

基于混合量子粒子群优化算法的 三维航迹规划

傅阳光¹, 周成平¹, 丁明跃^{1,2}

- (1. 华中科技大学图像识别与人工智能研究所, 多谱信息处理技术国防科技重点实验室, 武汉 430074;
2. 华中科技大学生命科学与技术学院, 图像信息处理与智能控制教育部重点实验室, 武汉 430074)

摘 要: 针对粒子群优化算法(PSO)存在的早熟收敛问题, 通过将种群的繁殖机制引入量子粒子群优化算法(QPSO), 提出了一种混合量子粒子群优化算法(HQPSO), 将该算法应用于无人飞行器的三维航迹规划。同时, 运用统计学方法, 通过仿真实验比较了 HQPSO 算法与 QPSO 算法以及带动态变化惯性权系数的 PSO 算法的性能。仿真实验结果表明, HQPSO 不但比 QPSO 和 PSO 具有更强的全局搜索能力, 而且比 QPSO 和 PSO 具有更快的收敛速度。

关键词: 无人飞行器; 航迹规划; 繁殖机制; 混合量子粒子群优化

中图分类号: V249

文献标识码: A

文章编号: 1000-1328(2010)12-2657-08

DOI: 10.3873/j.issn.1000-1328.2010.12.006

3-D Route Planning Based on Hybrid Quantum-Behaved Particle Swarm Optimization

FU Yang-guang¹, ZHOU Cheng-ping¹, DING Ming-yue^{1,2}

- (1. Institute for Pattern Recognition and Artificial Intelligence, State Key Laboratory for Multi-Spectral Information Processing Technologies, Huazhong University of Science and Technology, Wuhan 430074, China;
2. College of Life Science and Technology, "Image Processing and Intelligence Control" Key Laboratory of Education Ministry of China, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: In view of the premature convergence problem of particle swarm optimization (PSO), a hybrid quantum-behaved particle swarm optimization algorithm (HQPSO) is presented by introducing the breeding strategy into QPSO in this paper. A method of 3-D route planning for unmanned air vehicle (UAV) is set up on the basis of the proposed HQPSO algorithm. The performance of HQPSO is compared with the QPSO and PSO with inertial weighting coefficients by using a statistical method. Simulation results demonstrate that HQPSO not only has stronger global searching ability, but also achieves a faster convergence speed compared with QPSO and PSO.

Key words: Unmanned aerial vehicle (UAV); Route planning; Breeding strategy; Hybrid quantum-behaved particle swarm optimization (HQPSO)

0 引 言

无人飞行器航迹规划作为任务规划的重要组成部分, 无论是在军用还是民用方面都越来越受到人们的重视, 成为当前研究的热点。无人飞行器的三

维航迹规划可以描述为: 在给定起始点和目标点、战场环境以及无人飞行器自身的机动性能约束参数(如水平拐弯角, 爬升/下滑角, 飞行高度)等条件下, 采用优化算法, 规划一条从起始点到目标点的最优或次优航迹。

目前用于无人飞行器航迹规划的寻优算法大致可分成以下几类:(1) 路线图方法。主要包括 Voronoi 图法^[1]和随机路线图法^[2];(2) 启发式算法。如 A* 算法^[3], 稀疏 A* 算法^[4-5]以及 D* 算法^[6];(3) 进化算法^[7-8]。基于路线图的方法通常只适用于二维航路规划,且难以实现无人飞行器的各种机动约束。而 A* 算法和其动态版本 D* 算法作为逐点搜索算法,其搜索空间相当庞大。稀疏 A* 算法通过将各种约束加入到搜索过程中,虽然极大地修剪了搜索空间,但当规划区域过大时,规划时间将呈指数增长。

进化算法是一种基于种群的优化方法,在进化的开始,种群被初始化,然后通过事先设置的一系列算子来控制个体的进化,适应度高的个体进入下一代的概率也高。粒子群优化算法(PSO)^[9]同样也是基于种群的智能优化算法,但与其他进化算法相比,具有两个显著的不同特点:(1) 没有“适者生存”的机制,所有的个体(粒子)在迭代的过程中始终作为种群的成员保留;(2) 没有进化算子(如交叉算子,变异算子等)的作用,每个粒子通过跟踪自身迄今搜索到的最优位置(个体最优粒子,也叫个体极值) p_{best} 和整个种群迄今搜索到的最优位置(全局最优粒子,也叫全局极值) g_{best} 来控制粒子的运动。这些特点不但使算法的原理变得简单,而且大大减少了算法的运行时间。

PSO 自提出以来,已经被广泛应用到多元函数优化,神经网络训练,模糊系统控制等问题,但是 PSO 容易陷入局部最优,即早熟收敛。为了提高算法性能,国内外学者对算法的改进作了大量工作。Shi 和 Eberhart^[10]将惯性权系数引入到 PSO,并指出动态变化的惯性权系数比静态的惯性权系数要好。孙俊以 Clerc^[11]对粒子收敛性的研究为基础,从量子力学的观点提出了具有量子行为的量子粒子群优化算法(QPSO)^[12],并通过对基准函数的测试表明 QPSO 的性能优于 PSO。

