

基于遗传算法的八数码问题的设计及实现

贺计文¹, 宋承祥², 刘 弘¹

(1. 山东师范大学 信息科学与工程学院, 山东 济南 250014;

2. 山东省教育厅, 山东 济南 250011)

摘 要: 介绍了遗传算法(GA)在八数码问题中的应用。首先介绍了八数码问题及遗传算法的相关知识, 分析了求解八数码问题的传统解决方案; 然后给出了八数码问题的遗传算法模型, 并对此模型进行了算法的设计, 即确定编码的表示、选择算子、交叉算子、变异算子及适应度函数; 最后把此算法运用到基于八数码问题的拼图游戏求解过程的动态演示上。文中对此算法进行了多角度试验, 试验表明采用遗传算法解决八数码问题是有效的、稳定的, 具有较高的搜索效率。

关键词: 八数码问题; 遗传算法; 搜索算法

中图分类号: TP301.6

文献标识码: A

文章编号: 1673- 629X(2010)03- 0105- 04

Design and Implementation of Eight Puzzle Problem Based on Genetic Algorithms

HE Ji-wen¹, SONG Cheng-xiang², LIU Hong¹

(1. School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China;

2. Education Department of Shandong Province, Jinan 250011, China)

Abstract: Introduces the application of genetic algorithms in the eight puzzle problem. Firstly depicted the knowledge about the eight puzzle problem and the GA, analyzed the classical solutions. Then presented a model based on GA and designed the algorithm based on the model. Lastly implemented a game which can demonstrate the process of motion dynamically. This algorithm was tested with several aspects, it is proved that the algorithm is available and efficient with the higher search efficiency.

Key words: eight puzzle problem; genetic algorithms; search algorithms

0 引 言

八数码问题也称九宫排定问题, 是拼图游戏的基础, 是人工智能中的经典问题之一。该问题是在一个 3×3 的方格上, 放有数字1~8, 剩下一个格为空格, 空格可以和其上下左右的数字交换; 给定初始状态和目标状态, 求能使初始状态转化为目标状态的一系列交换序列。这个问题实际上是在一个巨大的状态空间(9!)中寻找解的过程。传统的做法是采用搜索算法解决^[1], 比如无启发信息的有界深度优先(DFS)算法、广度优先算法(BFS)、双向广度优先算法或者有启发式信息的A*算法、IDA*(Iterative Deepening)算法。文献[1]采用DFS、BFS、A*算法实现八数码问题的求

解, 并分析和比较了三者所表现出来的性能。另一种方法是采用产生式系统来求解^[2], 其中的控制策略同样需要使用搜索算法。搜索算法的本质是把这些状态空间看作一颗四叉树, 初始状态作为根节点, 求解的过程即为寻找一条从根节点到达目标节点的路径的过程, 很显然, 当目标节点的深度较深时, 会产生很多冗余节点, 而每个节点都需要保存当前状态, 这样会消耗巨大的内存空间, 需要极大的运算量。

遗传算法^[3~6]是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型, 它最初由美国Michigan大学J. Holland教授于1975年首先提出来的^[7, 8]。遗传算法体现了自然界中“物竞天择、适者生存”进化过程, 并有了坚实的理论基础。用遗传算法解决问题, 首先要对待解决问题的模型结构和参数进行编码, 把问题符号化、离散化。基于对自然界中生物遗传与进化机理的模仿, 针对不同的问题, 很多研究者设计了许多不同的编码方法来表示特定问题的解, 开发了许多种不同的遗传算子来模仿不同环境下的生

收稿日期: 2009-06-16; 修回日期: 2009-10-19

基金项目: 国家自然科学基金(60374054, 60743010); 山东省自然科学基金(Z2006G09)

作者简介: 贺计文(1981-), 男, 山东枣庄人, 硕士研究生, 研究方向为遗传算法、机器学习; 宋承祥, 兼职教授, 研究方向为计算机应用; 刘 弘, 博士, 教授, 研究方向为软件智能化、软件开发环境。

物遗传过程。由不同的编码方法和不同的遗传算子就构成了各种不同的遗传算法, 这些遗传算法都有共同的特点, 即通过对生物遗传和进化过程中选择、交叉、变异机理的模仿, 来完成对问题最优解的自适应搜索过程。

实践证明, 遗传算法对于组合优化中的 NP 问题非常有效, 已经被成功地应用在求解旅行商问题、背包问题、装箱问题、图形划分问题等方面, 此外遗传算法在生产调度问题、自动控制、机器人学、图象处理、人工生命、机器学习等方面也获得了广泛的运用。

文中尝试采用遗传算法解决八数码问题, 给出了遗传算法模型, 并实现了一个动态演示八数码解的拼图游戏。

1 遗传算法模型建立

定义 1 设 M 代表空格移动方式的集合, 向上记为 U , 向下记为 D , 向左记为 L , 向右记为 R , 则 M 可以表示为 $M = \{ U, D, L, R \}$ 。

