毕业论文

|  |  |
| --- | --- |
| 题目 | 基于遗传算法的tsp问题研究 |

|  |  |
| --- | --- |
| **学 院** | 计算机与科学技术 |
| **专 业** | 计算机科学与技术 |
| **学 号** | 200213137193 |
| **学生姓名** | 张 三 |
| **指导教师** | 李 四 |
| **日 期** | 二·〇〇八年六月 |

摘 要

TSP问题又称为货郎担问题。TSP是一个典型的优化组合问题，它需要求出旅行商从某一城市出发经过所有城市所走路程的最短路径，其可能的路径数与城市个数成指数关系增长。找出有效的近似求解算法具有重要的意义。

选择用遗传算法去解决TSP问题。本论文对各个算子分别选择的是基于序的评估函数、轮盘赌选择法、两点交叉法、两点区间随机排序变异法，并且通过30个城市的实际的例子来验证，结果求出最短路径为421.5977，优于二叉树描述法的结果428.90，启发式搜索法的结果436.01，表明遗传算法在求解TSP问题上是有效的。

关键词：组合优化 ；TSP问题 ；遗传算法 ；最短路径

Abstract

TSP problem is also known as the traveling salesman problem. TSP is a typical portfolio optimization problem and needs to calculate the shortest path that a traveling salesman goes through all cities. The number of the possible paths may grow with index of the number of cities. It is of great significance to find out an effective approximate algorithm.

It is used genetic algorithms to solve the TSP problem. In this paper, the operators are fitness function based on sequence choice, selection with the law of roulette gambling, two- point crossover, two- point random interval mutation. An actual example through 30 cities is got. The result is 421.5977 of the shortest path, which is better than binary tree description with the result of 428.90, heuristic search with the result of 436.01, and shows that the genetic algorithm for TSP is effective.

**Key words**: Combinatorial Optimization ; TSP ; Genetic Algorithm ; the Shortest Path

目 录

[目 录 4](#_Toc467333343)

[绪论 5](#_Toc467333344)

[1 遗传算法的设计思想 8](#_Toc467333345)

[2 系统实现 9](#_Toc467333346)

[3 程序的运行演示 12](#_Toc467333347)

[4 测试运行 13](#_Toc467333348)

[4.1 模块测试 13](#_Toc467333349)

[4.2 整体测试 13](#_Toc467333350)

[4.2.1 在测试过程中使用到的调试技术： 13](#_Toc467333351)

[4.2.2 评估运行的可靠性问题： 13](#_Toc467333352)

[4.3 测试结论 14](#_Toc467333353)

[结论 15](#_Toc467333354)

[参考文献 16](#_Toc467333355)

绪论

一、遗传算法概述和发展背景

（1）遗传算法简介

遗传算法（Genetic Algorithm，简称GA）是模拟达尔文的遗传选择和自然淘汰的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法，它是由美国Michigan大学J.Holland教授于1975年首先提出来的，并出版了颇有影响的专著《Adaptation in Natural and Artificial Systems》，GA这个名称才逐渐为人所知，J.Hilland教授所提出的GA通常为简单遗传算法（SGA）。遗传算法是从代表问题可能潜在的解集的一个种群（population）开始的，而一个种群则由经过基因（gene）编码的一定数目的个体(individual)组成。每个个体实际上是染色体(chromosome)带有特征的实体。染色体作为遗传物质的主要载体，即多个基因的集合，其内部表现（即基因型）是某种基因组合，它决定了个体的形状的外部表现，如黑头发的特征是由染色体中控制这一特征的某种基因组合决定的。这个过程将导致种群像自然进化一样，后生代种群比前代更加适应于环境，末代种群中的最优个体经过解码（decoding），可以作为问题近似最优解[1]。

（2）遗传算法特点

遗传算法是一类可用于复杂系统优化的具有鲁棒性的搜索算法，与传统的优化算法相比，主要有以下特点：

（1）遗传算法以决策变量的编码作为运算对象。传统的优化算法往往直接决策变量的实际植本身，而遗传算法处理决策变量的某种编码形式，使得我们可以借鉴生物学中的染色体和基因的概念，可以模仿自然界生物的遗传和进化机理，也使得我们能够方便地应用遗传操作算子。

