

机器人路径规划中的改进型遗传算法

唐国新,陈 雄,袁 杨

TANG Guo-xin, CHEN Xiong, YUAN Yang

复旦大学 信息学院,上海 200433

School of Information Science and Technology, Fudan University, Shanghai 200433, China

E-mail: 042021044@fudan.edu.cn

TANG Guo-xin, CHEN Xiong, YUAN Yang. Improved genetic algorithm for robotic path planning. *Computer Engineering and Applications*, 2007, 43(22): 67-70.

Abstract: An improved genetic algorithm designed for robotic path planning is proposed. Crossover operator and mutation operator are specifically optimized for the application of path planning, and two customized genetic operators are introduced as well. The evolutionary efficiency is noticeably enhanced by incorporating map information in genetic operators and better adaptive ability towards complex maps is realized by adopting the customized genetic operators. Demonstrated by simulation results, the proposed algorithm achieves considerable improvements, with respect to the basic genetic algorithm, in optimal solution output rate.

Key words: genetic algorithm; robotic path planning; crossover operator; mutation operator; customized genetic operator

摘 要:提出了一种应用于机器人路径规划的改进型遗传算法。针对机器人路径规划的实际应用,优化设计了交叉算子和变异算子,引入了自定义的插入和删除两种遗传操作。通过把地图特征信息作为参与决策的已知条件来约束遗传算子的操作过程,提高了算法的进化效率。自定义遗传算子的使用,使得算法对复杂地图也表现出良好的适应能力。计算机仿真实验证明该算法在最优解输出概率方面相对于基本遗传算法有了显著提高。

关键词:遗传算法;机器人路径规划;交叉算子;变异算子;自定义遗传算子

文章编号:1002-8331(2007)22-0067-04 文献标识码:A 中图分类号:TP18

1 引言

在机器人学领域,路径规划是最基本同时也是最重要的研究课题之一。作为机器智能的一部分,它的任务就是使机器人在其工作环境中能够自主的从起始点运动到目标点同时满足一定的约束条件。这些约束条件包括但不限于:(1)不与障碍物碰撞;(2)运动路径最短;(3)尽量远离障碍物;(4)机器人耗能最小;(5)运动轨迹包含某些设定的必经路点。

基于不同的基本思想,有多种算法出现。包括有人工势场法(APF)、臭虫法(BUG)、随机位图法(PRM)、快速随机树(RRT)、神经网络法(NN)等等。这些算法都在不同的方面有着各自的优势,然而在总体上,这些路径规划算法都还存在各自的不足。如计算复杂度、局部最优解、地图适应性、动态环境下有效性,全局及多目标优化能力。

随着遗传算法(GA)的提出,机器人路径规划算法也得到相应的发展。通过GA在路径规划中的应用^[1,2,7],使得机器人更加的智能,其运行路径也更加逼近理想的优化要求。尤其是其优秀的全局优化能力和较高的搜索效率使得GA成为处理复杂环境下全局优化要求的最佳算法之一。

早期的路径规划应用中,GA算法通常未对路径规划问题特别设计遗传操作,而使用的标准遗传操作又无法反映所求问

题的特定知识,所以算法的进化过程很大程度上属于盲搜索状态。较低的效率使得一条最终路径的产生需要耗费相当长的时间^[2]。为了利用路径规划问题中已有的知识来加快求解速度,常用的方法是先构造MAKLINK图来进行地图建模以找出机器人可以运动的自由空间,然后再使用Dijkstra算法得出MAKLINK图中的最短路径,最后,再应用GA对所得路径进行优化,以获得全局最短路径。这样的方法虽然可以有效的简化GA需要完成的优化工作,但它需要完成前期非常耗时的建模过程作为运行GA算法的前提。所以,其总体计算复杂性并没有得到有效地降低。后续过程对地图建模的依赖性使得算法面对不同的地图环境时缺乏灵活性。

