

## 遗传算法在路径规划中的应用

孟宪权<sup>1,2</sup>, 赵英男<sup>3</sup>, 薛 青<sup>1</sup>

(1. 装甲兵工程学院装备指挥与管理系, 北京 100072; 2. 总参陆航研究所, 北京 101114;

3. 吉首大学物理科学与信息工程学院, 吉首 416000)

**摘 要:** 遗传算法在路径规划中得到了成功应用, 但是已有算法对于不可行路径和置换比例的关注较少。该文对此进行研究, 将不可行路径进行优劣评价后添加到适应度函数中, 并在子代种群与父代种群的替换过程中运用了置换比例。试验结果表明两者均能提高遗传算法的性能, 因此, 在遗传算法设计中应对不可行路径和置换比例给予足够的重视。

**关键词:** 路径规划; 遗传算法; 适应度函数; 置换比例

## Application of Genetic Algorithm in Path Planning

MENG Xian-quan<sup>1,2</sup>, ZHAO Ying-nan<sup>3</sup>, XUE Qing<sup>1</sup>

(1. Department of Equipment Command and Management, Academy of Armored Force Engineering, Beijing 100072; 2. Army Aviation Institute,

General Staff Headquarters, Beijing 101114; 3. College of Physics Science & Information Engineering, Jishou University, Jishou 416000)

**【Abstract】** Genetic algorithm has been widely applied in path planning demonstrating good success. Traditionally, not-available paths and replacement proportion are deemed to be useless and always discarded directly by almost researchers in this field. However, this paper shows that this observation should be re-considered. By adding not-available path factor to fitness function after analyzing their performances and applying replacement proportion during the replacement process of father population by offspring population, very encouraging experimental results are achieved. All these observations suggest that more attention should be paid to traditionally abandoned not-available paths and replacement proportion for genetic algorithm in path planning application.

**【Key words】** path planning; Genetic Algorithm(GA); fitness function; replacement proportion

### 1 概述

遗传算法(Genetic Algorithms, GA)具有很强的全局寻优能力和适应性, 被广泛地应用于路径规划问题, 或者用来对原有路径规划方法进行改进。在路径规划应用中的遗传算法设计, 其研究内容主要集中在环境的表示方法、路径编码形式、初始群体的产生方法、适应度函数的确定、遗传算子的设计等方面。例如, 文献[1]实现了离散空间下机器人路径规划的遗传算法, 并且应用遗传算法来优化连续空间下机器人路径规划的 Ford 算法结果; 文献[2]将需规划的路径的二维编码简化成一维编码, 并把免碰撞要求和最短路径要求融合成一个适应度函数; 文献[3]讨论了 GA 的染色体编码方式、基于知识的初始种群生成方法与适应度函数, 基于领域知识设计了 5 种遗传算子; 文献[4]提出了一种建立在改进人工势场模型上的基于遗传算法的最优路径搜索方法。虽然在这一领域已经取得了一些研究成果, 但也存在着一定局限性。例如, 对不可行路径在适应度函数中的作用以及在子代种群替换父代种群过程中置换率的设置未给予足够的重视, 本文对此进行了研究, 试验结果表明两者均有助于提高算法的性能。

### 2 路径模型及编码方式

在路径规划中, 常用的路径模型有栅格法、单元树法和多边形表示法等。其中, 前 2 种方法属于离散化空间的表示方法, 这类方法存在一个局限性, 即若将空间划分过粗, 则规划精度较低; 若划分太细, 则数据量又会太大。而对空间的连续化描述则能避免这个问题, 多边形表示法即属于此类。文中采用该方法作为路径模型。

在二维虚拟环境中, 障碍物用多边形来表示。将多边形顶点表示为路径点, 如果路径点  $v$  和  $w$  可以互相看见(即这 2 点之间的连线不与环境中的任何障碍物的内部相交), 就在它们之间引入一条线段  $\overline{vw}$ 。对环境中的所有路径点之间都执行完这个操作后, 就得到了该环境的可见性图, 即将虚拟环境中的路径规划问题转化为路径点组合的寻优问题。这可以转化为图论中的最优路径问题, 路径模型如图 1 所示。

