



工业和信息化部“十二五”规划教材

管理建模与仿真

GUANLI JIANMO YU FANGZHEN

主编 姜金贵

HEUP 哈尔滨工程大学出版社

选题策划：卢尚坤
责任编辑：张晓彤
封面设计：恒润设计

管理建模与仿真

GUANLI JIANMO YU FANGZHEN

上架建议：管

ISBN 978-7-5661



9 787566 111005 >

定价：36.00元



工业和信息化部“十二五”规划教材

管理建模与仿真

GUANLI JIANMO YU FANGZHEN

主编 姜金贵

副主编 宋 艳 杜 蓉

HEUP 哈尔滨工程大学出版社

内 容 简 介

本书的特色是突出建模与仿真方法在管理领域中的应用,强化案例分析与操作,致力于提高读者的实践操作能力。本书将管理科学问题进行分类总结,在阐述系统建模、系统仿真等概念、内涵、基本理论、基本方法与技术的基础上,重点围绕基于 MATLAB 和 NetLogo 两种工具软件的管理科学问题的建模与仿真进行了详细实例讲解与分析。在 MATLAB 工具软件应用部分,主要围绕管理领域中的预测问题、分类问题、评价问题、最优化问题等,对包括神经网络、灰色系统、层次分析法、遗传算法、模拟退火算法、粒子群算法等在内的建模仿真方法的基本原理、应用步骤、实例、程序代码等进行了详细讲解;在 NetLogo 工具软件部分,强调了建模仿真技术的操作性,通过层层深入的实例逐步明晰了该建模仿真技术的操作步骤,最后结合网络危机信息传播演化仿真这一综合实例系统地讲解了如何对某一管理科学问题进行模型构建与仿真。

本书不仅可以作为管理类研究生和高年级本科生的教科书,同时也适用于对管理建模与仿真方法和技术有兴趣的读者。

图书在版编目(CIP)数据

管理建模与仿真 / 姜金贵主编. -- 哈尔滨: 哈尔滨工程大学出版社, 2015.5

ISBN 978-7-5661-1005-3

I. ①管… II. ①姜… III. ①管理学-建立模型
IV. ①C931

中国版本图书馆 CIP 数据核字(2015)第 103510 号

选题策划 卢尚坤

责任编辑 张晓彤

封面设计 恒润设计

出版发行 哈尔滨工程大学出版社
社 址 哈尔滨市南岗区东大直街 124 号
邮政编码 150001
发行电话 0451-82519328
传 真 0451-82519699
经 销 新华书店
印 刷 哈尔滨市石桥印务有限公司
开 本 787 mm × 1 092 mm 1/16
印 张 15.25
字 数 397 千字
版 次 2015 年 10 月第 1 版
印 次 2015 年 10 月第 1 次印刷
定 价 36.00 元

<http://www.hrbeupress.com>

E-mail: heupress@hrbeu.edu.cn

前 言

管理建模与仿真是一门综合运用管理知识、数学建模理论与方法、决策科学方法、现代计算机仿真技术研究解决管理实践问题的科学,是管理类专业的一门重要基础课程。本书是作者在长期的管理科学、建模与仿真领域的教学、科研实践基础上,对管理领域中主要研究问题进行的系统归纳总结,并广泛借鉴国内外管理建模与仿真方面文献凝练而形成的。书中将管理科学问题进行了分类总结,在阐述了系统建模、系统仿真等概念内涵、基本理论、基本方法与技术的基础上,重点围绕基于 MATLAB 和 NetLogo 两种工具软件的管理科学问题的建模与仿真结合实例进行了详细的讲解与分析。

本书中,在 MATLAB 工具软件应用部分,主要围绕管理领域中的预测问题、分类问题、评价问题、最优化问题等,对包括神经网络、灰色系统、层次分析法、遗传算法、模拟退火算法、粒子群算法等在内的建模仿真方法的基本原理、应用步骤、实例、程序代码等进行了详细讲解;在 NetLogo 工具软件部分,强调了建模仿真技术的操作性,通过层层深入的实例逐步明晰了该建模仿真技术的操作步骤,最后结合网络危机信息传播演化仿真这一综合实例系统地讲解了如何对某一管理科学问题进行模型构建与仿真。

本书不仅可以作为管理类研究生和高年级本科生的教科书,同时也适用于对管理建模与仿真方法和技术有兴趣的读者。本书的特色是,突出建模与仿真方法在管理领域中的应用,在系统梳理各类管理科学问题的基本原理、基本方法与流程的基础上,介绍了解决各类问题的建模与仿真方法及技术,并强化案例分析与操作,力图使读者能够通过案例的学习与操作做到触类旁通,从而掌握这些管理科学问题的建模与仿真方法,以便更好地应用到管理实践与科学研究。

在阐述方式上,本书力求简明扼要、深入浅出,运用大量的实例来讲解管理问题的建模与仿真技术及工具软件的使用,致力于提高读者的实践操作能力,为研究解决管理中出现的多类问题提供方法、技术、工具、实例的支持。

本书共分 9 章,其中第 1 章、第 2 章、第 7 章由宋艳执笔,第 3 章、第 5 章、第 6 章、第 8 章的 8.4 节、第 9 章的 9.1~9.3 节由姜金贵执笔,第 4 章、第 8 章的 8.1~8.3 节、第 9 章的 9.4~9.8 节由杜蓉执笔。全书最后由姜金贵负责统稿和审定。

本书的编写得到了工业和信息化部“十二五”规划教材基金资助,哈尔滨工程大学经济管理学院灾难与危机管理研究所团队的支持,在此向支持本书出版的领导、学者表示深深的谢意。

由于作者水平有限,书中不妥之处在所难免,恳请有关专家和广大读者批评指正。

编 者

2015 年 1 月

目录

第 1 章 导论	1
1.1 系统及其分类	1
1.2 系统建模	3
1.3 系统仿真	6
1.4 系统建模与仿真技术的应用	8
第 2 章 建模的基本理论	10
2.1 建模的原则与模型的有效性	10
2.2 演绎推理	12
2.3 系统辨识理论	14
2.4 相似理论	17
2.5 灰色系统理论	19
2.6 自组织理论	21
2.7 元胞自动机理论	25
第 3 章 仿真方法与技术	32
3.1 仿真技术的分类	32
3.2 仿真的一般过程	35
3.3 仿真技术的特点和作用	37
3.4 仿真技术的应用	38
3.5 仿真技术的发展	39
第 4 章 MATLAB 基础	41
4.1 MATLAB 的基本构成与操作	41
4.2 变量与函数	50
4.3 矩阵及其运算	53
4.4 M 文件与编程	58
4.5 基本绘图函数及其操作	59
第 5 章 神经网络模型及仿真	79
5.1 人工神经网络基本理论	79
5.2 BP 神经网络算法	84
5.3 BP 神经网络的 MATLAB 工具箱	86
5.4 组建神经网络的注意事项	93
5.5 基于 MATLAB 工具箱的神经网络建模仿真实例	96
第 6 章 灰色预测建模仿真	107
6.1 预测概述	107
6.2 灰色系统概论	119

6.3 灰色预测模型	121
6.4 灰色预测的 MATLAB 建模仿真实例	125
第 7 章 层次分析法建模仿真	134
7.1 评价概述	134
7.2 层次分析法的基本原理	138
7.3 层次分析法的 MATLAB 建模仿真实例	142
第 8 章 现代优化模型与仿真	146
8.1 最优化问题概述	146
8.2 遗传算法及其 MATLAB 仿真	149
8.3 模拟退火算法及其 MATLAB 仿真	159
8.4 粒子群算法及其 MATLAB 仿真	166
第 9 章 复杂适应系统建模与仿真	174
9.1 复杂适应系统理论的基本观点与概念	174
9.2 个体怎样适应和学习	179
9.3 从个体到全局——回声模型	180
9.4 复杂适应系统建模与仿真工具 NetLogo 简介	183
9.5 NetLogo 建模仿真框架	186
9.6 NetLogo 建模技术	188
9.7 NetLogo 建模仿真操作学习	189
9.8 实例分析:网络危机信息传播的演化仿真	212
参考文献	234

第1章 导 论

1.1 系统及其分类

1.1.1 系统的定义和特性

系统是由多个相互依赖、相互作用的要素按照某种规律组合起来、实现特定功能的有机整体,一个系统又可以由若干个子系统构成。系统有如下特征。

1. 整体性

整体性是系统最基本与本质的特征。系统由两个或两个以上要素(或子系统)构成,各个组成部分具有一定的独立性,又相互构成一个有机整体,系统的整体性主要表现在系统的整体功能。例如,计算机系统一般由中央处理器(CPU)、存储器、输入与输出设备等硬件构成,同时还包括操作系统、应用程序、数据库等软件,硬件与软件共同构成一个整体,从而发挥出了计算机系统的功能。

2. 相关性

各要素组成了系统,是因为它们之间存在相互联系、相互作用、相互影响的关系。这个关系也不是简单的累加,而是有可能地互相增强或互相减弱。一个要素的变化会这样或那样地引起全系统的变化。例如,构成人体的呼吸、消化、循环、排泄、神经等各组成部分通过特定的相互依存、相互制约关系有机地结合在一起,才使得人成为一种具有特殊高级功能和高度智慧的高等动物。如果某一器官出现了问题,就会影响其他部位器官的正常运行,从而导致人产生不舒服的感觉。

3. 目的性

系统具有能使各个要素结合在一起的共同目的(这类系统又称为组织界系统),各子系统就是为了完成大系统的既定目标而协同工作的。系统的目的一般通过更具体的目标来体现,复杂的系统通常不止一个目的,而是多重目的。

4. 阶层性

系统作为一个相互作用的诸要素的总体可以分解为一系列的子系统,并存在一定的层次结构。系统的阶层性表述了系统中不同层次子系统之间的从属关系或相互作用关系。在不同的层次结构中存在着动态的信息流和物质流,构成了系统的运动特性,为深入研究系统层次之间的控制与协调功能提供了条件。

5. 环境适应性

任何系统都有一定的边界和环境,它与周围的外部环境产生一定的联系和相互作用,从环境中接受各种影响(包括正常的输入和随机干扰),经过系统的转换,产生一定的输出,从而对外部环境产生一定的作用。外部环境及其影响是经常变化的,为了使系统达到最优化,必须对系统进行调节,使之适应环境的变化。

在定义一个系统时,首先要确定系统的边界。尽管世界上的事务是相互联系的,但当

我们研究某一对象时,总是要将该对象与其环境区别开来。边界确定了系统的范围,边界以外对系统的作用称为系统的输入,系统对边界以外的环境的作用称为系统的输出。

世界上的系统千差万别,但描述系统离不开“三个要素”,即实体、属性和活动。实体确定了该系统的构成,明确了描述一个系统的实体也就确定了系统的边界;属性也称为描述变量,描述每一实体的特征;活动定义了系统内部实体之间的相互作用,从而确定了系统内部发生变化的过程。

1.1.2 系统分类

系统的分类方法很多,按照不同的分类方法可以得到各种类型的系统。我们仅从建模与仿真研究的需要出发,对系统进行以下的分类。

1. 确定性系统和随机系统

按照系统输入与输出之间的关系,可以将系统分为确定性系统和随机系统。

确定性系统是指输出完全由系统的输入及相应的转换关系(包括决策、措施等)所决定的系统。这里,系统的输入、转换关系和输出都是确定的,只要知道输入,就可预先确定系统的输出。

随机系统在既定的输入下,系统的输出是非确定的,带有随机的性质。产生随机性的原因是由于在系统的输入过程和转换过程中存在多种难以预知的偶然因素的作用。然而,尽管随机系统的输出不能完全预知,但它们通常遵循一定的统计分布规律。确定系统输出(或输入)的统计分布以及对系统的输出进行估计,是系统仿真的主要任务之一。

大多数管理系统都属于随机系统。对于这类系统,当其复杂性超过一定限度时,运用数学分析方法建立系统模型并求解往往是很困难的,甚至是不可能的。这种情况下,仿真方法就显示出其优越性。

2. 连续系统和离散系统

在系统仿真中最重要的一种分类是按照系统中起主导作用的状态变量的变化是否连续,将系统分为连续系统和离散系统。我们把与特定时间和研究目的有关的描述系统所需变量的集合定义为系统的状态。例如,排队系统的研究中,状态变量可能是正忙着的服务员数、在系统中的顾客数以及每个顾客到达系统的时间。

连续系统是指系统的状态变量随时间变化而产生连续变化。这类系统的动态特性可以用微分方程或者一组状态方程来描述,也可以用一组差分方程或一组离散状态方程来描述。究竟采取哪一种描述方法,取决于研究者是对系统状态随时间连续变化的整个过程感兴趣,还是仅仅对某些时间点感兴趣,或者是所能得到的数据资料仅仅限于某些时间点。例如,在一些社会经济系统中,往往所能得到的数据也只有按月、季度,甚至是按年的。尽管这类系统实际的状态变化是连续的,但是也只能用差分方程和离散状态方程来描述。

离散系统是指系统的状态变量在某些离散的时间点瞬时变化。按照系统仿真的术语,称状态的瞬间变化为事件,将发生事件的时刻称为事件时间。如果事件时间是一些非均匀离散时点,这样的事件称为离散事件,相应的系统称为离散事件系统。

3. 线性系统和非线性系统

根据系统要素之间相互作用的性质,可以将系统分为线性系统和非线性系统。由于要素之间的关系最终会影响到系统输入与输出之间的依赖关系,所以只要系统中含有非线性环节就是非线性系统。



线性系统是状态变量和输出变量对于所有可能的输入变量和初始状态都满足叠加原理的系统。线性系统的状态变量(或输出变量)与输入变量间的因果关系可用一组线性微分方程或差分方程来描述,这种方程称为系统的数学模型。

一个系统,如果其输出不与其输入成正比,则它是非线性的。从数学上看,非线性系统的特征使叠加原理不再成立。叠加原理是指描述系统的方程的多个解之和仍为其解。叠加原理可以通过两种方式失效。其一,方程本身是非线性的;其二,方程本身虽然是线性的,但边界是未知的或运动的。

