

文章编号: 2095-2163(2021)06-0153-04

中图分类号: TP312

文献标志码: A

基于最短距离适应度的随机路径优化问题研究

张耀允¹, 谭振江¹, 周伟², 方大甲³

(1 吉林师范大学 计算机学院, 吉林 四平 136000; 2 吉林师范大学 数值模拟吉林省高校重点实验室, 吉林 四平 136000;

3 吉林师范大学 附属机构, 吉林 四平 136000)

摘要: 本文针对已有的以曼哈顿距离作为适应度函数而得出的非最佳路径, 通过采用以起点到终点间的直线距离作为适应度函数的遗传算法优化, 将其优化为效率更高、更为优秀的路径。经实验验证, 证实了此方法在普通二维地图路径优化问题中的优越性。

关键词: 适应度函数; 寻路; 路径优化; 遗传算法

Research on stochastic path optimization based on shortest distance fitness

ZHANG Yaoyun¹, TAN Zhenjiang¹, ZHOU Wei², FANG Dajia³

(1 School of computer Science, Jilin normal University, Siping Jilin 136000, China; 2 Key Laboratory of numerical simulation, Jilin Normal University, Siping Jilin 136000, China; 3 Affiliated institution of Jilin normal University, Siping Jilin 136000, China)

【Abstract】 Aiming at the existing non-optimal path obtained by taking the Manhattan distance as the fitness function, the genetic algorithm is optimized by using the straight line distance between the starting point and the end point as the fitness function. It is optimized into a more efficient and better path, and the superiority of this method in the ordinary two-dimensional map path optimization problem is proved by experiments.

【Key words】 fitness function; path finding; path optimization; genetic algorithm

0 引言

遗传算法^[1] (Genetic Algorithm, GA) 是一种模拟自然界进化规律的数学计算模型。通过模拟自然界繁衍过程中优秀的父代母代相交, 得到更优秀子代的方法, 同样是对“自然选择, 适者生存”原则的仿真过程。

该算法的主要特点是个体单元的选择和相应个体信息的交换和交集。与普通的搜索算法不同, 遗传算法从种群的初始解开始搜索过程^[2]。其是一种建立在自然选择学说和自然界遗传机制基础上的搜索算法。该算法模拟了自然界中生物体的整个繁殖、杂交和有较小概率会发生的突变过程。一个待优化问题的每一个可能解决方案都被编码为一个“染色体”, 也就是一个个体。每个个体在迭代过程中的不断更新并保留前代优秀部分的过程称为遗传。个体的优缺点通常采用适应度值来评价。算法

开始后随机生成一些初代, 也就是初始解, 通过适应度函数, 筛选出评价较高的个体进行繁衍, 而评价较差的个体则会被慢慢淘汰, 则新产生的这一代的个体, 在表现上理应优于上一代, 因为其继承了上一代的一些优秀品质。通过这种方式, 逐渐走向一个相对较好的方向, 问题的解也就慢慢趋向一个最优解。此算法的优点在于原理操作较为简单, 鲁棒性强, 不易受限制条件的约束, 具有一定的隐含并行性和较强的全局寻优能力^[3]。遗传算法流程如图1所示。

目前, 遗传算法常用于一些非线性、多模型、多目标的函数优化问题, 以及搜索空间较大的组合优化问题。例如在: 旅行商^[4]、装箱、布局优化、车间调度、生产线优化^[5]、生产调度等问题的求解中, 起到了很大的作用。同时, 遗传算法也非常适合解决传统的搜索算法难以适应的非线性问题。在路径规划问题中, Yagvalkya Sharma^[6] 将遗传算法与Dijkstra's 算法进行了对比, 结果显示遗传算法在该

基金项目: 教育部科技司赛尔网络下一代互联网技术创新项目 (NGII20180408); 吉林省教育厅项目 (JKH20200441SK); 吉林省教育教学改革研究课题 (JLJJ719920190723194557, 2017ZCZ045); 吉林师范大学教学成果培育项目 (“三基一新”型计算机专业人才培养模式研究与实践)。

作者简介: 张耀允 (1995-), 男, 硕士研究生, 主要研究方向: 智能信息技术; 谭振江 (1965-), 男, 博士, 教授, 博士生导师, 主要研究方向: 计算机网络、智能信息技术; 周伟 (1979-), 女, 博士研究生, 助理研究员, 主要研究方向: 计算机应用技术; 方大甲 (1981-), 男, 学士, 一级教师, 主要研究方向: 数学建模。

