

Route Planning Based on Improved Particle Swarm Optimization Algorithm

Zijie Li¹, Haiguang Wei², Zhipeng Zhou¹, Chengping Zhou¹

¹Institute for Pattern Recognition and Artificial Intelligence, Multi-Spectral Information Processing Technologies Laboratory, Huazhong University of Science and Technology, Wuhan

²Jiangsu Automation Institute, Lianyungang
Email: lizijie0419@163.com

Received: Dec. 5th, 2011; revised: Jan. 12th, 2012; accepted: Jan. 29th, 2012

Abstract: To reduce the computational complexity, route planning algorithms often use hierarchical planning strategy to deal with different constraints separately during the planning process. Hierarchical planning is divided into outer planning and inner planning. Inner planning is a local planning on the basis of outer planning. This paper proposes an improved particle swarm optimization algorithm for inner planning, introducing a variation factor into PSO algorithm. The improved algorithm designs a specific perturbation operator to enhance its search ability. The simulation results show that the improved method is more effective to get a satisfactory path in the same environment. It enhances the speed of the overall algorithm.

Keywords: Route Planning; Hierarchical Planning; PSO; GA; Variability Factor

基于改进粒子群算法的航迹规划方法

李自杰¹, 魏海光², 周志鹏¹, 周成平¹

¹华中科技大学图像识别与人工智能研究所, 多谱信息处理技术实验室, 武汉

²江苏省自动化研究所, 连云港
Email: lizijie0419@163.com

收稿日期: 2011年12月5日; 修回日期: 2012年1月12日; 录用日期: 2012年1月29日

摘要: 为了降低航迹规划的计算复杂度, 航迹规划算法时常采用分层规划策略, 在规划过程中分开处理不同性质的约束条件; 分层规划包括外层规划和内层规划, 内层规划是在外层规划的基础上进行的局部规划。本文提出了一种用于内层规划的改进粒子群算法, 在粒子群算法中引入变异因子, 设计了特定的扰动算子, 提高了航迹寻优能力。仿真实验表明, 在相同约束的规划环境中, 改进方法较基本粒子群算法、基本遗传算法可以更快搜索到满足条件的航迹, 提高了整体算法的规划速度。

关键词: 航迹规划; 分层规划; 粒子群算法; 遗传算法; 变异因子

1. 引言

航迹规划问题是在一定的规划区域中, 综合考虑各类约束条件, 寻找从起始点到目标点的最优或次优路径^[1]。航迹规划问题本质上是一类多目标优化问题, 在规划过程中要考虑多方面因素的约束, 包括飞行器自身物理约束、规划环境约束等等。

在实际应用过程中, 为了确保航迹在性能指标下的最优性以及复杂环境下的规划速度, 往往不是单一使用一种规划算法, 而是采用分层规划策略, 分阶段使用不同的规划算法来处理不同性质的约束条件, 提高算法的效率。刘新等^[2]提出一种基于分层策略的规划方法, 用于解决传统规划方法的时间复杂性过大问题。

无人飞行器通常采用惯性制导,同时采用匹配和GPS辅助制导方式来修正飞行过程中产生的惯导误差。匹配辅助制导有地形匹配和景象匹配多种方式。匹配辅助制导方式的一个共同点是匹配区有特定的匹配方向,并且在整个规划空间上是离散分布的。

匹配区的特点表明航迹规划是在离散解空间搜索可行解。有些算法如粒子群优化算法和蚁群算法应用在离散空间存在局限性,遗传算法^[3]不受此限制,可以在离散空间从多点进行并行搜索,通过多次迭代进行交叉和变异运算来获得新的个体,在全局区域内有效搜索到匹配区。在搜索到满足约束的匹配区之后,还需要通过一次规划寻找从一个匹配区到另一个匹配区的最优或次优路径,即内层规划。

内层规划不考虑匹配区,搜索局部连续空间的最优航迹段,规划速度对整体算法效率有重要影响。粒子群算法(Particle Swarm Optimization, PSO)易于实现,收敛速度快,适合用于局部搜索。但粒子群算法也存在早熟收敛的现象,算法过早的收敛,不利于寻优,所以还需要对算法进行改进。