最近,傅阳光^[13]等基于量子粒子群优化算法,通过将各种约束条件结合到航路的搜索过程中,为无人飞行器规划出一条平滑可飞的二维航路。本文在文献[13]研究的基础上,将种群的繁殖机制^[14]引入 QPSO 算法,对 QPSO 算法做了进一步改进,提出了混合量子粒子群算法(HQPSO),并将该算法应用

于无人飞行器的三维航迹规划。本文把无人飞行器必须满足的约束集成到了代价函数中,然后利用算法的优化功能,使航迹代价越来越小,驱使算法找到满足约束的最优航迹。运用统计学的方法,通过仿真实验比较了 HQPSO 算法与 QPSO 算法以及带动动态变化惯性权系数的 PSO 算法的性能。仿真实验结果表明, HQPSO 算法不但比 QPSO 算法和 PSO 算法具有更强的全局搜索能力,而且比 QPSO 算法和 PSO 算法具有更快的收敛速度。同时,基于 HQPSO 算法生成的三维航迹能有效实现威胁回避和地形跟随。

1 航迹表示与航迹代价模型

1.1 航迹编码

一个粒子表示解空间中的一个可能解,即搜索空间中的一条备选航迹,或称之为种群中的一个个体。因此本文对粒子,个体,航迹,可能解等名称不作区别。粒子中的相应分量代表航迹的节点。航迹的编码其本质是在种群中的每个粒子与搜索空间中的备选航迹之间建立一个一一映射关系。每个粒子代表唯一的一条航迹,每条航迹也对应唯一的一个粒子。为了公平地比较 PSO、QPSO 和 HQPSO 的性能,本文对三种算法采用同样的航迹编码方式,同样的初始化方法,同样的代价函数模型以及同样的战场环境(即相同的起始点,目标点和威胁环境),即除算法自身因素外其他条件均保持相同。

种群可以表示为一个矩阵 $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_m]^T$, 其中 m 为种群的大小, \mathbf{X}_i 为种群中的第 i 个个体($i = 1, \dots, m$)。由于每条航迹有相同的起始点和目标点,所以起始点与目标点不参与粒子的编码。设每条航迹除起始点和目标点外由 n 个航迹节点组成,为了记录每个航迹节点的空间位置信息(x, y, z),采用以下的定长实数编码粒子结构:

$\mathbf{X}_i = [x_{i1}, \dots, x_{in}, x_{i,n+1}, \dots, x_{i,2n}, x_{i,2n+1}, \dots, x_{i,3n}]$ 为 \mathbf{R}^{3n} 中的一个向量,其中 x_{i1}, \dots, x_{in} 和 $x_{i,n+1}, \dots, x_{i,2n}$ 分别记录了 n 个航迹节点在平面内的横坐标值 x 和纵坐标值 y ,类似地 $x_{i,2n+1}, \dots, x_{i,3n}$ 记录了 n 个航迹节点的在垂直方向上的飞行高度值 z 。基于上面的粒子结构,第 i 条航迹的第 j 个节点坐标为 $(x_{ij}, x_{i,j+n}, x_{i,j+2n})$, 其中 $j = 1, \dots, n$ 。即若航迹由 n 个节点组成,则粒子的维数为 $3n$ 。

1.2 解空间与初始化

与经典的函数优化问题类似,在明确了待优化的参数以后,下一步就是确定每个参数的范围,即定义解空间。在我们的问题中, $x_{ij}(i = 1, \dots, m; j = 1, \dots, 3n)$ 即为需要优化的参数,为此,下面分别给出它们的范围:

$$\begin{cases} 0 \leq x_{ij} \leq x_{\max}, & j = 1, \dots, n \\ 0 \leq x_{ij} \leq y_{\max}, & j = n + 1, \dots, 2n \\ z_{\min} + h \leq x_{ij} \leq z_{\max} + H, & j = 2n + 1, \dots, 3n \end{cases}$$

其中 x_{\max}, y_{\max} 分别为无人飞行器的最大航程在水平方向和垂直方向上的投影距离(也可以分别设为战场的长和宽), h 和 H 分别为地形的最小和最大高程值, z_{\min} 和 z_{\max} 分别为无人飞行器允许的最低和最高安全相对飞行高度(因为飞得过高容易被雷达发现,飞得过低容易与地面相撞)。从而问题的解空间被定义为一个长宽高分别为 x_{\max}, y_{\max} 和 $(z_{\max} + H - z_{\min} - h)$ 的长方体空间区域。