针对上述不足之处,本文采用了直接基于路点坐标值的可变长度的染色体编码方式,配合包含障碍物排斥子函数项的代价函数,有效避免了地图建模过程。同时针对路径规划问题的具体应用改进了交叉和变异两种标准遗传操作,并且还引入了插入和删除两种自定义遗传操作,大大提高了算法的搜索效率,实现了较高的收敛速度。计算机仿真结果证明了本方法较强的地图适应性和相对于标准遗传算法的性能提高。

本文内容安排如下,第2章介绍所提的改进型遗传算法,第3章给出算法在多种地图环境下的仿真结果,以及和普通遗传算法的性能对比。第4章作出总结。

2 改进型遗传算法

在本文的机器人路径规划中,目标是在一幅障碍物分布已知的二维地图上寻找一条最优路径使到达目标点距离最短,同时尽可能地最大化与障碍物的距离。为了简化讨论,机器人被作为一个质点来考虑,而障碍物的边界向外扩张半个机器人最大直径。

2.1 染色体编码

机器人的运动轨迹由若干直线段构成,每段直线段是机器人运动的基本单位,于是机器人到达目标点的整个路径可表示成

$$T=I_1+I_2+\dots+I_{n-1}$$

其中, I_i 为第 i 段直线段的矢量表示,它的两个端点分别为 p_i 和 p_{i+1} ,符号“+”表示把各矢量首位相连的连接运算。以 O 表示原点,于是有

$$I_i=OP_{i+1}-OP_i$$

进而,整个机器人运动轨迹可以表述为路点矢量的顺序集合

$$T=\{OP_1, OP_2, \dots, OP_n\}$$

设 p_i 点坐标为 (x_i, y_i) ,于是算法实现时,路径 T 存储为

$$T=[x_1+i^*y_1, x_2+i^*y_2, \dots, x_n+i^*y_n]$$

其中, i 为虚数单位。于是完成了直接基于路点坐标的矢量染色体编码。在后续算法过程中,固定 (x_1, y_1) 为机器人起始点坐标, (x_n, y_n) 为目标点坐标,所以路径 T 是一个连接起点和目标点的可能解。其中 n 为变数并满足 $2 \leq n \leq N_{\max}$, N_{\max} 为最大路点数。

2.2 适应度函数

每条路径的优劣评价通过适应度函数来给出。本文中,以路径长度和障碍物相交程度作为评价指标,并使所求解向这些指标渐小的方向进化,所以这里的适应度函数又被称为代价函数。使各项指标在代价函数中表现出来,于是该函数构造如下:

$$F(T_k)=a_1 \sum_{i=1}^{N-1} \rho_i + a_2 \sum_{i=1}^{N-1} \alpha_i$$

其中 a_1, a_2 为权重系数,分别强调了不同优化指标的重要性。式中第 1 项为表示路径 T_k 的总长度, ρ_i 表示路径 T_k 中第 i 段直线的长度。适应度函数中的第 2 项被称为障碍物排斥函数,表达为:

$$\alpha_i = \sum_{j=0}^M \beta_{ij}$$

其中, M 为障碍物的个数,而 β_{ij} 则是第 i 段直线与和第 j 个障碍物的排斥度。并定义为

$$\beta_{ij} = \begin{cases} \gamma + d_s, & \text{, 直线段与障碍物相交时} \\ \frac{1}{d_s} \cdot d_0^2 + (-2) \cdot d_0 + d_s, & \text{, 直线段距离障碍物 } d_0 \leq d_s \\ 0, & \text{, 直线段远离障碍物 } d_0 > d_s \end{cases}$$

其中 γ 为使直线段不与障碍物相交所要移动的最短距离, d_s 为直线段到障碍物的距离,称 d_s 为安全距离,当 $d_0 \geq d_s$ 后,算法将不再试图使路径进一步远离障碍物,称该线段和障碍物无排斥。

在后面的运行过程中,算法试图使代价函数最小化并认为使得该函数取得较小值的解为较优解。

2.3 遗传操作

遗传操作是决定 GA 性能的关键因素之一。本文针对路径规划问题的特点改进了标准交叉算子和变异算子,把地图信息作为参与决策的已知条件引入遗传操作以改善算法在解空间上的搜索效率。此外,除了改进标准遗传操作,本文还引入了自