Shi等^[4]提出了惯性因子 w 线性递减的改进算法,提高了算法性能;van den Bergh^[5]通过使粒子群中最佳粒子始终处于运动状态,保证算法收敛到局部最优;Kennedy等^[6]研究粒子群的拓扑结构,分析粒子间的信息流,提出了一系列的拓扑结构。Baskar等^[7]提出了协同PSO算法,通过多群粒子协同优化来改进基本算法。

本文改进方法在基本粒子群算法的基础上设计了特定的变异因子参与迭代过程,当选择迭代对象是粒子种群最优个体时,用变异因子作用于最优个体,不进行公式迭代。下边将先简单描述分层策略,再详细介绍本文改进粒子群算法,最后给出仿真实验结果。

2. 分层策略

2.1. 约束条件

航迹规划要考虑多种因素的约束,不同的航迹规划问题考虑的约束条件不尽相同。主要有如下约束条件:

- 1) 航程约束
- 2) 匹配区最大/小距离约束
- 3) 禁飞区/威胁区约束
- 4) 最大/小爬升角度约束

- 5) 最大/小下滑角度约束
- 6) 机动修正和稳定距离约束
- 7) 最小拐弯半径约束
- 8) 最短水平弧长约束
- 9) 最大拐弯角约束

其中 1)、2)、3)在第一层遗传算法全局搜索时考虑处理;4)、5)在航迹高度规划时考虑处理;本文所讨论方法仅考虑 3)、6)、7)、8)、9)项约束。

2.2. 分层算法

根据约束条件性质的不同,采用分层策略将整个规划算法分为平面航迹规划和高度规划。高度规划在平面航迹规划成功后进行,主要考虑爬升、下滑约束。为了简化运算,平面航迹规划采用双层规划算法。

双层规划算法的思想是通过第一层的遗传算法在全局规划区域内搜索有效的匹配区,主要考虑总航程约束、匹配区距离约束、禁飞区和威胁区约束,即在满足总航程要求的前提下间隔一定距离搜索一个匹配区且要严格避开禁飞区,尽量避开威胁区。

第二层规划算法以第一层算法搜索到的匹配区作为起始点和结束点,考虑机动修正和稳定距离约束、转弯半径约束、转弯角约束、弧长约束,严格躲避禁飞区,尽量避开威胁区,搜索匹配区之间的最优或次优航迹。本文讨论的主要内容为用于第二层规划的粒子群算法的改进问题。

3. 粒子群算法概述

在粒子群算法^[8]中,每个个体称为一个粒子。种群由若干个粒子组成,种群规模过大会影响算法的运算速度和收敛性。粒子群算法的基本原理是粒子种群在搜索空间以一定的速度飞行,每个粒子在搜索时,考虑自己搜索到的历史最优位置和种群内其他粒子的历史最优位置,在此基础上进行位置的变化。

设 $x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{in})$ 为第 i 个粒子的位置矢量, $v_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{in})$ 为粒子 i 的飞行速度, $pbest_i = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{in})$ 为粒子 i 搜索到的最优位置, $gbest = (g_1, g_2, g_3, \dots, g_n)$ 为整个粒子种群搜索到的最优位置。

粒子速度和位置迭代方程如下:

$$v_i^{k+1} = wv_i^k + c_1r_1(pbest_i - x_i^k) + c_2r_2(gbest - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

其中 i 为粒子序号, k 是迭代次数, r_1 和 r_2 是 0~1 之间的随机数, 这两个参数用来保持种群的多样性。 c_1 和 c_2 为学习因子, 使粒子具有自我总结和向种群中优秀个体学习的能力, 从而不断向自己的历史最优位置以及种群内的历史最优位置靠近。