（2）遗传算法直接以适应度作为搜索信息，无需导数等其它辅助信息。

二、关于TSP问题的简介

TSP（Traveling Salesman Problem）[2]问题又称为货郎担问题、旅行商问题，是出现在许多应用中的一个组合优化问题。该问题可简单的描述为：已知n个城市各城市间的相对距离，某一旅行商从某个城市出发访问每个城市一次且仅一次，最后回到出发城市，怎样安排才使其所走路线最短。TSP是一个典型的优化组合问题，它需要求出旅行商从某一城市出发经过所有城市所走路程的最短路径，其可能的路径数与城市个数成指数关系增长[3]。用图论的术语来说，假设有一个图G=(V,E)，其中V是顶点集，E是边集，设D＝(dij)是由顶点 i与顶点j间的距离所组成的距离矩阵，TSP就是求出一条通过所有顶点且每个顶点通过一次的最短路径的哈密尔顿回路[4]。通常说的 TSP都指对称型的 TSP。若对于n个城市 V＝｛V1，V2，…，Vｎ｝的一个访问顺序为 T＝(t1，t2，…，tn)，其中tｉ∈V（i＝1，2，…，n），且记 t n+1＝t1。则数学模型可以描述为公式（1.1）。

$$P\_{i}=F\_{i}/\sum\_{i=1}^{N}Fj(i=1,…,N)$$

（1.2）

三、本论文的研究内容

本次设计的研究内容包括：

（1）用C++ 写出基于遗传算法解决TSP问题的核心程序，本系统用的是两点交叉法和两点区间随机排序的变异法；

（2）用C++ 写出一个界面来调用所写的遗传算法程序解决给定的数据；

1 遗传算法的设计思想

以遗传算法的思想去解决TSP问题，即要在众多的城市路径中找到一个最短的，我们模拟生物进化的程序，即遗传的方式，我们先以一定的方式生成一个初始化群体，为每个染色体计算评价函数，然后群体竞争选择，种群交叉种群变异，如此迭代下去直到迭代代数达到要求找到最短的路径。

下面从遗传算法的具体方面来进行设计思想的分析：

一、算法设计

目前，求解TSP问题的较为常用的方法有二叉树描述法、启发式搜索法、最近邻法、神经网络法、模拟退火法、遗传算法等。遗传算法是模拟生物在自然界中的遗传和进化过程而形成的一种自适应全局概率搜索算法，具有良好的全局寻优能力，成为解决TSP问题的有效方法之一。

二、遗传编码

遗传算法的编码时将待求问题的解的形式变换成遗传算法所面对的基本编码窜对象，以便于遗传运算。对于最短路径问题，其可行解的形式一般为结点下标联结成的数字串，因此在遗传算法中的编码方式一般为符号编码。具体可以分为以下几种：近邻编码、序编码(Grefenstette 编码)、边编码、自然编码等。在本设计中用到的是自然编码。

2 系统实现

本系统使用vc++编写，将遗传算法和界面程序分开编写，这样修改起来就比较方便，并且程序的结构看起来也很清晰，理解也很容易。在本程序中有一个功能模块，通过这个模块我们可以读取已经保存好在文件中的城市坐标文件，然后可以通过运行遗传算法来计算给出的数据并得出所要求的结果。

本论文通过调用OpenDataFile（）函数读取城市坐标信息。该函数输入参数是strFileName，表示文件名称字符串引用，函数返回值是文件中所含城市个数。其中，城市坐标文件的格式要求为：

城市名称 X轴坐标 Y轴坐标

读取城市坐标文件的内容到一个容器名为vecCity的结构中，主要代码如图2.1所示。

if( FileType == 1 )

{

int ncityindex = 1;

while( DataFile.ReadString(strReadString) )

{

strReadString.TrimLeft();

strReadString.TrimRight();

CString cityName, citycodx, citycody;

int nspace = 0;

nspace = strReadString.Find(" ");

// 将得到文件串的左边子串传给cityName作为城市名；

if( nspace > 0 )

cityName = strReadString.Left( nspace );

strReadString = strReadString.Mid( nspace+1 );

nspace = strReadString.Find(" ");

// 将读到的字符串的左边串传给citycodx作为x坐标；

if( nspace > 0 )

citycodx = strReadString.Left( nspace );

