻找飞行器最优的三维航迹,首先需要寻找投影面内的最优二维航迹。

将二维飞行航迹表示为一组节点序列 $\{P_S,P_1,P_2,\cdots,P_{n-2},P_G\}$,共有 n 个航迹节点,那么在二维投影平面内,一条完整的航迹可以表示为如下粒子位置坐标:

$$\boldsymbol{X}_i=[x_{iS},y_{iS},x_{i1},y_{i1},\cdots,x_{i,n-2},y_{i,n-2},x_{iG},y_{iG}]$$

式中,(x_{iS},y_{iS}),(x_{i1},y_{i1}),$\cdots$,$(x_{i,n-2},y_{i,n-2})$,(x_{iG},y_{iG}) 分别记录了第 i 个粒子的航迹节点$\{P_S,P_1,P_2,\cdots,P_{n-2},P_G\}$ 在二维规划空间中的坐标。对于种群数量为 N 的粒子群来说,其位置信息可以用矩阵表示,矩阵的每个行向量代表一个粒子的位置坐标(也就是一条完整航迹),因为共有 N 个粒子,对应的矩阵就有 N 行,故粒子群的位置矩阵表示为:

$$\boldsymbol{X}_{N\times 2n}=\begin{bmatrix}\boldsymbol{X}_1\\ \boldsymbol{X}_2\\ \vdots\\ \boldsymbol{X}_N\end{bmatrix}=\begin{bmatrix}x_{1S} & y_{1S} & x_{11} & y_{11} & \cdots & x_{1,n-2} & y_{1,n-2} & x_{1G} & y_{1G}\\ x_{2S} & y_{2S} & x_{21} & y_{21} & \cdots & x_{2,n-2} & y_{2,n-2} & x_{2G} & y_{2G}\\ \vdots & & & & & & & & \\ x_{NS} & y_{NS} & x_{N1} & y_{N1} & \cdots & x_{N,n-2} & y_{N,n-2} & x_{NG} & y_{NG}\end{bmatrix} \tag{7}$$

对粒子群位置矩阵(7)的行向量 $\boldsymbol{X}_i$ 前后元素之间进行相应的约束限制,使其满足飞行航迹节点均在威胁区域外、航迹段与威胁区域无交点并且相邻航迹段满足最大转弯角约束。自然地,粒子群在这个限制范围内搜索最优航迹,得到的二维航迹既能够满足飞行器自身的物理约束,同时也能够保证飞行器避开威胁区域。这就意味着,用于评价三维航迹优劣的航迹长度 L、飞行高度 H、威胁指数 T 3 个性能指标,此时只须考虑飞行航迹长度[20]。因为飞行高度的性能指标已由“最小威胁曲面”的规划思想保证了飞行高度的最优,而威胁指数则直接靠限制粒子群位置矩阵保证飞行“零威胁”,所以只剩飞行航迹长度一个性能指标,而航迹段可表示为 $L_i(i=1,2,\cdots,s)$,故本文的航迹代价可以简化为:

$$\min J=\sum_{i=1}^{s}L_i \tag{8}$$

4　基于改进粒子群算法的航迹规划与重规划

4.1　基于改进粒子群算法的离线航迹规划

按照上述粒子群位置信息编码方式,航迹规划转化为在满足航迹约束条件下,搜索航迹代价最小的粒子。下面给出基于改进粒子群算法的离线航迹规划的具体步骤:

Step1　对航迹规划空间在 x 轴上做 n 等分(对应 n 个航迹节点坐标参数),设置算法参数,粒子群数 N、最大速度 $V_{\max}$、最大最小惯性权重 $\omega_{\max}$ 和 $\omega_{\min}$、加速因子 c_1 和 c_2、最大迭代次数 $T_{\max}$ 等。

Step2　令迭代次数 $t=0$,从约束条件出发,初始化粒子群位置矩阵(7)。

Step3　利用(8)式计算每一个粒子的适应值,比较每一个粒子当前适应值和历史最优适应值 p_i。如果当前值小于 p_i,则将当前粒子的位置和适应值更新历史最优(首次迭代时 p_i 为当前位置)。确定所有粒子的适应度最小的群体全局最优解 p_g,如果当前全局最优小于历史全局最优,则更新历史全局最优。