实际的系统大多是非线性系统,而将其看作线性系统不过是一种抽象和简化。线性系统模型比较简单,可以应用业已成熟的线性系统分析进行研究。由于线性模型在建模、分析和求解等方面的优点,我们在构造系统模型时,总是尽可能地使模型线性化,即使对于一些无法将其完全线性化的非线性关系,我们也设法使之分段线性化。但这种线性化的简化不应改变系统原有的主要特征,由此引入的误差不应超过允许的限度。

1.2 系统建模

1.2.1 模型及其基本性质

在系统仿真中,被研究的实际系统或未来的想定系统称为原型,而原型的等效替身则称为模型。模型通常借助文字、符号、图表、实物或数学表达式等提供关于系统要素、要素间关系以及特征或变化规律的信息和知识。

有效的模型必须能够反映原型的主要表征、特性及功能,并具备如下性质。

1. 普遍性

指模型能够描述多个相似系统。

2. 相对精确性

指模型的近似度和精度都不可超出应有限度和许可条件。

过于粗糙的模型将失去原型的过多信息和特征;太精确的模型则易造成模型过于复杂而导致研究困难,甚至终不得其解。所以模型应具有考虑诸多条件折中下的精确性。

3. 可信性

指模型必须经过校核、验证和确认,使其具有满意的可信度。

4. 异构性

指对于同一个系统,模型可以具有不同形式和结构,即模型不是唯一的。

5. 通过性

指模型可视为“黑箱”,能够利用输入/输出实验数据辨识出其结构和参数。

1.2.2 模型的分类

模型通常分为物理模型、概念模型和数学模型。

1. 物理模型

定义:以实物或图形直观地表达对象特征所得的模型。

物理模型是根据一定的规则对系统进行简化、描述或按照一定比例缩小、放大而得到

的仿制品。通常要求物理模型与实物高度相似,能够逼真地描述实物原型。

如风洞实验所用的飞行器外形和船体外形、飞机模拟驾驶系统、人工模拟太空环境以及人工制作或绘制的 DNA 分子双螺旋结构模型等都是用来描述实物原型的物理模型。在大型水利工程、土木工程项目设计、施工或飞行器、轮船研制过程中,常常要运用物理模型。

2. 概念模型

定义:对现实世界及其活动进行概念抽象与描述的结果称为概念模型。

概念模型是基于人们的经验、知识背景和思维直觉形成的,是人的大脑活动的产物。概念模型基于对所研究系统相关概念的抽象并通过对抽象概念相互关系的概括和描述得到,通常用语言、符号、框图等形式表达。

概念模型可看成是现实世界到数学模型或计算机仿真系统的一个中间层次。

3. 数学模型

定义:用来描述系统要素之间以及系统与环境之间关系的数学表达式。

各种不同的数学表达式均可以作为数学模型的基本形式。数学模型主要用来描述系统运行行为特征和基本规律,是研究系统、认识系统的重要工具和手段。

数学建模的目标是要确定系统的模型形式、结构和参数,获得正确反映系统表征、特征和功能的最简数学表达式。

数学建模的过程可遵循如下步骤:

观察和分析实际系统→提出问题→做出假设→系统描述→构筑形式化模型→模型求解→模型有效性分析→修改模型→最终确认→模型使用。

1.2.3 建模方法与步骤

1. 明确任务,建立概念模型

2. 模型设计

- ①确定模型变量;
- ②设定模型的数学形式;
- ③分析模型参数的符号和大致的变化范围。

3. 确定统计指标,收集、整理数据

对收集到的原始数据进行适当加工、整理才能用于建立模型。

数据加工整理包括甄别分类、汇总、归并、拆分、补缺、调整统计口径等,保证数据完整、准确、可靠,且满足可比性和一致性的要求。

常用的统计数据主要有以下三种分类方式。

(1)按计量尺度分类

①分类数据

分类数据是对事物进行分类的结果,数据表现为类别,是用文字来表示的。如在描述人口性别中:可用“1”表示男性,“0”表示女性。

②顺序数据

顺序数据也是对事物进行分类的结果,但这些类别是有顺序的。如产品等级可划分为一等品、二等品、三等品、次品等。

③数值型数据

数值型数据是使用自然或度量单位对事物进行计量的结果,其结果表现为具体的



数值。

(2) 按收集方法分类

① 观测数据

观测数据是通过调查或观测而收集到的数据,这类数据是在没有对事物人为的控制的前提下而得到的,有关社会经济现象的统计数据几乎都是观测数据。

② 实验数据

实验数据是在实验中控制实验对象而收集到的数据,如新药疗效实验。

(3) 按时间状况分类

① 时间序列数据

时间序列数据是按照时间先后顺序排列的数据。时间频率一般取年、季度、月、日、时、分、秒等,同一模型中各个变量的事件频率应保持一致。

② 截面数据

截面数据是指不同观测对象在某一时间的观测数据,如人口普查数据、审计调查数据等。

③ 面板数据(Panel Data)

面板数据是截面数据与时间序列数据的综合。其有时间序列和截面两个维度,当这类数据按两个维度排列时,是排在一个平面上,与只有一个维度的数据排在一条线上有着明显的不同,整个表格像是一个面板。

4. 估计模型参数

在数据收集、整理工作完成之后,即可将数据用于估计模型参数。一般可根据经济变量的性质、不同估计方法的特性、方法本身的难易程度和所需费用等因素选择适当的估计方法对模型参数进行估计。

5. 模型检验

模型检验是对已得到的参数估计值进行评价,研究其在理论上是否有意义,统计上是否显著,进而研究模型是否正确反映系统诸要素之间的关系。只有通过检验的模型才能应用于实际系统分析。常用的检验方法有以下几种。

(1) 经济意义检验

经济意义检验主要是检验参数估计值的符号及大小在经济意义上是否合理。例如,在需求函数中,需求量一般与收入正相关、与价格负相关,因此收入变量的参数估计值应为正值,价格变量的参数估计值应为负值;在消费函数中,消费一般随着收入的提高而上升,且消费的增幅通常会低于收入的增幅,因此收入变量的参数估计值应在0~1之间。如果参数估计值与经济理论不相符,则应分析原因并采取相应的办法加以解决。

(2) 统计学检验

统计学检验主要是根据数理统计学中的统计推断准则,对模型的可靠性进行检验。常用的统计学检验方法有拟合优度检验、t检验、F检验,分别用来检验模型和解释变量估计值的显著性。必须指出,统计学检验是为模型的经济意义服务的,对于一个违背经济理论的模型,不必进行统计学检验。

(3) 计量经济学检验

计量经济学检验主要是以数理统计为基础发展起来的一类检验方法,用于检验模型的计量经济学性质。例如,回归方程的假设条件检验,模型的识别性检验等。常用的计量经

济学方法主要有关于随机扰动项序列相关性的检验、异方差检验和解释变量多重共线性检验等。

(4) 预测性能检验

预测性能检验侧重于检验模型的稳定性,以及模型对样本期之外客观事实的描述和推断能力,即模型的超样本特性。具体检验方法如下:

①增加样本容量或更换新的样本数据重新估计模型参数,将新的估计值与原估计值比较,并检验二者间是否有显著差异。

②利用所建立的模型对样本期之外某一时期进行预测,通过比较预测值与实际值的误差,检验模型的预测能力。

6. 模型应用

通过检验的模型可视为实际系统的缩影,因此可据此对实际系统进行研究。

1.3 系统仿真

1.3.1 系统仿真的定义

“仿真”一词译自英文 Simulation,也称为“模拟”。1961年 G. W. Morgenthler 首次对“仿真”一词做了技术性的解释,他认为“仿真”意指在实际系统尚不存在的情况下,对于系统或活动本质的实现。

1978年,科恩(Korn)在《连续系统仿真》一书中给出技术性定义:“用能代表所研究的系统的模型做实验”。1982年,斯普瑞特(Spriet)进一步将仿真的内涵加以扩充:“所有支持模型建立与模型分析的活动即为仿真活动”。1984年,奥伦(Oren)给出了仿真的基本概念框架:“建模—实验—分析”,如图1-1所示。

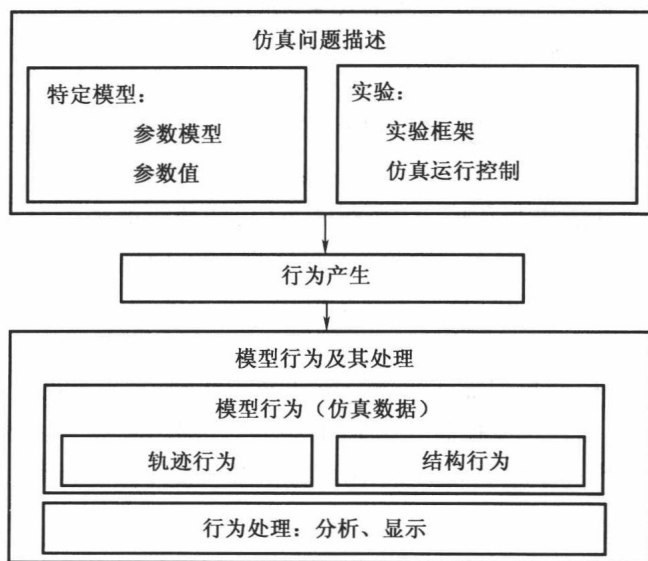


图 1-1 仿真的概念框架



1.3.2 系统、模型、仿真三者之间的关系

“系统、模型、仿真”三者之间联系密切。系统是研究的对象;模型是系统的抽象,是实际系统的替代物和模仿品;仿真则是基于系统建模和实验分析研究系统的过程。

这里我们将系统仿真定义为:通过对替代物或模仿品的实验分析对与之相似的原型系统进行研究的过程。

系统仿真技术随着计算机技术的发展而迅速发展。例如,将按一定比例缩小的飞行器模型置于风洞中吹风,测出飞行器的升力、阻力、力矩等特性;要建设一个大水电站,先建一个规模缩小的小水电站来取得建设水电站的经验及其运行规律;指挥员利用沙盘来指挥一个战役或一个战斗。

系统仿真是分析和研究各种(复杂)系统的重要工具。

计算机仿真技术的三要素及基本活动如图 1-2 所示。

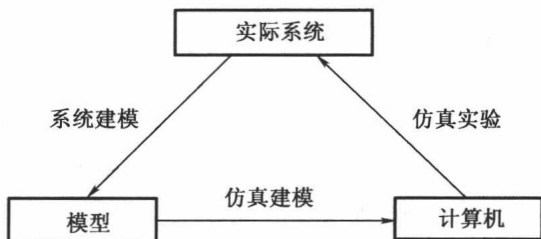


图 1-2 计算机仿真的三要素及三个基本活动

“系统建模”:系统辨识技术范畴。

“仿真建模”:即针对不同形式的系统模型研究其求解算法。

“仿真实验”:包括检验和致效。

检验(Verification)——“仿真程序”的检验;

致效(Validation)——将仿真结果与实际系统的行为进行比较。

现代仿真技术将仿真活动扩展到上述三个方面,并将其统一到同一环境中。

1. 系统建模

系统建模是通过对实际系统的观测或检测,在忽略次要因素及不可检测变量的基础上,用物理或数学逻辑的方法进行描述,从而获得实际系统的简化近似模型。

2. 仿真建模

仿真建模反映了系统模型(简化模型)同仿真器或计算机之间的关系,它应能为计算机所接受,并能进行运行。因此仿真建模就是对系统的数学模型进行一定的算法处理,使其在变成合适的形式之后能在计算机上进行数字仿真的“可计算/可执行模型”。

3. 仿真实验

仿真实验是指在计算机上通过对模型的不断运行、分析其结果,进一步认识系统,得到问题解答的过程。

1.4 系统建模与仿真技术的应用

系统建模与仿真技术的突出优点在于,能够在一个系统实际运行之前对它的运行性能进行模拟,并在不干扰实际系统的情况下比较各种方案的优劣,为决策提供科学依据。应用系统建模与仿真技术,人们可以用较小的投资大幅度降低决策风险。

应用上的可靠性和经济性以及计算机技术的突飞猛进,推动了系统建模与仿真技术的广泛和大规模应用。尤其是对投资大、风险大的大型项目,如航空、航天、战略武器系统、计算机集成制造、并行工程等领域,其应用效益非常显著。

1.4.1 航空航天领域

航空航天领域产品十分复杂且造价昂贵,具有投资大、风险大等大型项目的典型特征。人们正是在经历沉痛的失败教训之后才开始认识到系统仿真实验体系的重要性。例如,美国航空宇航局 1958 年进行了四次人造卫星发射,全部以失败而告终,1959 年发射成功率也仅为 57%。通过不断总结经验教训,为有效化解风险,提高项目成功概率,美国航空宇航局逐步建立了一套发射系统仿真实验体系,使发射成功率迅速提高。20 世纪 60 年代发射成功率已达 79%,到 20 世纪 70 年代达到 91%。随着空间发射系统仿真实验体系的逐步完善,美国的空间发射计划失败的情形近年来已经很少发生了。据测算,飞机研制项目采用系统仿真技术,通常可使研制投入减少 20% 左右,同时研制周期缩短 10% 以上。

1.4.2 电力工业

随着单元发电机组容量的不断增加,电力系统成为一个高度复杂的系统,人们对电力系统运行安全性、稳定性的要求也越来越高。因此,电站仿真系统已成为电站建设和运行管理中必不可少的配套装备,同时,系统仿真技术也是优化电力系统负荷配置、实现瞬态稳定性控制、确保系统安全运行的重要手段。

1.4.3 石油、化工、冶金工业

由于石油、化工、冶金工业生产过程十分复杂,产品设计、工业流程规划、生产计划安排、生产过程控制、产品性能分析和质量检验、原材料、产品的物流和库存管理等,涉及的因素众多,系统仿真技术在提高石油、化工、冶金工业生产和管理效率方面发挥了重要作用。

1.4.4 政策仿真

政策仿真是信息时代多学科交叉的产物,已被广泛应用于国家发展政策、地方社会经济政策和企业发展政策制定等方面。人们通过建模、仿真辅助决策部门完成政策方案的制定和政策效果的评估,起到对历史发展政策进行反思、对现实发展过程进行监测预警、对未来发展前景进行预测分析的功效。

