



(12) 发明专利申请

(10) 申请公布号 CN 102880186 A

(43) 申请公布日 2013. 01. 16

(21) 申请号 201210274571. 5

(22) 申请日 2012. 08. 03

(71) 申请人 北京理工大学

地址 100081 北京市海淀区中关村南大街 5 号

(72) 发明人 耿庆波 刘建英 刘世岳

(51) Int. Cl.

G05D 1/10(2006. 01)

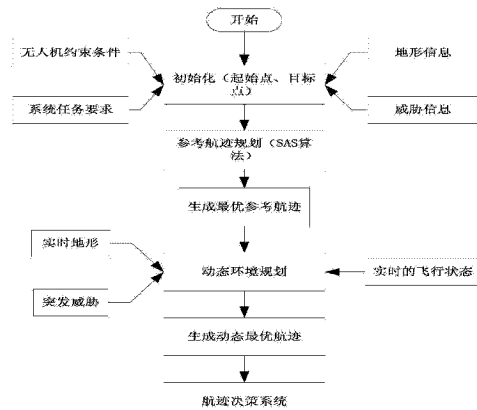
权利要求书 4 页 说明书 10 页 附图 4 页

(54) 发明名称

基于稀疏 A* 算法和遗传算法的航迹规划方法

(57) 摘要

本发明涉及一种基于稀疏 A* 算法和遗传算法的航迹规划方法,属于无人机航迹规划技术领域。本方法针对航迹规划的特点,首先利用 SAS 算法规划出初始参考航迹,它把约束条件结合到算法搜索中去,可以有效的删除搜索空间中的无用节点,缩短了搜索时间;当无人机实时飞行出现突发威胁时,利用遗传算法进行实时的航迹规划,生成局部最优或者近似最优的航迹,直到威胁消除,UAV 返回到原全局最优参考航迹继续飞行。该发明提出的方法具有较好的实时性和快速性,搜索到的航迹更逼近实际的无人机最优航迹;可应用于复杂环境下的机器人路径规划、城市车辆路径规划等技术领域。



1. 基于稀疏 A* 算法和遗传算法的航迹规划方法,其特征在于:包括如下步骤:

步骤 1:飞行环境建模;

UAV 飞行环境规划的三维规划空间表示为几何空间区域 $\{(x, y, z) | 0 \leq x \leq X_{max}, 0 \leq y \leq Y_{max}, 0 \leq z \leq Z_{max}\}$;采用数字化栅格的方法将规划空间离散为若干个单元栅格;

步骤 2:设置无人机航迹规划的初始条件,包含规划的起始点、目标点、威胁分布及地形信息,具体实现方法如下:

首先,给定经步骤 1 栅格化后的飞行环境规划空间中每个单元的高程信息;

然后,将飞行任务中的威胁模型化:将威胁的地理位置、高度和影响范围等威胁指数转化为离散化规划空间中的单元信息;

最后,在飞行环境模型和威胁模型建立后,根据任务要求,在离散化规划空间中找到起始点 S 和目标点 $G(x_g, y_g, z_g)$;所述起始点 S 位于栅格顶点;

步骤 3:根据任务要求,利用 SAS 算法生成全局最优的参考航迹;具体实现方法为:

步骤 3.1,在与当前航迹点相邻的栅格顶点中,排除不满足约束条件的点,将其余的相邻栅格顶点作为下一步可能的航迹点;

在初步判断出无人机下一步的可能航迹点后,从当前航迹点开始,计算下一步可能航迹点的代价值 g_i :

$$g_i = w_1 l_i^2 + w_2 h_i^2 + w_3 f_{TAI} \quad (3)$$

式中 l_i 表示第 i 段航迹的长度, h_i 为第 i 个飞行节点的飞行高度, f_{TAI} 表示第 i 段航迹的威胁指数; w_1 为 l_i 的系数、 w_2 为 h_i 的系数、 w_3 为 f_{TAI} 的系数;

步骤 3.2,计算步骤 3.1 得到的可能的航迹点到目标位置的代价估值 u_i , (x_i, y_i, z_i) 为第 i 点的坐标值, (x_g, y_g, z_g) 为目标点的坐标;

$$u_i = (x_i - x_g)^2 + (y_i - y_g)^2 + (z_i - z_g)^2 \quad (4)$$

步骤 3.3,计算步骤 3.1 得到的各个可能的航迹点总的航迹代价 f_i :

$$f_i = g_i + u_i \quad (5)$$

将 f_i 按大小顺序排列,放入 OPEN 表中,从建立的 OPEN 表中选择 f_i 最小的点作为下一个航迹点放入 CLOSED 表中;在初始化时,将起始点 S 放入 OPEN 表, CLOSED 表置空;

步骤 3.4,将步骤 3.3 得到的下一个航迹点作为下一步循环的当前航迹点,继续按照步骤 3.1 至步骤 3.3 所述方法寻找下一航迹点,直到某一点与目标点 G 的距离小于 L 为止,航迹搜索过程结束;

步骤 3.5,从得到的 CLOSED 表中的最后航迹点开始向上回溯,直到起始点,再加入目标点,最终得到从起始到目标的全局最优飞行参考航迹;

步骤 4:UAV 沿步骤 3 确定的最优参考航迹飞行,当出现突发威胁时,启动基于遗传算法的 UAV 三维航迹规划算法进行在线局部航迹规划,生成局部最优或者近似最优的航迹;具体实现过程如下:

步骤 4.1,根据突发威胁的情况,确定局部航迹规划的起始点、终止点和威胁指数;

步骤 4.2,根据步骤 4.1 确定的起始点、终止点,随机生成 m 条染色体,每条染色体表示一条飞行航迹;所有航迹的第一个和最后一个节点的坐标相同;所述 m 条染色体包含了可行航迹和不可行航迹;

步骤 4.3,计算步骤 4.2 得到的每一条航迹 J 的适应值 $F(J)$

$$F(J) = \begin{cases} C(J) & J \text{可行} \\ C_{\max} + \text{约束违背量} & J \text{不可行} \end{cases} \quad (6)$$

其中, $C(J)$ 为航迹代价, n 为航迹的节点个数;

$$C(J) = \sum_{i=1}^n (w_1 f_i^2 + w_2 h_i^2 + w_3 f_{TAi}) \quad (7)$$

C_{\max} 为 m 条染色体中所有可行航迹中的最大代价;

对于可行航迹,只需要根据 m 条染色体计算它的航迹代价;对于不可行航迹的适应值,与它本身的约束违背量以及 m 条染色体有关,如果 m 条染色体没有可行航迹, C_{\max} 为 0;本发明采用的航迹评价方法,不但包含航迹的代价,还要考虑航迹的各种约束条件;约束违背量为根据航迹约束条件进行正规化处理后得到;

步骤 4.4,选取局部最优航迹;具体方法为:

按照比例适应度分配的选取方法,从 m 条染色体中选取 S 个染色体组成繁殖池;对于某个染色体 i ,其适应度为 F_i ,则其被选择的概率 P_i 为

$$P_i = \frac{F_i}{\sum_{i=1}^m F_i} \quad (8)$$

步骤 4.5,根据预先设计的概率机制,选择相应的进化算子作用于被选出的 S 个染色体;由于每次进化都要利用平滑算子和定向扰动算子,所以,平滑算子和定向扰动算子被选择的概率为 1;

步骤 4.6,通过进化算子的作用生成新个体,将新生成的个体加入到染色体中,按照步骤 4.3 的方法计算新个体的适应值;

步骤 4.7,将扩展后的染色体中适应值较小的 S 个染色体删除,使其恢复到原来的染色体大小;