定义的插入和删除操作,进一步提高算法的搜索能力。

2.3.1 交叉算子

交叉操作作用于两个操作对象,分别对它们进行随机分割,然后再重新组合从而获得两个新个体。根据分割点的数量,可分为单点交叉或是多点交叉。其原理都可用多点交叉过程加以说明,对每条路径随机选择 m 个无重复的交叉点,在交叉点之间的变量间断地相互交换,产生两条新的路径,完成交叉操作。如图 1。

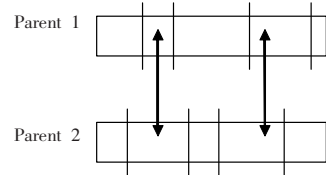


图 1 多点交叉操作示意

图 1 中,两个父个体均被 4 个分割点分为 5 部分,其中标有箭头的部分进行相互交换。

显然,由于分割点位置的不同,当交叉操作完成后,新产生的两个子代个体的染色体长度很有可能是不相等的并且相对于父代也发生变化。同时,其对应的代价函数值也将改变。所以,本文对交叉算子进行改进,在分割点个数确定后,寻找合适的分割点位置,使得新产生的子代个体之一获得尽可能低的代价函数值。

为此,将路径 T 写为

$$T=I_1+I_2+\dots+I_{n-1}$$

$$I_i=OP_{i+1}-OP_i$$

记路径上每条线段的相交函数为:

$$f(I_i) = \begin{cases} 0, & \text{第 } i \text{ 段直线段与所有障碍物不相交} \\ 1, & \text{第 } i \text{ 段直线段与障碍物相交} \end{cases}$$

从而

$$g_i = f(I_{i+1}) - f(I_i)$$

指示出路径上各线段与障碍物相交状态的变化。当 g_i 为 1 时,说明 $i+1$ 点的前一线段未与障碍物相交而其后一线段与障碍物发生相交。反之,当 g_i 为 -1 时,说明 $i+1$ 点的前一线段与障碍物相交,而其后一线段未与障碍物相交。当 g_i 等于 0 时,说明在 $i+1$ 点前后的两段线段与障碍物的相交状态相同。

选择 g_i 为 1 时对应的变化点作为 1 号父个体的第一分割点,选择紧随该点之后使得 g_i 为 -1 的点作为第二分割点。对于 2 号父个体,选择过程恰好相反,选择 g_i 为 -1 时对应的变化点作为 2 号父个体的第一分割点,选择紧随该点之后使得为 1 的变化点作为第二分割点。更多的分割点同理可得。

至此,对分割点位置的优化得以完成。但是,对于分割点数量的选择依然需要考虑。由于机器人路径规划问题自身的特点,以及本文采用的可变长度染色体编码方案,染色体长度在算法初始阶段会较长,所以采用多点交叉以改善算法搜索性能。然而,如前所述,非定点的交叉操作使得算法的进化后期染色体长度可能会变得较短。于是,在进行多点交叉操作时,同时考虑根据不同的进化阶段,不同的染色体长度设定不同的交叉点数。在算法设计中,本文采用如下分段设定:

$$\text{crossnum} = \begin{cases} 1, & 2 < \text{cLen} \leq 5 \\ 2, & 5 < \text{cLen} \leq 20 \\ 4, & 20 < \text{cLen} \leq 35 \\ 6, & 35 < \text{cLen} \leq N_{\max} \end{cases}$$

其中, crossnum 表示交叉点数, cLen 表示染色体长度, N_{\max} 则为本

算法中限定最大染色体长度。

2.3.2 变异算子

变异过程中,个体中的分量以很小的概率或步长产生转移。对于给定路径,该操作对路径上的各路点 p_i 以一定的概率 P_{mi} 改变其坐标。

在标准变异过程中并没有对已知的地图信息加以利用。使得变异操作成为了随机搜索过程,于是变异常常导致路径劣化。为此,改进型变异算子优先选取和障碍物相交的线段的端点进行变异,同时限制变异所得的路点坐标在障碍物之外,并且使变异所得的路点新坐标满足