(1)式是粒子根据它上一次迭代的速度、它当前位置和自身最好位置与种群最好位置之间的距离来更新速度。然后粒子根据(2)式飞向新的位置。

4. 基于改进粒子群算法的内层规划

本文针对粒子群算法容易过早收敛到局部最优解、算法迭代后期搜索能力不足的缺陷以及规划时考虑禁飞区约束的特点提出改进粒子群算法, 引入变异因子。当局部规划区域中存在较大禁飞区域时, 整个粒子种群都会陷入禁飞区搜索不到可行解, 通过变异因子作用于种群最优个体可以搜索到禁飞区外的可行解。另外, 在算法迭代后期变异因子可以避免粒子种群收敛到局部最优解, 保持粒子种群的搜索能力。

4.1. 评价函数设计

评价函数用来给出粒子种群每个个体的适应值。所有约束条件均满足的个体称为可行个体, 违反任何一项约束条件的个体称为不可行个体。评价函数设计时遵循如下准则:

- 1) 可行个体的代价值为正, 代价值越小越优;
 - 2) 不可行个体的代价值为负, 约束违背量越小负代价值越大, 约束违背量越大负代价值越小;
 - 3) 可行个体总是优于不可行个体。
- 依照上述准则设计评价函数:

$$C_- = \sum_{i=1}^n (w_1 l_{Fi} + w_2 \theta_i + w_3 d_i) \quad (3)$$

$$C_+ = \sum_{i=1}^n (w_4 l_{Ti}) + L_{total} \quad (4)$$

$$F = \begin{cases} C_- & C_- \neq 0 \\ C_+ & C_- = 0 \end{cases} \quad (5)$$

(3)式用于计算不可行代价, l_{Fi} 表示个体第 i 段穿越禁飞区的长度, θ_i 表示转弯角违背约束量值, d_i 表示距离违背量值, w_1 、 w_2 、 w_3 为权系数, 均为负值, 禁飞代价为最严厉的违背代价, 所以 w_1 较 w_2 、 w_3 在绝对值上要大得多;

(4)式用于计算可行代价, l_{Ti} 表示个体第 i 段穿越威胁区的长度, w_4 为正的权系数, L_{total} 为个体的总航迹长度;

(5)式为个体评价函数公式, 在有负代价存在时仅考虑个体负代价, 负代价为零时才考虑个体正代价。

4.2. 粒子种群初始化

内层规划是在外层规划的基础上进行的, 所以内层规划的起始点和结束点都是匹配区点。内层规划不需要考虑匹配制导, 规划空间是连续的, 可以插入若干辅助节点来进行规划。

4.2.1. 转弯控制点

转弯控制点是内层规划中起辅助作用的节点。转弯控制点的前一节点(匹配区点或转弯控制点)与其连线方向为转弯前飞行方向, 转弯控制点与下一节点(匹配区点或转弯控制点)的连线方向为转弯后飞行方向。所以第一个转弯控制点必须在起始匹配区点的飞行方向上, 最后一个转弯控制点必须在结束匹配区点飞行方向反向延长线上, 其余转弯控制点可以在内层规划空间中任意分布(如图 1)。

4.2.2. 转弯控制点确定转弯

每一个转弯控制点可以确定一个转弯。如图 2 示, 起始匹配区点的飞行方向为 α , 结束匹配区点的飞行方向为 β , 转弯角 $\theta = \beta - \alpha$, r 为约束条件限定的

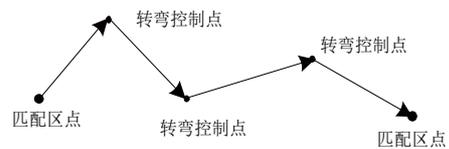


Figure 1. Distribution of control turning points
图 1. 转弯控制点分布示意图

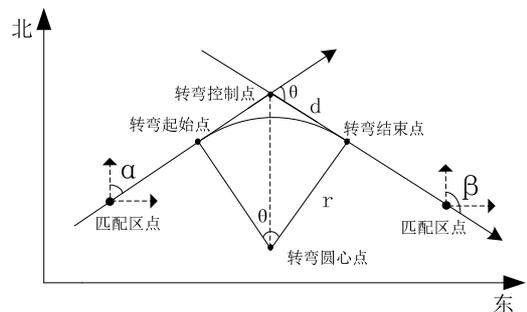


Figure 2. Determine turning via control turning point
图 2. 转弯示意图

最小转弯半径, 由公式 $d = r * \tan(\theta/2)$ 可以确定以最小转弯半径转弯时的距离 d , 再根据转弯控制点的位置和转弯前后的飞行方向可以确定转弯起始点、结束点的位置。