通讯作者: 谭振江 Email: 78770170@qq.com

收稿日期: 2021-03-23

类问题上的优越性;在文献[7]中,首先模拟多目标游戏地图作为实验平台,然后以路径长度、路径安全程度和对游戏角色的耗费为评估目标,提出了一种基于多目标遗传算法的路径规划方法;在机器人路径规划问题中,郑美英^[8]等验证了遗传算法较蚁群算法更智能、更稳定的优点。

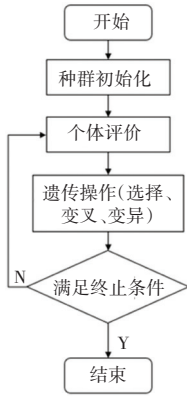


图1 遗传算法流程图

Fig. 1 Flow chart of genetic algorithm

1 工作基础

多数方法中,对于遗传算法的适应度函数均选择使用曼哈顿式。在将地图网格化后,若取 A 、 B 两点分别作为起点和终点(如图2),根据其计算公式:

$$d(i, j) = |x_i - x_j| + |y_i - y_j|. \quad (1)$$

其中, x 、 y 分别是两点的横坐标与纵坐标值。

选用曼哈顿式得出的最优路径则为 a 路径,其路径长度较 A 、 B 两点间的直线距离 b 路径要大一些。因此,若使用两点间直线距离作为路径规划过程中的适应度函数,得出的路径应较使用曼哈顿式得出的路径所费代价更低。

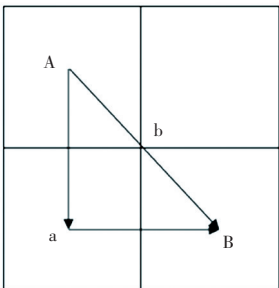


图2 曼哈顿距离与最短距离

Fig. 2 Manhattan distance and shortest distance

2 算法设计

根据上述算法思想基础,将常用的适应度函数由曼哈顿式改进为以最短距离公式为核心的新的适

度函数。进而以传统遗传算法为基础,经过遗传算法中的交叉、变异、产生新子代等过程,对随机路径进行优化。优化算法实现过程如下:

2.1 初始化基因库

定义一个新的基因类,即初始化一个基因库,作为遗传的初代。基因的内容在可行解范围内随机选取,在这里初代即为所要优化的搜索路径, `enerateOnePath` 即是已完成的一个起点为 S 终点为 E 的寻找可行路径的函数。

2.2 适应度函数的选择

由于适应度函数的选取直接影响整个算法的遗传优化性能,并且寻路中可能涉及到左上、左下、右上、右下4个方向,所以选取更为简单直观的两点间的直线距离作为寻路算法的适应度函数。根据距离公式,利用函数将其计算出来并转换为坐标值。

$$\rho = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad |x| = \sqrt{x_2^2 - y_2^2}. \quad (2)$$

2.3 编码

遗传算法是根据问题的内容和属性对问题中的变量进行编码,常用的编码方法有二进制编码、实数编码、整数编码、数据结构编码等。若使用二进制编码表示个体,则二进制数转化为十进制数的解码公式为:

$$F(b_{i1}, b_{i2}, \dots, b_{il}) = R_i + \frac{T_i - R_i}{2^l - 1} \sum_{j=1}^l b_{ij} 2^{j-1}. \quad (3)$$

路径优化过程中的主要变量即是每次前进的方向,在二维地图中,可以大致将每一步的方向概括为上、下、左、右、左上、右上、左下、右下8个方向,采取整数编码方式,将这8个方向编码存入数组中可得: $[(-1, -1), (0, -1), (1, -1), (-1, 0), (1, 0), (-1, 1), (0, 1), (1, 1)]$ 。

2.4 遗传操作

遗传操作是将问题代代优化,最终找到满足要求的最佳后代。其可以分为选择、交叉和变异3个基本操作过程。前两种方法可以实现遗传算法的大部分优化功能,变异操作则提高了遗传算法的寻优能力。

2.4.1 选择

从队伍中选出最优淘汰弱者,这种方法是基于适应度评价的。适应度越大,则被选择的可能性就越大,被选择的个体被分配进入设置好的数组或配对数据库中,等待进一步的操作。目前常用的选择方法有轮盘赌法、最佳个体保留法、期望值法、排名