美国、澳大利亚、加拿大、日本、德国等国家都曾投入大量资金建立了具有一定规模的仿真系统,对进出口、投资、消费、汇率、利率、能源、环境、就业、住房、技术变化、财政、税收、人口政策、社会福利、农业政策等进行仿真分析,为政府决策提供支持。近年来,我国学者



和政府决策咨询机构在政策仿真系统研究开发领域也开展了富有成效的工作。

1.4.5 经济安全分析

2009年初,美国国防部在马里兰米德堡军事基地的战争分析实验室内组织进行了一场模拟经济大战。巨幅电视墙上,经济形势瞬息万变:“朝鲜瓦解、俄罗斯垄断天然气价格、台湾海峡形势危急……”V型台前,代表美国、俄罗斯、中国、东亚和世界其他国家的5队人马,为争夺世界经济主导权展开经济博弈。

这场模拟的经济大战是美国五角大楼首次举行的以经济安全为主题的军事演习。美国《政治周报》网站2009年4月9日披露了演习的部分内容。据报道,演习计划早在2008年9月全球金融危机爆发前就已制订,而金融危机的蔓延更凸显了这次演习的重要性。参加演练的包括对冲基金经理、大学专家、投资银行高管等,他们被分成5支队伍,在世界面临经济灾难时模拟各方表现。另外,还有一组人担任裁判,评估各方行为的后果。他们的一举一动都被美军和情报人员记录下来并进行详细分析。《政治周报》评论说,对经济战的模拟绝非科幻,它反映出“9·11”事件之后,美军为捍卫美国的世界地位考虑的范围越来越广泛,五角大楼开始运用军事思维研究大国间的经济冲突。五角大楼模拟经济战反映了美国国内一部分人对美国国家经济安全的担忧。

第2章 建模的基本理论

2.1 建模的原则与模型的有效性

2.1.1 建模的原则

建立能够较全面、集中、精确地反映系统的状态、本质特征和变化规律的模型是系统建模的关键。事实上,能够直接用数学公式描述的事物是很有限的。因此,在大多数情况下数学模型不可能与实际现象完全吻合。为保证模型尽可能逼近实际系统,建立模型应遵循以下原则。

1. 简单性

从实用的观点看,由于在建模过程中忽略了一些次要因素和某些非可测变量的影响,因此,实际的模型已是一个简化了的近似模型。一般而言,在实用的前提下,模型越简单越好。

2. 清晰性

一个复杂的系统是由许多子系统组成的,因此对应的系统模型也是由许多子模型构成的。建立模型时,在子模型之间除为了研究目的所必需的信息联系外,互相耦合要尽可能少,结构尽可能清晰。

3. 相关性

模型中应该只包括系统中与研究目的有关的那些信息。虽然与研究目的无关的信息包括在系统模型中可能不会有很大的危害,但是它会增加模型的复杂性,从而使得在求解模型时增加额外的工作,所以应该把与研究目的无关的信息尽量排除在外。

4. 准确性

建立模型时,应该考虑所收集的、用以建立模型的信息的准确性,包括确认所对应的原理和理论的正确性及其使用范围,同时检验建模过程中针对系统所作的假设的正确性。

5. 可辨识性

模型结构必须具有可辨识的形式。所谓可辨识性是指系统的模型必须有确定的描述或表示方式,而在这种描述方式下与系统性质有关的参数必须具有可识别的解。若一个模型结构中具有无法估计的参数,则此模型就无实用价值。

6. 集合性

建立模型还需要进一步考虑的一个因素,是模型的集合性,亦即是否能够把若干个实体组成更大的实体。例如,对防空导弹系统的研究,除了能够研究每枚导弹的发射细节和飞行规律之外还可以综合计算多枚导弹发射时的作战效能。



2.1.2 模型的有效性

2.1.2.1 模型的可信度

模型的可信度本身是一个非常复杂的问题,它一方面取决于模型的种类,另一方面又取决于模型的构造过程。根据构造模型的难易程度,通常可以把模型的可信度水平分为三种:

①在行为水平上的可信度,即模型是否能复现真实系统的行为。

②在状态结构水平上的可信度,即模型能否与真实系统在状态上互相对应,通过这样的模型对未来的行为进行有效的预测。

③在分解结构水平上的可信度,即模型能否表达真实系统内部的工作情况。

有时这些可信度水平又分别称为重复性、重复程度和重构性。查看这些情况的一条可行途径是将每个水平视为一种对真实系统知识的索取。随着认识水平的提高,这种索取变得更加强烈。

不论在哪一种可信性水平上,都应当充分考虑在整个建模过程中及以后各阶段的可信性。

1. 在演绎中的可信性

演绎分析要求逻辑关系正确、数学过程严谨。在这种条件下,数学表示的可信性将取决于先验知识的可信性。而先验知识的可信性往往寓于正确性和普遍性之中,不易被认可和接受。数学模型的可信性还可以从以下两个途径进行分析:

①通过对前提条件正确性的研究来分析模型本身是否可信。

②通过对其他结果的验证来分析信息以及由此得到的模型的可信性。

2. 在归纳中的可信性

首先,可以检查归纳程序是否符合逻辑关系正确、数学过程严谨的要求,然后通过对比模型行为与真实系统行为判断模型的可信性。

在检验过程中,可将真实系统视为数据源,通过观测输入输出,获得系统行为数据。真实系统的输入输出关系通常用 R_1 表示。由于有效的实验数据是有限的,即在某一时刻 t ,能够观测、记录的数据仅仅是全部潜在的可获得数据的一部分,记作 R_2 。

模型本身也是数据 R_1 的来源。在某一时刻 t ,模型可信就意味着 $R_1 = R_2$ 。除此之外,还可以通过分析模型数据与真实系统数据的偏离程度来判断可信性。

对于具有某种统计特性的数据,或运用随机过程表示的模型,往往基于模型数据与真实系统数据的偏离程度判定其可信性。人们习惯于运用统计检验方法判断实际系统与模型之间的偏离程度。

3. 在目的方面的可信性

从实践的观点出发,假如运用一个模型能达到预期的目的,那么这个模型就是成功的、可信的。

2.1.2.2 模型的验证

仿真结果的有效性取决于系统模型的可靠性。因此,模型验证是一个不可或缺环节,甚至应贯穿于“系统建模—仿真实验”这一过程的始终,直到仿真实验取得满意的结果。

1. 模型验证过程中应注意的问题

进行模型验证应注意以下几个问题。

(1) 模型验证是一个重要环节

模型验证也是建模者对所研究问题由感性认识上升到理性认识的一个过程,往往需要多次反复才能完成。

(2) 模型验证过程具有不确定性

由于系统模型是实际系统的一种相似或近似,其相似或近似程度具有一定的不确定性。模型验证与建模者对实际系统的认识与理解程度有关。因此,建模过程不完全具备可重复性,“对于同一个问题,不同的建模者所建的模型可能有所不同”。

(3) 对模型进行全面验证往往是不可能的

尤其是对于一些复杂的系统模型与仿真问题(如社会系统、生态问题、飞行器系统等),模型验证所需要的大量统计数据难以获得,无法对模型进行全面验证。

2. 模型验证的基本方法

(1) 基于机理建模的必要条件法

对于采用机理建模法建立的模型,在模型验证过程中主要是检验模型的可信性。所谓必要条件法,就是通过对实际系统的各种特性、规律和现象进行仿真实验,然后根据仿真结果与必要条件的吻合程度来验证系统模型的可信性和有效性。通常,模型验证需要进行实验设计,其实验结果是人们可以判定的,正确的结果是正确的模型所应具备的必要性质。

(2) 基于实验建模的数理统计法

所谓数理统计法又称为最大概率估计法,它是数理统计学中描述一般随机状态(或过程)发生的可能性大小的一种数学方法。

由于实验建模中所依据的“数据”往往带有一定的随机性与不确定性,因此所得模型的可信性与准确性往往也是不确定的。所以,在实验由于建模时,应该选取那些概率最大的数据进行建模,以保证所建模型具有较高的可信性。

综上所述,对基于实验建模法建立的系统模型,可通过考察在相同输入条件下,系统模型与实际系统的输出结果在一致性、最大概率性、最小方差性等数理统计特性的对比来综合判断其可信性与准确性。

(3) 实物模型验证法

对于机电系统、化工过程系统以及工程力学等一类可依据相似原理建立实物模型的仿真问题,应用实物(或半实物)仿真技术能够在一定条件下实现较高精度的模型验证,但这种验证有时代价很高。这也是为什么在产品开发和飞行器研制过程中,人们总是把实物仿真作为产品定型和批量生产前的最高级仿真实验的原因。

2.2 演绎推理

2.2.1 演绎推理的定义

演绎推理是从一般性的前提出发,通过推导即“演绎”,得出具体陈述或个别结论的过程。演绎推理的定义包含如下内涵:



- ①演绎推理是从一般到特殊的推理;
- ②它是前提蕴涵结论的推理;
- ③它是前提和结论之间具有必然联系的推理;
- ④演绎推理就是前提与结论之间具有充分条件或充分必要条件联系的必然性推理。

演绎推理的逻辑形式对于理性的重要意义在于,它对人的思维保持严密性、一贯性有着不可替代的校正作用。这是因为演绎推理保证推理有效的根据并不在于它的内容,而在于它的形式。演绎推理的最典型、最重要的应用,通常存在于逻辑和数学证明中。

演绎推理是严格的逻辑推理,一般表现为大前提、小前提、结论的三段论模式,即从两个反映客观世界对象的联系和关系的判断中得出新的判断的推理形式。例如,自然界中的一切物质都是可分的,基本粒子是自然界的物质,因此基本粒子是可分的。

演绎推理的基本要求是:大、小前提的判断必须是真实的;推理过程必须符合正确的逻辑形式和规则。演绎推理的正确与否首先取决于大前提的正确与否,如果大前提错了,结论自然不会正确。

2.2.2 演绎推理的形式

演绎推理有三段论、假言推理、选言推理、关系推理等形式。

1. 三段论

三段论是由两个含有一个共同项的性质判断作前提,得出一个新的性质判断为结论的演绎推理。

三段论是演绎推理的一般模式,包含三个部分:

大前提——已知的一般原理;

小前提——所研究的特殊情况;

结论——根据一般原理,对特殊情况做出判断。

例如:知识分子都是应该受到尊重的,人民教师都是知识分子,所以,人民教师都是应该受到尊重的。

其中,结论中的主项叫作小项,用“ S ”表示,如上例中的“人民教师”;结论中的谓项叫作大项,用“ P ”表示,如上例中的“应该受到尊重”;两个前提中共有的项叫作中项,用“ M ”表示,如上例中的“知识分子”。在三段论中,含有大项的前提叫作大前提,如上例中的“知识分子都是应该受到尊重的”;含有小项的前提叫小前提,如上例中的“人民教师是知识分子”。三段论推理是根据两个前提所表明的中项 M 与大项 P 和小项 S 之间的关系,通过中项 M 的媒介作用,从而推导出确定小项 S 与大项 P 之间关系的结论。

2. 假言推理

假言推理是以假言判断为前提的推理。假言推理分为充分条件假言推理和必要条件假言推理两种。

(1) 充分条件假言推理的基本原则

小前提肯定大前提的前件,结论就肯定大前提的后件;小前提否定大前提的后件,结论就否定大前提的前件。如下面的两个例子:

①如果一个数的末位是0,那么这个数能被5整除;这个数的末位是0,所以这个数能被5整除。

②如果一个图形是正方形,那么它的四边相等;这个图形四边不相等,所以,它不是正

方形。

两个例子中的大前提都是一个假言判断,所以这种推理尽管与三段论有相似的地方,但它不是三段论。

(2) 必要条件假言推理的基本原则

小前提肯定大前提的后件,结论就要肯定大前提的前件;小前提否定大前提的前件,结论就要否定大前提的后件。如下面的两个例子:

①只有肥料足,菜才长得好;这块地的菜长得好,所以这块地肥料足。

②育种时,只有达到一定的温度,种子才能发芽;这次育种没有达到一定的温度,所以种子没有发芽。

3. 选言推理

选言推理是以选言判断为前提的推理。选言推理分为相容的选言推理和不相容的选言推理两种。

(1) 相容的选言推理的基本原则

大前提是一个相容的选言判断,小前提否定了其中一个(或一部分)选言支,结论就要肯定剩下的一个选言支。

例如:这个三段论的错误,或者是前提不正确,或者是推理不符合规则;这个三段论的前提是正确的,所以这个三段论的错误是推理不符合规则。

(2) 不相容的选言推理的基本原则

大前提是个不相容的选言判断,小前提肯定其中的一个选言支,结论则否定其他选言支;小前提否定除其中一个以外的选言支,结论则肯定剩下的那个选言支。例如下面的两个例子:

①一个词,要么是褒义的,要么是贬义的,要么是中性的。“结果”是个中性词,所以“结果”不是褒义词也不是贬义词。

②一个三角形,要么是锐角三角形,要么是钝角三角形,要么是直角三角形。这个三角形不是锐角三角形和直角三角形,所以它是个钝角三角形。

4. 关系推理

关系推理是前提中至少有一个是关系命题的推理。

下面简单举例说明几种常用的关系推理:

①对称性关系推理,如 $1\text{ m} = 100\text{ cm}$,所以 $100\text{ cm} = 1\text{ m}$;

②反对称性关系推理,如 a 大于 b ,所以 b 不大于 a ;

③传递性关系推理,如 $a > b, b > c$,所以 $a > c$ 。

2.3 系统辨识理论

2.3.1 系统辨识定义及基本要素

系统辨识最早的定义是 Zadeh 于 1962 年提出的。Zadeh 对辨识提出如下定义:“系统辨识是在对被辨识系统进行输入和输出观测的基础上,在设定的一类系统中,确定一个与被辨识系统等价的系统。”



1978年,瑞典学者 Ljung 进一步就系统辨识的定义作了补充:“系统辨识就是在一系列模型类中,按照某个准则,选择一个与被辨识系统的观测数据拟合得最好的模型。”

上述两个定义中,Zadeh 的定义较严格,但是要找出一个与实际系统完全等价的模型是比较困难的。而按 Ljung 的定义,辨识的实质可理解为数据拟合和优化,比较切合实用。系统辨识的基本原理如图 2-1 所示。