$$l_i^{new} = OP_{i+1} - OP_i^{new}$$

$$l_{i-1}^{new} = OP_i^{new} - OP_{i-1}$$

$$f(l_{i-1}^{new}) + f(l_i^{new}) \leq f(l_{i-1}) + f(l_i)$$

通过这样的约束条件保证了每次变异对路径优化的非负面效果。

2.3.3 插入算子

该算子在其所作用路径上增加路点。考虑路径上某一直线段 $l_i = OP_{i+1} - OP_i$ 与障碍物相交,并且有端点坐标 p_i 处于障碍物外部空间。于是,通过在 p_i 和 p_{i+1} 之间插入合适的端点 p_{i+1}^{new} ,一定可以得到 $l_i^{new} = OP_{i+1}^{new} - OP_i$ 不与障碍物相交。同理,对于 p_{i+1} 处于障碍物外部空间时,一定可以有 $l_{i+1}^{new} = OP_{i+1}^{new} - OP_{i+1}$ 不与障碍物相交。对于 p_i 和 p_{i+1} 均位于障碍物内部的情况,该算子将随机生成坐标值,满足 p_{i+1}^{new} 位于所有障碍物的外部空间。

2.3.4 删除算子

该算子在所操作路径上记录所有位于障碍物内部空间的路点,随机选择其中之一并予以删除。对于不和障碍物相交的路径,该算子则在其全体路点中随机选择删除点。

2.4 算法流程

(1)种群初始化。本算法中初始种群采用随机方式产生,路径包含路点数在 $[2, N_{max}]$ 上服从均匀分布,路径上各路点坐标值在整个地图坐标范围内随机取值,服从均匀分布。种群规模记为 Gpn 。

(2)采用锦标赛选择法挑选父个体,并按一定的概率分布随机选择不同的遗传算子以执行遗传操作。

(3)获得子代个体后检查子代个体染色体长度是否超过 N_{max} ,如果没有超过则保留,否则丢弃。

(4)重复(2)、(3)步直到获得的新个体数量与父代群体数量相等。

(5)父代群体和子代群体构成规模为 $2 * Gpn$ 的并集,通过代价函数给出该合集中每个个体的代价值,选择代价值较低的前 Gpn 个个体构成新的父代群体,取代原有父代群体。

(6)检查算法停止条件。本文中,当进化世代数 Evn 达到 50 时,算法结束并选择当前父代群体中的代价值最小个体作为输出。否则跳转至第(2)步,算法继续进行。

3 仿真结果

本文通过多种地图情况下的计算机仿真来验证所提改进算法的性能。在仿真中,算法相关参数设定如下:种群规模 100,进化世代数 50, a_1, a_2 分别为 1、1 000。