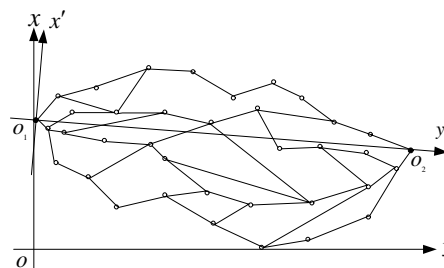


图1 最短路径规划示意图

图中的  $o_1$  为起点,  $o_2$  为终点, 在绝对坐标系  $xoy$  中, 路径点表示为  $v_i$ , 坐标为  $(x_i, y_i)$ , 共有  $n$  个路径点和  $m$  条线段。以  $o_1$  为原点,  $o_1o_2$  为坐标系横轴建立相对坐标系  $x'o_1y'$ ,  $v_i$  在

**基金项目:** 国家“863”计划基金资助项目(2004AA115130-05); 国家自然科学基金资助项目(70471089)

**作者简介:** 孟宪权(1973 - ), 男, 博士研究生, 主研方向: 智能体的建模与仿真; 赵英男, 讲师、博士; 薛 青, 教授、博士生导师

**收稿日期:** 2007-09-26 **E-mail:** mxq9973@tom.com

该坐标系下的坐标为  $(x'_i, y'_i)$ 。路径点下标的确定是根据  $x'_i$  的大小对  $v_i$  从小到大进行排序得到的, 即  $(v_1, v_2, \dots, v_n)$  对应着  $(x'_1, x'_2, \dots, x'_n)$ , 且  $x'_1 \leq x'_2 \leq \dots \leq x'_n$ 。

文中遗传算法的种群由路径构成, 一个个体就是一条路径。根据路径模型, 个体的编码采用符号编码方式。路径表示为  $P = a_1 a_2 \dots a_i \dots a_{node}$ ,  $node$  是该条路径中包含中间路径点的数量。  $a_i = 1, 2, \dots, n$  是  $v_i$  的下标。并且不同个体的  $node$  值是不一定相同的, 即染色体长度是变化的。

为描述方便, 特别说明如下定义:

**定义 1** 若路径点  $v_i$  和路径点  $v_j$  可以相互看见, 就称它们为可见的, 即连接它们的线段  $\overline{v_i v_j}$  不与环境中任何障碍物的内部相交, 且  $\overline{v_i v_j}$  称为一条可见边; 相反, 若  $v_i$  和  $v_j$  不可以相互看见, 则称它们为不可见的, 线段  $\overline{v_i v_j}$  称为一条不可见边, 不可见边在图 1 中没有表示出来。全部由可见边组成的路径称为可行路径, 存在不可见边的路径称为不可行路径。

**定义 2** 个体(路径)相对于  $\overline{o_1 o_2}$  的偏移距离  $D_{dep}$ :

$$D_{dep} = \frac{1}{node} \sum_{i=1}^{node} |y'_i| \quad (1)$$

其中,  $y'_i$  为路径点  $v_i$  的相对坐标系下的纵坐标。

**定义 3**  $v_i$  和  $v_j$  2 点间的距离  $D_{ij}$ :

$$D_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (2)$$

显然,  $D_{ij} = D_{ji}$ 。

个体的路径长度用  $L_i$  表示,  $i = 1, 2, \dots, N$ , 且  $N \ll M$ ,  $N$  是种群中可行路径数量,  $M$  是种群大小。不可行路径的长度设为无穷大。

### 3 适应度函数的确定

适应度是遗传算法中用来度量群体中各个个体在优化计算中有可能达到或接近于或有助于找到最优解的优良程度, 遗传算法根据适应度来决定个体在种群中的去或留, 它是遗传算法收敛性和稳定性的重要影响因素。度量个体适应度的函数称为适应度函数。

路径规划的首要问题就是得到的路径长度最短, 因此, 路径长度是适应度函数中首先需要考虑的内容。种群中除了可行路径, 还有不可行路径, 并且不可行路径也有优劣之分, 可以用不可见边的相对数量来衡量。还可以通过计算路径偏离  $\overline{o_1 o_2}$  的程度来衡量路径的优劣<sup>[5]</sup>。因此, 在本试验中, 影响路径优劣的因素主要有路径长度、不可见边数量和偏移距离。