在满足各项约束的情况下, 可以对半径进行适当的调整来丰富解空间。

飞行器从匹配区出来需要一段机动修正距离, 进入匹配区前和完成一次转弯机动以后需要一段稳定距离。通常可以算出当前转弯控制点与前一点(匹配区点或转弯结束点)的距离 d_1 以及与下一点(转弯控制点或匹配区点)的距离 d_2 , 若下一点是转弯控制点则令 $d_2 = d_2/2$, 为下次转弯机动留出距离, adjustDistance 是机动修正(或稳定)距离, 在 $d_1 > \text{adjustDistance} + r * \tan(\theta/2)$ 且 $d_2 > \text{adjustDistance} + r * \tan(\theta/2)$ 同时满足的情况下, 按照如下步骤调整转弯起始点、结束点的位置以及转弯半径值:

- 1) d_{\min} 取 d_1, d_2 中的最小值;
- 2) 用 d_{\min} 减去 adjustDistance , 即 $d_{\min} = d_{\min} - \text{adjustDistance}$, 通常为 d 设定个上限值 D , 若是 $d_{\min} > D$, 则令 $d_{\min} = D$;
- 3) 令 d 为 $r * \tan(\theta/2) \sim d_{\min}$ 范围内的随机数值, 即 $d = r * \tan(\theta/2) + \text{rand}() * (d_{\min} - r * \tan(\theta/2))$;
- 4) 根据转弯控制点位置和 d 值确定转弯起始点、结束点的位置, 转弯半径为 $r = d * \cot(\theta/2)$ 。

4.2.3. 粒子个体初始化

粒子种群由长度相同的若干个体组成, 转弯控制节点数可以根据匹配区距离约束、转弯参数约束、机动修正和稳定距离约束来确定, 通常为 3~6 个。设两导航点的直线距离为 D_{pp} , 粒子长度为 n , 令 $\text{interval} = D_{pp} / (n - 1)$, 粒子个体初始化步骤如下:

- 1) 粒子头节点初始化为起始匹配区点;
- 2) 第一个转弯控制点初始化在起始匹配区点的飞行方向上, 距离起始匹配区点的距离为 $\text{distance} = \text{adjustDistance} + \text{specialStep} + \text{randis}$, adjustDistance 为飞出匹配区后的机动修正距离, specialStep 为距离裕量, 第三项表示在前两项距离的基础上在飞行方向上一定范围内随机节点;
- 3) 中间第 $i (1 < i < n - 1)$ 个转弯控制点的初始化:
 - a) 以两匹配区点连线上距离第一个匹配区点 $i * \text{interval}$ 远处节点为圆心;

b) 在 360 度范围内随机一个角度, 再随机一个半径值, 尽可能覆盖整个内层规划空间;

c) 将由圆心、半径、角度随机到的节点初始化为第 i 个转弯控制点;

4) 最后一个转弯控制点初始化在结束匹配区点飞行方向的反向延长线上, 距离结束匹配区点距离 $\text{distance} = \text{adjustDistance} + \text{specialStep} + \text{randis}$, adjustDistance 为进入匹配区前的稳定距离, 其他同 2);

5) 粒子末节点初始化为结束匹配区点;

6) 重复上述过程直至粒子种群初始化完成。

4.3. 扰动算子设计

扰动算子作用于种群最优个体 gbest 。扰动算子分为大扰动算子和小扰动算子。

在粒子群算法迭代初期种群最优个体 gbest 通常存在违背约束条件的情况, 例如穿越禁飞区或者转弯角度远大于限定的最大转弯角等等; 在算法迭代后期 gbest 是局部最优解, 粒子种群大量聚集于一个很小的区域内, 陷入局部最优。所以分阶段使用大、小扰动算子。大扰动算子: 1) 迭代初期扰动 gbest 向满足约束的方向进化以获得可行解; 2) 迭代后期扰动 gbest 跳出局部最优; 小扰动算子: 迭代后期扰动 gbest 逐步改进最优个体。迭代后期通过概率选择大、小扰动算子。