在程序运行的初期,整个种群需要初始化,尽管一些更精细的初始化方法,如 Sobol 序列产生器方法,非线性单纯形法,骨架化方法等,能从整体上进一步提高算法的性能,但为了简便起见,本文采用常用的随机初始化方法,即每个粒子被初始化为解空间中的一个随机位置。对 PSO 而言,还需要为每个粒子初始化一个随机速度。种群大小以及航迹节点个数(等价于粒子维数)是与算法相关的需要用户确定的参数。显然,航迹节点的个数越多,生成的结果航迹的精度就越高,但是所花费的计算时间也相应增加。

1.3 航迹代价模型

将优化算法用于实际物理问题时,设置合理的代价函数是至关重要的。代价函数使算法与实际物理问题紧密联系在一起。因此,对于无人飞行器的航迹规划这一特殊应用背景,其核心就是建立尽可能精确的代价函数模型。一旦代价函数模型建立后,代价函数就成为航迹优劣或个体好坏的唯一评价标准。航迹代价越小则表明航迹越优,反之表明航迹越差。

无人飞行器的航迹规划不同于传统的机器人二维路径规划,更不同于最短路径问题。无论是机器人路径规划还是无人飞行器航迹规划,现有文献中的代价函数通常至少包含路径/航迹长度代价和障

碍/威胁代价。一方面希望最小化航迹长度,另一方面希望最大化避开威胁。然而这两个方面似乎相互矛盾,最小化航迹长度将导致威胁代价增加,甚至可能穿越威胁区。同样地,最大化避开威胁将得到一个相对长的航迹长度。常用的办法是设置一个加权系数来权衡最小化航迹长度与最大化威胁回避。本文除了考虑航迹长度代价与威胁代价外,还把无人飞行器必须满足的约束(如水平拐弯角,爬升/下滑角,飞行高度等)集成到代价函数中,即在代价函数中还考虑水平方向拐弯代价、垂直方向爬升/下滑代价和高度代价^[15]。

综合考虑这些代价,对给定航迹 X_i , 其总代价可表示为这些代价的加权和,即

$$J(X_i) = \sum_{k=1}^5 w_k J_k(X_i) \quad (1)$$

其中 $J_1 - J_5$ 分别为航迹的长度代价、威胁代价、拐弯代价、爬升/下滑代价和高度代价, w_k 为它们相应的权值($k = 1, \dots, 5$)。

航迹长度代价、威胁代价和拐弯代价的具体计算方法可参见文献[13],航迹的爬升/下滑代价与拐弯代价的计算方法类似。如果飞行器的离地高度在 $[z_{\min}, z_{\max}]$ 内,则航迹的高度代价为0,否则(对应离地高度大于 z_{\max} 和离地高度小于 z_{\min} 两种情形)施加相应的惩罚。

2 基于 HQPSO 算法的航迹规划

2.1 算法原理

在 PSO 中,每个粒子除了对应于一个位置向量 $X_i = [x_{i1}, x_{i2}, \dots, x_{i,3n}]$,还伴随着一个速度向量 $V_i = [v_{i1}, v_{i2}, \dots, v_{i,3n}]$,每个粒子根据自己的飞行经历和种群中其它粒子的飞行经历来调整自己的飞行,通过跟踪个体最优粒子 $P_i = [p_{i1}, p_{i2}, \dots, p_{i,3n}]$ 和全局最优粒子 $P_g = [p_{g1}, p_{g2}, \dots, p_{g,3n}]$ 来控制粒子的运动。粒子的速度和位置更新方程为^[9]

$$\begin{cases} v_{ij}^{k+1} = w v_{ij}^k + c_1 r_{1i}^k (p_{ij}^k - x_{ij}^k) + c_2 r_{2i}^k (p_{gj}^k - x_{ij}^k) \\ x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1}, \quad (i = 1, 2, \dots, m; j = 1, 2, \dots, 3n) \end{cases}$$

等价的向量形式速度和位置更新方程为

$$\begin{cases} V_i^{k+1} = w V_i^k + c_1 r_{1i}^k (P_i^k - X_i^k) + c_2 r_{2i}^k (P_g^k - X_i^k) \\ X_i^{k+1} = X_i^k + V_i^{k+1}, \quad (i = 1, 2, \dots, m) \end{cases}$$