3.1 地图 1

地图中,机器人起始点为 $(0,0)$,目标点 $(50,10)$,算法运行

多次,典型结果如图 2,其中的折线轨迹即为算法的输出路径,其代价函数值为 109.956 1,路径全长 109.956 1。

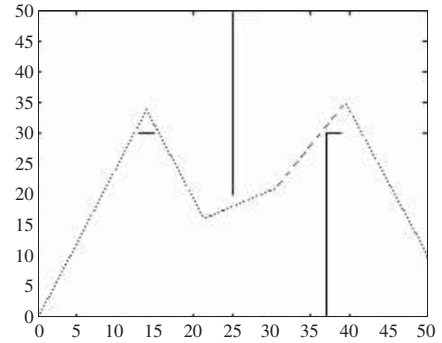


图 2 算法输出结果 1

3.2 地图 2

本实验中机器人起始点 $(0,0)$,目标点 $(40,30)$,算法运行多次,典型结果如图 3,其代价函数值为 80.083 5,路径全长 80.083 5。

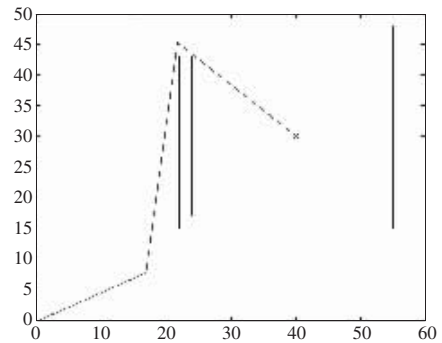


图 3 算法输出结果 2

3.3 地图 3

本实验中机器人起始点 $(0,0)$,目标点 $(42,60)$,算法运行多次,典型结果如图 4,其代价函数值为 76.141 2,路径全长 76.141 2。

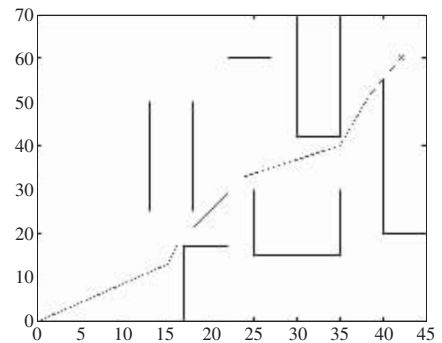


图 4 算法输出结果 3

3.4 算法对比

本实验通过使用不同遗传算子,比较由此导致的算法性能差异。对比实验中,算法分为 3 种,第 1 种为标准遗传算法,第 2 种使用改进型遗传算子,但不包含本文中所提的自定义算子,第 3 种同时使用了本文中提出的改进型遗传算子和自定义遗传算子。其它算法设置完全相同。针对上述 3 幅不同的地图,每种算法运行 100 次,当输出结果和前 3 个实验中对应的典型结果的代价值相差 50%以内时,认为算法成功。记录算法 100 次运行中的成功率、平均代价值、代价值的标准差,作为评价算

法优劣的标准。统计结果如表 1~表 3 所示。

表 1 成功率对比

| | 地图 1 | 地图 2 | 地图 3 |
|------|------|------|------|
| 算法 1 | 57 | 41 | 0 |
| 算法 2 | 72 | 59 | 0 |
| 算法 3 | 100 | 99 | 92 |

表 2 平均代价对比

| | 地图 1 | 地图 2 | 地图 3 |
|------|-----------|----------|----------|
| 算法 1 | 113.234 2 | 82.580 9 | NA |
| 算法 2 | 111.324 2 | 81.128 3 | NA |
| 算法 3 | 109.617 8 | 82.647 8 | 78.248 5 |

表 3 标准差对比

| | 地图 1 | 地图 2 | 地图 3 |
|------|---------|---------|----------|
| 算法 1 | 4.147 1 | 1.449 4 | NA |
| 算法 2 | 7.560 3 | 1.454 8 | NA |
| 算法 3 | 5.672 1 | 3.183 7 | 10.681 9 |

4 总结

上述的实验数据证明了本文所提出的改进型遗传算法的有效性。在 3 幅不同的地图上都达到了 90% 以上的算法成功率,并且相对其它算法有明显提高。随着地图的不同,各算法的成功率均出现不同程度波动,但改进型遗传算法波动幅度最小,保持了较好的稳定性,体现出良好的地图适应能力。

本文通过直接基于坐标值的染色体编码方案和有效的代价函数构造使得路径规划中的地图信息被成功引入到遗传操作的实现过程,并由此提高了算法的成功率。值得指出,参数 N_{\max} 采用了手动设定的方式来确定。由前面的算法描述可知,过大的 N_{\max} 会加大所需的计算量,从而导致输出结果耗时的延长,而过小的 N_{\max} 又使算法无法充分的搜索解空间,影响解的优化质量,甚至无法输出有效解。显然,有效的参数优化,将会进一步提高改进型遗传算法的性能。与此类似的还有多点交叉的点数,不同遗传操作的选中概率,染色体变异概率,代价函数

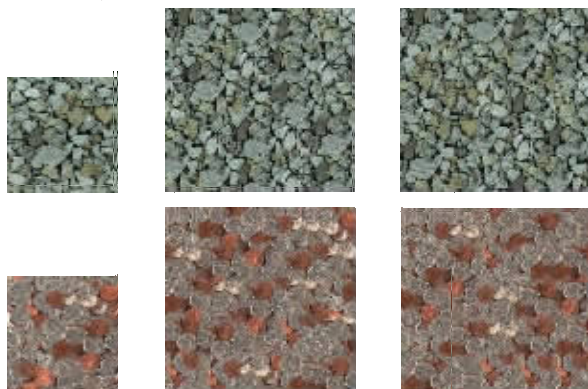
中的权值系数。如何让算法自适应的调整并最优化这些参数的取值,将是本算法的进一步改进方向。