Step4　判断是否达到结束条件(一般为达到最大迭代次数 $T_{\max}$,或最优值收敛至指定精度)。如果达到约束条件,则转到 Step12,否则转到 Step5。

Step5　令 $i=0$。

Step6　令 $j=0$。

*Step*7　按照(2)式与(6)式更新粒子的速度与位置,若 $|v_{ij}|\geqslant|v_{\max}|$,则取 $|v_{ij}|=|v_{\max}|$,方向不变。

Step8　$j=j+1$。如果 $j>n$,转到 Step9;否则转到 Step7。

Step9　$i=i+1$。如果 $i>N$,转到 Step10;否则转到 Step6。

Step10　检查更新后粒子的有效性,无效的重新进行初始化。

Step11　若上次迭代与本次迭代的全局最优适

应度值之差小于 ε,则保留当前全局最优位置 p_g,同时重新初始化粒子群位置。

Step12 $t=t+1$。若 $t>T_{\max}$,转到 Step13;否则转到 Step3。

Step13 迭代结束,选取一条最优航迹。

4.2 基于改进粒子群算法的在线航迹重规划

当飞行器沿着参考航迹飞行时,假设在 t_1 时刻机载设备探测到前方一定范围内有新的威胁,若机载计算机采用改进的粒子群算法进行一次航迹规划所用最长时间为 $t_{p\max}$,在 $t_2=t_1+t_{p\max}$ 时刻飞行器在参考航迹上的位置可以根据速度信息推算出来,则 t_2 时刻所处的位置 (x_E,y_E) 就是开始沿着新规划航迹飞行的最早起始点。假设新的突发威胁与参考航迹的交点坐标分别为 (x_{C1},y_{C1}) 和 (x_{C2},y_{C2}),其中 (x_{C1},y_{C1}) 更接近 (x_E,y_E),那么点 (x_{C1},y_{C1}) 就是开始沿着新规划航迹飞行的最晚起始点。最早起始点和最晚起始点都不宜选作新航迹规划起始点,因为前者会带来航迹规划运算时间太长的问题,后者可能无法保证飞行器的安全性。所以,新起始点应该在两者之间进行选择。本文定义重规划起始点权重系数 δ 如下

$$\delta=\frac{x_S-x_E}{x_{C1}-x_E} \tag{9}$$

式中,x_S 表示被选择的新的起始点,其取值范围为 $x_E<x_S<x_{C1}$。x_S 越接近最早起始点 x_E,δ 越接近于 0;x_S 越接近最晚起始点 x_{C1},δ 越接近于 1。由此可见,δ 实际上是一个平衡 2 种航迹重规划思想的权值。当 δ 较小时,航迹规划运算时间较长,但重规划所得的航迹更优;当 δ 较大时,航迹规划运算时间较短,但重规划所得的航迹代价值更大。在实际的航迹规划中,需要根据飞行器发现新威胁时距离威胁的距离、自身的机动性能、机载计算机的处理速度、航迹规划任务侧重来确定权重系数 δ 的值。

航迹重规划新的目标点 x_G 与新的起始点 x_S 可以对称选择,即:

$$x_G=x_{C2}+(1-\delta)(x_{C1}-x_E) \tag{10}$$

也就是说,航迹重规划新的起始点 x_S 与 x_{C1} 之间的距离和新的目标点 x_G 与 x_{C2} 之间的距离是相等的。如此就可以通过只调整一个权值 δ,达到既调整重规划新起始点又调整新目标点 2 个点的目的。