选择法、竞赛法、线性标准化法等,其中轮盘赌法是最实用、最方便的方法,也是最常用的方法。若要在父代中进行随机选取并开始繁衍,正常情况下进化方向是由选择这一步骤所决定的,而杂交以及变异是不会过度影响当前种群的整体进化方向的。为了满足“越优秀被选到的概率越大”的原则,选取最符合条件的赌轮选择法。每个个体被选择的概率与其适应度大小成正比,轮盘上面积代表对应父代基因的适应度,适应度越大的区域被选中作为父代的概率就越大。计算适应度的实现过程中,为方便之后的比较以及其它操作,将适应度数值添加到数组中。根据适应度函数计算,当前点四周8个方向的适应度值分别与其和做商,结果存入 `pArr` 数组中。在选择过程中随机选出一个小数作为概率,`pArr` 数组中的结果大于这个概率的就会被选中。

2.4.2 交叉

如图3所示,杂交是指将双亲个体的一部分结构重新组合,其目的是从优良父母中选出一部分优良基因,共同形成新一代的个体。交叉是遗传算法获得优良个体最重要的操作。交叉操作是按照一定的交叉概率在数组或数据库中随机的选取两个个体进行的,交叉位置也是随机的,可根据一定概率随机选取。交叉概率一般取得很大,大致范围在0.6~0.9。本文中,根据交叉原理判断后,在满足交叉条件 $cross_p = 0.6$ (一般在实现过程中随机生成)的前提下,可在基因上随机确定一个交叉点 `index1`,并把父代交叉点前后的基因相互交叉,组成一个新的子代。新生成的子代就可能同时保留了父代的优点,若出现自交等特殊情况,可以直接将父代基因存入子代,等待下一轮交叉。

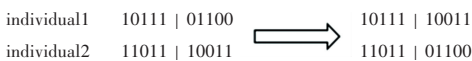


图3 交叉过程

Fig. 3 Cross process

2.4.3 变异

变异是以很小的变异概率 P_m ,随机地改变种群中个体的某些基因值,利用随机数函数产生一个 $[0,1]$ 之间的随机数 `rand`,若 $rand < P_m$,则进行变异操作。变异因其加强了算法的随机性,可使算法维持种群多样性,在接近最优解时可以加速向最优解收敛。但是,变异的概率一定要取一个较小的值,因当概率大于0.5时使得算法随机性大大增加,遗传算法就退化为随机搜索。本文中,在交叉结束后,对新产生的子代基因进行变异操作,若子代基因满

足预先设定的变异概率条件 $mutation_p = 0.4$,则进行变异。如,选取随机数 r ,并与给定概率0.4进行大小的比较,在 $r < mutation_p(0.4)$ 的情况下,随机选取一个变异点 `index2` 进行变异,变异规则是将变异节点处的基因值改变为原基因值的相反数,若为0,则基因不变,进一步降低了变异后随机性。

3 实例分析

针对一个随机生成的路径,对新算法进行测试,在分别对算法中的重要程序段进行解释说明的基础上,选取并编写相应的程序实验环境,对算法效率进行测试。

首先,生成一个长和宽都是100个单位的正方形地图,并在该地图中随机抽取2个点,分别作为路径的起点 S 和终点 E 。在此使用以智能路径算法中常用的曼哈顿距离作为从起点到目标点的适应度函数,在周围8个方向上随机抽取一个节点后,比较该节点与起点距离终点的曼哈顿距离,若小于起点则将该节点直接加入路径,并作为下一个起点加入进一步的搜索过程,否则重复上述过程直到节点中寻到终点。

(1)根据最短距离公式,编写计算适应度的函数

```
def distance(p1, p2):
    return math.sqrt((p1[0] - p2[0]) * * 2
+ (p1[1] - p2[1]) * * 2)。
```

```
def markposition(mark):
    return (mark % width, int(mark / width))
```

```
def posmark(position):
    return position[1] * width + position[0]
```

```
def init(count):
    res = []
    for i in range(0, count):
        res.append(generateOnePath(S, E))
    return res
```

(2)遗传操作中的选择过程

```
def choose(pArr, count):
    res = []
    for i in range(count):
        p = random.random()
        for j in range(len(pArr)):
            if p <= pArr[j]:
                res.append(paths[j])
        break
```

```
return res
```