小扰动算子步骤:

1) 若是第一个(或最后一个)转弯控制点, 则以当前转弯控制点为中心点, 在当前飞行方向上中心点前后较小 s 距离内随机一个节点取代当前转弯控制点;

2) 若是其他转弯控制点, 则以当前转弯控制点为圆心, 在较小 r 范围内随机一个半径值, 在 0~360 范围内随机一个角度值, 将由圆心、半径、角度随机到的节点作为新的转弯控制点。

大扰动算子步骤:

1) 若是第一个(或最后一个)转弯控制点, 则以起始匹配区点为起点, 方向为飞行方向(或以结束匹配区点为起点, 方向为其飞行方向反方向), 以距离 $\text{distance} = \text{adjustDistance} + \text{specialStep} + \text{randis}$ 随机到的新节点取代当前转弯控制点;

2) 若是其他转弯控制点, 设 O 是前一控制点与下一控制点连线在起始、结束匹配区点连线上的投影段的中点, 以 O 为圆心, 在较大 r 范围内随机一个半

径值，在 0~360 范围内随机一个角度值，将由圆心、半径、角度随机到的节点作为新的转弯控制点。

扰动算子步骤如下：

- 1) 判断整条航迹是否穿越禁飞区，是则找到相应穿越禁飞区航迹段的转弯控制点，转 5)；
- 2) 判断整条航迹中是否有转弯角度大于最大转弯角，存在则找到相应转弯航迹段的转弯控制点，转 5)；
- 3) 判断整条航迹中是否有距离约束(机动修正或稳定距离)不满足的航迹段，存在则找到相应航迹段的转弯控制点，转 5)；
- 4) 若是整条航迹不存在违背上述约束的情况则为可行解，随机选择要扰动的转弯控制点和扰动算子，对转弯控制点施加扰动，转 6)；
- 5) 对选中的转弯控制点施加加大扰动；
- 6) 连接扰动后的个体，评价连接成功的个体，若是代价上比原先的 *gbest* 个体更优，则取代原先的 *gbest* 作为最优个体；
- 7) 扰动操作结束。

4.4. 改进后算法框架

改进后的 PSO 算法的流程图如图 3。

5. 实验结果与分析

本文方法在 Intel Core 2 Duo(2.93 GHz)PC 机上进行仿真实验，运行环境为 Windows XP，编程环境为 Visual Studio 2005。规划环境中禁飞区和威胁区用严格凸多边形来表示(禁飞区为红色，威胁区为黄色)，地形高程信息以栅格形式保存为地形高程图(DTED)，绿色区域为匹配区。仿真环境如图 4。

仿真实验用双层规划算法对五个不同的航迹规划任务进行规划，外层算法为基本遗传算法，内层分别用三种方法进行局部规划。实验对比如下表，规划时间为十次规划实验耗时的平均值。规划环境越复杂，