其中 w 为惯性权系数, c_1 和 c_2 分别为认知项系数和社会项系数,也称为学习因子或加速因子。 r_1 和 r_2 为

[0,1] 之间的随机数,服从均匀分布。上标 k 表示第 k 次迭代。

在量子力学中,不可能同时用粒子位置和速度的确定值来描述粒子的量子状态,因为粒子具有波粒二象性,粒子的坐标和动量不可能同时具有确定值。因此,在 QPSO 中,粒子的状态不再由粒子的位置和速度来描述,取而代之的是粒子的波函数。波函数在空间中某一点的强度(即波函数振幅绝对值的平方)和粒子在该点出现的概率成正比。同时,控制粒子运动的方程不再是经典力学中的牛顿运动方程,则是量子力学中的薛定谔方程。孙俊通过求解薛定谔方程,得到了粒子在空间中某点出现的概率密度函数^[12],

$$f(x_{ij}^{k+1}) = (1/l_{ij}^k) \exp(-2|x_{ij}^{k+1} - q_{ij}^k|/l_{ij}^k)$$

其中局部吸引子 $q_{ij} = rp_{ij} + (1-r)p_{gj}$, r 为 [0,1] 之间的随机数; l_{ij} 为 Delta 势阱的特征长度。同时, q_{ij} 和 l_{ij} 分别为上述对称指数分布的位置参数和尺度参数。通过简单计算可以发现, x_{ij} 的数学期望 $E(x_{ij}) = q_{ij}$, 其物理意义为粒子位置分量 x_{ij} 将收敛到其相应的局部吸引子 q_{ij} 。

为了实现从量子搜索空间到经典解空间的变换,蒙特卡罗随机模拟被用来求解粒子在经典空间的位置更新方程,

$$x_{ij}^{k+1} = q_{ij}^k \pm (l_{ij}^k/2) \ln(1/u)$$

其中

$$l_{ij}^k = 2b | \text{mbest}_j^k - x_{ij}^k |, \text{mbest}^k = \sum_{i=1}^m P_i^k / m$$

从而粒子的位置更新方程可以表示为

$$x_{ij}^{k+1} = rp_{ij}^k + (1-r)p_{gj}^k \pm b | \sum_{i=1}^m p_{ij}^k / m - x_{ij}^k | \ln(1/u) \quad (2)$$

其中 u 为 [0,1] 上的服从均匀分布的随机数, b 为收缩扩张系数。

在每次迭代中,粒子的位置更新后,个体极值和全局极值的相应更新公式为

$$P_i^{k+1} = \underset{1 \leq j \leq k+1}{\text{argmin}} J(X_j^k) = \begin{cases} P_i^k, & \text{if } J(P_i^k) \leq J(X_i^{k+1}) \\ X_i^{k+1}, & \text{if } J(P_i^k) > J(X_i^{k+1}) \end{cases} \quad (3)$$

$$P_g^{k+1} = \underset{1 \leq i \leq m}{\text{argmin}} J(P_i^{k+1}) = \begin{cases} P_g^k, & \text{if } J(P_g^k) \leq J(P_i^{k+1}) \\ P_i^{k+1}, & \text{if } J(P_g^k) > J(P_i^{k+1}) \end{cases} \quad (4)$$

其中 J 为(1)式中定义的代价函数。

与 PSO 和 QPSO 算法不同的是, HQPSO 算法在整个种群中粒子位置更新后并不直接进入下一代,而是有选择性的对种群中的粒子进行繁殖操作。在整个种群中粒子位置更新后进入下一次迭代之前,以概率 pa 将粒子放进繁殖池。为了保护最优个体不被破坏,每次迭代中种群中的最优个体 $gbest$ 不参与繁殖操作。为了保持种群的多样性,在将粒子放进繁殖池前并不对种群中的粒子基于代价值进行排序,即一个粒子能否进入繁殖池与其本身的代价值无关,而只依赖于概率 pa 。

一旦繁殖池生成后,就以繁殖池中的个体为父个体,按照如下准则生成新个体。在繁殖池中随机选择两个个体作为父个体,并将父个体从繁殖池中删除,直到繁殖池中只有一个粒子或繁殖池为空。子个体的位置向量定义为父个体的位置向量随机加权,且满足两个子个体的位置向量之和与父个体的位置身量之和相等。为了保持种群大小不变,生成的新个体将替代原种群中的相应父个体。如图 1 所示,记这两个父个体分别为 p_1 和 p_2 , 生成的两个子个体分别为 s_1 和 s_2 , 则 p_1 和 p_2 、 s_1 和 s_2 分别为 \mathbf{R}^{3n} 中的两组向量,且 p_1 和 p_2 、 s_1 和 s_2 分别为 \mathbf{R}^{3n} 中同一个向量的两种不同分解形式,即

$$\begin{cases} s_1 = pb \cdot p_1 + (1-pb) \cdot p_2 \\ p_1 + p_2 = s_1 + s_2 \end{cases} \quad (5)$$