// 将读到的字符串的右边串传给citycody作为y坐标；

strReadString = strReadString.Mid( nspace+1 );

citycody = strReadString;

SYCity tmpCity;

// 得到城市名；

tmpCity.m\_strName = "城市 "+cityName;

tmpCity.m\_nIndex = ncityindex;

// 得到城市的x轴坐标；

tmpCity.m\_Coordinate.m\_fcodx = atof( citycodx );

// 得到城市的y轴坐标；

tmpCity.m\_Coordinate.m\_fcody = atof( citycody );

// 将得到的城市放在城市列表中；

vecCitys.push\_back( tmpCity );

// 城市的索引号往后移一位；

ncityindex++;

}

}

图2.1读取城市坐标代码

轮盘赌方式竞争选择染色体的模型如图2.2所示。



图2.2 一个表现轮盘赌游戏选择的图标，在这个轮盘中可以被选择的数m是6。圆弧中的那些数字对应的是事物被选择中的可能性。

3 程序的运行演示

1. 运行GAApp.exe，界面如图3.1所示。



图3.1 程序运行界面

1. 选择“城市坐标文件”，界面如图3.2所示。



图3.2 打开“打开文件”对话框

4 测试运行

测试在程序设计阶段有两个时期，通常在编写每个模块后做单元测试，另一个时期是对程序的综合测试。

4.1 模块测试

在模块测试时我们主要从以下几个方面考虑：

模块接口；

局部数据结构；

重要执行通道；

出错处理通道；

影响上述方面的边界条件。

测试时进行代码审查，从数据类型，变量声明，数据结构进行审查，然后进行功能测试，从输入一些简单的数据开始执行一遍，观测运行期间变量的变化，运行中值的变化范围。改变测试方案来变换另一个角度进行测试，发现错误并记录，修改代码，测试条件使程序通过多层分支，判别运行结果从而完成模块测试。

4.2 整体测试

4.2.1 在测试过程中使用到的调试技术：

采用debug调试语句，跟踪数据；

嵌入打印语句，输出中间结果；

利用Visual Studio 6.0中调试工具，从调试窗口观测变量的变化；

设置断点，观察程序在断点附近的状况。

4.2.2 评估运行的可靠性问题：

结果正确；

运行速度；

空间利用率；

算法的可行性。

4.3 测试结论

本系统经过测试，发现每个模块都是正确的，结果也是正确的，运行速度合理，空间利用率高，证明算法是可行的。

结论

本文系统地分析了基于遗传算法解决TSP问题开发背景以及开发过程，其具体内容如下：

介绍了遗传算法的发展背景及遗传算的解决问题的思想和流程；

阐述整个遗传算法解决TSP问题的结构及解决原理；确定了遗传算法所选择的基因选择方法、交叉和变异方法等。

分析并解决实现中的若干技术问题，建立完整的遗传算法解决TSP问题的程序，通过程序能够进行实际例子的计算。

本系统在开发中尚有待完善的地方，例如，如果用matlab来绘制所走的城市路径会使结果更加直观，未对防止局部最优解进行讨论，而陷入局部最优解是遗传算法在优化过程中比较容易出现的问题。

参考文献

[1] 周明．遗传算法原理及应用[M]．北京：国防工业出版社，1～20

[2] 陈国良，王煦法，庄镇泉，王东升．遗传算法及其应用[M]．北京：人民邮电出版社，1996．1～30

[3] Michalewicz Z.　How to Solve It—Modern Heuristick[M].　German: Berlin Heidelberg, 2000.

[4] 潘正君，康立山．演化计算[M]．北京：清华大学出版社，2004．1～30

致谢

在本程序的开发期间，我得到了老师和同学的很多帮助。因此，才得以完成了本程序的开发。

首先，在程序的设计过程中，老师给予了我充分而又耐心的指导，并对我开发的程序给予了肯定和建议，使得我的设计能够不断得到完善。

其次，在毕业设计期间，同组同学的鼓励和他们对课题设计的积极态度给了我很大帮助，在此对他们致以诚挚的谢意。

最后，对于引用和学习的那些著作和论文的作者也非常感谢，这些前辈们的成果使得我受益匪浅，让我可以站在巨人的肩上，利用前辈们的成果使得我的程序可以顺利地完成。