(3) 遗传操作中的交叉与变异过程

```
def cross():
    n = len(paths)
    res = []
    for i in range(1, n, 2):
        p = crossOneTimes(paths[i], paths[i-1])
    res.extend(p)
    return res
def crossOneTimes(path1, path2)
    index = random.randint(0, len(paths))
    p1 = path1[:index]
    p2 = path2[index:]
    p3 = p1.extend(p2)
```

在交叉结束后,对新产生的子代基因进行变异操作,若子代基因满足预先设定的变异概率条件,则进行变异。抽取到的随机数与给定概率的大小比较方法如下;

```
def mutation(path):
    r = random.random()
    if r < mutation_p:
        return path
    else:
        sz = len(path)
        if sz <= 2:
            return path
```

在满足条件的情况下随机选取一个变异点进行变异。

```
index = random.randint(1, sz - 2)
if paths[index] == 1:
    paths[index] = -1;
if paths[index] == -1:
    paths[index] = 1;
else
    paths[index] = 0;
return paths
```

实验结果显示,初始路径经过 64 次移动结束寻路抵达终点,而经过本文改进后的算法优化后,得出的新路径仅需进行 56 次移动即可成功抵达路径目标,较优化前的初始路径大幅提升了效率。

4 结束语

经过使用两点间距离作为适应度函数的遗传算法对初始路径进行优化后,明显减少了寻路的搜索

次数,进而加强了寻路的效率。当地图数据量过于复杂时,想要找到所需路径时间会很长,只有通过多代遗传来寻找最优解,遗传代数往往会变得很大,有时会出现程序无限制的循环,为了限制这种情况,只能强行限制遗传代数,超过限制则被视为无解。同时,受限于遗传算法的新空间搜索能力有限,在搜索后期效率会出现下降,有时只能得到局部最优解,进而无法得到所需的全局最优。在计算中,众多的随机数,如交叉率、变异率等参数也干扰了路径的选取方式,进而使得在程序中会出现反向搜索,需要通过限制变异率的数值来使其无法演变为随机搜索,来控制这种情况的发生。

由于遗传算法中的编码由简单的数字编码构成,不能完全、准确的表达复杂问题的复杂属性。受编码的属性影响,寻路算法中无法针对带有复杂地形、复杂障碍物等结构复杂的地图进行搜索优化。随着越来越多的复杂 3D 场景的不断涌现,原有传统的智能寻路技术难以适应 3D 场景的复杂性^[9],特别是在复杂环境下,往往需要耗费大量时间才能规划出可行路径^[10]。随着深度学习、神经网络等智能优化算法的发展,将遗传算法与最前沿的优秀算法思想结合,传统遗传算法的各个缺陷也会得到相对应的优化,进而使得该算法在更多的领域中获得更广阔的发展。

参考文献

- [1] HOLLAND J H. Adaptation in natural and artificial system[M]. Ann Arbor: The University of Michigan Press, 1975.
- [2] 李岩,袁弘宇,于佳乔,等. 遗传算法在优化问题中的应用综述[J]. 山东工业技术, 2019(12): 242-243, 180.
- [3] 王坎河,高辉,刘欢. 基于遗传算法的共享助力车调度问题研究[J]. 工业工程, 2021, 24(1): 90-96.
- [4] 张立毅,高杨,费腾,等. 求解旅行商问题的搜寻者遗传算法[J]. 数学的实践与认识, 2019, 49(7): 115-122.
- [5] PEZZELLA F, MORGANTI G, CIASCETTI G. A genetic algorithm for the flexible job-shop scheduling problem[J]. Computers and Operations Research, 2008, 35(10): 3202-3212.
- [6] SHARMA Y, SAINI S C, BHANDHARI M. Comparison of Dijkstra's Shortest Path Algorithm with Genetic Algorithm for Static and Dynamic Routing Network[J]. international journal of electronics & computer science engineering, 2012(12): 416-425.
- [7] 刘大瑞,冯锦. 基于多目标遗传算法的游戏路径规划研究[J]. 软件导刊, 2014, 13(1): 49-51.
- [8] 郑美英,董航. 机器人路径规划算法研究[J]. 辽宁高职学报, 2021, 23(2): 75-79.
- [9] 高天寒,刘文成. 面向 3D 场景智能寻路技术综述[J]. 计算机工程与应用, 2017, 53(1): 16-22.
- [10] 桑和成,宋栓军,邢旭朋,等. 自适应遗传算法在移动机器人路径规划中的应用[J]. 西安工程大学学报, 2021, 35(1): 44-49, 56.