其中 pb 为 [0,1] 上的服从均匀分布的随机数。

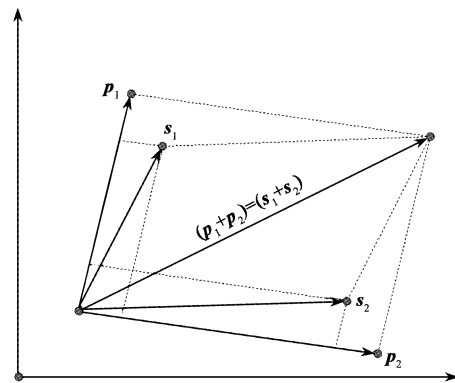


图 1 粒子繁殖示意图

Fig. 1 Illustration for breeding of particles

2.2 算法分析

QPSO 的一大亮点是在 PSO 的个体极值(个体最优粒子) $pbest$ 和全局极值(全局最优粒子) $gbest$

的基础上又引入了平均极值(平均最优粒子) $mbest$ 和局部吸引子 $Q_i = [q_{i1}, q_{i2}, \dots, q_{i,3n}]$ 。在每次迭代中,平均极值为种群中所有个体极值的算术平均,局部吸引子为个体极值和全局极值的随机加权平均。平均极值和局部吸引子的引入,使得粒子之间有了更多的信息交换,这也是 QPSO 性能比 PSO 性能优越的原因之一。

在 PSO 中,由于粒子的状态是由粒子的位置和速度来共同描述的,粒子的收敛以轨道的形式实现。尽管粒子速度的更新公式中有一定的随机因素,但粒子的速度总是在一个有限的范围内,因此粒子只能在一个有限的区域(解空间的一个子集)中搜索。正因为 PSO 不是在整个解空间中搜索,所以很容易陷入局部最优。而在 QPSO 中,谈论粒子的轨迹是毫无意义的,因为粒子的状态是由粒子的波函数来描述的,粒子能够以一定的概率搜索整个解空间中任意一个位置。根据量子力学中的隧道效应,粒子能以较小的概率运动到远离势阱的中心(即局部吸引子 Q_i)的某个位置,而这个位置可能比当前群体中的全局极值(全局最优粒子) $gbest$ 具有更好的适应值。因此,QPSO 比 PSO 具有更强的全局搜索能力。

而 HQPSO 是在 QPSO 的基础上的加入了繁殖机制,且没有破坏由 QPSO 算法得到的全局最优个体 $gbest$,从而由 HQPSO 得到的 $gbest$ 至少不会比由 QPSO 得到的 $gbest$ 差。设想这样一种情形,当繁殖池中的两个父个体分别为两个局部最优位置,根据繁殖原则,新生成的子个体为原父个体的随机加权和,所以新生成的子个体更容易跳出局部最优。因此,从理论上讲,HQPSO 算法的性能应该优于 QPSO 算法。

事实上,设在第 k 次迭代中,HQPSO 算法进行繁殖操作前的全局最优粒子为 $gbest1$,在繁殖操作过程中得到的全局最优粒子为 $gbest2$,下面分两种情况进行讨论:

(1) $J(gbest1) \leq J(gbest2)$, 即 $gbest1$ 优于 $gbest2$, 根据全局极值更新规则, HQPSO 算法在此次迭代中经过繁殖操作后整个种群的全局最优粒子仍然为 $gbest1$, 但这并不意味着繁殖为无效操作, 由(5)式知繁殖操作有助于增加种群的多样性和跳出局部最优;

(2) $J(gbest1) > J(gbest2)$, 即 $gbest2$ 优于

$gbest1$, 根据全局极值更新规则, HQPSO 算法在此次迭代中经过繁殖操作后整个种群的全局最优粒子由繁殖前的 $gbest1$ 更新为 $gbest2$, 即通过繁殖找到了代价更小的全局最优粒子。

而在第 k 次迭代中,由 QPSO 算法得到的全局最优粒子为 $gbest1$ 。综合(1)和(2)知,HQPSO 算法比 QPSO 算法具有更强的全局搜索能力。同时,本文的实验结果也从另一个角度验证了这一点。

2.3 算法步骤

(1) 选择合理的种群大小,粒子维数及最大迭代次数等算法参数;

(2) 读入战场环境信息和(1)中选定的各个参数;

(3) 随机初始化种群,得到一系列初始航迹;

(4) 根据(1)式中代价函数评价每条航迹;

(5) 分别计算局部吸引子和平均极值,并根据(2)式更新粒子的位置;

(6) 根据(3)-(4)式更新个体极值和全局极值;

(7) 根据(5)式进行繁殖操作;

(8) 重复(4)-(7),直到最大迭代次数条件满足;

(9) 输出全局极值作为生成的最优航迹;

3 实验结果与分析

3.1 实验数据与参数

本文是在 Intel Core2 Duo E7200, 2.53GHz, 2G 内存的 PC 机上进行仿真实验,运行环境为 Windows XP,编程环境为 MATLAB 7.6。实验使用图像大小为 700×700 ,分辨率为 $100m \times 100m$ 的数字地形高程图(DTED)和模拟生成的威胁数据。威胁的覆盖范围用圆表示。本文实验采用以下参数值:无人飞行器最低安全相对飞行高度为 20m,最高安全相对飞行高度为 100m;最大水平拐弯角和最大爬升/下滑角分别为 60° 和 30° ;PSO 中的惯性权系数从 0.9 到 0.4 线性递减,认知项系数和社会项系数均取为 2.0;QPSO 与 HQPSO 中的收缩扩张系数从 1.0 到 0.5 线性递减;HQPSO 中的繁殖概率为 0.5;记号“☆”为无人飞行器的起始点,位于战场的左上角;“◇”为目标点,位于战场的右下角;“•”为航迹节点。