步骤 4.8,重复步骤 4.4 至步骤 4.7,判断是否满足终止条件;迭代过程进行到预先给定的最大次数,或者最优个体在若干次迭代中其适应值保持不变;当满足了终止条件,则从 m 条染色体中选出适应值最小的染色体作为所求航迹;

步骤 5:UAV 沿步骤 4 得到的局部最优航迹飞行,直到威胁消除,UAV 返回到步骤 3 得到的全局最优参考航迹继续飞行;

至此,就实现了基于 SAS 和遗传算法的航迹规划过程。

2. 根据权利要求 1 所述的基于稀疏 A* 算法和遗传算法的航迹规划方法,其特征在于:在实际规划过程中,将 x 、 y 、 z 坐标轴分别按不同分辨率划分单元大小,从而得到离散化的规划空间。

3. 根据权利要求 1 所述的基于稀疏 A* 算法和遗传算法的航迹规划方法,其特征在于:所述的进化算子包括如下形式:

(1) 交叉算子

交叉算子是指将两条航迹重新组合,生成两条新的航迹;被作用的航迹可以是可行的,也可以是不可行的;作为父个体的两个染色体节点数可以相同也可以不同;

(2) 扰动算子

扰动算子是指随机地改变航迹的一个中间节点的坐标;该航迹可以是可行的,也可以是不可行的;如果所选航迹是可行的,则在可行范围内加以较小扰动,以提高航迹的适应

值 ;如果所选航迹是不可行的,则可适当增大扰动幅度,以期获得可行的航迹 ;

(3) 插入算子

插入算子是指在两个相邻的航迹节点中间随机地插入一个新的航迹节点 ;一般地,在违背最低飞行高度约束,或者是穿越威胁区域的航迹段,采用该算子得到的子个体其性能可能提高 ;

(4) 删除算子

删除算子是指删除航迹的一个中间节点 ;该航迹可以是可行的,也可以是不可行的 ;如果所选航迹是不可行的,该中间节点可以随机的选取 ;如果所选航迹是可行的,则节点的选取需要给予某些启发式信息 ;如果没有信息表明需要删除节点,则以一个很小的概率来确定是否删除 ;

(5) 交换算子

交换算子是指交换任意两个相邻节点的先后顺序以减小拐弯角 ;该算子只作用于不可行航迹 ;如果两个相邻航迹节点不满足最大拐弯角约束,通过交换它们的顺序可能减小拐弯角 ;如果两相邻节点处航迹拐弯角越大,选择它们进行交换的概率越大 ;

(6) 平滑算子

平滑算子是指通过“切除尖角”将航迹平滑化 ;该算子随机选择一航迹节点,在与该航迹节点相连的两航迹段上各插入一新的节点,然后删除开始选择的节点 ;如果某节点处航迹拐弯角越大,选择它进行平滑的概率越大 ;该算子只作用于不可行航迹 ;

(7) 定向扰动算子

该算子为实现有效的目标进入方向而设计,且只作用于目标的前一个节点 ;如果要求满足有效的目标进入方向,则所有目标的航迹都将按该方向进行初始化 ;在进化过程中,目标的前一个节点只能用该算子进行作用 ;作用后的坐标由下式给出 :

$$\begin{cases} x'_{n-1} = x_n + t(x_{n-1} - x_n) \\ y'_{n-1} = y_n + t(y_{n-1} - y_n) \\ z'_{n-1} = z_n + t(z_{n-1} - z_n) \end{cases} \quad (9)$$

其中, x'_{n-1} , y'_{n-1} , z'_{n-1} 为经定向扰动算子作用后第 $n-1$ 点的坐标, $t > 0$ 是一个随机实数。

4. 根据权利要求 1 所述的基于稀疏 A* 算法和遗传算法的航迹规划方法,其特征在于 :所述 L 为根据栅格边长而预先给定的一个常数。

5. 根据权利要求 1 所述的基于稀疏 A* 算法和遗传算法的航迹规划方法,其特征在于 :所述约束条件为 :

1) 受到无人机最大爬升 / 下滑角的限制,当前航迹点的相邻点中正上和正下方的点不做为下一步的可能航迹点 ;

2) 最低飞行高度约束 :根据任务要求计算出航线离地高度的最小值 H_{min} ;搜索过程中对每一个给定的节点,只有当它与当前点之间的路径上的每一点的离地高度都大于或等于 H_{min} 时,则把它作为可能的航迹点 ;若小于 H_{min} ,则判断此点是无效航迹点 ;

3) 航迹距离约束条件 :搜索过程中长度大于航迹的最大允许长度 d_{max} 的航迹被认为是无效航迹,具体判断方法为

$$D(x) + SL(x) \leq d_{max} \quad (1)$$

其中 $D(x)$ 是从起始位置到当前点经过的真实距离, $SL(x)$ 是从当前点到达目标的直线距离; 不满足航迹距离约束条件的点不能作为下一航迹点;

4) 最大拐弯角约束: 受到最大拐弯角的限制, 下一步可能的航迹点必须保证无人机的实际拐弯角大于允许的最大拐弯角;

5) 最小航迹约束: 记最小航迹长度为 l_{\min} , i 表示第 i 段航迹, 该约束条件可表示为:

$$l_i \geq l_{\min} (i = 1, 2, \dots, n-1) \quad (2)。$$

基于稀疏 A* 算法和遗传算法的航迹规划方法

技术领域

[0001] 本发明涉及一种基于稀疏 A* 算法和遗传算法的航迹规划方法,属于无人机航迹规划技术领域。

背景技术

[0002] 无人机 (Unmanned Aerial Vehicle, UAV) 具有自动起降 (发射)、自动驾驶、自动导航、自动快速准确定位、自动信息采集与传送等功能,特别适合代替人在危险、恶劣和极限的环境下完成特定的工作和任务,于是在军事、测绘、航空航天、商业等领域有着广泛应用。作为任务规划系统 (Mission Planning System) 的核心之一,航迹规划是一门伴随现代信息技术而发展起来的高新技术。准确地说,无人机航迹规划就是在综合考虑 UAV 到达时间、燃料消耗、威胁以及飞行区域等因素的前提下,为无人机规划出一条最优,或者是最满意的飞行航迹,以保证圆满完成飞行任务。飞行器航迹规划问题的目标函数复杂,涉及到大量信息的处理,为了简化计算,航迹规划常采用分层规划:先在整个搜索空间进行参考航迹规划,然后缩小搜索空间,在以参考航迹为中线的走廊内进行细致航迹规划。

[0003] 无人机航路规划有多种方法,如动态规划法、A* 搜索法、Voronoi 图算法、人工势场法、蚁群算法、遗传算法等。它们之间各有优缺点,其中动态规划可以得到问题的最优解,但具有维数爆炸的特性,计算任务艰巨;A* 算法计算简单,容易实现,但启发函数的选取限制了了解的全局最优性;Voronoi 图法一般应用于低维路径规划中;人工势场法和蚁群算法则容易陷入局部最优。近年来,遗传算法因其具有大规模全局搜索能力而受到重视。

[0004] (1) A* 算法,2000 年,Robert J Szcaerba 等提出了一种改进的 A* 算法,称为稀疏 A* 算法 (Sparse A* Search),简称 SAS 算法。在规划环境中把不可行的航迹点列为不寻优区域,再将可行的航迹空间分割为多个子空间,在每个子空间中根据航迹代价函数来决定哪些点应该加入到 OPEN 表中。这种算法的搜索时间比以往的搜索时间大大的缩短,节省很大的计算机内存空间。