利用改进粒子群算法进行在线航迹重规划的流程如图 1 所示。

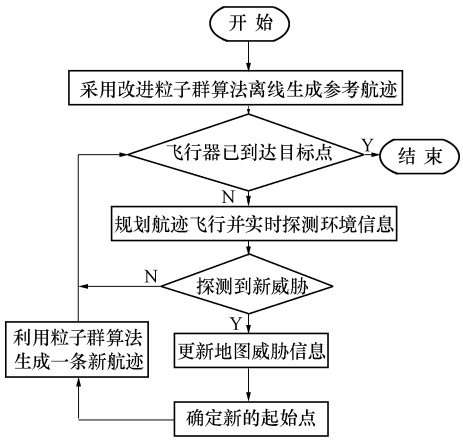


图 1 局部航迹在线重规划流程图

5 仿真分析

本文借鉴前人有关生成飞行区域中数字高程地图的函数模拟方法,建立飞行区域及其中分布的威胁模型,分别使用本文提出的改进粒子群算法 IMPSO-1 和惯性权值线性递减策略下的粒子群算法 IMPSO-2,完成无人机航迹规划的仿真分析。

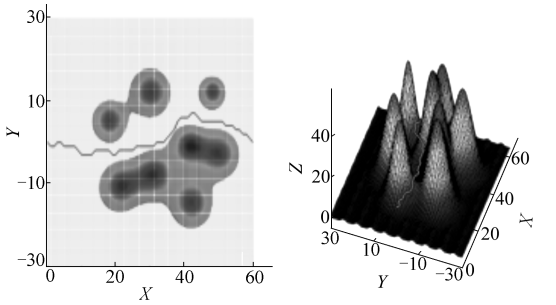
5.1 无人机的三维离线航迹规划

粒子群算法参数设置:IMPSO-1 与 IMPSO-2 的粒子群总数量 $N=200$,最大迭代次数 $T_{\max}=1\,000$,$\omega_{\max}=0.9$ 且 $\omega_{\min}=0.4$,$c_1=c_2=2$。根据本航迹规划仿真示例的特点与最大迭代次数,IMPSO-1 中的参数取 $\lambda=0.7$,$\varepsilon=0.01$。分别使用 IMPSO-1 和 IMPSO-2 在数字高程地图(横向为 X 坐标,纵向为 Y 坐标,高度为 Z 坐标)最小威胁曲面(求解最小威胁曲面高度值属于另外一个优化问题,本文假设取 $h_c=15$ km)水平投影下规划获得二维航迹,并在此基础上进行反投影到最小威胁曲面上获得三维航迹,结果如图 2、图 3 所示。2 种方案下的性能评价指标函数式(8)对应的值见表 1 所示。

表 1 2 种算法改进下的性能评价指标函数值

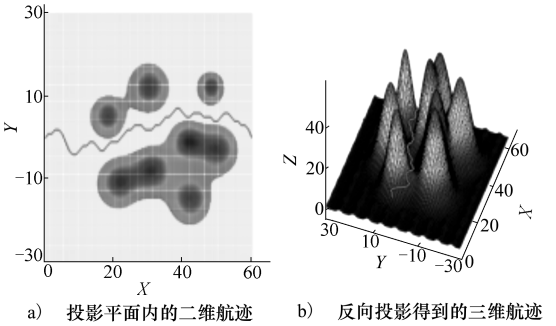
IMPSO Number	IMPSO-1	IMPSO-2
Performance index	70.348 7	76.963 0

对比图 2、图 3 以及表 1 中 2 个航迹规划仿真实验结果可以看出,IMPSO-1 的优化性能较 IMPSO-2 更佳,验证了本文提出的改进算法 IMPSO-1 在无人机离线航迹规划中比 IMPSO-2 具有更好的寻优性能。