运用归一化方法设定适应度函数, 并采用加权求和的方式, 形式如下:

$$F = \sum_{i=1}^s K_i \left(1 - \frac{C_i}{C_i}\right) \quad (3)$$

其中,  $s = 3$ , 表明该函数由 3 项组成;  $K_i$  为各项的权重, 表明该项在函数中的贡献程度。其他各参数的定义如下:

$$c_1 = L_j; j = 1, 2, \dots, M; C_1 = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n D_{ij}$$

$$c_2 = N_{\text{breakline}}; C_2 = node + 1$$

$$c_3 = D_{depj}; j = 1, 2, \dots, M; C_3 = \max(|y'_1|, |y'_2|, \dots, |y'_n|)$$

其中,  $c_1$  是个体的路径长度  $L_i$ ,  $C_1$  是全部路径线段的总和;  $c_2$  是个体中不可见边的数量,  $C_2$  是个体中边的总数;  $c_3$  是个体距离  $\overline{o_1 o_2}$  的平均长度,  $C_3$  是全部路径点中距离  $\overline{o_1 o_2}$  的最大长度。为描述方便, 称式(3)中第 1 项为路径长度项, 第 2 项

为不可见边项, 第 3 项为偏移距离项。

本文采用的适应度函数解决了对不可行路径的适应度评价问题。在已有方法中, 有的是将不可行路径的适应度设为 0<sup>[2]</sup>, 有的是置不可行路径的适应度为某一极小值<sup>[3,6]</sup>。第 1 种方法使得不可行路径参与进化的可能性非常微小, 如图 2 中曲线 0 所示。第 2 种方法对于不可行路径没有区别对待, 不利于种群的进化。笔者采用的适应度评价函数可以根据不可行路径的优劣给出相应的适应度, 取得了良好的效果。

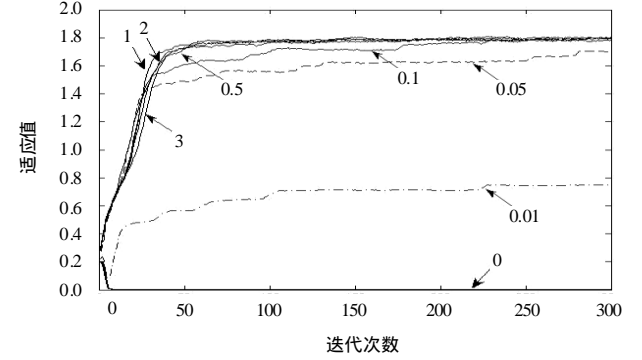


图 2 不可见边项对算法性能的影响

通过试验来考察适应度函数中各项对遗传算法的性能影响程度。试验的路径模型见图 1, 共有 35 个路径点(不包括起点和终点)。首先必须确定比较的标准, 可以用每代中出现的最短路径长度或者可行路径的平均长度, 但是不可行路径的长度没办法确定, 并且可行路径在种群中的比例是逐渐增大的。当可行路径在种群中是少数时, 受单个个体的影响很大, 会使得平均值不能正确反映种群的优劣。另一个比较标准可以选用种群的平均适应值, 该值可以正确反映种群的优劣, 但是当试验的考察对象是适应度函数时, 由于不同试验中适应度函数中的参数不同, 这样会带来同一个个体在不同试验中的适应度不一样, 进而造成不同试验中种群的平均适应值不具有可比性。为了解决这个问题, 为每个个体设置了 2 个适应度函数, 一个函数的参数是可调的, 计算得到的适应度用来参与算法运算; 而另一个函数的参数是固定的, 得到的适应度只用来进行种群间平均适应度的比较, 保证了不同试验中相同的个体具有相同的适应度。

图 2 所示为验证式(3)中不可见边项作用的试验结果, 箭头尾标注为  $K_2$  的值。

试验的参数设置情况为:  $\{M, T, p_c, p_m\} = \{200, 300, 0.9, 0.1\}$ 。其中,  $M$  为种群大小;  $T$  为终止进化代数;  $p_c$  为交叉概率;  $p_m$  为变异概率。  $K_1 = 1$ ,  $K_3 = 0.05$ ,  $K_2 = \{0, 0.01, 0.05, 0.1, 0.5, 1, 2, 3\}$ 。每种设置下进行 30 次试验, 然后求出每代种群的平均适应度作为这种设置下算法的性能表示。显然, 随着  $K_2$  的逐渐减小, 算法的收敛变慢, 性能随之下降。特别是当  $K_2 = 0$  时, 就不能找到可行解, 由此可见不可见边项在算法中的重要性。然而,  $K_2$  并不是越大越好, 当  $K_2 > 1$  时, 算法性能则基本没有改进。