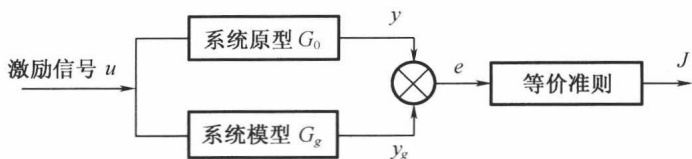


图 2-1 系统辨识基本原理示意图

图 2-1 中规定等价准则为 $J(y, y_g)$, 它是误差 e 的函数, 系统原型 G_0 和系统模型 G_g 在同一机理信号 u 作用下, 产生系统原型输出信号 y 和系统模型输出信号 y_g , 二者之间的误差为 e 。经等价准则计算后, 取修正模型参数, 然后再反复进行, 直到误差满足代价函数最小为止, 其数学表达式为

$$J(y, y_g) = f(e) \quad (2-1)$$

其中, $f(e)$ 为准则函数表达式。而辨识的目的则是找出一个模型 $G_g \in \phi$, ϕ 为给定的模型类, 使 $J(y, y_g) \rightarrow \min$, 则有 $G_g = G_0$, 此时即称为系统被辨识。

这个定义明确了系统辨识过程中的三大要素:

- ① 输入输出数据 (u, y, y_g) ;
- ② 模型类 (G_g) ;
- ③ 等价准则 $(J(y, y_g))$ 。

而模型的精度由 $J(y, y_g)$ 决定, 也即由 e 来决定。其中, 数据是辨识的基础; 准则是辨识的优化目标; 模型类是寻找模型的范围。从上述可知, 辨识的实质就是按照某种等价准则, 从一组模型类中选择一个模型, 使之能最好地拟合所研究的实际过程的动态特性。

2.3.2 系统辨识准则

在系统辨识过程中, 一个很重要的概念是, 要求被辨识对象和模型在各种可能的相同输入下, 它们的输出近似相等, 即两个系统“等价”, 这个概念是用来衡量模型接近实际过程的标准。而通常被辨识对象和模型的等价性是通过引入评价函数定义的, 这个评价函数称为等价准则函数。对某种相同的输入 u , 若实际系统 G_0 的输出为 y , 模型 G_g 的输出为 y_g , 而被辨识对象输出和模型输出之间的偏离(误差)为 $e = y - y_g$ 时, 通常采用的准则函数如下。

连续信号下, 其准则函数为

$$J(y, y_g) = \int_{t-T}^t [y(t) - y_g(t)]^2 dt = \int_{t-T}^t e^2(t) dt \quad (2-2)$$

离散信号下, 其准则函数为

$$J(y, y_g) = \sum_k^N |y - y_g|^2 = \|e\|^2 \quad (2-3)$$

在给定的模型类中, 当模型 G_g 使准则函数最小时, 定义 G_g 与对象等价。因此, 辨识就是

求使准则函数最小的模型 G_g 的一个最优化问题。当模型类采用参数模型描述时,辨识就归结为参数优化问题。

在给定的模型中,准则函数 $J(y, y_g)$ 是表述实际系统输出 y 和模型输出 y_g 之间关系的函数,常用的方式是表示成误差 e 的函数,写成 $J(y, y_g) = f(e)$ 。其具体表达方式中,平方误差准则 $f(e) = e^2$ 用得最多。而根据误差的定义形式,辨识方法又可以分为输入误差、输出误差和广义误差等形式。

2.3.3 系统辨识的模型类

一个动态系统按其描述方法和分析、定义域的不同,可用不同的数学模型来表达,具体可分为连续模型和离散模型、时间域模型和频率域模型、参数型模型和非参数型模型等。经典控制理论中的微分方程和现代控制理论中的状态空间方程都属于时域范畴,离散模型中的差分方程和离散状态空间方程也是如此。传递函数和传递函数矩阵属于频域范畴。一般在经典控制理论中采用频域传递函数建模,而在现代控制理论中则采用时域上的状态空间方程建模。

所谓参数型模型,是指用模型的参数来描述系统特性的模型,如微分方程和传递函数中的系数 a_i, b_i , 状态空间方程中的系数矩阵 A, B, C 和 D , 即为模型的参数。非参数型模型是指用响应曲线来描述系统特性的模型,如时域中的脉冲响应模型、频域中的频率响应模型。

2.3.4 系统辨识的步骤

实现系统辨识的过程可以分为以下几个步骤。

- ①选定一类代表被辨识系统的数学模型。
- ②在系统上加一个适当选择的实验信号,并记录其输入-输出数据。如果系统处在连续运行中,并且不允许加实验信号,则利用正常系统运行数据来辨识。
- ③完成参数辨识,在给定的类型中选定模型,使其最好地拟合统计数据。
- ④进行有效性检验,以考核所选模型对于最终的辨识对象来说是否适当地代表了该系统的特性。
- ⑤如果有效性检验通过,则辨识过程结束,否则必须选择另一类模型并且重复步骤②到④,直到获得有效的模型为止。

图 2-2 概略地展示了实际系统及其输入-输出数据的采集、转换和辨识,直至获得描述系统输入-输出之间关系的模型。

系统辨识的结果是得到一个表示输入和输出关系的数学表达式。为此,可利用各种输入信号去激励系统,并测量其输出。但是,这些输入和输出信号通常都为噪声所污染,辨识技术就是通过处理这样的带有噪声的数据,而获得模型的。

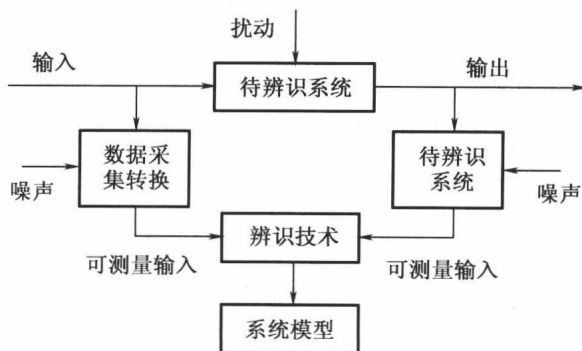


图 2-2 系统辨识示意图



2.4 相似理论

2.4.1 相似概念

自然界和人类社会中广泛存在着“相似概念”,最普通的是几何相似和现象相似。几何相似最简单而直观,如平面几何中多边形相似和三角形相似等。现象相似是几何相似的延伸和扩展,如相似的物理现象就表明了一个系统中所发生的过程的各物理量与另一个系统中所发生的过程的相应物理量之间存在着固定比例系数 m_i 。这表明现象相似中必然存在着类似几何相似的关系,即

$$V' = \frac{V}{m_v}, \quad t' = \frac{t}{m_t}, \quad u' = \frac{u}{m_u} \quad (2-4)$$

式中 m_v, m_t, m_u ——固定比例系数。

这是“相似”的一层含义,“相似”还具有另一层含义,就是人们对客观事物认识上的相似性。这是由于人的感受器官在接收信息及其变化时,经过转化单元及各级神经机构的整合,每一级所得到的中间结果都是相似的、有局限性的。同时,人在分析解决问题时还需经过多级相似,从而丢失许多信息和产生部分失真,因此最后得到的结论自然不可能与客观存在完全相同,而只能相似。可见,在任何情况下,相似是绝对的,全部映射或完全相同几乎是不可能的。

2.4.2 相似分类与方式

按照相似程度,相似可分为绝对相似、完全相似、不完全相似(或局部相似)、近似相似和模糊相似等。

所谓绝对相似是指所研究的两个系统(如原型与模型)全部几何尺寸和其他相应参数在时空域上产生的全部变化(或整个过程)都是相似的。

如果两个系统相应方面的过程在时空域上相似,则称为完全相似。例如,当我们研究发电机的电流和电压问题时,所建立的发电机数学模型应保证与实际发电机在电磁现象方面是完全相似的。这时就无需追究热工或机械过程方面的相似,即没有必要强求模型与原型的绝对相似。

不完全相似(或部分相似)是一种仅保证研究部分的系统相似。而非研究和不要部分的过程可能被歪曲,这是为研究目的所容许的。

在某些简化假设下的现象相似称为近似相似。对于建模来说,这种近似相似必须保证其模型的有效性。

模糊相似是指在两个系统之间保证它们的隶属函数相似或各自白化后的完全相似。

相似方式是保证上述系统相似的方方面面相似,采用何种相似方式取决于建模目标和仿真任务。通常,有如下相似方式供我们选取:几何相似、参量比例相似、结构相似、性能相似、感觉相似、生理相似、心里相似、思维方法相似、行为过程相似、群体行为相似和环境相似等。

2.4.3 相似关系

数学上的关系一般包括普通数相等关系、大小关系、图形全等关系,几何之间的包含关系、对应关系(即函数关系)以及相似关系等。

相似关系可作如下定义:

设 X, Y 是两个非空集,如果存在一个从 X 到 Y 的关系 S_s ,使得 X 中的每一个元素 x ,有 Y 中的唯一确定的元素 y 与之对应,即 $X \xrightarrow{S_s} Y$,这时则称关系 S_s 是从 X 到 Y 的相似关系。

同时,有 S_s 的定义域

$$D[S_s] = \{x; x \in X, \text{且对某一 } y \in Y, \text{有 } (x, y) \in S_s\}$$

及值域

$$R[S_s] = \{y; x \in X, (x, y) \in S_s, \text{对某一 } x \in S_s\}$$

2.4.4 相似定理

1. 相似第一定理

相似系统应具有同样的相似判据,即相似指标应当等于“1”。

该定理的数学形式为

$$\pi = idem$$

式中 π ——相似判据符号;

$idem$ ——表示所观测的一系列系统是相同的。

上述相似第一定理是在研究力学问题时由牛顿提出的。对于两个力学系统,如果用 F, M, l 和 t 分别表示力、质量、长度和时间, m_F, m_M, m_l 和 m_t 分别表示力、质量、长度和时间的相似常数,则有

$$\frac{F_2}{F_1} = m_F, \quad \frac{M_2}{M_1} = m_M, \quad \frac{l_2}{l_1} = m_l, \quad \frac{t_2}{t_1} = m_t \quad (2-5)$$

可得, $\frac{m_M m_l}{m_t^2 m_F} = 1$ 就是力学系统的相似指标。它进一步表明,相似系统的相似指标必然等于“1”,即任何相似系统必须满足相似第一定理。

2. 相似第二定理

相似第二定理又叫 π 定理。它表明,相似系数的相似判据为 π 。

假定任意物理系统是由 n 个不同量纲的物理量所组成的,物理过程的关系决定于方程 $F(j_1, j_2, \dots, j_n)$,若式中 n 个物理量有 k 个是相互独立的,任选出 $r_1, r_2, \dots, r_k$ 作为基本量,则其余 $n-k$ 个物理量与选定的基本量所组成的 $n-k$ 个无量纲的比例数 $\pi_1, \pi_2, \dots, \pi_{n-k}$ 可以用算式完全表达出来,而这些无量纲比例数 π 就是相似判据。

可见, π 定理的实质是如何寻求相似系统的相似判据。或者说, π 定理指出了如何利用量纲分析法找到一个物理现象(或系统)的相似判据个数,并确定出这些相似判据的表达式。

应强调指出:

①物理过程的量纲分析必须具备两个条件,即

- a. 给出参与物理过程的物理量;
- b. 已知各物理量的量纲式。

②物理量 Q 的量纲式一般形式为

$$\dim Q = L^\alpha M^\beta T^\gamma$$

式中 α, β, γ ——量纲指数;

L, M, T ——给定的三个基本量(l, m, t)的量纲。

3. 相似第三定理

若两个现象(系统)单值条件相似,并且从单值条件引出的相似判据相等,则这两个现象(系统)相似。

该定理是第一、第二定理的补充,它表明了判断现象(系统)相似的充要条件。

所谓单值条件是指一个现象(系统)从一群现象(系统)里区别出来的所需要的条件。通常,单值条件由现象(系统)的几何特性、物理参数、起始条件、边界条件等构成。从此角度讲,相似第三定理的实质是要求现象(系统)相似必须是单值条件相似,即几何相似、物理参数相似、起始条件相似、边界条件相似,当然还包括时间相似。

2.5 灰色系统理论

2.5.1 灰色系统基本原理

灰色系统理论是由我国著名学者邓聚龙创立的。1982年,他的论文《灰色控制系统》的发表,标志着灰色系统理论的问世。

灰色系统理论重点研究“小子样”“贫信息”不确定性系统或问题。邓聚龙教授通过对部分已知信息的生成、开发寻求实际系统运行行为和演化规律。其突出特点是“少数据建模”。

灰色系统理论经过三十多年的发展,不仅丰富了本身原理内涵,而且基本建立起了它的学科结构体系。

灰色系统基本原理可用如下诸公理来描述与概括:

【公理 2.1】“差异”是信息,凡信息必有差异——差异信息原理。

【公理 2.2】信息不完全、不确定的解是非唯一的——解的非唯一原理。

【公理 2.3】灰色系统理论的特点是充分开发利用已占有的“最少信息”——最少信息原理。

【公理 2.4】信息是认知的根据——认知根据原理。

【公理 2.5】新信息对认知的作用大于老信息——新信息优先原理。

【公理 2.6】信息不完全(灰)是绝对的——灰性不灭原理。

2.5.2 灰色系统理论体系结构

目前,灰色系统已形成如下较完善的学科体系结构。

1. 理论体系

理论体系以灰色朦胧集、灰色代数系统、灰色矩阵、灰色方程等为理论基础。

2. 分析体系

分析体系包括灰色关联分析、灰色聚类 and 灰色统计评估等。

3. 方法体系

统一在序列算子概念下的灰色序列生成,包括缓冲算子、均值生成算子、级比生成算子、累加生成算子和累减生成算子等。

4. 模型体系

以灰色模型(GM)为核心,模型按照五步建模思想来构建,即通过灰色生成或序列算子的作用弱化随机性,挖掘潜在的规律,经过灰色系统差分方程与灰色微分方程之间的互换实现利用离散的数据序列建立连续的动态微分方程。

5. 技术体系

技术体系以系统分析、评估、建模、预测、决策、控制、优化为主体。其中,灰色预测是基于模型做出的定量预测,并按照其功能分为数列预测、区间预测、突变预测、季节突变预测、波形预测及系统预测等模型。灰色决策包括灰靶决策、灰色关联决策、灰色统计决策、灰色局势决策和灰色层次决策等。灰色控制包括本特性灰色系统控制、灰色关联控制和预测控制等。灰色优化技术包括灰色线性规划、灰色非线性规划、灰色整数规划和灰色动态规划等。