a)　投影平面内的二维航迹　　b)　反向投影得到的三维航迹

图 2　基于 IMPSO-1 的航迹规划仿真结果图

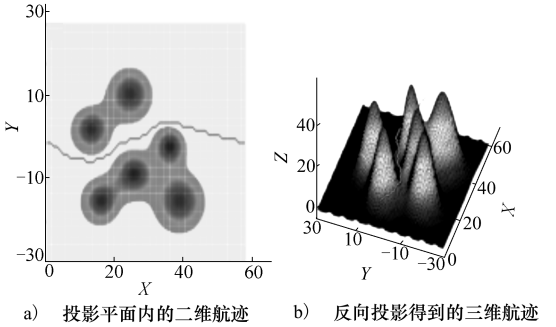


a)　投影平面内的二维航迹　　b)　反向投影得到的三维航迹

图 3　基于 IMPSO-2 的航迹规划仿真结果图

5.2　无人机的三维在线航迹重规划

首先,使用 IMPSO-1 生成无人机的三维离线航迹。IMPSO-1 算法参数设置:粒子种群总数量 N = 200,最大迭代次数 $T_{\max}$ = 1 000,$\omega_{\max}$ = 0.9 且 $\omega_{\min}$ = 0.4,$c_1 = c_2 = 2$。根据本航迹规划示例的特点与最大迭代次数,IMPSO-1 中参数取 λ = 0.7,ε = 0.01。在水平投影下规划获得的二维航迹,以及反投影到最小威胁曲面上获得的离线三维航迹如图 4 所示。



a)　投影平面内的二维航迹　　b)　反向投影得到的三维航迹

图 4　预先已知地图的离线航迹规划仿真结果图

假如飞行器发现新威胁 T 时,恰好飞至图 5 参考航迹中的点 $N(38,5)$,此时机载计算机立即开始进行局部航迹在线重规划的运算。在图 4 的航迹规划中获得整个参考航迹的平均运算时间为 21 s,而此例中发现新威胁时飞行器已飞至最后 1/3 段航程,局部重规划的时间显然要比生成整个航迹所用的 21 s 更短,但为安全起见,仍旧将重规划的时间 t_{rp} 设为 21 s,给重规划留有充分的时间裕度。设飞行器的飞行速度为 200 m/s,那么在重规划时间 t_{rp} = 21 s 内飞行器由 $N(38,5)$ 飞至 $E(42,4)$,参考航迹与新威胁 T 的交点分别为 $C_1(47,3)$、$C_2(56,1)$。通过权衡飞行器发现新威胁 T 时与威胁之间的距离和计算机的运算时间等信息,本文从保证航迹更优性的侧重出发,选择重规划起始点权值 δ = 0.4,根据(9)式确定重规划的新起始点为 $S(44,4)$,由(10)式确定新的目标点为 $G(59,0)$。

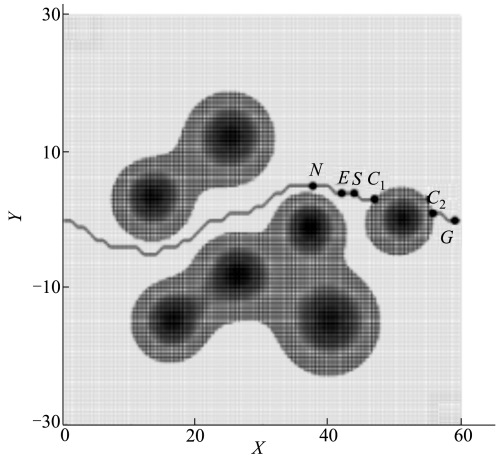
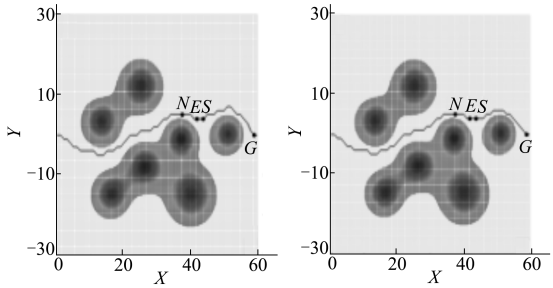