(收稿日期:2006 年 12 月)

参考文献:

- [1] Ayala-Ramirez V, Perez-Garcia A, Montecillo-Puente F J, et al. Path planning using genetic algorithms for mini-robotic tasks[C]//Proceedings IEEE SMC'2004 Conference, Hague Netherlands, Oct 2004, 4:3746-3750.
- [2] Tu J, Yang S. Genetic algorithm based path planning for a mobile robot[C]//Proc of IEEE Intl Conf on Robotics and Automation, Taiwan, September 2003:1221-1226.
- [3] Qi Yuan-qing, Sun De-bao, Li Ning, et al. Path planning for mobile robot using the particle swarm optimization with mutation operator [C]//Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, August 2004:2473-2478.
- [4] Hocaoglu C, Sanderson C. Planning multiple paths with evolutionary speciation[J]. IEEE Trans on Evolutionary Computation, 2001, 5(3): 169-191.
- [5] Farritor S, Dubowsky S. A genetic planning method and its application to planetary exploration[J]. ASME Journal of Dynamic Systems, Measurement and Control, 2002, 124(4):698-701.
- [6] Wang Chun-miao, Soh Y C, Wang Han, et al. A hierarchical genetic algorithm for path planning in a static environment with obstacles [C]//IEEE Canadian Conference on Electrical and Computer Engineering, Canada, 2002, 3:1652-1657.
- [7] Hu Yan-rong, Yang S X. A knowledge based genetic algorithm for path planning of a mobile robot[C]//Proceedings of the 2004 IEEE International Conference on Robotics Automation, New Orleans, April 2004:4350-4355.
- [8] Gerke M. Genetic path planning for mobile robots[C]//Proc of the 1999 American Control Conference. San Diego CA:IEEE Press, 1999, 4:2424-2429.

(上接 33 页)



(a)输入的 128x128 的样本纹理 (b)运用 PSO 算法输出的 200x200 的目标纹理 (c)运用 QPSO 算法输出的 200x200 的目标纹理

图 5 两种算法的合成结果比较

搜索过程中具有较大的优势,其收敛速度是标准 PSO 算法的数倍。这是因为 QPSO 所基于的量子系统具有非线性性和不确定性,所以在这样一个系统中,一个粒子能够以某一确定的概率出现在整个可行的搜索空间中任意一个位置,而非限定在某一个固定区域内,因而增加了找到更好适应值的可能。

将 QPSO 引入了纹理合成算法中,与先前的 PSO 纹理合成

比较起来,有收敛速度快,控制参数减少,合成结果的质量更高等优点。但其仍具有许多缺点,如仍需要人为的设定块的大小,迭代次数,以及粒子的数目,对于某些特定的纹理实现效果不佳,以及块算法自身的缺陷。

在未来的工作中,可以将以自适应算法为方向发展,结合 QPSO 算法,实现更好的纹理合成效果。

(收稿日期:2007 年 3 月)

参考文献:

- [1] Guo Bai-ning, Liang Lin, Liu Ce, et al. Real-time texture synthesis by patch-based sampling[J]. ACM Transaction on Graphics, 2001, 20(3).
- [2] Shum H Y, Szeliski R. Panoramic image mosaics[R]. Microsoft Research.
- [3] Kennedy J, Eberhart R C. Particle swarm optimization[C]//Proc IEEE Int'l Conf on Neural Networks Vol IV. IEEE Service Center, 1995.
- [4] Sun J, Feng B, Xu W. A global search strategy of quantum behaved particle swarm optimization[C]//IEEE Proc of Conference on Cybernetics and intelligent system, 2004.
- [5] Zhang Yan, Meng Yu, Li Wen-hui, et al. A fast algorithm for image analogy using particle swarm optimization [C]//Proc of the 3rd International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004.