目录

[第一章 前言 2](#_Toc417336788)

[1. 背景 2](#_Toc417336789)

[2. 解决方案 2](#_Toc417336790)

[3. 遗传算法的发展历史 3](#_Toc417336791)

[4. 本文的内容介绍 4](#_Toc417336792)

[第二章 平台和工具介绍 5](#_Toc417336793)

[1. C++模板 5](#_Toc417336794)

[2. MFC框架 5](#_Toc417336795)

[3. Visual Studio 2010的一个MFC实例 7](#_Toc417336796)

[第三章 遗传基因算法 9](#_Toc417336797)

[1. 算法简介 9](#_Toc417336798)

[2. 遗传算法涉及的几个名词 10](#_Toc417336799)

[3. 遗传算法应用的关键操作 10](#_Toc417336800)

[第四章 基于遗传基因算法的TSP问题 10](#_Toc417336801)

[第五章 实验数据和分析 11](#_Toc417336802)

[第六章 结论和总结 12](#_Toc417336803)

# 前言

## 背景

随着经济的发展，物流正处于高速发展阶段，日常生活离不开物流，物流市场竞争日趋激烈。如何应对物流的全球化，信息化高速发展趋势，为顾客提供更高效快捷的物流服务，路径规划显然占据极为重要的地位，这对路径规划提出了更加科学和高效管理的要求。

物流行业在我国的新兴经济产业中占据了重要了地位，以成为推动经济快速增长的“加速器”。而物流配送作为物流系统的关键一环，影响着物流的整个运作过程和运输企业的发展趋势。采用科学、高效、合理的方法来进行物流配送，是物流配送领域的主要重要研究内容。近来，国内外均有大量的学者对物流配送过程中最优路径选择的问题，开展了大量深入的研究，从早期货运路径问题研究，到根据提出各种约束模型和条件不断变化的货运最优路径研究，以及随着计算机学科的不断发展从而提出的针对解决物流配送路径最优化的模型（TSP问题）和算法等方面，都取得丰硕的研究成果。

配送是物流系统中与消费者相关联的一个重要环节，是一个货物从物流节点到达收货人的过程。运输系统是配送系统中一个重要的子系统，运输费用约占整体物流损耗的50%，降低物流的成本应该首先考虑从降低货运成本开始。选取适当的路线是减少货运成本开销的关键所在，选取合适的运输流线，可以使客户需求的响应速度加快，并提高服务质量，增强物流系统的满意度，大幅降低成本。

路径规划问题与tsp（旅行商问题）求解有着密不可分的关系。路径规划的核心问题就是tsp问题，可以借助求解tsp的技术，研究最短路径决策系统的解决方案。

TSP问题是组合优化问题中最典型的问题之一，并且是一个np难问题，吸引了数学，运筹学，物理，人工智能等多个领域的研究者。TSP问题具有很强的应用背景，展示了组合优化的特点，并因此成为多种算法的首要测试实例，如模拟退火、禁忌搜索、神经网络以及进化方法等等。很多实际问题最终都可以转化为TSP问题，本文研究的问题就是一个最鲜明的例子，解决TSP问题具有重要的理论和实践意义。

## 解决方案

本文通过使用遗传算法来解决TSP问题。遗传算法是借鉴生物界生存法则发展起来的随机搜索算法，其基本操作是基于概率。物竞天择,适者生存，是生物界的自然法则，遗传算法将其运用于对现实问题的研究中。

除了核心问题由遗传算法来解决外，用户操作接口将使用mfc框架来实现。在后面的章节将详细介绍遗传算法和mfc的具体实现。

## 遗传算法的发展历史

遗传算法（genetic algorithm）是近年来迅速发展起来的一种随机搜索优化算法，其思想是基于达尔文的进化论和Mendel的遗传学说。后由密歇根大学教授Holland以及其学生于1975年创建。而后，遗传算法得到很多国内外学者的广泛关注。

1967年，Holland的学生J.D.Bagley在博士论文中首次使用了“遗传算法”。 1971年，R.B.Hollstien在其博士论文中首次使用遗传算法运用于函数优化问题。1975年Holland出版了著名专著《自然系统和人工系统的自适应》，这是一本系统论述遗传算法的书籍。同年，K.A.De Jong在他的博士论文《一类遗传自适应系统的行为分析》中把Holland的模式理论和实验计算结果结合，他的研究工作和结论对于遗传算法的应用打下了坚实的基础。