定义 2 设 P 代表 n 条路径的集合, 则 P 可以表示为 $P = \{ P_1, P_2, \dots, P_i, \dots, P_n \}$ 。

定义 3 路径 P_i 是由一系列的空格移动方式 (U, D, L, R) 组成的序列, 则 P_i 可表示为 $P_i = (UDRULDR \dots)$, 路径 P_i 的长度记为 l 。

根据以上定义, 八数码问题的数学模型表示如下:

$$P_i = \text{Search}(P) \tag{1}$$

$$F(P_i) = \sum (100 - \text{归位数字} \times 10) \tag{2}$$

式(1)表示从路径集合 P 中寻找一条可以从初始状态到达目标状态的路径 P_i 。

式(2)表示从初始状态按路径 P_i 移动空格后得到状态的已经归位数字的权值之和, 即适应度评价函数。数字越小权值越大, 步长为 10, 依次为 90, 80, \dots , 10。

2 遗传算法的设计

遗传算法是一种通过模拟自然进化过程搜索全局最优解的方法, 其本质是一种基于概率的随机搜索算法, 其方法是先采用随机的方式建立待解决问题的若干染色体 (又称个体), 即尝试解, 所有染色体称为种群。然后去测试每个染色体能否解决问题, 并评价出每个染色体解决问题的能力, 即适应度。如果不存在解决问题的染色体, 则在种群中根据染色体的适应度选出两个染色体作为父代, 遗传到下一代或通过配对交叉产生两个后代, 两个后代再按一个比较小的概率进行变异, 重复上述过程, 直到得到和父代数量一样的一组后代染色体, 再用这组染色体去测试, 直到得到一

个能够解决问题的染色体。

算法描述如下:

- a. 产生初始种群;
- b. 对每个染色体判断是否为解, 并计算适应度, 如为解退出, 否则进入 c;
- c. 选择;
- d. 交叉;
- e. 变异;
- f. 返回 b。

2.1 编码的表示

遗传算法首要解决的问题是对染色体进行编码, 即确定染色体的基因。具体到八数码问题, 把空格的移动方式 M 作为基因, 因此共有四种基因, 采用两个二进制数表示, 如表 1 所示。

表 1 编码

基因	编码
U	00
D	01
L	10
R	11

染色体表示为基因序列 ($M_1 M_2 \dots M_i \dots M_l$), 其中 M_i 表示第 i 步的移动方式。这种染色体表示一次尝试路径, 如表 2 所示。其中 M_i 在生成初始种群时随机产生。

表 2 染色体样例

染色体	编码	路径
P_1	01 10 11 10 11...	(DLRLR...)
P_2	10 11 00 01 10...	(LRUDL...)

2.2 选择算子

选择算子的目的是把适应度高的染色体遗传到下一代或通过配对交叉产生新的染色体遗传到下一代, 选择算子的操作建立在染色体的适应度基础上。常用的有基于排序的如锦标赛选择方法和基于适应度数值的如轮盘赌选择方法。

文中采用轮盘赌选择方法, 即采用概率的方法从种群中选择一个染色体, 选中的几率正比于染色体的适应度; 若某染色体 i , 其适应度为 f_i , 染色体个数为 n , 则选择概率为:

$$P_i = \frac{f_i}{\sum_{i=1}^n f_i}$$

算法描述如下:

随机取 0 到整个种群的适应度内的一个浮点数; 对种群中的每一个染色体, 进行以下操作:

累加已考察过的染色体的适应度;
如果累加值大于随机取的浮点数, 则选择该染色体。

2.3 交叉算子

交换操作是遗传算法中最主要的遗传操作。交叉算子是把两个父代染色体的部分结构加以替换重组而生成两个新染色体, 在遗传算法中起核心作用。交叉发生的概率由预设的交叉率确定, 如果不发生杂交, 则两个父代直接复制为子代。交叉算子的设计一般与所求解的具体问题有关, 而且要和编码方式相协调。就八数码而言, 可以采用遗传算法标准的一点交叉、两点交叉、均匀交叉等等。两点交叉就是在染色体中随机选取两点, 然后交换两点中的一段基因链; 均匀交叉为从父化染色体中以一定概率随机选取等位基因而构成两个子代染色体。

文中采用一点交叉, 即把染色体看作二进制流, 在每次应用交叉算子时, 首先在 0 到 2l 内随机取一个整数作为交叉点, 然后产生交叉掩码并应用。

例如, 假设采用选择算子得到两个染色体 P_1 : 01 10 1 1 10 11...(DLRLR...), P_2 : 10 11 0 0 01 10...(LRUDL...), 交叉点为 5, 则交叉掩码为 1111100000..., 交叉后的后代为 B_1 : 10 11 01 10 11...(LRDLR...), B_2 : 01 10 10 01 10...(DLLDL...).