[0005] (2) 模拟退火算法, Kirkpatrick 等在 1982 年将退火的思想引入到了组合优化邻域,特别是解决 NP 完全组合优化问题。模拟退火算法源于对固体的物理退火过程的模拟,采用 Metropolis 接受准则,用冷却进度表的参数控制算法的进程,使算法在多项式时间里给出一个近似最优解。正是利用了该方法能够解决局部极值特点,在航迹规划中将许多具有全局优化能力的算法与模拟退火算法相结合,都得到了较好的效果。

[0006] (3) 蚁群算法 (Ant Algorithm),蚁群算法是通过蚂蚁的信息交流和相互协作来实现路径搜索,包括适应阶段和协作阶段。在适应阶段待选解根据已有的信息不断调整自身结构,协作阶段候选解之间通过信息交流来产生性能好的解。

[0007] (4) 粒子群优化算法 (Particle Swarm Optimization, PSO),粒子群优化算法是一种群体智能算法,是模拟鸟群和鱼群等觅食过程中迁徙和聚集的行为。PSO 算法是利用个体之间的相互协作来搜索最优解,利用的是生物群体信息共享的思想,既具有容易实现又具有深刻的智能背景,既适合科学研究,又特别适合工程应用的特点。在航迹规划方面这两种

方法也应用的比较多。

[0008] (5) 遗传算法, 遗传算法 (GeneticAlgorithm GA) 是基于自然选择和基因遗传学原理的搜索算法。遗传算法是一种新的全局最优搜索算法, 简单实用, 适合并行处理, 在许多邻域特别是在求解组合优化的问题中得到了广泛的应用, 成为求解函数优化问题的强有力的工具。遗传算法的基本操作包括编码、群体生成、适应度函数的构造、遗传操作等。遗传算法的优化速度受到搜索空间的大小、编码方式、适应度函数的复杂度、群体的规模、编码的长度、交叉及变异的概率以及代沟的影响。目前许多学者对于算法的改进大多都是在这几方面进行的。

[0009] 基于 A* 算法的 SAS 算法, 能规划一条全局最优的航迹, 但存在陷入局部搜索的问题。利用遗传算法则可以很好地解决上述问题, 遗传算法对全局的把握能力较强, 通过进化算子可以很快实现航迹的变化, 避免反复的局部搜索。但遗传算法是基于种群的, 容易陷入局部最优解。

发明内容

[0010] 本发明的目的是为解决无人机航迹规划容易存在陷入局部搜索的问题, 提出一种基于稀疏 A* 算法 (SAS) 和遗传算法的航迹规划方法。

[0011] 本发明将 SAS 与遗传算法结合起来进行航迹规划。在任务执行前, 先利用 SAS 算法生成一条全局最优的航迹。然后 UAV 一边沿生成的航迹向目标飞行, 一边探测环境信息。一旦 UAV 探测到新的环境信息妨碍到飞行任务的执行, 它立即将航迹前面一定距离的节点作为新的起始点进行局部航迹规划生成新的航迹。然后 UAV 沿新的航迹飞行, 知道安全时再返回到原来的航迹上沿原航迹飞行。

[0012] 本发明的目的是通过下述技术方案实现的。

[0013] 步骤 1 : 飞行环境建模。

[0014] UAV 飞行环境规划的三维规划空间表示为几何空间区域 $((x, y, z) | 0 \leq x \leq X_{max}, 0 \leq y \leq Y_{max}, 0 \leq z \leq Z_{max})$ 。本发明中采用数字化栅格的方法将规划空间离散为若干个单元栅格。在实际规划过程中, 将 x、y、z 坐标轴分别按不同分辨率划分单元大小, 从而得到离散化的规划空间。

[0015] 步骤 2 : 设置无人机航迹规划的初始条件, 包含规划的起始点、目标点、威胁分布及地形信息, 具体实现方法如下 :

[0016] 首先, 给定经步骤 1 栅格化后的飞行环境规划空间中每个单元的高程信息。

[0017] 然后, 将飞行任务中的威胁模型化 : 将威胁的地理位置、高度和影响范围等威胁指数转化为离散化规划空间中的单元信息。

[0018] 最后, 在飞行环境模型和威胁模型建立后, 根据任务要求, 在离散化规划空间中找到起始点 S 和目标点 $G(x_g, y_g, z_g)$ 。所述起始点 S 位于栅格顶点。

[0019] 步骤 3 : 根据任务要求, 利用 SAS 算法生成全局最优的参考航迹。具体实现方法为 :

[0020] 步骤 3.1, 在与当前航迹点相邻的栅格顶点中, 排除不满足约束条件的点, 将其余的相邻栅格顶点作为下一步可能的航迹点。所述约束条件为 :

[0021] 1) 受到无人机最大爬升 / 下滑角的限制, 当前航迹点的相邻点中正上和正下方的

点不做为下一步的可能航迹点。

[0022] 2) 最低飞行高度约束:根据任务要求计算出航线离地高度的最小值 H_{\min} 。搜索过程中对每一个给定的节点,只有当它与当前点之间的路径上的每一点的离地高度都大于或等于 H_{\min} 时,则把它作为可能的航迹点。若小于 H_{\min} ,则判断此点是无效航迹点。

[0023] 3) 航迹距离约束条件:搜索过程中长度大于航迹的最大允许长度 d_{\max} 的航迹被认为是无效航迹,具体判断方法为

$$[0024] \quad D(x)+SL(x) \leq d_{\max} \quad (1)$$

[0025] 其中 $D(x)$ 是从起始位置到当前点经过的真实距离, $SL(x)$ 是从当前点到达目标的直线距离。不满足航迹距离约束条件的点不能作为下一航迹点。

[0026] 4) 最大拐弯角约束:受到最大拐弯角的限制,下一步可能的航迹点必须保证无人机的实际拐弯角大于允许的最大拐弯角。

[0027] 5) 最小航迹约束:记最小航迹长度为 l_{\min} , i 表示第 i 段航迹,该约束条件可表示为:

$$[0028] \quad l_i \geq l_{\min} (i = 1, 2, \dots, n-1) \quad (2)$$

[0029] 所述 1)、2)、3) 4) 和 5) 的约束条件需同时满足。

[0030] 在初步判断出无人机下一步的可能航迹点后,从当前航迹点开始,计算下一步可能航迹点的代价值 g_i :

$$[0031] \quad g_i = w_1 l_i^2 + w_2 h_i^2 + w_3 f_{TAi} \quad (3)$$

[0032] 式中 l_i 表示第 i 段航迹的长度, h_i 为第 i 个飞行节点的飞行高度, f_{TAi} 表示第 i 段航迹的威胁指数。 w_1 为 l_i 的系数、 w_2 为 h_i 的系数、 w_3 为 f_{TAi} 的系数。

[0033] 步骤 3.2,计算步骤 3.1 得到的可能的航迹点到目标位置的代价估值 u_i , (x_i, y_i, z_i) 为第 i 点的坐标值, (x_g, y_g, z_g) 为目标点的坐标。

$$[0034] \quad u_i = (x_i - x_g)^2 + (y_i - y_g)^2 + (z_i - z_g)^2 \quad (4)$$

[0035] 步骤 3.3,计算步骤 3.1 得到的各个可能的航迹点总的航迹代价 f_i :

$$[0036] \quad f_i = g_i + u_i \quad (5)$$

[0037] 将 f_i 按大小顺序排列,放入 OPEN 表中,从建立的 OPEN 表中选择 f_i 最小的点作为下一个航迹点放入 CLOSED 表中。特殊地,在初始化时,将起始点 S 放入 OPEN 表,CLOSED 表置空。