1985年，美国第一次召开遗传算法国际会议，并成立国际遗传算法学会，每两年举行一次。

1989年，Holland的学生D.E.Goldberg出版了专著《搜索、优化和机器学习中的遗传算法》。该书系统的总结了遗传算法研究的主要成果，对遗传算法及其应用进行全面的介绍和论述。同年，美国斯坦福大学的Koza创造性地提出了用层次化的计算机程序来解决问题的遗传算法程序设计方法，并成功地解决了许多问题。

在欧洲，1990年开始，每隔一年都会举办一次Parallel Problem Solving from Nature学术会议，会议主要内容之一正是遗传算法。

1991年，L.Davis出版了《遗传算法手册》，这本书中包含了遗传算法在工程技术以及生活中的大量应用实例。

1992年，Koza发表了他的专著《遗传程序设计:基于自然选择法则的计算机程序设计》。1994年，他继续出版了《遗传程序设计，第二册:可重用程序的自动发现》加深了对遗传程序设计的研究，在程序设计自动化方面展现新局面。关于遗传算法的学术论文也不断在《Artificial Intelligence》、《Parallel Computing》、《Information science》等杂志上发表。

1993年，MIT出版社创刊了《Evolutionary Computation》。

1997年，IEEE又创刊了《Transactions on Evolutionary Computation》。《Advanced Computational Intelligence》杂志即将发刊，由模糊集合的创始人L.A.Zadeh教授作为名誉主编。

目前，关于遗传算法研究的热潮持续渐进，越来越多从事于不同领域的研究者和学者已经投入到有关遗传算法的研究或应用之中。遗传算法无论是理论研究还是应用领域都成了十分热门的课题。而遗传算法在应用领域的研究则显得更加活跃，很多领域都开始使用这一方面的研究成果。除此之外，一些新的理论成果和解决方案在应用领域的研究中同样得到了迅速的发展，这些都是遗传算法的灵活性大大提高。遗传算法在应用方面变得更新颖、更工程化。

## 本文的内容介绍

第一章主要讲解本文研究的问题的背景，然后提出问题的解决方案遗传算法，进而对遗传算法的出现发展作了简要概述。

第二章主要讲解问题研究过程中使用的平台和工具，本次研究课题所有实验将通过C++以及mfc框架来实现。

第三章主要讲解遗传算法的具体实现。

第四章使用遗传算法来解决TSP问题，并详细介绍遗传算法操作过程中，关于tsp问题的具体细节。

第五章讲解遗传算法在TSP问题中的实验结果，数据分析。

第六章结论和总结

# 平台和工具介绍

本文所涉及算法的所有代码将使用C++和windows提供的mfc框架来实现。

## C++模板

C++是在C的基础上开发的一种集成面向对象和面向过程、泛型编程的编程语言。应用较为广泛，是一种静态数据类型检查的，支持多重编程的通用程序设计语言。C++有很多的优点，但这里并不一一介绍，这里只简要介绍我们使用到的C++模板，为什么要使用模板，并且不涉及具体语法细节（语法细节可以参见《C++primer》，由于其他语言同样支持相应的解决方案，如C#泛型（从功能上看）。

模板是根据参数类型生成函数和类的机制（有时称为“参数决定类型”），通过使用模板，可以只设计一个类来处理多种类型的数据，而不必为每一种类型分别创建类。譬如C++标准库中的大部分容器，为了适应不同类型的需要，而将各种容器定义为模板类。