2.5.3 灰数概念、运算及灰色关联度分析

1. 灰数概念

所谓灰数是指信息不完全的数,是灰色系统的标志之一,记为 $\otimes$ 。其相应的白化数为 $\tilde{\otimes}$ 。灰数被分为离散灰数和连续灰数。

如果 $\otimes$ 是离散灰数,则有

$$\forall \tilde{\otimes} \in \otimes \Rightarrow \tilde{\otimes} \in A = \{x(k) \mid k \in K = (1, 2, \dots, n)\}$$

如果灰数 $\otimes$ 中的白化数 $\tilde{\otimes}$ 是按区间连续分布的,则有

$$\forall \tilde{\otimes} \in \otimes \Rightarrow \tilde{\otimes} \in It(a, b) \in \{[a, b], (a, b), [(a, b), (a, b)]\}$$

2. 灰数运算与白化

关联度是事物(系统)之间的关联相似性的测度。所谓灰色关联度分析是指根据序列曲线几何形状的相似程度来判断其联系是否紧密,曲线越接近,则相应序列之间的关联度就越大;反之亦然。或者说,关联度分析法是一种按照待分析系统的各种特征参量序列所构成的曲线几何形状、发展变化趋势接近程度来度量待分析系统之间关联程度的方法。

在灰色系统建模中,主因素与各子因素之间的关联相似程度不同,因此需要用关联度分析来寻找这种相似程度,即关联度。关联度是根据原始数据的几何相似性通过数据处理方法求得的,而不是数据具有某种分布,这是与统计分析法的根本区别。

关联度分为绝对值关联度、速度关联度、斜率关联度等,通常采用绝对值关联度,绝对值关联度的原理如下。

设母序列(即主因素的量化值)为 $X_0^{(0)}$, $X_i^{(0)}$ 为第 i 个子序列(即子因素的量化值),那么

$$X_0^{(0)} = \{x_0^{(0)}(1), \dots, x_0^{(0)}(n)\}$$

$$X_i^{(0)} = \{x_i^{(0)}(1), \dots, x_i^{(0)}(n)\}$$

式中 上标(0)——未经累加生成的原始数据;

n ——采样点数。

则关联系数为

$$\xi_i(k) = \frac{\min \min |x_0^{(0)}(k) - x_i^{(0)}(k)| + \rho \max \max |x_0^{(0)}(k) - x_i^{(0)}(k)|}{|x_0^{(0)}(k) - x_i^{(0)}(k)| - \rho \max \max |x_0^{(0)}(k) - x_i^{(0)}(k)|} \quad (2-6)$$

式中, $\rho \in [0, 1]$ 是给定的辨识系数, 一般取 $\rho = 0.5$ 。要比较各子序列对母序列的关联度大小, 通常取共同参考点, 如坐标原点。故上式的分子第一项为零, 即

$$\xi_i(k) = \frac{\rho \max \max |x_0^{(0)}(k) - x_i^{(0)}(k)|}{|x_0^{(0)}(k) - x_i^{(0)}(k)| - \rho \max \max |x_0^{(0)}(k) - x_i^{(0)}(k)|} \quad (2-7)$$

称数 r_i 为对于 $x_0^{(0)}(k)$ 的关联度, $r_i = \frac{1}{n} \sum_{k=1}^n \xi_i(k)$ 。所有的 r_i 将构成一个关联度集 R , $R = \{r_1, r_2, \dots, r_n\}$ 。这时, 根据 R 的大小顺序即可选出重要或次要影响因素。在此基础上, 再根据关联度选择模型的说明标量。关联度分析的全过程如图 2-3 所示。



图 2-3 关联度分析的全过程流程图

2.6 自组织理论

2.6.1 系统自组织概念

自组织是客观世界普遍存在的状态现象。在系统实现空间的、时间的或功能的结构过程中, 如果没有外界的特定干扰, 仅是依靠系统内部的相互作用来达到, 则称系统是自组织的。

人们的自组织思想由来已久。从系统学角度讲, 所谓系统的自组织是指系统的各部分无需外界输入特殊的组织指令, 自行结合为一个有机整体。这种自行结合是一个不可逆的动态过程。

自组织理论不是一个独立的理论体系,而是一个学科群,它是由协同学(Synergetic)、超循环理论(Hypercycle Theory)、突变论(Morphogenesis)、混沌理论(Chaotic Theory)和分形理论(Fractal Theory)等若干关于系统演化理论共同组成的,以系统的发生、发展为重点,揭示了组成一个宏观系统的大量子系统,如何自行组织,实现从无序到有序(或者从低级有序到高级有序)演化的一般条件、机制和规律性,深化了人们对系统(包括自然系统、社会系统、经济系统等)认识的程度。

自组织理论体系如图2-4所示,耗散结构理论深刻地揭示了系统自组织演化的客观环境条件,协同学以系统要素的竞争与协同效应解释自组织形成的内在机制,超循环理论阐述了系统自组织演化的具体形式以及交叉作用于更大的循环链的发展过程,而突变论则着重剖析了系统自组织演化的局部渐变到整体突变的实现途径,混沌理论和分形理论则对系统自组织演化过程中的时间复杂性和空间结构与特性进行了解释和描述。以上这些理论形成了当今自然科学探索复杂系统自组织演化的理论前沿。

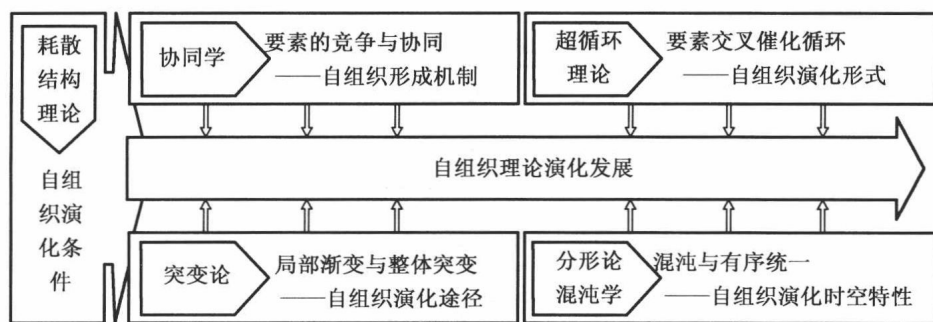


图2-4 自组织理论体系

2.6.2 耗散结构论

耗散结构是开放的复杂系统所具有的一种动态有序结构,是由比利时科学家普利高津(Prigogine)于1969年首先提出的。研究表明,一个远离平衡态的开放系统,通过不断与外界交换物质和能量,在外界环境对系统的影响程度达到一定阈值时,就会形成一种新的有序结构,且这种结构要依靠耗散外界的物质和能量来维持,故称之为耗散结构。耗散结构是一个动态的有序结构,可以从一种耗散结构向另一个耗散结构跃迁。

基于上述研究成果,耗散结构论指出,在开放条件下,非线性动力系统与外界的物质、能量和信息交换,若系统远离平衡并在大量微观粒子自组织协同一致时,将会在整个上涌现出新的有序结构——耗散结构。因此,耗散结构论实质上就是远离平衡的非线性动力学系统自组织理论。

耗散结构论认为,系统要形成耗散结构必须具备如下基本条件。

1. 系统属于开放系统

也就是说,耗散结构是在开放环境下生成的,若系统与外界环境隔绝,便不可能形成耗散结构,即使原系统存在耗散结构也终会瓦解。

2. 系统远离平衡态

由于耗散结构优势是在平衡条件下生成的,所以要求系统在外界环境影响下,能超过

非平衡的线性区,处于远离平衡态的非线性区域。

3. 系统组分间存在非线性反馈

它是指只有系统内部存在非线性反馈才能使系统稳定到耗散结构上。

4. 系统中存在涨落

所谓涨落就是对系统的稳定的平均状态的偏离,即系统的不稳定状态。涨落对系统演化起着触发作用,通过涨落的有序和放大可使系统实现从无序到有序的转变,从低级有序向高级有序进化。涨落并不是无条件地发展为序参量的,只有远离平衡且当控制参数达到分支点附近,才有这种可能。

耗散结构的形成就是系统的自组织过程,耗散结构理论第一次科学地揭示了自然界从简单到复杂、从无序到有序、从非生命到生命的演化,解决了使经典科学陷入困境的进化与退化、确定性与不确定性的宇宙学佯谬,阐明了系统自组织的内外部条件,揭示了系统“非平衡不可逆性是组织之源、有序之源”,“通过涨落达到有序”的演化过程。

2.6.3 协同学理论

协同学是德国物理学家哈肯于1969年提出的。它在自组织理论的动力学方法论中占有相当重要的地位,旨在研究自组织活动里的竞争、协同、支配及序参量等。

协同学的理论要点是,一个系统从无序到有序转变的关键不在于系统是平衡还是非平衡,也不在于离平衡态有多远,而是系统内部各子系统间通过非线性相互作用和协调,在一定条件下,能自发地产生在时间、空间或功能上稳定的有序结构,这就是系统的自组织能力。协同学还指出,系统在临界点(分支点)附近的行为仅由少数慢变量决定,系统的慢变量支配快变量,即所谓的支配原理。

协同学告诉人们:制定一定的规则,以一定的参数进行调节,然后放手让子系统自己相互作用,产生序参量运动模式,从而推动整个系统演化是非线性、自组织的最好管理方式。

2.6.4 超循环理论

德国物理学家曼弗雷德·艾根(Manfred Eigen)将物理学和化学的普遍性原理推广到生物学领域,研究生物大分子自组织过程中生物信息起源,建立了关于生命起源的超循环模型,开创了超循环理论,不仅在生命科学界,而且在系统科学界乃至哲学社会科学界都引起了广泛的重视。

与耗散结构相比,超循环理论从另一个角度阐述了系统的自组织实现过程。如图2-5

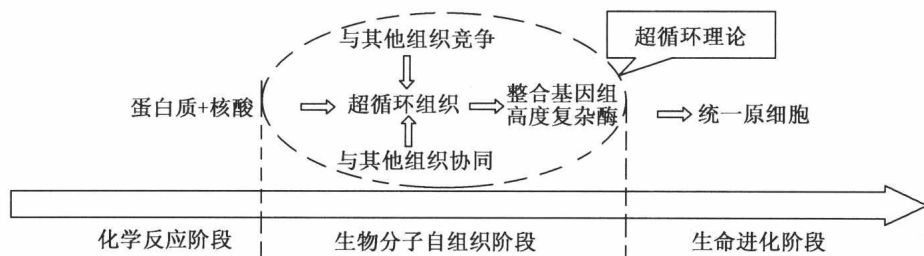


图2-5 超循环理论生物分子自组织阶段分形示意图

所示,超循环理论认为,生命的起源可以分为三个阶段;化学反应阶段、生物分子的自组织阶段和生命进化阶段。其中,生物分子的自组织阶段一直被生物进化理论学者所忽视,而这个阶段正是生命演化的内在本质所在,生物大分子的自组织形成了超循环组织结构,提供分子自组织所需要的全部物质,发生在超循环组织之间的竞争和协同效应进一步扩大了循环链圈,使得这种超循环行为逐渐扩展到整个细胞内部,并形成了具有整合基因组、高度复杂酶、统一细胞结构的原细胞,最终实现了生物的进化。由此可见,超循环是生物大分子自组织导致生命起源以及进化的不可避免之路。

生物系统是一个复杂的巨系统,将超循环理论抽象到对整个复杂系统的研究之中,为人类探索复杂系统的演化过程开辟了一种新的思路和途径。耗散结构强调的是系统对来自外部环境能量的耗散,而超循环理论则更侧重于系统内部发生的非平衡的、非线性的随机不可逆的因果多种循环关系,揭示了系统自组织演化的普遍形式。

2.6.5 突变论

突变论是专门研究自然界连续的量变怎样引起质变并用数学模型描述这种变化现象的学科,其范畴属于静态分支点的问题,即平衡点之间相互转换。

突变论认为系统内部任何变化都可以分为量变和质变,即渐变和突变。渐变可以引起突变,而突变是系统形态结构改变的根源;突变的产生是由于系统内部的控制因素的变化所引起的,在整个突变过程中,控制因素 λ 始终处于临界值,如图 2-6 所示。系统控制因素的阈值小于 λ_1 ,在 O 点到 λ_1 点区间,系统则沿着 $O \rightarrow b \rightarrow a$ 轨迹演化,当系统的外界输入和内部输入导致控制因素 λ 无限接近于 λ_1 点时,即系统演化到控制因素阈值的临界点 b 时,系统则改变演变途径,发生跳跃性的变化 $b \rightarrow b'$,从 b' 开始沿 $b' \rightarrow c \rightarrow d$ 轨迹演化,此时控制因素随之上升到新的临界值状态,即在 λ_1 点到 λ_2 点区间,并且小于 λ_2 。而当系统演化到临界点 c , λ 无限接近于 λ_2 时,系统再次发生跳跃现象。由此可见,系统的突变发生在控制因素无限接近阈值的临界状态。

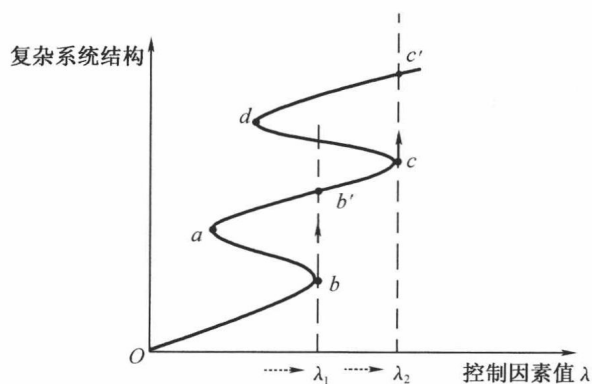


图 2-6 复杂系统突变过程

突变论超越了以往“自然界无飞跃”的渐进进化思想,揭示出了原因连续的作用有可能导致结果的突然变化,加深了我们对于系统的有序与无序转化的方式和途径的多样性的理解。

2.6.6 分形理论

分形理论是自欧几里得以来几何学最大的一次变革,分形理论的创始人曼德尔布洛特(B. B. Mandelbrot)创造了与以往传统数学和其他学科截然不同的方法,通过这些方法和简单规则,产生出复杂性的分形图样,从而使得以往被几何学排斥在外的大量不规则几何对象得以描绘,跨越一切尺度的“穿透力”,揭示出深藏在惊人复杂形状中从大尺度到小尺度的几何规律,而且建立了图形生长的模型,探索分支产生的整个结构。