图 5　投影面内局部航迹重规划任务图

分别使用 IMPSO-1 和 IMPSO-2 对图 5 中 $E \sim G$ 段进行局部航迹重规划,其中参数设置为:粒子群数量 N = 50,最大迭代次数 $T_{\max}$ = 300,$\omega_{\max}$ = 0.9 且 $\omega_{\min}$ = 0.4,$c_1 = c_2 = 2$。根据本航迹规划示例的特点与最大迭代次数,取 λ = 0.6,ε = 0.01。投影面内规划的二维航迹分别如图 6 所示。2 种方案下的性能评价指标函数式(8)所对应的值如表 2 所示。



a)　基于 IMPSO-1 算法的投影面内的二维航迹　　b)　基于 IMPSO-2 算法的投影面内的三维航迹

图 6　投影面内的局部航迹重规划仿真结果对比图

表 2 2种方案下的性能评价指标函数值

IMPSO Number	IMPSO-1	IMPSO-2
Performance index	69.941 1	71.177 2

对比图 6 中 2 幅图,并由 2 种改进算法下的航迹规划仿真实验结果可以看出,IMPSO-1 在局部三维航迹的在线重规划下的优化性能依旧较 IMPSO-2 更佳,再次验证本文提出的改进算法 IMPSO-1 在进行无人飞行器的航迹规划中具有更好的优化性能。

由图 6 可以看出,飞行器顺利地避开了突发威胁,满足航迹规划的实时性要求。而且在粒子群数目、最大迭代次数相等的情形下,使用 IMPSO-1 比 IMPSO-2 规划获得了航迹性能评价指标更优的航迹。这就意味着,当飞行器发现突发威胁时,前者能够比后者在更短的时间内找到相同航迹代价要求的航迹,也就是说在局部航迹重规划中具有更优的实时性。

6 结 论

本文所提出的改进粒子群算法相比其他复杂的改进方案,惯性权值的调整形式和为跳出局部最优解而执行的操作均更加简单、快捷,其中由于可调参数 λ 和精度控制参数 ε 的存在,使得算法具有更好的通用性和适应性;改进算法不仅可以显著地提高原算法的优化精度和稳定性,还可以通过调整最大迭代次数来调节结果向最优解逼近的程度。航迹规划是一个复杂的优化问题,涉及众多约束条件,不存在绝对意义上的最优航迹,只存在某种意义下的满意解。本文提出的改进粒子群算法可以在满足航迹约束的前提下,快速对无人机航迹完成近似最优规划。仿真实验表明,该算法不仅可以生成更优的航迹,而且可以满足在线实时航迹规划的要求。

参考文献:

[1] 郑昌文,严平,丁明跃. 飞行器航迹规划研究现状与趋势[J]. 宇航学报,2007,28(6):1441-1446
Zheng Changwen, Yan Ping, Ding Mingyue. Rsearch Status and Trend of Route Planning for Flying Vehicles[J]. Journal of Astronautics, 2007, 28(6): 1441-1446 (in Chinese)

[2] Michael Steinbrunn, Guido Moerkotte, Alfons Kemper. Heuristic and Randomized Optimization for the Join Ordering Problem [J]. The VLDB Journal,1997,6(3):191-208

[3] Khatib O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots[M]. Autonomous Robot Vhicles, Springer, New York, 1986:396-404

[4] Holland J H. Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence[M]. University of Michigan Press, 1975: 126-137

[5] Colorni A, Dorigo M, Maniezzo V. Distributed Optimzation by Ant Colonies[C]//European Conference on Artificial Life, 1991

[6] Kennedy J, Eberhart R C. Particle Swarm Optimization[C]//. IEEE International Conference on Neural Networks, 1995: 1942-1948

[7] 董方. 粒子群算法研究及其在动态优化中的应用[D]. 杭州:浙江大学,2014
Dong Fang. Researches on Particle Swarm Optimizer and Its Applications in Dynamic Optimization[D]. Hangzhou, Zhejiang University, 2014 (in Chinese)

[8] Li Jianping. Improved Quantum-Behaved Particle Swarm Optimization[J]. Open Journal of Applied Sciences, 2015(5): 240-250

[9] 姚灿中,杨建梅. 基于变惯性权重及动态邻域的改进 PSO 算法[J]. 计算机工程, 2011,37(21): 20-22
Yao Canzhong, Yang Jianmei. Improved PSO Algorithm Based on Variety Inertia Weight and Dynamic Neighborhood[J]. Computer Engineering, 2011,37(21): 20-22 (in Chinese)