2.4 变异算子

变异算子是对染色体的某个基因值作变动, 以维持种群的多样性。变异首先在种群中随机选取一个个体, 对选中的个体以一定的概率随机地改变染色体某个基因值, 同生物界一样, GA 中变异发生的概率很低, 通常取值在 0.001 至 0.01 之间。文中对染色体的每一个二进制位随机取一个浮点数, 和变异率比较后决定是否对该位变异。由于染色体采用的是二进制编码, 所以只需要对需要变异的位求反即可。算法描述如下:

对染色体的每一位(共 2l 个), 进行以下操作:
随机取 0 到 1 内的一个浮点数;
如果该浮点数小于预设的变异率, 则对该位求反。
例如, 对采用交叉算子得到的染色体 B_1 : 10 11 01 10 11...(LRDLR...) 的第一位变异, 假设该位符合变异条件, 则变异后为 B_1 : 00 11 01 10 11...(URDLR...).

2.5 适应度计算

适应度函数的构造是非常重要的, 它决定了算法是否收敛到全局极值, 是对解的质量的评价。不同的问题, 适应性函数的定义方式也不同, 需要经验和不断的尝试。

笔者在大量的试验基础上, 提出了一个对归位数字权值求和的适应度函数, 试验表明这个适应度函数的收敛速度较快, 能较好地解决八数码问题:

$$F(P_i) = \sum (100 - \text{归位数字} \times 10)$$

对种群中的每一条路径 P_i , 从初始状态按该路径移动空格, 忽略掉空格不能到达的移动方式以及重复的移动方式, 得到一个状态, 对该状态使用上面的公式求值, 得到该条路径的适应度。路径的适应度越高, 说明越接近目标状态。

例如, 假设从初始状态使用某一路径 P_i 移动空格得到的状态为 1 4 5 8 9 2 7 6 3 (注: 9 代表空格), 则归位数字为 1、9, $F(P_i) = 90 + 30 = 120$ 。

3 试验结果及结论

拼图游戏是一种益智游戏, 游戏使用 VC6.0 编制, 既可以拼图也可以拼数字。目标状态为正确的图形或者 1~8 数字顺序排列。由于任意给定一个随机的初始状态, 可能没有解, 因此为了确保总是有解, 采取在目标状态的基础上通过随机地移动空格与其相邻的数字得到初始状态, 并把移动的次数作为游戏的难易程度。

创建一个线程, 采用文中的算法求八数码的解, 在此把找到解或者遗传代数达到 500 作为终止条件。空格每移动一次都检查是否为目标状态, 并使用一个数组纪录路径, 用来动态演示空格移动过程。采用文中所述的遗传算法进行问题求解的拼图游戏的界面, 如图 1 所示。

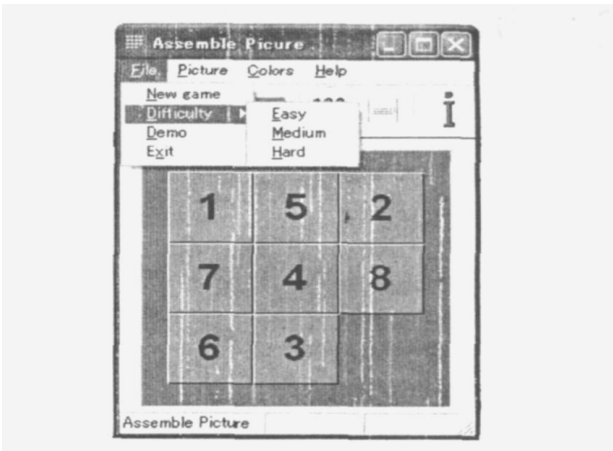


图 1 求解八数码问题的游戏界面

文中设置以下参数进行试验: 种群大小设为 140, 染色体基因数设为 50(移动的次数), 变异率设为 0.75, 交叉率设为 0.09。

试验 1: 分别设置不同的难度级别, 随机选取了一些初始状态, 得到的相应解的遗传代数, 步长如表 3 所示。

表 3 随机取一些初始状态取得解的代数 and 步长

初始状态	Difficulty	遗传代数	步长
4 1 5 8 9 2 7 6 3	Easy(50)	2	12
4 1 9 3 8 6 7 2 5	Easy(50)	12	20
7 1 2 8 9 3 6 4 5	Easy(50)	2	16
4 6 5 7 3 1 9 8 2	Medium(100)	261	20
7 1 2 6 9 8 4 5 3	Medium(100)	3	24
7 1 5 6 9 2 3 4 8	Medium(100)	68	18
1 8 4 3 9 5 7 2 6	Hard(150)	83	22
3 5 1 8 7 2 4 6 9	Hard(150)	101	20
6 7 2 5 9 4 8 1 3	Hard(150)	500	26