图 3 所示为验证式(3)中偏移距离项作用的试验结果, 箭头尾标注为  $K_3$  的值。试验的参数设置情况与图 2 中不同的是:  $M = 150$ ,  $K_2 = 1$ ,  $K_3 = \{0, 0.01, 0.05, 0.1, 0.5, 1, 2\}$ 。由图可见,  $K_3 = 0.01$  比  $K_3 = 0$  时的算法性能有了提高, 但不是很大。并且, 随着  $K_3$  的增大, 算法性能出现了下降。因此, 适应度函数中的偏移距离项对算法性能的改进作用不大。

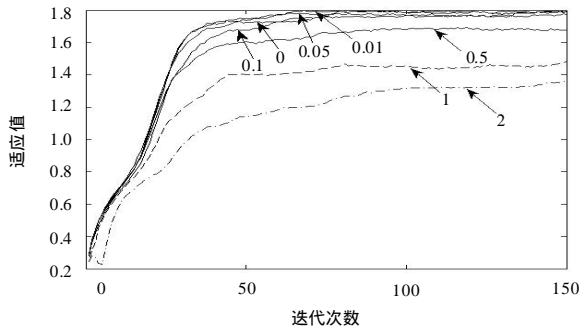


图3 偏移距离项对算法性能的影响

#### 4 置换方法和置换比例

$t$  时刻, 父代种群  $X(t)$  通过选择、交叉、变异、插入等遗传操作得到了子代种群  $X'(t)$ , 为了保持种群大小  $M$  不变, 必须制定  $X(t)$  和  $X'(t)$  的置换方法, 进行相应的置换运算, 以形成  $t+1$  时刻的父代种群  $X(t+1)$ 。遗传算法中种群通过进化会产生越来越多的优良个体, 但是由于遗传算子操作过程的随机性, 有可能破坏了父代种群中适应度较好的个体, 并且不能保证每个子代个体都会优于父代个体。因此, 应该通过保留父代种群中的较优个体来加快算法的收敛速度。但是另一方面, 这种方法也容易使得局部最优个体不易被淘汰反而快速扩散, 从而导致算法的全局搜索能力不强<sup>[1]</sup>。因此, 如何制定和选择置换方法是一个重要的研究内容。

文献[7]提出了杰出者选择和父子混合选择这 2 种保证 GA 收敛性的置换方法。杰出者选择的过程是: 首先根据父代种群  $X(t)$  执行通常的遗传操作, 生成子代种群  $X'(t)$ , 然后从  $X'(t)$  中选择  $M-1$  个个体, 从  $X(t)$  中选择一个最优个体共同组成新一代种群  $X(t+1)$ 。父子混合选择是在父代种群  $X(t)$  与子代种群  $X'(t)$  的并  $X(t) \cup X'(t)$  中选择  $M$  个个体作为新一代母体种群  $X(t+1)$ 。文献[1]中将最优保存策略(即杰出者选择)视为选择操作的一部分, 然而考虑到本文中的置换运算与选择运算在计算时序上的显著差异(即  $X(t)$  在进行完选择、交叉、变异、删除、插入运算后才进行置换运算)和其在遗传算法中的重要性, 将置换运算单独列出并对其实现方法进行研究。

笔者提出 2 种方法来执行父子混合选择。方法 1: 根据个体适应值, 对  $X(t) \cup X'(t)$  进行降序排列, 选择排序后的前  $M$  个个体作为  $X(t+1)$ 。方法 2: 分别对  $X(t)$  和  $X'(t)$  进行降序排列以一定比例用  $X'(t)$  中适应值较高的个体来替代  $X(t)$  中适应值较低的个体后得到  $X(t+1)$ 。将这个比例定义为置换比例  $P_u$ , 那么,  $X(t)$  被替换的个体数量为  $M \times P_u$ 。显然, 方法 2 中包含杰出者选择置换方法, 例如, 对于  $M=200$ , 当  $P_u=0.995$  时, 置换个体数量为 199, 那么置换过程中  $X(t)$  的适应值最高个体都会被保留。