综合考虑规划时间与航迹精度,通过反复实验,设定种群大小为 40;航迹节点为 5,即粒子维数为 15;最大迭代次数为 100。每种算法进行 30 次仿真

实验,分别记录每次实验迭代结束时生成的最优航迹的代价和每次实验所花费的时间。将 30 次实验花费的总时间先对实验次数取平均后再对迭代次数取平均,从而得到每次迭代的平均时间(秒)。设第 i 次实验生成的最优航迹的代价为 J_i ,则所有的 J_i 组成一个样本空间,样本容量为 30,即样本空间

$$\Omega = \{J_1, J_2, \dots, J_{30}\}$$

对这 30 个样本进行统计分析,分别求出样本均值 mean、样本极差 range 和样本标准差 std,相应的实验结果数据见表 1,

$$\begin{cases} \text{mean} = \sum_{i=1}^{30} J_i / 30, \text{ range} = \max_{1 \leq i \leq 30} \{J_i\} - \min_{1 \leq i \leq 30} \{J_i\} \\ \text{std} = \left(\sum_{i=1}^{30} (J_i - \text{mean})^2 / (30 - 1) \right)^{1/2} \end{cases}$$

样本均值的物理意义为平均最优航迹代价,样本极差的物理意义为最优航迹代价的最大值与最小值之差,样本标准差则从统计意义上反映了样本空间中的样本偏离样本均值的程度,即刻画了样本的波动程度。样本标准差越小,表明样本取值越集中,波动幅度越小;反之,表明样本取值越分散,波动幅度越大。

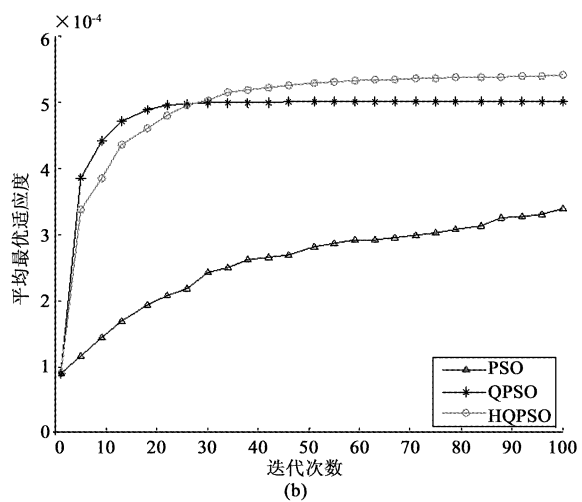
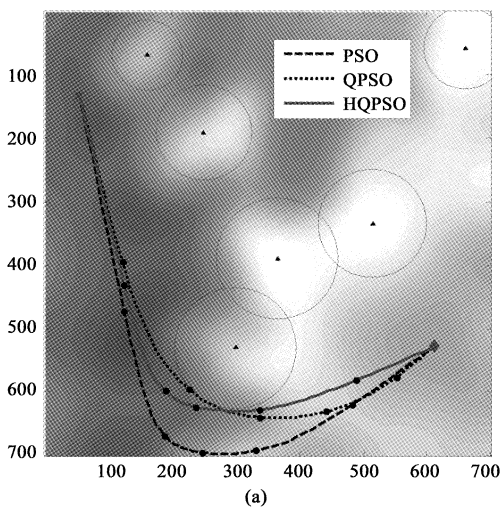


图 2 PSO 算法、QPSO 算法和 HQPSO 算法生成的最优航迹的平面投影及平均最优适应度的收敛曲线

Fig. 2 The best paths' plane projections and convergence curves of mean best fitness generated by the PSO, QPSO, and HQPSO

3.2 算法性能分析与比较

本文主要从以下两个方面来比较算法的性能:

(1) 解的质量,即由算法得到的最优平均航迹代价的大小。算法找到的解的质量越好,表明算法的全局搜索能力越好,不容易陷入局部最优。(2) 收敛

速度的快慢。在相同的条件下,经过较小的迭代次数即可得到同样质量的解的算法的收敛速度更快,反之,则收敛速度慢。从表 1 中可以看出,采用 HPQSO 算法得到平均最优航迹代价比采用 PSO 算法和 QPSO 算法得到的平均最优航迹代价都要小。

表 1 PSO 算法、QPSO 算法和 HQPSO 算法实验结果比较

Table 1 Comparison of experiment results produced by the PSO, QPSO, and HQPSO

算法	均值	极差	标准差	时间
PSO	2952.9	5482.1	1358.9	0.163932
QPSO	1997.6	593.3	155.2	0.158959
HQPSO	1849.8	436.1	118.1	0.162315