分形理论认为:具有不规则的、破碎形状的、同时部分又与整体具有某种方式下的相似性,其维数不必是整数的复杂系统,可以通过对部分的自相似性描述,揭示复杂系统的内部结构。分形理论的研究揭示出系统部分和整体的相似性,试图找到介于有序-无序、宏观-微观、整体-部分之间的新秩序。

分形理论不仅突破了以往几何学只是研究同一尺度中空间关系及规则图形的局限,突破了以往全部线性科学的正整数维空间概念,而且突破了以往数学限于逻辑推理和抽象形式的传统,并和混沌学一起揭示了系统自组织演化的复杂性。

2.6.7 混沌学

混沌概念是李天岩和约克(Yorke)于1975年首先提出的。混沌现象是非线性所产生的复杂非本质随机性不规则行为,它普遍存在于人类社会和自然界中。

混沌学通过非线性不确定性方程所蕴涵的蝴蝶效应,揭示了分形结构的动力学根源。分形理论和混沌学同为非线性科学的不同分支,它们之间存在很大联系性,它们是复杂系统在时间和空间上的统一,混沌系统在空间上表现为分形结构,而分形是描述系统混沌结构的有力工具,复杂系统的混沌现象在不同的时间标度上表现出相似的变化模式,具有时间上的标度不变性。分形的自相似性是一种空间的标度不变性,所以可以说混沌在时间尺度上反映了系统的复杂性,而分形则在空间尺度上反映了系统的复杂性。

2.7 元胞自动机理论

2.7.1 元胞自动机的定义

元胞自动机具有较为宽松甚至模糊的构成条件,但作为一个物理模型,元胞自动机有着严格的科学定义。它是物理学家、数学家、计算机科学家和生物学家共同的工作结晶。物理学家将其视为离散的、无穷维的动力学系统;数学家将其视为描述连续现象的偏微分方程的对立体,是一个时空离散的数学模型;计算机科学家将其视为新兴的人工智能、人工生命的分支;而生物学家则将其视为生命现象的一种抽象。下面给出几种常见的定义。

1. 物理学定义

元胞自动机是定义在一个由具有离散、有限状态的元胞组成的元胞空间上,并按照一定局部规则,在离散的时间维上演化的动力学系统。不同于一般的动力学模型,元胞自动机不是由严格定义的物理方程或函数确定,而是由一系列的演化规则构成。大量元胞通过

简单的相互作用而构成动态系统的演化。

2. 数学定义

美国数学家 L. P. Hurd 和 K. Culik 等人在 20 世纪 90 年代初,对元胞自动机分别从集合论和拓扑学等角度进行了严格的描述和定义。

基于集合论的定义:设 d 代表空间维数, k 代表元胞状态,在有限集合 S 中取值, r 代表元胞的邻居半径, $\mathbf{Z}$ 是整数集, t 代表时间,将整数集 $\mathbf{Z}$ 上的状态集 S 上的分布记作 $S_{\mathbf{Z}}$ 。对于一维元胞自动机,元胞自动机的动态演化就是在时间上状态组合的变化,可记作 $S_{\mathbf{Z}}^t \rightarrow S_{\mathbf{Z}}^{t+1}$,这个动态演化又由各个元胞的局部函数 f 决定;局部函数 f 通常被称为局部规则。对于局部规则 f 来讲,函数的输入、输出集均为有限集合,实际上,它是一个有限的参照表。对元胞空间内的元胞,独立施加上述局部函数,则可得到全局的演化: $F(S_{\mathbf{Z}}^{t+1}) = f(S_{i-r}^t, \dots, S_i^t, \dots, S_{i+r}^t)$,其中 S_i^t 表示 t 时刻在位置 i 处的元胞。

基于拓扑学的定义:为描述和理解方便,同样假定维数 d 为 1。设 S 为 k 个符号的有限集, $\mathbf{Z}$ 为整数全体的集合,称 $\mathbf{Z}$ 到 S 的映射的全体 $S_{\mathbf{Z}}$ 为构形空间。显然 $S_{\mathbf{Z}}$ 就是用 S 中的符号组成的双侧无限的符号序列的全体,即一维元胞自动机的所有构形的集合。称 $a = (\dots, a_{-1}, a_0, a_1, \dots)$ 为构形空间中的点。在 $S_{\mathbf{Z}}$ 中引进任意两点 x 和 y 之间的距离: $\text{dist}(x, y) = \sum \delta(x_i, y_i) 2^{-|i|}$, 其中当 $x_i = y_i$ 时, $\delta(x_i, y_i) = 0$, 当 $x_i \neq y_i$ 时, $\delta(x_i, y_i) = 1$ 。然后,在 $S_{\mathbf{Z}}$ 中可以建立起开、闭、紧等拓扑概念。其中定义移位算子 δ 为 $\delta(x_i) = x_{i-1}, i \in \mathbf{Z}$ 。若连续映射 $F: S_{\mathbf{Z}}^t \rightarrow S_{\mathbf{Z}}^{t+1}$ 与 δ 可交换,即 $F\delta \rightarrow \delta F$, 或对任意的 $x \in S_{\mathbf{Z}}$ 有 $F[\delta(x)] \rightarrow \delta[F(x)]$, 则称 F 为元胞自动机。

2.7.2 元胞自动机的构成

元胞自动机最基本的组成包括四个部分:元胞 (Cell)、元胞空间 (Lattice)、邻居 (Neighbor) 及规则 (Rule)。简单地讲,元胞自动机可以视为由一个元胞空间和定义于该空间的变换函数所组成。元胞自动机组成之间的关系如图 2-7 所示。

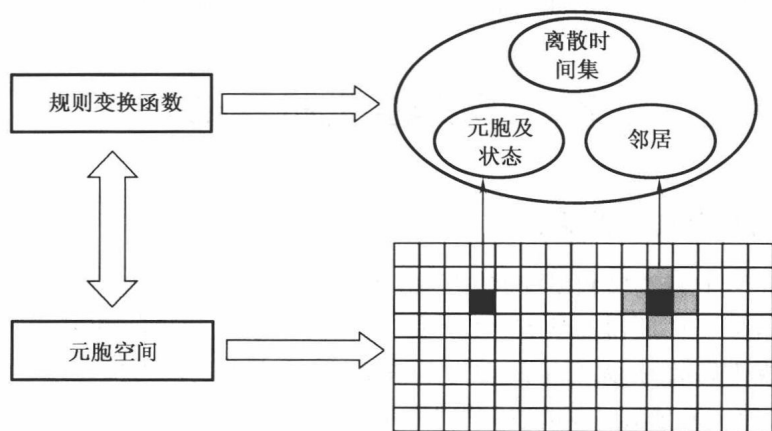


图 2-7 元胞自动机的构成示意图

1. 元胞 (Cell)

元胞又可称为单元或基元,是元胞自动机的最基本的组成部分。元胞分布在离散的一



维、二维或多维欧几里得空间的晶格点上。某一时刻每个元胞都具有自己的状态。状态可以是 $\{0,1\}$ 的二进制形式,或是 $\{S_0, S_1, \dots, S_i, \dots, S_k\}$ 整数形式的离散集。理论上元胞只能有一个状态变量,但在实际的应用中,常常要添加其他的状态量。

2. 元胞空间(Lattice)

元胞分布所在的空间网点集合就是元胞空间。它可以是任意维数的欧几里得空间的规则划分。但大多数的研究集中在一维和二维元胞自动机上,三维或者三维以上的元胞自动机研究相对较少。一维元胞自动机的元胞空间划分只有一种,如图2-8所示;而高维元胞自动机空间划分可以有多种形式。最为常见的是二维元胞自动机,二维元胞空间通常可按三角、四方或六边形三种网格排列,如图2-9所示。

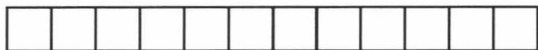
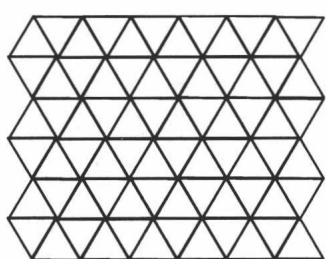
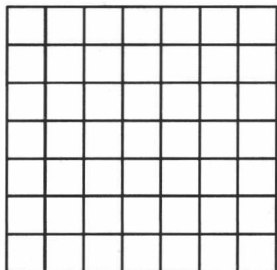


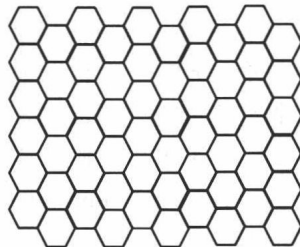
图2-8 一维元胞自动机的空间结构



(a)



(b)



(c)

图2-9 二维元胞空间的三种常见划分方式

(a)三角网格;(b)方网格;(c)六边形网格

这三种规则的元胞空间划分在构模时各有优缺点:

三角网格的优点是,拥有相对较少的邻居数目,这在某些时候很有用;其缺点是,对于计算机的表达与显示不方便,需要转换为四方网格。

四方网格的优点是,直观而简单,而且特别适合于在现有计算机环境下进行表达显示;其缺点是,不能较好地模拟各向同性的现象,例如格子气模型中的HPP模型。

六边形网格的优点是,能较好地模拟各向同性的现象,因此模型能更加自然而真实,如格子气模型中的FHP模型;其缺点是,同三角网格一样,在表达显示上较为困难、复杂。

3. 边界条件(Boundary)

在理论上,元胞空间通常在各维向上是无限延展的,这有利于在理论上的推理和研究。但是在实际应用过程中,我们无法在计算机上实现这一理想条件,因此我们需要定义不同的边界条件。归纳起来,边界条件主要有三种类型:周期型、反射型和定值型。有时,在应用中,为更加客观、自然地模拟实际现象,还有可能采用随机型,即在边界实时产生随机值。

周期型边界条件(Periodic Boundary)是指相对边界连接起来的元胞空间。对于一维空间,元胞空间表现为一个首尾相接的“圈”。对于二维空间,上下相接,左右相接,而形成一个拓扑圆环面(Torus),形似车胎或甜点圈。周期型空间与无限空间最为接近,因而在理论

探讨时,常以此类空间型作为实验。

反射型边界条件(Reflective Boundary)是指在边界外邻居的元胞状态是以边界为轴的镜面反射。例如,在一维空间中,当半径 $r=1$ 时的边界情形:



定值型边界条件(Constant Boundary)是指所有边界外元胞均取某一固定常量,如 0, 1 等。

需要指出的是,这三种边界类型在实际应用中,尤其是二维或更高维数的构模时,可以相互结合。如在二维空间中,上下边界采用反射型,左右边界可采用周期型(相对边界中,不能一方单方面采用周期型)。

4. 邻居(Neighbor)

元胞自动机演化是局部的,对指定元胞的状态进行更新时只需要知道其邻近元胞的状态。某一元胞状态更新时所搜索的空间域叫作该元胞的邻居。一般,元胞邻居的大小没有限制,但所有元胞的邻居大小都要相同。而实际上往往只有邻接的元胞构成邻居,如果邻居太多,演化规则会变得过于复杂。一维元胞自动机中,通常用半径 r 来确定邻居,距离一个元胞 r 内的所有元胞都认为是该元胞的邻居。二维元胞自动机的邻居定义比较复杂,通常有 Von Neumann 型、Moore 型、扩展的 Moore 型、Margolus 型等,如图 2-10 所示。图 2-10 中黑色元胞为中心元胞,灰色为其邻居,即它们的状态将决定下一时刻中心元胞的状态。Von Neumann 型中一个元胞的上、下、左、右相邻的四个元胞为该元胞的邻居,这里,邻居半径 r 为 1。Moore 型中一个元胞的邻居半径 r 同样为 1,邻居数量扩展为与之相连的八个元胞。扩展的 Moore 型是将 Moore 型的邻居半径 r 扩展为 2 或者更大而得到的邻居。Margolus 型图例类似于扩展的 Moore 型,但是其每次将一个 2×2 的元胞做统一处理,而之前三种邻居模型中,每个元胞是分别处理的。

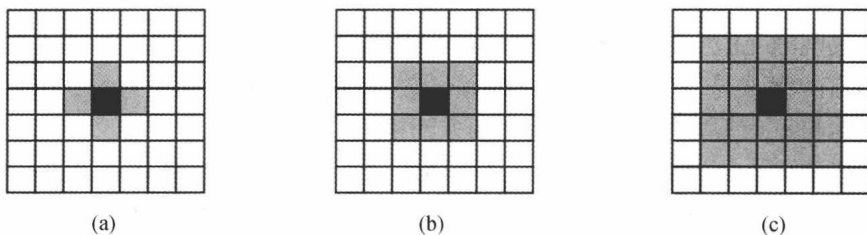


图 2-10 元胞自动机的邻居模型

(a) Von Neumann 型; (b) Moore 型; (c) 扩展的 Moore 型

5. 规则(Rule)

根据元胞当前状态及其邻居状况确定下一时刻该元胞状态的动力学函数,即状态转移函数。将一个元胞的所有可能状态连同负责该元胞的状态变换的规则一起称为一个变换函数。这个函数构造了一种简单的、离散的空间/时间的局部物理成分。要修改的范围里采用这个局部物理成分对其结构的“元胞”重复修改。这样,尽管物理结构的本身每次都不发展,但是状态在变化。它可以记为 $f: S_i^{t+1} = f(S_i^t, S_N^t)$, S_N^t 为 t 时刻的邻居状态组合, f 为元

胞自动机的局部映射或局部规则。

6. 时间(Time)