通过试验研究以上方法对算法性能的影响, 试验的参数设置情况为  $\{M, T, p_c, p_m\} = \{200, 500, 0.9, 0.1\}$ ,  $K_1=1$ ,  $K_3=0.05$ ,  $K_2=1$ , 每种情况的试验次数为 30 次。试验结果如图 4 所示, 其中, 曲线 1:  $P_u=0.005$ ; 曲线 2:  $P_u=0.3$ ; 曲线 3:  $P_u=0.6$ ; 曲线 4:  $P_u=1$ ; 曲线 5:  $P_u=0.995$ ; 曲线 6: 方法 1。曲线 2 和曲线 6 很接近, 但是曲线 6 的水平部分没有波动, 也就是适应值平均值收敛得很快, 实际寻优结果也没有曲线 2 好。显然, 当置换比例较大, 也就是较多的父代个体被子代个体替代时, 算法出现了明显的早熟(早期收

敛)现象, 如曲线 3~曲线 5(即杰出者选择置换方法); 当置换比例较小时, 算法的收敛速度比较慢, 如曲线 1。因此, 应该兼顾算法的收敛速度和寻优能力, 选择合适的置换比例, 如曲线 2。

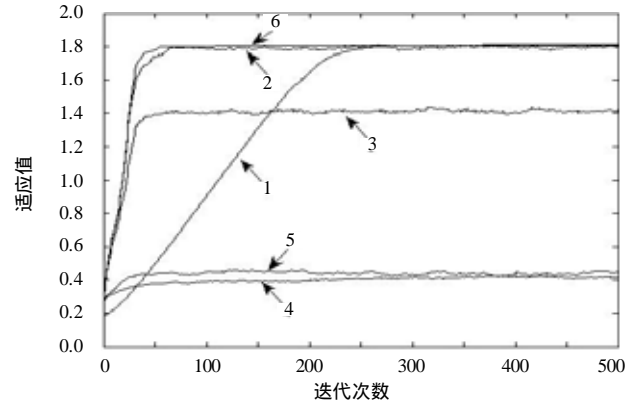


图4 置换比例的影响

#### 5 其他参数的确定

选择算子采用比例选择算子。

交叉算子采用重合点交叉的方式, 也就是只有当 2 个体存在相同路径点时才进行交叉操作, 交换这个相同路径点后的基因。当存在多个相同路径点, 随机选择其中一点进行交叉。

变异算子使用基本位变异算子, 若在个体第  $i$  个基因位进行变异操作, 则随机产生整数  $a'_i$  ( $a'_i=1, 2, \dots, n$ ) 来替换  $a_i$ 。

删除运算, 若在个体的第  $i$  个基因位进行删除操作, 则判断 2 点  $v_{a_{i-1}}$  和  $v_{a_{i+1}}$  是否可见, 若可见则删除  $a_i$ , 否则不进行删除操作。

插入运算, 若  $\overline{v_{a_i} v_{a_{i+1}}}$  是可见边, 则该边是  $v_{a_i}$  和  $v_{a_{i+1}}$  之间的最短距离, 因为 2 点间直线距离最短, 所以不需要执行插入运算; 若  $\overline{v_{a_i} v_{a_{i+1}}}$  是不可见边, 则随机产生整数  $a'_i$  ( $a'_i=1, 2, \dots, n$ ) 插入到  $a_i$  和  $a_{i+1}$  之间。由于插入运算是针对不可见边的操作过程, 可以消除不可行路径, 因此加快了算法收敛。图 5 是插入运算对算法性能的影响结果, 该试验中的  $P_u=0.3$ , 其他设置同第 4 节中的试验。可见插入运算的作用很明显。

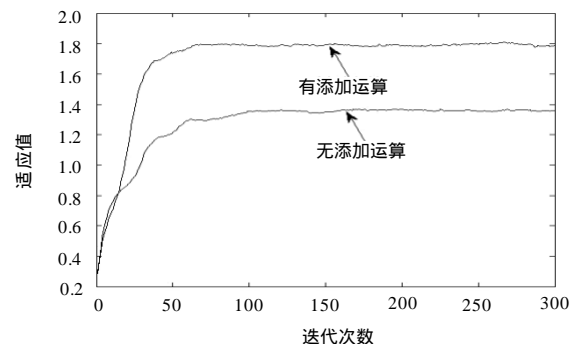


图5 插入运算的影响

#### 6 结束语

本文对应用于路径规划的遗传算法进行了较为深入的试验研究。文章作出的贡献如下: (1) 采用归一化加权求和方法建立一个具体的适应度函数, 特别是解决不可行路径的适应

(下转第 220 页)