速度的快慢。在相同的条件下,经过较小的迭代次数即可得到同样质量的解的算法的收敛速度更快,反之,则收敛速度慢。从表 1 中可以看出,采用 HPQSO 算法得到平均最优航迹代价比采用 PSO 算法和 QPSO 算法得到的平均最优航迹代价都要小。

由 QPSO 和 HQPSO 得到的极差和标准差比由 PSO 算法得到的极差和标准差提高了一个数量级,这表明量子机制的引入使得 PSO 算法的稳定性得到进一步提高。由于 QPSO 和 HQPSO 算法中没有速度的更新操作,且它们的收敛速度都比 PSO 算法快(如图 2(b)所示),从而 QPSO 和 HQPSO 算法每代的迭代时间都比 PSO 算法小。而 HQPSO 算法的每次迭代需要的时间比 QPSO 算法多的原因是在

HQPSO 中粒子在位置更新后还进行了繁殖操作。从图 3 可以看出,三种算法均能很好地实现威胁回避,但由 QPSO 算法和 HQPSO 算法生成的航迹能更好地进行地形跟随。本文对不同的起始点和目标点以及不同的威胁环境进行了大量实验,实验结果表明在不同的战场环境下,由 HQPSO 算法生成的航迹始终优于由 PSO 算法和 QPSO 算法生成的航迹。

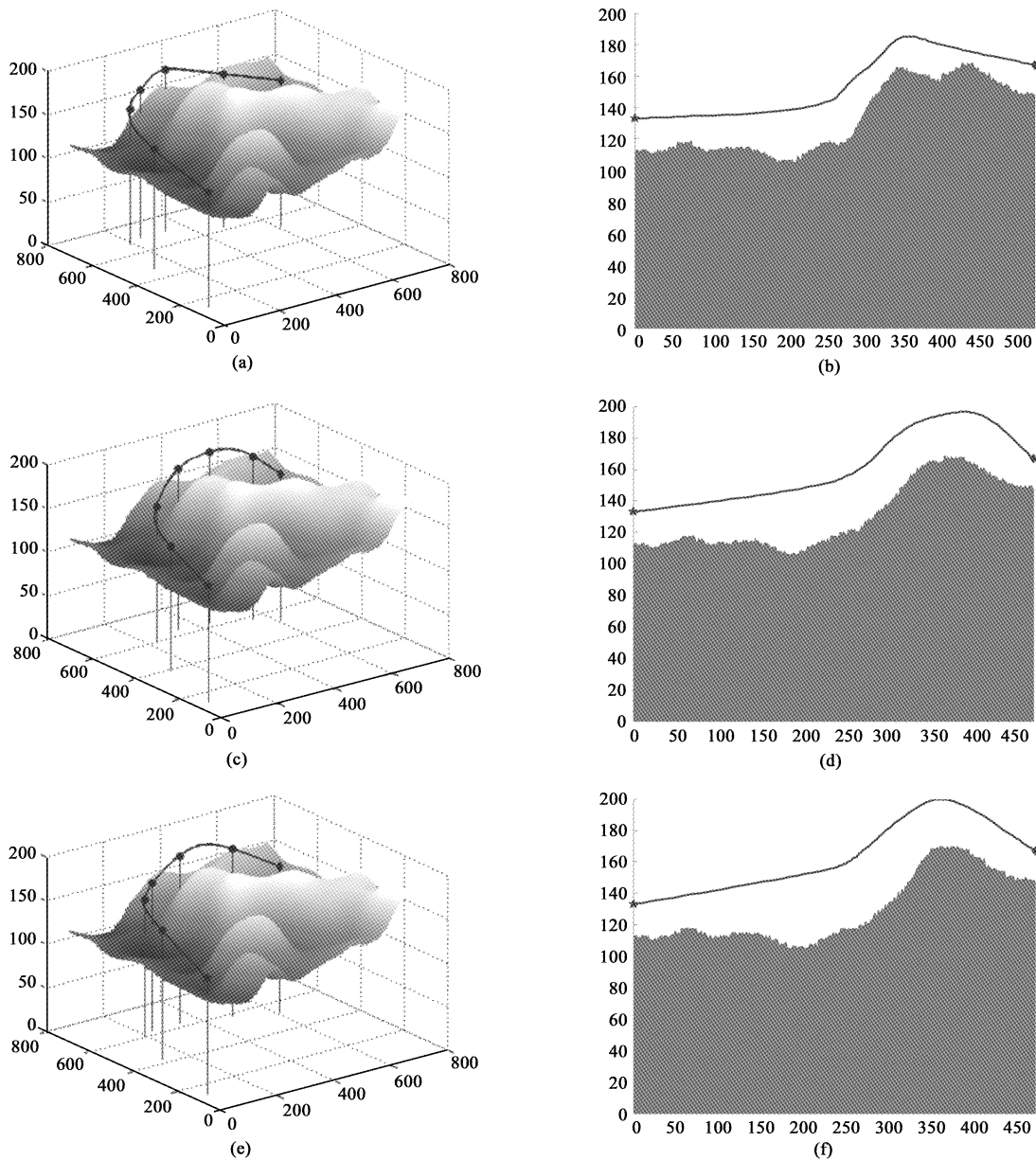


图 3 PSO 算法、QPSO 算法和 HQPSO 算法生成的最优航迹的三维显示与剖面显示

Fig. 3 The 3-D displays and vertical profiles of the best paths generated by the PSO, QPSO, and HQPSO