试验 2: 难度从 50 连续设置到 150, 每个难度随机取 5 个状态, 取得解的遗传代数的分布图如图 2 所示。

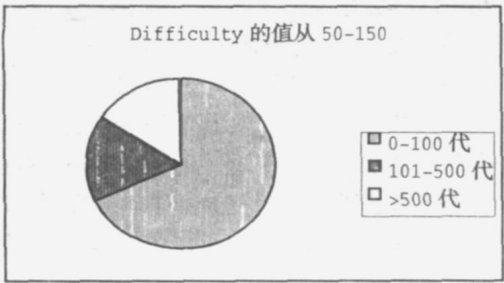


图 2 难度的值从 50~150, 取得解的遗传代数分布图

试验 3: 具体到该游戏的易中难三个难度级别, 每个级别随机取 200 个状态, 取得解的遗传代数的分布图如图 3、图 4、图 5 所示。

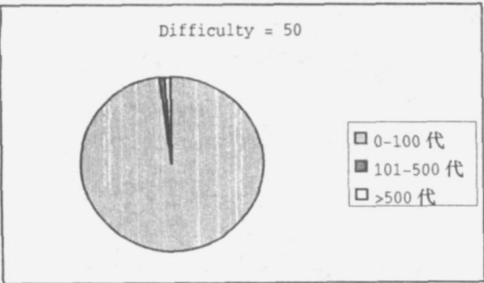


图 3 难度为 50, 取得解的遗传代数分布图

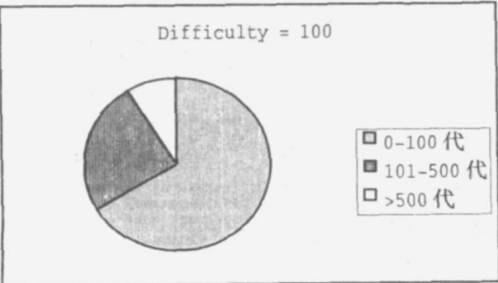


图 4 难度为 100, 取得解的遗传代数分布图

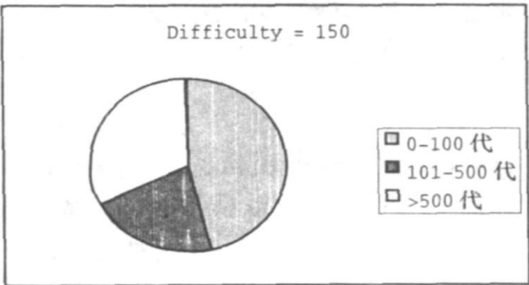


图 5 难度为 150, 取得解的遗传代数分布图

从试验结果看, 每个解的路径步长在 30 步之内, 大部分为最优解。

从图 2 可以看出 85%左右的初始状态可以在 500 代之内得到解。

从图 3、图 4、图 5 可以看出, 随着难度的增加, 500 代之内可以得到解的概率在下降。

4 结束语

文中采用遗传算法对八数码问题的求解进行了设计及实现, 试验结果表明采用遗传算法解决八数码问题是有效的、稳定的, 并且只消耗固定的内存空间。但是, 经试验当 Difficulty 的值设置为 300 以上时, 遗传代数会迅速增加, 只有 50%左右的初始状态可以在 500 代之内得到解。因此, 如何设计出更有效的编码方式以及适应度函数成为提高求解效率的关键, 这为以后的研究指明了方向。

参考文献:

[1] 詹志辉, 胡晓敏, 张军. 通过八数码问题比较搜索算法的性能[J]. 计算机工程与设计, 2007, 28(11): 2505—2508.

[2] 卢格尔. 人工智能: 复杂问题求解的结构和策略[M]. 史忠植, 张银奎, 赵志崑等译. 北京: 机械工业出版社, 2006: 64—108, 147—150.

[3] 陈国良, 王煦法, 庄镇泉. 遗传算法及其应用[M]. 北京: 人民邮电出版社, 2001.

[4] 吴佳英, 郑金华. 遗传算法的研究与发展动向[J]. 衡阳师范学院学报: 自然科学版, 2003, 24(3): 30—33.

[5] 徐清振, 肖成林. 遗传算法的研究与应用[J]. 现代计算机, 2006(5): 19—22.

[6] Davis L. Handbook of genetic algorithms[M]. New York: Van Nostrand Reinhold, 1991: 47—49.

[7] Holland J H. Adaptation in natural and artificial systems[M]. Ann Arbor: University of Michigan press, 1975.

[8] Holland J H. Genetic algorithms[J]. Scientific American, 1992, 9(7): 44—50.