[0038] 步骤 3.4,将步骤 3.3 得到的下一个航迹点作为下一步循环的当前航迹点,继续按照步骤 3.1 至步骤 3.3 所述方法寻找下一航迹点,直到某一点与目标点 G 的距离小于 L 为止,航迹搜索过程结束。所述 L 为根据栅格边长而预先给定的一个常数。

[0039] 步骤 3.5,从得到的 CLOSED 表中的最后航迹点开始向上回溯,直到起始点,再加入目标点,最终得到从起始到目标的全局最优的飞行参考航迹。

[0040] 生成的飞行航迹表示为三维空间中的一系列航迹点,相邻航迹点之间用直线段连接。一条航迹实际上是一组节点序列 $\{S, P_1, \dots, P_{n-1}, G\}$, P_1, \dots, P_{n-1} 为中间航迹节点。

[0041] 步骤 4:UAV 沿步骤 3 确定的最优参考航迹飞行,当出现突发威胁时,启动基于遗传算法的 UAV 三维航迹规划算法 (Evolutionary Route Planner, ERP) 进行在线局部航迹规划,生成局部最优或者近似最优的航迹。具体实现过程如下:

[0042] 步骤 4.1,根据突发威胁的情况,确定局部航迹规划的起始点、终止点和威胁指数。

[0043] 步骤 4.2, 根据步骤 4.1 确定的起始点、终止点, 随机生成 m 条染色体, 每条染色体表示一条飞行航迹。所有航迹的第一个和最后一个节点的坐标相同。所述 m 条染色体包含了可行航迹和不可行航迹。

[0044] 步骤 4.3, 计算步骤 4.2 得到的每一条航迹 J 的适应值 $F(J)$

[0045]

$$F(J) = \begin{cases} C(J) & J \text{ 可行} \\ C_{\max} + \text{约束违背量} & J \text{ 不可行} \end{cases} \quad (6)$$

[0046] 其中, $C(J)$ 为航迹代价, n 为航迹的节点个数。

$$C(J) = \sum_{i=1}^n (w_1 l_i^2 + w_2 h_i^2 + w_3 f_{TAI}) \quad (7)$$

[0048] C_{\max} 为 m 条染色体中所有可行航迹中的最大代价。

[0049] 对于可行航迹, 只需要根据 m 条染色体计算它的航迹代价; 对于不可行航迹的适应值, 与它本身的约束违背量以及 m 条染色体有关, 如果 m 条染色体没有可行航迹, C_{\max} 为 0。本发明采用的航迹评价方法, 不但包含航迹的代价, 还要考虑航迹的各种约束条件。

[0050] 所述航迹的各种约束条件在步骤 3.1 中提到, 具体包括最小航迹长度、最大拐弯角、最大爬升/下滑角、航迹距离约束和最低高度限制。约束违背量为根据航迹约束条件进行正规化处理后得到。

[0051] 步骤 4.4, 选取局部最优航迹。具体方法为:

[0052] 按照比例适应度分配的选取方法, 从 m 条染色体中选取 S 个染色体组成繁殖池。对于某个染色体 i , 其适应度为 F_i , 则其被选择的概率 P_i 为

$$P_i = \frac{F_i}{\sum_{i=1}^S F_i} \quad (8)$$

[0054] 步骤 4.5, 根据预先设计的概率机制, 选择相应的进化算子作用于被选出的 S 个染色体。由于每次进化都要利用平滑算子和定向扰动算子, 所以, 平滑算子和定向扰动算子被选择的概率为 1。

[0055] 所述的进化算子包括如下形式:

[0056] (1) 交叉算子

[0057] 交叉算子是指将两条航迹重新组合, 生成两条新的航迹。被作用的航迹可以是可行的, 也可以是不可行的。作为父个体的两个染色体节点数可以相同也可以不同。

[0058] (2) 扰动算子

[0059] 扰动算子是指随机地改变航迹的一个中间节点的坐标。该航迹可以是可行的, 也可以是不可行的。如果所选航迹是可行的, 则在可行范围内加以较小扰动, 以提高航迹的适应值; 如果所选航迹是不可行的, 则可适当增大扰动幅度, 以期获得可行的航迹。

[0060] (3) 插入算子

[0061] 插入算子是指在两个相邻的航迹节点中间随机地插入一个新的航迹节点。一般地, 在违背最低飞行高度约束, 或者是穿越威胁区域的航迹段, 采用该算子得到的子个体其性能可能提高。

[0062] (4) 删除算子

[0063] 删除算子是指删除航迹的一个中间节点。该航迹可以是可行的, 也可以是不可行

的。如果所选航迹是不可行的,该中间节点可以随机的选取;如果所选航迹是可行的,则节点的选取需要给予某些启发式信息。如果没有信息表明需要删除节点,则以一个很小的概率来确定是否删除。

[0064] (5) 交换算子

[0065] 交换算子是指交换任意两个相邻节点的先后顺序以减小拐弯角。该算子只作用于不可行航迹。如果两个相邻航迹节点不满足最大拐弯角约束,通过交换它们的顺序可能减小拐弯角。如果两相邻节点处航迹拐弯角越大,选择它们进行交换的概率越大。

[0066] (6) 平滑算子

[0067] 平滑算子是指通过“切除尖角”将航迹平滑化。该算子随机选择一航迹节点,在与该航迹节点相连的两航迹段上各插入一新的节点,然后删除开始选择的节点。如果某节点处航迹拐弯角越大,选择它进行平滑的概率越大。该算子只作用于不可行航迹。

[0068] (7) 定向扰动算子

[0069] 该算子为实现有效的目标进入方向而设计,且只作用于目标的前一个节点。如果要求满足有效的目标进入方向,则所有目标的航迹都将按该方向进行初始化;在进化过程中,目标的前一个节点只能用该算子进行作用。作用后的坐标由下式给出:

$$[0070] \quad \begin{cases} x'_{n-1} = x_n + t(x_{n-1} - x_n) \\ y'_{n-1} = y_n + t(y_{n-1} - y_n) \\ z'_{n-1} = z_n + t(z_{n-1} - z_n) \end{cases} \quad (9)$$

[0071] 其中, x'_{n-1} , y'_{n-1} , z'_{n-1} 为经定向扰动算子作用后第 $n-1$ 点的坐标, $t > 0$ 是一个随机实数。

[0072] 步骤 4.6,通过进化算子的作用生成新个体,将新生成的个体加入到染色体中,按照步骤 4.3 的方法计算新个体的适应值。

[0073] 步骤 4.7,将扩展后的染色体中适应值较小的 S 个染色体删除,使其恢复到原来的染色体大小。

[0074] 步骤 4.8,重复步骤 4.4 至步骤 4.7,判断是否满足终止条件:迭代过程进行到预先给定的最大次数,或者最优个体在若干次迭代中其适应值保持不变。当满足了终止条件,则从 m 条染色体中选出适应值最小的染色体作为所求航迹。

[0075] 步骤 5:UAV 沿步骤 4 得到的局部最优航迹飞行,直到威胁消除,UAV 返回到步骤 3 得到的全局最优参考航迹继续飞行。

[0076] 至此,就实现了基于 SAS 和遗传算法的航迹规划过程。

[0077] 有益效果

[0078] 使用本发明将 SAS 算法与遗传算法相结合进行航迹规划。采用稀疏 A* 算法 (SAS) 进行参考航迹规划,得到全局最优的航迹规划。由于 SAS 算法是 A* 算法的改进,它将无人机的物理约束条件结合到搜索算法中,删除了一些不可用节点,在一定程度上也减小了无人机航迹规划的计算量。