元胞自动机是一个动态系统,它在时间维上的变化是离散的,即时间 f 是一个整数值,而且连续等间距。假设时间间距 $dt=1$,若 $t=0$ 为初始时刻,那么 $t=1$ 为其下一时刻。在上述转换函数中,一个元胞在 $t+1$ 时刻只(直接)决定于 t 时刻的该元胞及其邻居元胞的状态,虽然在 $t-1$ 时刻的元胞及其邻居元胞的状态间接(时间上的滞后)影响了元胞在 $t+1$ 时刻的状态。

由以上对元胞自动机的组成分析,我们可以更加深入地理解元胞自动机的概念。用数学符号来表示标准的元胞自动机是一个四元组,即 $A=(L_d, S, N, f)$: A 代表一个元胞自动机系统; L 表示元胞空间; d 是一正整数,表示元胞自动机内元胞空间的维数; S 是元胞的有限的、离散的状态集合; N 表示一个所有邻域内元胞的组合(包括中心元胞),记为 $N=(s_1, s_2, \dots, s_n)$,即包含 n 个不同元胞状态的一个空间矢量。

2.7.3 元胞自动机的基本特征

从元胞自动机的构成及其规则上分析,标准的元胞自动机具有以下特征。

1. 同质性、齐性

同质性反映在元胞空间内的每个元胞的变化都服从相同的规律,即元胞自动机的规则,或称为转换函数;而齐性指的是元胞的分布方式相同,大小、形状相同,空间分布规则整齐。

2. 空间离散

元胞分布在按照一定规则划分的离散的元胞空间上。

3. 时间离散

系统的演化是按照等间隔时间分步进行的,时间变量 t 只能取等步长的时刻点,形似整数形式的 $t, t+1, t+2, \dots$ 。而且, t 时刻的状态构形只对其下一时刻,即 $t+1$ 时刻的状态构形产生影响,而 $t+2$ 时刻的状态构形完全决定于 $t+1$ 的状态构形及定义在上面的转换函数。元胞自动机的时间变量 t 区别于微分方程中的时间变量 t ,那里 t 通常是个连续值变量。

4. 状态离散有限

元胞自动机的状态只能取有限个(k 个)离散值($s_1, s_2, \dots, s_k$)。相对于连续状态的动力系统,它不需要经过粗粒化处理就能转化为符号序列。而在实际应用中,往往需要将有些连续变量进行离散化,如分类、分级,以便于建立元胞自动机模型。

5. 同步计算(并行性)

各个元胞在 $t+1$ 时刻的状态变化是独立的行为,相互没有任何影响。若将元胞自动机的构形变化看成是对数据或信息的计算或处理,则元胞自动机的处理是同步进行的,特别适合于并行计算。

6. 时空局部性

每一个元胞的下一 $t+1$ 时刻的状态,取决于其周围半径为 r 的邻域(或者其他形式邻居规则定义下的邻域)中的元胞的当前时刻 t 的状态,即所谓时间、空间的局部性。从信息传输的角度来看,元胞自动机中信息的传递速度是有限的。

7. 维数高

由于任何完备元胞自动机的元胞空间是定义在一维、二维或多维空间上的无限集,因

此每个元胞的状态便是这个动力学系统的变量。那么,元胞自动机是一类无穷维动力系统。

2.7.4 元胞自动机的分类

元胞自动机的应用领域广泛,模型种类繁多,且它们的构建又没有固定的数学公式,规则稍许的变化会构成一个完全不同的元胞模型,因此对元胞自动机的分类难度也较大。目前,对于元胞自动机分类的研究仍然是元胞自动机的一个重要的研究课题和核心理论。基于不同的出发点出现了两种最常用的划分:基于维数的元胞自动机分类和 Wolfram 的动力学行为的元胞自动机分类。

1. 基于维数的元胞自动机分类

任意一个元胞自动机都定义在一定的元胞空间上,按照一个元胞自动机所处的元胞空间维数可将其分为一维、二维、三维和高维元胞自动机。

(1) 一维元胞自动机

元胞按等间隔方式分布在一条向两侧无限延伸的直线上,每个元胞具有有限个状态 s , $s \in S = \{s_1, s_2, \dots, s_k\}$, 定义邻居半径 r , 元胞的左右两侧共有 $2r$ 个元胞作为其邻居集合 N , 定义在离散时间维上的转换函数 $f: S^{2r+1} \rightarrow S$ 可以记为 $S_i^{t+1} = f(S_{i-r}^t, \dots, S_{i-1}^t, S_i^t, S_{i+1}^t, \dots, S_{i+r}^t)$, S_i^t 为第 i 个元胞在 t 时刻的状态。称上述 $A = \{S, N, f\}$ 三元组为一维元胞自动机。

对一维元胞自动机的系统研究最早,相对来讲,其状态、规则等较为简单,往往其所有可能的规则可以一一列出,易于处理,研究也最为深入。目前,对于元胞自动机的理论研究多集中在一维元胞自动机上。Wolfram 对元胞自动机的动力学分类也是基于对一维初等元胞自动机的分析研究得出的。它的最大的一个特征在于容易实现元胞自动机动态演化的可视化,可视化以二维显示,一个维度显示其空间构形,即空间维;另一个维度显示其发展演化过程,即时间维。

(2) 二维元胞自动机

元胞分布在二维欧几里得平面上规则划分的网格点上,通常为方格划分。以 J. H. Conway 的“生命游戏”为代表,应用最为广泛。由于世界上很多现象是二维分布的,还有一些现象可以通过抽象或映射等方法,转换到二维空间上,所以二维元胞自动机的应用最为广泛,多数应用模型都是二维元胞自动机模型。

(3) 三维元胞自动机

目前, Bays 等人在这方面做了若干实验性工作,包括在三维空间上实现了“生命游戏”,延续和扩展了一维和二维元胞自动机的理论。

(4) 高维元胞自动机

只在理论上进行少量的探讨,实际的系统模型较少。

2. 基于动力学行为的元胞自动机分类

Wolfram 在详细分析研究了一维元胞自动机的演化行为,并在大量的计算机实验的基础上,将所有元胞自动机的动力学行为归纳为四大类。

(1) 平稳型

自任何初始状态开始,经过一定时间运行后,元胞空间趋于一个空间平稳的构形,这里空间平稳即指每一个元胞处于固定状态,不随时间变化而变化。



(2) 周期型

经过一定时间运行后,元胞空间趋于一系列简单的固定结构(Stable Patterns)或周期结构(Periodical Patterns)。由于这些结构可以看作一种滤波器(Filter),故可应用到图像处理的研究中。

(3) 混沌型

自任何初始状态开始,经过一定时间运行后,元胞自动机表现出混沌的非周期行为,所生成的结构的统计特征不再变化,通常表现为分形分维特征。

(4) 复杂型

出现复杂的局部结构,或者说是局部的混沌,其中有些会不断地传播。

第3章 仿真方法与技术

3.1 仿真技术的分类

3.1.1 根据模型的构造方法分类

根据模型的构造方法、结构和实现手段,模型分为物理模型和数学模型,因此系统仿真可分为五种:物理仿真、数学仿真、半实物仿真、人在回路中仿真、软件在回路中仿真。

1. 物理仿真

按照真实系统的物理性质构造系统的物理模型,并在物理模型上进行实验的过程称为物理仿真。物理仿真要求模型与原型有相同的物理数学基础。物理仿真的优点是直观、形象、模型能真实全面地体现原系统的特性。物理仿真的缺点是模型制作复杂、成本高、周期长、模型改变困难,实验限制多,投资较大。

2. 数学仿真

对实际系统进行抽象,并将其特性用数学关系加以描述而得到系统的数学模型,对数学模型进行实验的过程称为数学仿真。计算机技术的发展为数学仿真创造了环境,使得数学仿真变得方便、灵活、经济,因而数学仿真也称为计算机仿真。数学仿真的缺点是受限于系统建模技术,即系统的数学模型不易建立。

3. 半实物仿真

半实物仿真又称物理-数学仿真,准确称为硬件(实物)在回路仿真,这种方法是将数学模型与物理模型甚至实物联合起来进行实验。对系统中比较简单的部分或对其规律比较清楚的部分建立数学模型,并在计算机上加以实现;而对比较复杂的部分或对其规律尚不十分清楚的系统,其数学模型的建立比较困难,则采用物理模型或实物。仿真时将两者连接起来完成整个系统的实验。

4. 人在回路中仿真

人在回路仿真是操作人员、飞行员或航天员在系统回路中进行操作的仿真实验。这种仿真实验将对象实体的动态特性通过建立数学模型,编程在计算机上运行,此外要求有模拟生成人的感觉环境的各种物理效应设备,包括视觉、听觉、触觉、动感等人能感觉的物理环境的模拟生成。由于操作人员在回路中,人在回路仿真系统必须实时运行。

5. 软件在回路中仿真

软件在回路仿真又称为嵌入式仿真,这里所指的软件是实物上的专用软件。控制系统、导航系统和制导系统广泛采用数字计算机,通过软件进行控制、导航和制导的运算,软件的规模越来越大,功能越来越强,许多设计思想和核心技术都反映在应用软件中,因此软件在系统中的测试愈显重要。这种仿真实验将系统用计算机与仿真计算机通过接口对接,进行系统实验。接口的作用是将不同格式的数字信息进行转换。软件在回路中仿真一般情况下要求实时运行。



3.1.2 根据仿真所采用的计算机类型分类

根据所采用的仿真计算机类型可将仿真分为三种:模拟计算机仿真、数字计算机仿真和数字模拟混合仿真。

1. 模拟计算机仿真

模拟计算机仿真以相似原理为基础的,实际系统中的物理量,如距离、速度、角度和质量等都按一定比例变换的电压来表示,实际系统某一物理量随时间变化的动态关系和模拟计算机上与该物理量对应的电压随时间的变化关系是相似的。因此,原系统的数学方程和模拟计算机上的方程是相似的。只要原系统能用微分方程、代数方程(或逻辑方程)描述,就可以在模拟计算机上求解。

模拟计算机仿真具有以下特点:

①能快速求解微分方程。模拟计算机运行时各运算器是并行工作的,模拟计算机的解題与原系统的复杂程度无关。

②可以灵活设置仿真实验的时间标尺。模拟计算机仿真既可以进行实时仿真,也可以进行非实时仿真。

③易于和实物相连接。模拟计算机仿真是用直流电压表示被仿真的物理量,因此和连续运动的实物系统连接时一般不需要模拟信号到数字信号(A/D)、数字信号到模拟信号(D/A)的转换装置。

④模拟计算机仿真的精度由于受到电路元件精度的制约和易受外界干扰,所以一般低于数字计算机仿真,且逻辑控制功能较差,自动化程度也比较低。

2. 数字计算机仿真

数字计算机仿真是将系统数学模型用计算机程序加以实现,通过运行程序来得到数学模型的解,从而达到系统仿真的目的。数字计算机的基本组成包括存储器、运算器、控制器和外围设备等。由于数字计算机只能对数码进行操作,因此任何动态系统在数字计算机上进行仿真时,都必须将原系统变换成能在数字计算机上进行数值计算的离散时间模型。故数字计算机仿真需要研究各种仿真算法,这是数字计算机仿真与模拟计算机仿真的最基本的差别。

数字计算机仿真的特点如下:

①数值计算的延迟。任何数值计算都有计算时间的延迟,其延迟的大小与计算机本身的存取速度、运算器的解算速度、所求解问题本身的复杂度及使用的算法有关。

②仿真模型的数字化。数字计算机对仿真问题进行计算时采用数值计算,仿真模型必须是离散模型,如果原始数学模型是连续模型,则必须转换成适合数字计算机求解的仿真模型,因此需要研究各种仿真算法。

③计算精度高。特别是在工作量很大时,与模拟计算机相比更显其优越性。

④实现实时仿真比模拟仿真困难。对复杂的快速动态系统进行实时仿真时,对数字计算机本身的计算速度、存取速度等要求高。

⑤利用数字计算机进行半实物仿真时需要 A/D 和 D/A 的转换装置与连续运动实物连接。

3. 数字模拟混合仿真

本质上,模拟计算机仿真是一种并行仿真,即仿真时代表模型的各部件是并发执行的。早

期的数字计算机仿真则是一种串行仿真,因为计算机只有一个中央处理器(CPU),计算机指令只能逐条执行。为了发挥模拟计算机并行计算和数字计算机强大的存储记忆及控制功能,以实现大型复杂系统的高速仿真,20世纪60年代至70年代,在数字计算机技术还处于较低水平时,产生了数字模拟混合仿真,即将系统模型分为两部分,其中一部分在模拟计算机上运行,另一部分在数字计算机上运行,两个计算机之间利用A/D和D/A转换装置交换信息。

数字模拟混合仿真系统的特点如下:

①数字模拟混合仿真系统可以充分发挥模拟仿真和数字仿真的特点。

②仿真任务同时在模拟计算机和数字计算机上执行,这就存在按什么原则分配模拟计算机和数字计算机的计算任务的问题:一般模拟计算机承担精度要求不高的快速计算任务,而数字计算机则承担高精度、逻辑控制复杂的慢速计算任务。

③数字模拟混合仿真的误差包括模拟计算机误差、数字计算机误差和接口操作转换误差,这些误差在仿真中均应予以考虑。

④一般数字模拟混合仿真需要专门的混合仿真语言来控制仿真任务的完成。

随着数字计算机仿真技术的发展,其计算速度和并行处理能力逐步提高,模拟计算机仿真和数字模拟混合仿真已逐步被全数字计算机仿真取代。因此,现在的计算机仿真一般指的就是数字计算机仿真。

3.1.3 根据仿真时钟与实际时钟的比例关系分类

实际动态系统的时钟称为实际时钟,而系统仿真时模型所采用的时钟称为仿真时钟。根据仿真时钟与实际时钟的比例关系,系统仿真可分为实时仿真、亚实时仿真、超实时仿真。

1. 实时仿真

即仿真时钟与实际时钟完全一致,也就是模型仿真的速度与实际系统运行的速度相同。当被仿真的系统中存在物理模型或实物时,必须进行实时仿真,如各种训练仿真器就是这样,因此有时也称为在线仿真。

2. 亚实时仿真

即仿真时钟慢于实际时钟,也就是模型仿真的速度小于实际系统运行的速度。对仿真速度要求不苛刻的情况一般采用亚实时仿真,如大多数系统离线研究与分析,因此有时也称为离线仿真。