航程越长，规划耗时越多。任务规划耗时随着禁飞区和威胁区数量的增加而增加。规划结果对比如表 1。

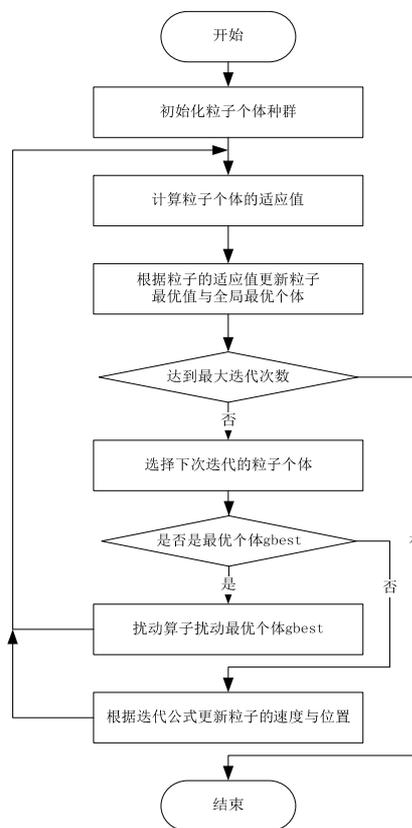


Figure 3. Flow char of improved PSO method
图 3. 改进粒子群算法流程图

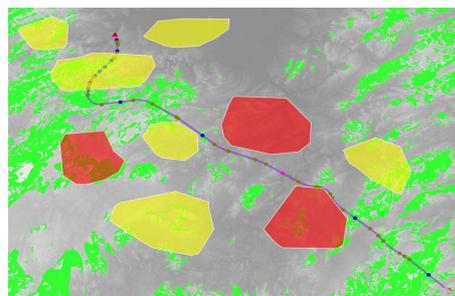


Figure 4. Simulation environment for path planning
图 4. 航迹规划仿真环境

Table 1. Comparison of planning result using different method
表 1. 规划结果比较

样例	禁飞区数量	威胁区数量	代价			规划时间/s		
			遗传算法	粒子群算法	本文方法	遗传算法	粒子群算法	本文方法
任务 1	6	5	2082.38	1647.55	1028.12	855.87	342.56	284.98
任务 2	11	6	1819.08	1830.22	1716.45	1206.41	1015.38	627.31
任务 3	11	6	2125.68	2699.71	2003.26	997.53	1062.27	559.59
任务 4	11	6	2342.91	3019.07	1690.43	922.06	736.83	523.81
任务 5	11	7	2458.36	3270.41	1351.17	1118.81	723.36	658.73

仿真实验结果表明对同一个任务进行规划时, 内层规划算法选择使用本文方法比基本遗传算法和基本粒子群算法在代价和规划耗时上都要优。本文算法很快就可以搜索到可行解, 并在此基础上快速向最优解靠近, 收敛速度快, 收敛结果更好, 整体规划耗时间少, 提高了整体算法规划速度。

在复杂的规划环境中局部搜索使用本文方法较其他两种方法可以通过较少次数的迭代更快地搜索到较优解, 而且本文方法对任务的复杂程度的变化适应性较好, 比较稳定, 这对于提高整体算法的效率很关键。

6. 结束语

本文基于基本粒子群算法提出一种改进方法, 在基本粒子群算法中设置变异因子。改进方法实现简单, 收敛速度快, 收敛效果好。仿真实验表明本文方法适合用于分层规划策略中的局部搜索, 可以通过较少次数的迭代搜索到较优的可行解, 明显提高了规划

速度, 使规划效率更高。

参考文献 (References)

- [1] 闵昌万, 袁建平. 军用飞行器航迹规划综述[J]. 飞行力学, 1998, 16(4): 14-19.
- [2] 刘新, 周成平, 俞琪等. 基于分层策略的三维航迹快速规划方法[J]. 宇航学报, 2010, 31(11): 2524-2529.
- [3] J. Holland. Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press, 1975.
- [4] Y. Shi, R. C. Eberhart. A modified particle swarm optimizer. Proceedings of IEEE International Conference on Evolutionary Computation, Anchorage, 1998, 69-73.
- [5] F. van den Bergh. An analysis of particle swarm optimizer. Proceedings of the 1998 conference of Evolutionary computation. Piscataway: IEEE Press, 1998: 67-73.
- [6] R. Mendes, J. Kennedy. The full informed particle swarm: Simpler, maybe better. IEEE Transaction on Evolutionary Computation, 2004, 8(3): 204-210.
- [7] S. Baskar, P. N. Suganthan. A novel concurrent particle swarm optimization. Proceedings of the 2004 Congress on Evolutionary Computation. Piscataway: IEEE Press, 2004: 792-796.
- [8] J. Kennedy, R. C. Eberhart. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, Perth, 1995: 1942-1948.