[0079] 在线航迹规划中,对无人机的实时性要求很严格,采用遗传算法进行在线航迹规划,得到航迹的局部调整,满足无人机对实时性的要求。

[0080] 传统 A* 算法要收敛到最优解需要很长的时间和极大的内存需求,通常随规划区域的增大呈指数增长,而且生成的环境不能满足无人机的约束条件。本发明采用 SAS 算法

进行 UAV 的三维环境搜索,在扩展节点时,通过把约束条件结合到搜索算法中去,有效地减小了搜索空间,缩短了搜索时间;搜索到的航迹更逼近实际的无人机最优航迹。同时,本发明也可应用于复杂环境下的机器人路径规划、城市车辆路径规划等技术领域。

附图说明

- [0081] 图 1 为本发明的航迹规划流程图;
- [0082] 图 2 为具体实施方式中数字化栅格的规划空间表示,图中数字是高程信息;
- [0083] 图 3 为本发明的在线航迹规划流程图,表示了在线航迹规划过程的具体过程;
- [0084] 图 4 为具体实施方式中遗传算法中的染色体结构,包括三个坐标值和无人机的状态值;
- [0085] 图 5 为具体实施方式中遗传算法中的七种进化算子;
- [0086] 图 6 为具体实施方式中静态环境中航迹规划的航迹图;
- [0087] 图 7 为具体实施方式中动态环境中航迹规划的航迹图。

具体实施方式

[0088] 为了更好的说明本发明的目的和优点,下面结合附图和实施例对本技术方案做进一步详细说明。

[0089] 步骤 1:飞行环境建模。

[0090] UAV 飞行环境规划的三维规划空间表示为几何空间区域 $((x, y, z) | 0 \leq x \leq X_{max}, 0 \leq y \leq Y_{max}, 0 \leq z \leq Z_{max})$ 。本发明中采用数字化栅格的方法将规划空间离散为若干个单元。在实际规划过程中,将 x, y, z 坐标轴划分为单位 1 的单元,即每个栅格的变长为单位 1,从而得到离散化的规划空间。

[0091] 步骤 2:设置无人机航迹规划的初始条件,包含规划的起始点、目标点、威胁分布及地形信息,具体实现方法如下:

[0092] 首先,给定经步骤 1 栅格化后的飞行环境规划空间中每个单元的高程信息。如图 2 所示。

[0093] 然后,将飞行任务中的威胁模型化:将威胁的地理位置、高度和影响范围等威胁指数转化为离散化规划空间中的单元信息。

[0094] 最后,在飞行环境模型和威胁模型建立后,根据任务要求,在离散化规划空间中找到起始点 S 和目标点 $G(x_g, y_g, z_g)$ 。所述起始点 S 位于栅格顶点。目标点不一定在栅格顶点上。

[0095] 步骤 3:根据任务要求,利用 SAS 算法生成全局最优的参考航迹。具体实现方法为:

[0096] 步骤 3.1,当前航迹点的下一步可能的航迹点为与其相邻的栅格顶点,但有一些是不满足约束条件的,不能做为下一步的可能航迹点,判断依据为:

[0097] 1) 无人机最大爬升/下滑角的限制,当前航迹点的相邻点中正上和正下方的点不做为下一步的可能航迹点。因此,所述当前航迹点的相邻点为离散化规划空间中的栅格顶点,且不包括正上和正下方的点。假定最大允许爬升/下滑角为 θ ,该约束表示为:

$$[0098] \quad \frac{|z_i - z_{i-1}|}{|a_i|} \leq \tan(\theta) \quad (i = 1, 2, \dots, n) \quad (10)$$

[0099] 2) 最大拐弯角的限制, 记 $a_i = (x_i - x_{i-1}, y_i - y_{i-1})$, 设最大允许拐弯角为 Φ , 则最大拐弯角约束表示为:

$$[0100] \quad \frac{a_i^T a_{i+1}}{|a_i| |a_{i+1}|} \leq \cos(\Phi) \quad (i = 2, \dots, n) \quad (11)$$

[0101] 3) 最低飞行高度约束: 根据任务要求计算出航线离地高度的最小值 H_{\min} 。搜索过程中对每一个给定的节点 i , 只有当它与当前点之间的路径上的每一点的离地高度都大于或等于 H_{\min} 时, 则把它作为可能的航迹点。若小于 H_{\min} , 就认为此点是无效航迹点。则该约束可表示为:

$$[0102] \quad H_i \geq H_{\min} \quad (i = 1, 2, \dots, n) \quad (12)$$

[0103] 4) 航迹距离约束条件为: 搜索过程中长度大于航迹的最大允许长度 d_{\max} 的航迹被认为是无效航迹。其中 $D(x)$ 是从起始位置到当前点经过的真实距离, $SL(x)$ 是从当前点到达目标的直线距离。具体判断方法为

$$[0104] \quad D(x) + SL(x) \leq d_{\max} \quad (13)$$

[0105] 5) 最小航迹约束的限制, 记最小航迹长度为 l_{\min} , i 表示第 i 段航迹, 该约束条件可表示为:

$$[0106] \quad l_i \geq l_{\min} \quad (i = 1, 2, \dots, n-1) \quad (14)$$

[0107] 根据 1)、2)、3)、4) 和 5) 的判断条件, 在当前航迹点的相邻点中排除了无效航迹点, 初步判断出无人机下一步的可能航迹点。

[0108] 然后从当前航迹点开始, 计算每一个下一步可能航迹点的代价值 g_i :

$$[0109] \quad g_i = w_1 l_i^2 + w_2 h_i^2 + w_3 f_{TAi} \quad (15)$$

[0110] 式中 l_i 表示第 i 段航迹的长度, h_i 为第 i 个飞行节点的飞行高度, f_{TAi} 表示第 i 段航迹的威胁指数。 w_1 为 l_i 的系数、 w_2 为 h_i 的系数、 w_3 为 f_{TAi} 的系数。通过 w_1 对 l_i 的限制来减少飞行距离, 缩短飞行器在战区飞行的时间, 同时又能节省油耗; 通过 w_2 对 h_i 的调节能够利用地形的遮挡作用减小被雷达发现的概率, 达到隐蔽飞行的目的; w_3 限制 UAV 不要与已知的地面威胁距离太近, 使得 UAV 尽量在威胁较小的区域飞行。

[0111] 步骤 3.2, 计算每一个下一步可能航迹点到目标点 $G(x_g, y_g, z_g)$ 的代价估值 u_i , (x_i, y_i, z_i) 为第 i 点的坐标值。

$$[0112] \quad u_i = (x_i - x_g)^2 + (y_i - y_g)^2 + (z_i - z_g)^2 \quad (16)$$

[0113] 步骤 3.3, 计算每一个下一步可能航迹点的总的航迹代价 f_i

$$[0114] \quad f_i = g_i + u_i \quad (17)$$

[0115] 将 f_i 按大小顺序排列, 放入 OPEN 表中, 从建立的 OPEN 表中选择 f_i 最小的点作为下一个航迹点放入 CLOSED 表中。初始化时, 将起始点 S 放入 OPEN 表, CLOSED 表置空。

[0116] 步骤 3.4, 将步骤 3.3 得到的下一个航迹点作为当前航迹点, 继续按照步骤 3.1 至步骤 3.3 所述方法寻找下一航迹点, 直到某一点与目标点 G 的距离小于 L 为止, 航迹搜索过程结束。

$$[0117] \quad L = \frac{\sqrt{2}}{2} \quad (18)$$

[0118] 步骤 3.5, 从得到的 CLOSED 表中的最后航迹点开始向上回溯, 直到起始点, 再加入目标点, 最终得到从起始到目标的全局最优的参考航迹。

[0119] 生成的飞行航迹表示为三维空间中的一系列航迹点, 相邻航迹点之间用直线段连接。一条航迹实际上是一组节点序列 $\{S, P_1, \dots, P_{n-1}, G\}$, P_1, \dots, P_{n-1} 为中间航迹节点。