[10] Chananes Akjiratikarl, Pisal Yenradee, Paul R Drake. An Improved Particle Swarm Optimization Algorithm for Care Worker Scheduling[J]. Industrial Engineering and Management Systems, 2008, 7(2): 171-181

[11] 张英男. 改进的粒子群优化算法(APSO 和 DPSO)研究[D]. 大连:大连理工大学,2008
Zhang Yingnan. The Improved Particle Swarm Optimization Algorithms: APSO and DPSO[D]. Dalian, Dalian University of

Technology,2008 (in Chinese)
[12] Maurice Clerc, James Kennedy. The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space [J]. IEEE Trans on Evolutionary Computation, 2002, 20(1): 1671-1676
[13] 程军. 基于生物行为机制的粒子群算法改进及应用[D]. 广州:华南理工大学,2014
Cheng Jun. Improvement and Application of Particle Swarm Optimization Algorithm Based on Biological Behavior Mechanism [D]. Guangzhou, South China University of Technology, 2014 (in Chinese)
[14] Zheng C, Zhou C, Ding M. Real-Time 3D Route Planner for Unmanned Air Vehicles[C]//International Society for Optics and Photonics, 2001: 167-172
[15] 杨维, 李歧强. 粒子群优化算法综述[J]. 中国工程科学, 2004, 6(5):87-94
Yang Wei, Li Qiqiang. Survey on Particle Swarm Optimization Algorithm[J]. Engineering Science, 2004, 6(5):87-94 (in Chinese)
[16] Shi Y, Eberhart R C. A Modified Partical Swarm Optimizer[C]//IEEE World Congress on computational Intelligence,1998: 69-73
[17] 陈贵敏,贾建援,韩琪. 粒子群优化算法的惯性权值递减策略研究[J]. 西安交通大学学报,2006,40(1): 53-56
Chen Guimin, Jia Jianyuan, Han Qi. Study on the Strategy of Decreasing Inertia Weight in Particle Swarm Optimization Algorithm[J]. Journal of Xi'an Jiaotong University, 2006,40(1): 53-56 (in Chinese)
[18] 周敏, 李太勇. 粒子群优化算法中的惯性权值非线性调整策略[J]. 计算机工程, 2011, 37(5):204-206
Zhou Min, Li Taiyong. Nonlinear Adjustment Strategy of Inertia Weight in Particle Swarm Optimization Algorithm[J]. Computer Engineering, 2011, 37(5):204-206 (in Chinese)
[19] Menon P K A, Kim E. Optimal Trajectory Synthesis for Terrain-Following Flight[J]. Journal of Guidance Control and Dynamics, 1991, 14(4): 807-813
[20] 唐必伟,方群. 基于改进蚁群算法的无人飞行器二维航迹规划[J]. 西北工业大学学报,2013,31(5): 683-687
Tang Biwei, Fang Qun. Effective 2D Route Planning of UAV Based on Improved Ant Colony Algorithm[J]. Journal of Northwestern Polytechnical University,2013,31(5): 683-687 (in Chinese)

3D Route Planning for UAV Based on Improved PSO Algorithm

Fang Qun, Xu Qing
(School of Aeronautics , Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Route planning is a key technology in reconnaissance and combat missions for UAV, and the performance of planning algorithm directly affects its quality. For the optimality and real-time problem in route planning, this paper proposes an improved particle swarm optimization (PSO) algorithm including inertia weight adjustment strategy based on step function and escaping strategy from local minimum. Introducing the concept of SOMR (Surface of Minimum Risk) and defining a specific particle swarm coding representation, the searching space is reduced. For the re-planning online problem, a weight coefficient for re-starting point is defined to balance the conflict between re-planning time and the route optimality. The simulation results demonstrate that 3D route planning for UAV based on this improved PSO can find a more optimal route than typical improvement strategy of PSO and meet the request of re-planning in real time.

Keywords: constrained optimization; cost functions; particle swarm optimization (PSO); unmanned aerial vehicles