4 结 论

本文在 QPSO 算法的基础上提出了 HQPSO 算法,并成功地将该算法应用于无人飞行器的三维航迹规划。与基于 QPSO 算法和基于 PSO 算法的三维航迹规划方法相比,基于 HQPSO 算法的三维航迹规划方法能以更快的收敛速度找到更优的航迹,且该方法生成的三维航迹能够有效地实现威胁回避和更好地进行地形跟随。但是当航迹节点数较大时(如前文中的航迹节点数设为 100),将导致粒子维数急剧增加,无论是 PSO 算法和 QPSO 算法还是 HQPSO 算法都会增加规划失败的概率,即当待处理问题维数(即粒子维数)过高时,三种算法都面临规划失败的风险。如何改进现有算法使其适应高维问题是以后进一步研究的主要方向。

参 考 文 献

- [1] McLain T, Beard R. Coordination variables, coordination functions, and cooperative-timing missions[J]. *Journal of Guidance Control and Dynamics*, 2005, 28(1): 150 - 161.
- [2] Pettersson P O, Doherty P. Probabilistic roadmap based path planning for an autonomous unmanned helicopter[J]. *Journal of Intelligent and Fuzzy Systems*, 2006, 17(4): 395 - 405.
- [3] Melchior P, Orsoni B, Laviolle O, et al. Consideration of obstacle danger level in path planning using A* and fast-marching optimization: comparative study[J]. *Signal Processing*, 2003, 83(11): 2387 - 2396.
- [4] Szczerba R, Galkowski P, Glicktein I, et al. Robust algorithm for real-time route planning[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2000, 36(3): 869 - 878.
- [5] 李春华, 郑昌文, 周成平, 等. 一种三维航迹快速搜索方法[J]. *宇航学报*, 2002, 23(3): 13 - 17. [Li Chun-hua, Zheng Chang-wen, Zhou Cheng-ping, et al. Fast search algorithm for 3D-route planning[J]. *Journal of Astronautics*, 2002, 23(3): 13 - 17.]
- [6] Stentz A. Optimal and efficient path planning for partially-known environments[C]. *IEEE International Conference on Robotics and Automation*, San Diego, USA, May 8 - 13, 1994.
- [7] Zheng C, Ding M, Zhou C. Real-time route planning for unmanned air vehicle with an evolutionary algorithm[J]. *International Journal of Pattern Recognition and Artificial Intelligence*, 2003, 17(1): 63 - 81.
- [8] Zheng C, Li L, Xu F, et al. Evolutionary route planner for unmanned air vehicles[J]. *IEEE Transactions on Robotics*, 2005, 21(4): 609 - 620.
- [9] Kennedy J, Eberhart R. Particle swarm optimization[C]. *IEEE International Conference on Neural Networks*, Perth, Australia, November 27 - December 1, 1995.
- [10] Shi Y, Eberhart R. A modified particle swarm optimizer[C]. *IEEE World Congress on Computational Intelligence*, Anchorage, USA, May 3 - 8, 1998.
- [11] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(1): 58 - 73.
- [12] Jun S, Bin F, Wenbo X. Particle swarm optimization with particles having quantum behavior[C]. *The Congress on Evolutionary Computation*, Portland, USA, June 19 - 23, 2004.
- [13] Fu Y G, Ding M Y, Zhou C P, et al. Path planning for UAV based on quantum-behaved particle swarm optimization[C]. *SPIE International Symposium on Multispectral Image Processing and Pattern Recognition*, Yichang, China, October 30 - November 1, 2009.
- [14] Lovbjerg M, Rasmussen T, Krink T. Hybrid particle swarm optimizer with breeding and subpopulations[C]. *The Genetic and Evolutionary Computation Conference*, San Francisco, USA, July 7 - 11, 2001.
- [15] 丁明跃, 郑昌文, 周成平, 等. 无人飞行器航迹规划[M]. 北京: 电子工业出版社, 2009.

作者简介:傅阳光(1984 -),男,博士研究生,研究方向为飞行器任务规划、智能优化算法等。

通信地址:华中科技大学图像识别与人工智能研究所(430074)

电话:(027)87544512

E-mail:yangguangfu@yahoo.com.cn

(编辑:张宇平)