[0120] 当搜索空间很大时, 搜索算法所需要的内存空间是巨大的。本发明通过限制 OPEN 表的大小减少了内存需求。将 OPEN 表中的节点进一步按从小到大的顺序排列, 当这个固定大小的 OPEN 表被装满了数据而又有新的节点需要存储是, 比较 OPEN 表中代价最大的节点和新节点的代价。若新节点的代价小, 则删除代价最大的节点, 并将新节点按代价大小插入 OPEN 表中。否则丢弃新节点。

[0121] 步骤 4: UAV 沿步骤 3 确定的最优参考航迹飞行, 当出现突发威胁时, 启动基于遗传算法的 UAV 三维在线航迹规划算法进行在线局部航迹规划, 在线航迹规划的流程如图 3 所示, 生成局部最优或者近似最优的航迹。具体实现过程如下:

[0122] 步骤 4.1, 根据突发威胁的情况, 确定局部航迹规划的起始点、终止点和威胁指数。

[0123] 步骤 4.2, 根据步骤 4.1 确定的起始点、终止点, 随机生成 m 条染色体, 每条染色体表示一条飞行航迹。所有航迹的第一个和最后一个节点的坐标相同。所述 n 条染色体包含了可行航迹和不可行航迹。

[0124] 本发明设计了一种变长度的实值基因编码方式, 染色体的每一个节点除了记录航迹节点的空间坐标 (x_i, y_i, z_i) 外, 还包括状态变量 b_i 。如图 4 所示。

[0125] 步骤 4.3, 计算步骤 4.2 得到的每一条航迹 J 的适应值 $F(J)$

[0126]

$$F(J) = \begin{cases} C(J) & J \text{ 可行} \\ C_{\max} + \text{约束违背量} & J \text{ 不可行} \end{cases} \quad (19)$$

[0127] 其中, $C(J)$ 为航迹代价, n 为航迹的节点个数。

$$[0128] \quad C(J) = \sum_{i=1}^n (w_1 l_i^2 + w_2 h_i^2 + w_3 f_{TAi}) \quad (20)$$

[0129] C_{\max} 为 m 条染色体中所有可行航迹中的最大代价。在上式中, 不可行航迹的适应值不但与它本身的约束违背量有关, 还与 m 条染色体有关, 对于可行航迹, 只需要计算它的航迹代价即可。如果 m 条染色体没有可行航迹, C_{\max} 为 0。本发明采用的航迹评价方法, 不但包含航迹的代价, 还要考虑航迹的各种约束条件。

[0130] 所述航迹的各种约束条件具体包括最小航迹长度、最大拐弯角、最大爬升 / 下滑角、航迹距离约束和最低高度限制。

[0131] 步骤 4.4, 按照比例适应度分配的选取方法, 从 m 条染色体中选取 S 个染色体组成繁殖池。对于某个染色体 i , 其适应度为 F_i , 则其被选择的概率 P_i 为

$$[0132] \quad P_i = \frac{F_i}{\sum_{i=1}^S F_i} \quad (21)$$

[0133] 步骤 4.5, 根据预先设计的概率机制, 选择相应的进化算子作用于被选出的 S 个染色体。由于每次进化都要利用平滑算子和定向扰动算子, 所以, 平滑算子和定向扰动算子被

选择的概率为 1。

[0134] 所述的进化算子包括如下形式,其具体作用形式如图 5 所示:

[0135] 1. 交叉算子

[0136] 交叉算子是指将两条航迹重新组合,生成两条新的航迹。被作用的航迹可以是可行的,也可以是不可行的。作为父个体的两个染色体节点数可以不同。

[0137] 2. 扰动算子

[0138] 扰动算子是指随机地改变航迹的一个中间节点的坐标。该航迹可以是可行的,也可以是不可行的。如果所选航迹是可行的,则在可行范围内加以较小扰动,以提高航迹的适应值;如果所选航迹是不可行的,则可适当增大扰动幅度,以期获得可行的航迹。

[0139] 3. 插入算子

[0140] 插入算子是指在两个相邻的航迹节点中间随机地插入一个新的航迹节点。一般地,在违背最低飞行高度约束,或者是穿越威胁区域的航迹段,采用该算子得到的子个体其性能可能提高。

[0141] 4. 删除算子

[0142] 删除算子是指删除航迹的一个中间节点。该航迹可以是可行的,也可以是不可行的。如果所选航迹是不可行的,该中间节点可以随机的选取;如果所选航迹是可行的,则节点的选取需要给予某些启发式信息。如果没有信息表明需要删除节点,则以一个很小的概率来确定是否删除。

[0143] 5. 交换算子

[0144] 交换算子是指交换任意两个相邻节点的先后顺序以减小拐弯角。该算子只作用于不可行航迹。如果两个相邻航迹节点不满足最大拐弯角约束,通过交换它们的顺序可能减小拐弯角。如果两相邻节点处航迹拐弯角越大,选择它们进行交换的概率越大。

[0145] 6. 平滑算子

[0146] 平滑算子是指通过“切除尖角”将航迹平滑化。该算子随机选择一航迹节点,在与该航迹节点相连的两航迹段上各插入一新的节点,然后删除开始选择的节点。如果某节点处航迹拐弯角越大,选择它进行平滑的概率越大。该算子只作用于不可行航迹。

[0147] 7. 定向扰动算子

[0148] 该算子为实现有效的目标进入方向而设计,且只作用于目标的前一个节点。如果要求满足有效的目标进入方向,则所有目标的航迹都将按该方向进行初始化;在进化过程中,目标的前一个节点只能用该算子进行作用。作用后的坐标由下式给出:

$$[0149] \quad \begin{cases} x'_{n-1} = x_n + t(x_{n-1} - x_n) \\ y'_{n-1} = y_n + t(y_{n-1} - y_n) \\ z'_{n-1} = z_n + t(z_{n-1} - z_n) \end{cases} \quad (22)$$

[0150] 其中, x'_{n-1} , y'_{n-1} , z'_{n-1} 为经定向扰动算子作用后第 $n-1$ 点的坐标, $t > 0$ 是一个随机实数。

[0151] 步骤 4.6,通过进化算子的作用生成新个体,将新生成的个体加入到染色体中,计算新个体的适应值。

[0152] 步骤 4.7,将扩展后的染色体中适应值较小的 S 个染色体删除,使其恢复到原来的染色体大小。

[0153] 步骤 4.8,重复步骤 4.4 至步骤 4.7,判断是否满足终止条件,即迭代过程进行到预

先给定的最大次数,或者最优个体在若干次迭代中其适应值不变,若满足,则从 m 条染色体中选出适应值最小的染色体作为所求航迹;若不满足,则修改迭代的最大次数值,重复步骤 4.4 至步骤 4.7。

[0154] 步骤 5: UAV 沿步骤 4 得到的局部最优航迹飞行,直到威胁消除, UAV 返回到步骤 3 得到的全局最优参考航迹继续飞行。

[0155] 实施例:

[0156] 首先建立飞行环境模型,选取相关参数。利用 SAS 算法进行离线航迹规划。本实施例采用以下参数值:

[0157] ● 代价函数中系数 $W_1=W_2=W_3=1/3$, $L = \frac{\sqrt{2}}{2}$;

[0158] ● 最小航迹段长度为 4km;最低飞行高度为 35m;最大拐弯角和最大爬升/下滑角分别为 30° 和 35° ;