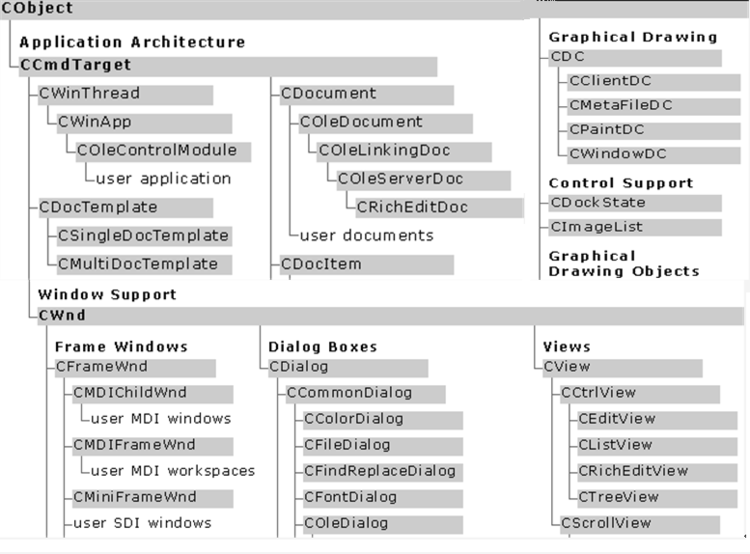
对于遗传算法，为了使算法易于扩展，模板类是一个好方法。遗传算法的整体框架是一套固定的流程，犹如自然界的发展规律，繁殖，变异，更新换代，并且文明程度不断向前发展，但其中的每种生物的演化过程却截然不同。自然界中，各种物种繁殖后代的方式截然不同，在不同的环境下发展情况也有所不同。类似的，遗传算法不仅可以解决TSP问题，也可以解决作业调度问题，那么遗传算法可以看成是自然规律，而TSP和作业调度问题可以看成自然界中的一种生物，这种生物符合大自然的发展规律。但个中的演化过程却各自不同，这里举一点，譬如遗传算法中个体的适应度，TSP的适应度是以路径的总长度来权衡，而作业调度则是总体时间消耗，计算方式截然不同。（关于适应度，后续会详细讲解）。

因此，我们使用C++模板定义了一套遗传算法的框架，当我们使用TSP时，则定义好TSP对应于遗传算法的各种操作，然后传给遗传算法的模板。具体操作可以是这样定义一个GA<Chromosome\_Type>的模板类，GA中定义了遗传算法的整体框架，Chromosome\_Type是一个染色体类型，如果我们解决TSP，那么我们传入的就是TSP的染色体，如果我们解决的是作业调度，那我们将传入作业调度的染色体，在Chromosome\_Type中，我们将定义TSP和作业调度应用于遗传算法的细节。这样当每次我们需要使用遗传算法时，就可以重复使用GA<Chromosome\_Type>，再定义对应问题的细节，大大提高代码的可重用性。

## MFC框架

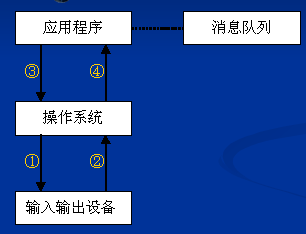
MFC库是开发Windows应用程序的C++接口。MFC提供了面向对象的框架，采用面向对象技术，将大部分的Windows API 封装到C++类中，以类成员函数的形式提供给程序开发人员调用。

MFC类结构图如下：



在我们的实验程序中，我们主要使用MFC来提供用户操作接口，也就是界面的实现。MFC为我们提供了一套非常简单便捷的接口，使我们对于界面的实现变得更加简单方便，这里简要讲解一下MFC的消息映射机制。

Windows程序是一种事件驱动方式和程序设计模式，主要是通过消息机制来实现的。操作系统将每个接受到的事件都封装成一个消息的数据结构来传递给应用程序，而应用程序则循环不断地从消息队列中取出消息，进行响应。MFC正是基于这种实现机制。



MFC中的消息映射机制的实现方法：在每个能接收和处理消息的类中，定义一个消息和消息的处理函数静态对照表，即消息映射表。在消息映射表中，消息与对应的消息处理函数指针成对出现。某个类能处理的所有的消息及其对应的消息处理函数的地址都列在这个类所对应的静态表中。当有消息需要处理时，程序只要搜索该消息静态表，查看表中是否函数该消息，就可以知道该内能够处理此消息。如果能，则依照静态表找到对应的消息处理函数。否则，将消息传给其父类。

除此之外，MFC框架提供了良好的编程接口，捕获窗口的各种事件，只需要简单的几个步骤就可以实现。本文的核心是讨论算法，因此我们尽量选择更加简便的工具来实现无关紧要的东西，因此，这里并不详细介绍编程细节，在我们的程序中，我们直接使用编译器提供的类向导中的add handler功能来添加处理函数，对界面的事件处理将更加简单，快捷，方便。下一节中，我将通过一个简单的实例在讲解如何添加事件响应函数的。