[0159] ● 最小航迹段长度为 2km;

[0160] ● 最大路径距离为起始点和目标点之间直线距离的 1.5 倍;

[0161] ● 交叉算子的概率选为 0.25,扰动算子、插入算子、删除算子和交换算子的概率选为 0.1,平滑算子和定向扰动算子的概率选为 1;

[0162] ● 种群大小为 60,每次迭代选取 30 条航迹进行操作,即 $p = 60$ 、 $S = 30$;

[0163] 结合本发明前面的内容,建立飞行空间,并结合上述给定的约束条件建立搜索空间。对待搜索区域中的每一个栅格的顶点,判断是否满足步骤 3.1 约束条件,如果满足则计算其总的航迹代价,并将它按代价的大小插入到 OPEN 表中。从中选出最小代价的节点,放入 CLOSED 表中,然后将其父节点指针指向当前节点。否则舍弃。反复这个寻找的过程,直到到达目标点为止。

[0164] 初始参考航迹规划完成后,无人机将按参考航迹飞行,当出现突发状况,使无人机偏离预先设定的航迹或者出现突发威胁,将启动遗传算法进行在线航迹再规划。将航迹前面一定距离的节点作为新的起始点进行航迹规划生成新的航迹,随机生成确定大小的 p 种群,评价每一条航迹,如果满足终止条件,则结束,否则随机选取 S 个个体组成繁殖池,根据预先给定的概率机制选择进化算子作用于被选出的个体,并将新的个体加入到种群中,计算新个体的适应值,再将扩展后的种群中最差的 S 个个体删除,使其恢复到原来种群的大小。反复迭代,直到找到满足约束条件的航迹为止。

[0165] 仿真与分析

[0166] 1) 静态环境航迹规划:静态环境航迹规划指飞行空间中只有固定障碍物,静态障碍物的位置分别为 $(2, 2, 3)$, $(3.5, 2.5, 4)$, $(4.5, 3.5, 6)$, $(5.5, 5.5, 7)$ 和 $(6, 5, 7)$, 仿真结果如图 6 所示。

[0167] 2) 动态环境航迹规划:动态环境航迹规划指飞行空间中不仅包含静态障碍物还包括突发的动态移动障碍物,在图 7 中有两个移动障碍物,仿真结果如图 7 所示。

[0168] 从以上仿真结果中可以看出,运用本发明中的方法,不仅可以迅速的到达目标位置,还能有效地避碰。当存在动态障碍物时,尽管规划的时间增长,但仍能有效的避免碰撞,在满足要求的时间内到达目标,并且从中可以看到规划不会陷入局部最小点。

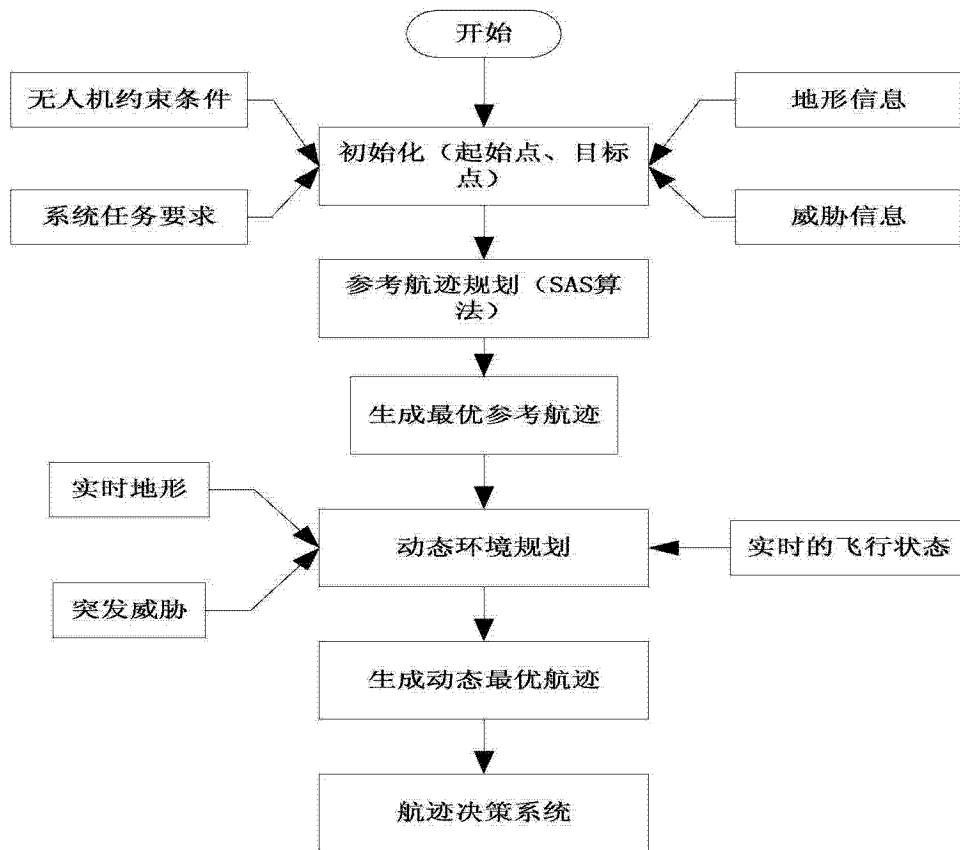


图 1

98	96	94	89	90	92
99	95	90	87	89	88
100	97	94	92	90	92
95	96	95	93	90	88
97	95	93	90	88	91

图 2

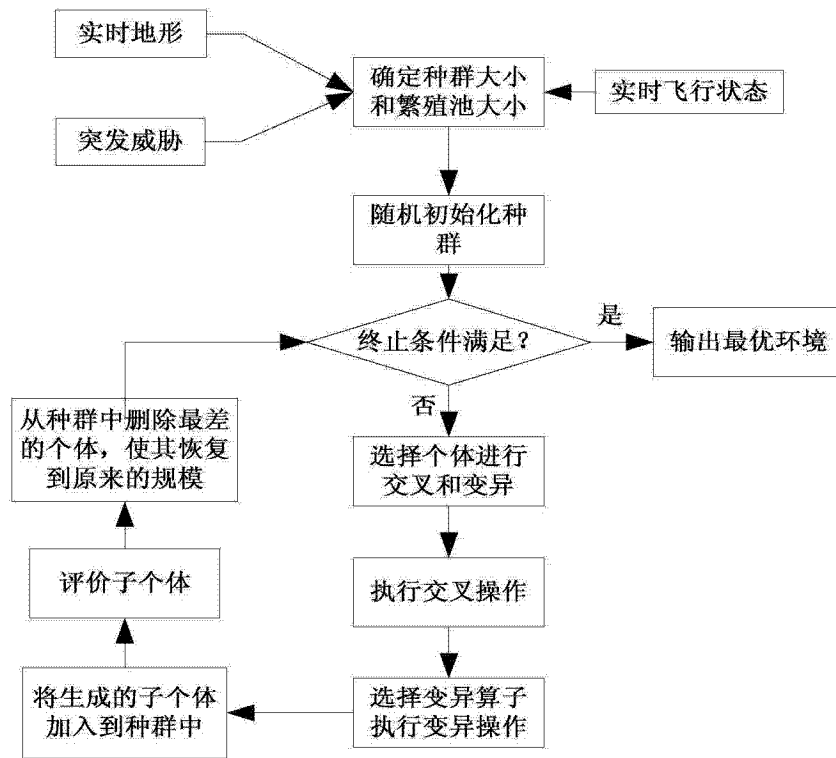


图 3

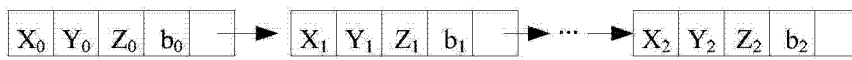


图 4

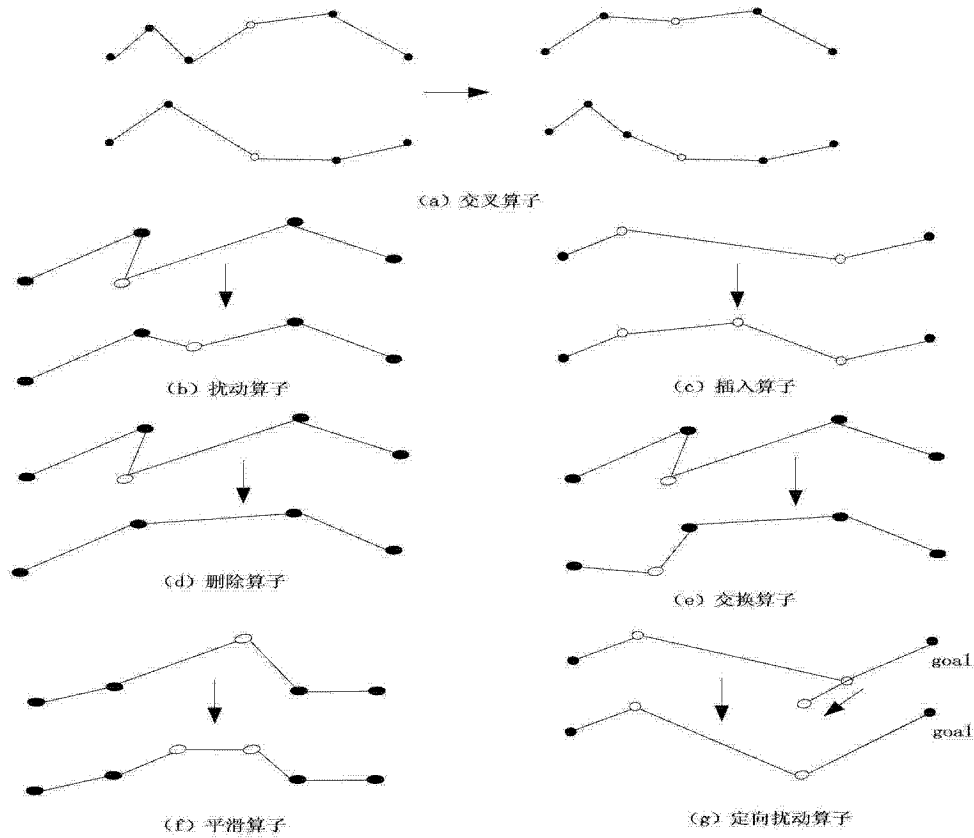


图 5

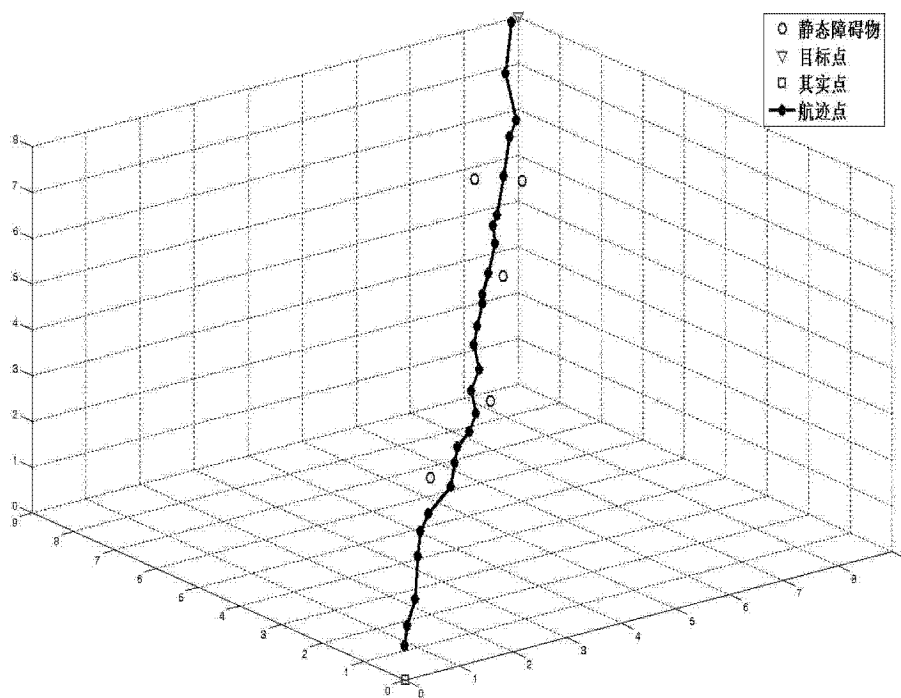


图 6

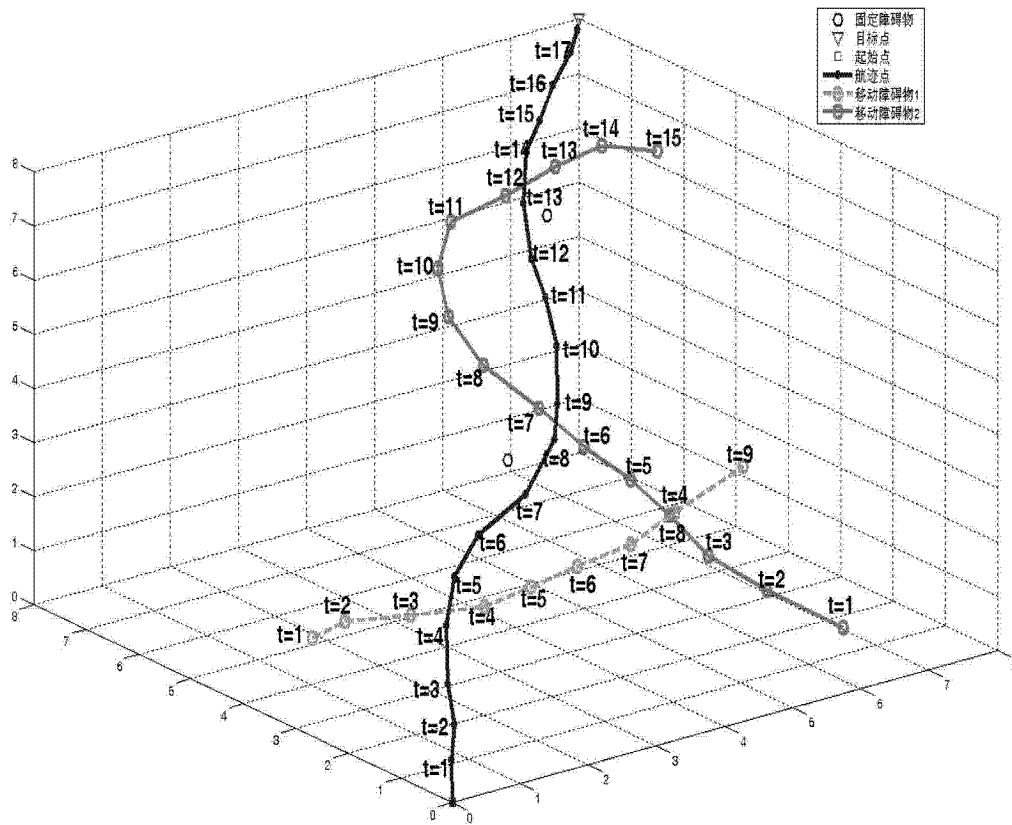


图 7