除此之外，微软提供的C#.net同样提供了类似的解决方案。但我们不一一细讲，因为本文的主角是基于遗传算法的TSP问题。

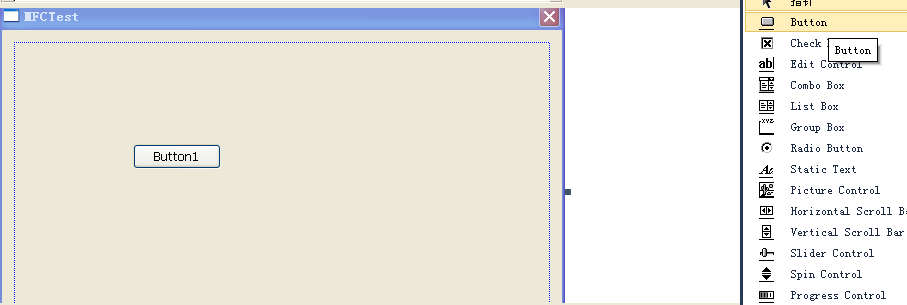
## Visual Studio 2010的一个MFC实例

Visual Studio 2010是微软的一个开发环境，继承多种windows框架，对于MFC有很好的支持。

我使用Visual Studio 2010创建一个MFC应用程序，选择好保存路径后点击“下一步”，会出现如下窗口，



后我们选择基于对话框，直接点击完成（其他细节，请参考MFC的相关书籍）。然后我们在右边的工具栏中拖放一个button到窗体上，然后右键->添加处理函数->选择消息类型和类->然后点击添加编辑，消息类型主要是选择事件的处罚方式，可以是单击或双击或其他，类则是负责监听这个消息的类，前面有关于消息映射的简要介绍。





然后我们添加代码

void CMFCTestDlg::OnBnClickedButton1()

{

// TODO: 在此添加控件通知处理程序代码

MessageBox(\_T("Test in MFC"));

}

当我们编译运行后，我们点击按钮时，将会弹出一个提示框。后续程序的界面，我们基本上是以这种操作来实现我们的界面。

# 第三章 遗传基因算法

TSP问题：有N个城市，寻找一条路径，从一个城镇出发，经过所有城市一次，最后回到出发城市，要求距离最短。

## 算法简介

遗传算法是一种借鉴自然遗传规律的算法，是一种随机搜索算法。遗传算法的规律和大自然类似，他不知道自然界中个体的演化规律，而只关注遗传过程中最终生存下来的个体，物尽天择适者生存，只有更强的适应能力才能不断得到发展。遗传算法正是如此，遗传算法中的个体是染色体，是由所研究对象按照某种条件产生的一组随机的数字编码，初始情况下，我们产生一组染色体作为遗传算法的初始种群，在这个种群中，根据具体问题提供对应的适应度计算公式，淘汰适应度低的染色体，选择适应度高的染色体，这些染色体作为父代，按照自身的规律，（染色体之间的差别好比鸭子生蛋，而老虎是胎生的，都是遗传后代，但有所区别）进行一系列的遗传操作后，形成新的一代种群，按照这种规律，不断产生更加优秀的染色体，直至逼近最优解。算法结构图如下:

开始

输入配置参数

初始化种群

计算适应度

选择优秀的染色体

交叉、变异生成新的种群

满足退出条件

否

是

输出最优染色体

结束

## 遗传算法涉及的几个名词

* 染色体:染色体是遗传算法中的个体,我们算法中操作交叉,变异都是对应得对象都是染色体.
* 串:串是染色体的表现形式,是以提取染色体的特性来形成具体的串,是一些特定长度的编码集合。
* 基因：基因是串中的元素，是编码集合中的一个编码，也就是说，串是由基因对应的数字组成的。
* 种群：染色体的集合，也就是说，在算法中，我们主要以多个串来组成种群。
* 种群大小：染色体集合中个体的个数，遗传操作中每一代的成员数。
* 基因位置：这是指基因对因的编码在串中的位置，通俗来讲，如果串是用素组，那么位置则是下标
* 串结构空间：串中，基因随机组合而成的集合。
* 适应度：表示染色体对环境的适应程度，在具体问题中，则可以考虑为权衡最优解的某种属性，如损耗最小，距离最大，花费金额等等。

## 遗传算法应用的关键操作

### 染色体的编码方式：

在遗传算法中，编码是指将问题的可行解从其描述空间转化为可由遗传算法理解的搜索空间。编码是遗传算法的一个关键步骤，选择正确的编码方式对后续的操作交叉、变异操作都会产生影响。常用的编码方式有以下几种：二进制编码、浮点数编码、格雷码、参数编码

# 第四章 基于遗传基因算法的TSP问题

# 第五章 实验数据和分析

# 第六章 结论和总结