3. 超实时仿真

即仿真时钟快于实际时钟,也就是模型仿真的速度大于实际系统运行的速度,如大气环流的仿真、交通系统的仿真、生物进化(宇宙起源)的仿真等。

3.1.4 根据系统模型的特性分类

仿真基于模型,模型的特性直接影响着仿真的实现。从仿真实现的角度来讲,系统模型特性可分为连续系统和离散事件系统。由于这两类系统固有运动规律的不同,因而描述其运动规律的模型形式就有很大的差别,相应的,系统仿真技术也分为连续系统仿真和离散事件系统仿真。

1. 连续系统仿真

连续系统仿真是指基于连续系统模型的仿真,连续系统仿真的目的是要得到状态变量的动态变化过程(行为的点轨迹),并由此分析系统的性能。需要说明的是,离散时间变化

模型中的差分模型归为连续系统仿真范畴。原因在于,当用数字仿真技术对连续系统仿真时,其原有的连续形式的模型必须进行离散化处理,并最终也变成差分模型。

2. 离散事件系统仿真

离散事件系统是指基于离散事件系统模型的仿真,由于离散事件系统模型很难用人们熟悉的数学方程形式(如微分方程或差分方程等)来描述,而一般只能借助于活动图或者流程图,这样无法得到系统动态过程的解析表达。因此离散事件系统仿真的主要目标是获得系统行为的统计性能,而不是状态变量的动态变化过程(行为的点轨迹)。

3.2 仿真的一般过程

概括地说,完成计算机仿真需要经过三个活动,即系统建模、仿真建模和仿真实验。更具体地说,系统仿真需要经过问题描述和系统定义、建立系统模型、收集和整理数据资料、建立仿真模型,并编制程序、调试程序、确认模型、实验设计、计算机仿真运行、分析仿真结果、建立文档、实施仿真决策等步骤,仿真的一般过程如图3-1所示。

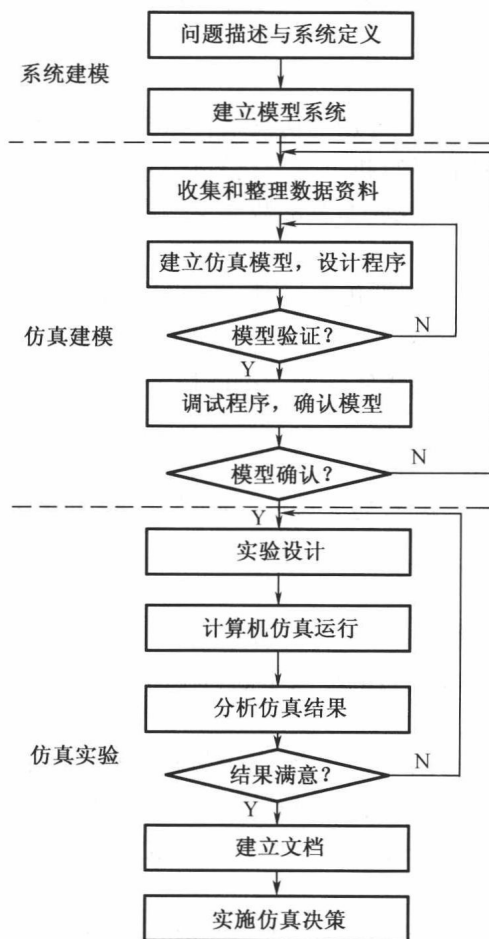


图 3-1 仿真的一般过程框图

1. 问题描述和系统定义

这是系统仿真的首要阶段。在进行系统仿真前,必须对将要研究的问题进行正确的定性或定量分析,明确系统仿真的目的和任务。在这一阶段中要正确定义所研究的系统边界、组成和环境,正确决定评价仿真方案优劣的准则,即衡量系统性能或输出结果的目标函数。这个阶段工作的好坏将直接影响到系统仿真的质量和效率。

2. 建立系统模型

根据系统结构、管理决策原则和作业规则,分析系统及各组成部分的状态变化和参数之间的数学逻辑关系,在此基础上建立所研究系统的数学逻辑模型。系统模型应该正确反映实际系统的本质,并且要繁简适宜。模型过于简化,无法全面、真实地反映实际系统,达不到研究系统的目的。模型过于详细,又会使模型显得繁琐,弱化了影响系统状态变化的主要因素,问题难以求解。因此,在建立系统模型时,可先建立系统主要因素较为简单的模型,以后再逐步进行补充和完善。

3. 收集和整理数据资料

在系统仿真中,需要输入大量的数据,这些数据的正确性直接影响仿真输出结果的正确性,因此正确地收集和整理数据资料是系统仿真的一个重要步骤。数据资料包括数学模型计算所需的参数以及基础资料、各项随机变量的分布函数及其相应参数、系统各组成部分的相互关系的定性分析资料。

4. 建立仿真模型,设计程序

运用通用的计算机程序语言或专用的仿真语言,将系统的数学模型转变为主要由计算机程序组成的仿真模型,以便在计算机上进行仿真运行。在这个阶段,要进行模型的验证,考察计算机仿真程序是否正确地反映系统的数学逻辑模型,在必要时要进行相应的修改。

5. 调试程序,确认模型

调试计算机程序,并进行调试性仿真运行,分析调试结果,进行模型的确认,主要检验所设计的数学模型是否正确地反映了现实系统的本质。在模型确认的过程中,需要相应地修改模型和调整计算机程序,直至数学逻辑模型的精度满足要求。

6. 实验设计

根据仿真任务的目的,建立系统仿真运行的实验条件。主要是指合理地设计具有不同可控变量与参数组合的仿真方案,设定系统的初始条件,确定仿真运行长度,决定随机样本大小和独立仿真运行次数等。

7. 计算机仿真运行

根据设定的仿真运行实验条件,进行计算机仿真运行,以获得丰富的仿真输出资料。应该详细、准确地记录每次仿真运行的输入参数和输出结果,供分析时使用。

8. 仿真结果分析

仿真结果的分析主要进行以下几方面:由于仿真结果是一种实验数据,因此要计算样本均值、均值方差以及置信区间等指标来分析仿真结果的统计特征;进行灵敏度分析,考察输入参数变化对输出结果的影响;根据设定的目标函数,选择较优的方案。

9. 建立文档

将经过验证、确认和运行考核的系统模型、计算机仿真程序以及相应的输入、输出资料和分析结论,建立文档以备查询。这些文档一方面供系统分析与设计人员改进和深化完善模型作参考,另一方面供管理决策人员改善决策和实施决策之用。

10. 实施仿真决策

将经过计算机仿真实验辅助做出的管理决策进行实施。

3.3 仿真技术的特点和作用

3.3.1 仿真技术的特点

仿真技术是伴随着计算机技术的发展而发展的。现代仿真技术均是在计算机支持下进行的,因此目前我们所说的仿真或系统仿真通常是指计算机仿真。计算机仿真有如下显著特点。

1. 模型参数任意调整

模型参数可根据要求通过计算机程序随时进行调整、修改或补充,使人们能掌握各种可能的仿真结果,为进一步完善研究方案提供了极大的方便。这正是计算机被称为“计算机实验”的原因,这种“实验”与通常的实物实验相比,具有运行费用低、无风险以及方便灵活等优点。

2. 系统模型快速求解

借助于先进的计算机系统,人们在较短时间内就能知道仿真运算的结果(数据或图像),从而为人类的实践活动提供强有力的指导。这是通常的数学模型方法所无法实现的。

3. 运算结果准确可靠

只要系统模型、仿真模型和仿真程序是科学合理的,那么计算机的运算结果一定准确无误(除非机器有故障)。因此,人们可毫无顾虑地应用计算机仿真的结果。

4. 仿真结果形象直观

计算机仿真的结果易于通过图形、图像来形象直观地表现。把仿真模型、计算机系统和物理模型及实物联结在一起的实物仿真(有些还同时是实时仿真),形象十分直观,状态也很逼真。

3.3.2 仿真技术的作用

仿真方法的独特作用主要表现在以下几方面。

1. 优化系统设计

对于复杂系统的研究,一般要求达到最优化,为此必须对系统的结构和参数反复进行修改和调整。这只有借助计算机仿真方法才能方便、快捷地实现。

2. 降低实验成本

对于复杂的工程系统,如果直接进行实物实验,则费用会很高。而应用计算机仿真方法就可大大降低费用。以航空工业为例,一般单次试飞的成本为1万至1亿美元(依不同机型而定)。若用仿真手段,费用仅为上述成本的1/10~1/5,且设备可重复使用。

3. 减少失败风险

对于一些难度高、风险大的复杂工程系统,如载人宇宙飞行,若直接实验,一旦失败则无论在经济上还是在政治上都是难以承受的。为了减少风险,必须先进行计算机仿真实验,以提高成功率。

4. 提高预测能力

对于各种非工程复杂系统,如经济、军事、社会等系统,几乎不可能进行直接实验研究,因而也很难准确预测其发展趋势。但计算机仿真实验却可以在给定的边界条件下,推演出此类系统的变化趋势,从而为人们制定对策提供可靠的依据。

3.4 仿真技术的应用

20 世纪中期以来,计算机技术和信息技术的发展,推动了仿真技术的发展与应用。利用某种建模方法描述所研究的对象,采用一定的求解方法和计算机平台计算模型进而求得问题的解答,成为分析和解决复杂问题的一种普遍而有效的途径。可以说,继理论研究、实验研究之后,仿真实验正在成为人类认识世界和改造世界的“第三种手段”,而且这种手段的应用范围越来越广阔。

理论上系统仿真可以解决任何系统分析的问题,系统仿真的应用最有魅力的地方正是其不拘一格的应用对象和近乎神奇的手段方法。美国在调停波黑参战三方时就进行了仿真,将三方的地图、地区、人员情况、武器装备、司令部、弹药库的位置,以及采用的战术等建立起系统模型,再运用逻辑模型模拟整个战斗的过程。模拟的结果是三败俱伤。看完仿真演示后,三方立即在停战协议上签了字。1990 年美国一栋大楼失火,根据失火前大楼的初始状态,火熄灭后的终止状态,以及火与烟的运动规律归纳出大楼着火的逻辑模型,根据线索找到了起火点。在演示了着火的全过程后,法官据此找到着火原因,做出了相应的判决。20 世纪 90 年代为制订更加科学的人口生育计划,以利中国人口计划生育以及中国经济的发展,建立了人口模型,模拟各种计划生育策略对我国人口增长、经济发展产生的影响,为我国制定计划生育策略提供了帮助。此外,宇航仿真系统用于供宇航员进行地面训练,反导弹仿真系统用来研究反导弹系统的性能等。

以上实例中的系统虽然各不相同,研究目标各不相同,但它们的共同之处则是:系统都十分复杂,问题非常棘手,用现有的方法无法着手进行研究。这些例子说明了一个共同的事实:在研究处于山穷水尽的时候,系统仿真往往成了唯一可求助的方法。这就是系统仿真给我们的启示。它提醒人们,问题越是复杂,越是无计可施,不妨试试用系统仿真技术来解决。

随着专业仿真语言的产生,计算机运行性能的提高、费用降低,仿真方法的进步,系统仿真在应用需求的牵引和有关学科发展的推动下,已经发展成为一项能在工业、农业、商业、教育、军事、交通、社会、经济、医学、娱乐、生活、服务等众多领域广泛采用的技术,并发展成为人们认识和改造客观世界的一项通用型技术。国际上一致认为系统仿真是“迄今为止最为有效、经济的综合集成方法”,是“推动科技进步的战略技术”。20 世纪 90 年代初,美国提出 22 项国家关键技术,系统仿真技术被列为第 16 项。美国国防部提出 21 项国防关键技术,系统仿真技术被列为第 6 项。美国已严格规定所有重要的武器研究,必须进行仿真实验后才可投入正式生产和使用。根据 20 世纪 80 年代末统计,日本企业运用系统工程解决管理与决策问题时,采用系统仿真方法的已经超过 80%。据国外应用统计,运用系统仿真优化系统设计规划减少投资 30%,在库存控制方面可减少库存 15%。英国制造业也普遍采用系统仿真方法解决物料控制、人力配置、调度评估、投资策略及均衡生产等问题。



尽管从理论上讲系统仿真可以解决任何系统分析的问题,但是由于先验知识和手段所限,迄今为止,并不是任何系统都能运用系统仿真分析得到满意的结果或效果。下面为一些不适合仿真应用的场合:

(1) 当问题可以通过普通的、解析的方法解决时,就不应该使用仿真。例如,一个理发店系统,顾客以 10 人/小时的速率随机抵达,理发师的服务速度是平均 4 人/小时,为决定最少所需的理发师数量,并不需要用仿真,因为只需计算 $10/4 = 2.5$,就表明需要三个或更多的理发师。

(2) 如果直接进行实验更为简便,则不应使用仿真。例如,为一个可以免下车的路边快餐店接收订单的人员配备手持终端及语音通信设备,为确定再增加一个订单接收站对客户等待时间的影响,实验的开销会更低。

(3) 如果成本超过仿真节省的开销,则不应使用仿真。完成一个仿真需要很多步骤,需要投入一些人力、物力,需要花费一些时间和成本。如果仿真成本估计为 2 万元,而仿真只节省 1 万元,则建议不要将资金投入仿真系统中。类似的如果需要在两周内做出决定,而仿真需要花费一个月的时间,则也不建议进行仿真研究。

(4) 如果系统行为太复杂或不可定义,无法建立有效的模型,则不适合使用仿真。如对人类行为建立数学模型有时是极为复杂的,很难建立一个有效的系统或仿真模型。

(5) 仿真需要数据,有时需要大量的数据。如果无数据可用,甚至数据都无法估计,则不建议使用仿真。

3.5 仿真技术的发展

由于近年来计算机、网络、图形图像、多媒体、软件工程、信息处理等技术发展的促进,仿真技术发展得十分迅速,其发展的方向主要集中在以下几方面。

1. 面向对象仿真(Object-Oriented Simulation)与基于 Multi-Agent 技术的仿真

面向对象仿真在理论上突破了传统仿真方法的观念,它根据组成系统的对象及其相互作用关系来构造仿真模型,模型的对象通常表示实际中相应的实体,从而弥补了模型与实际系统之间的差距,而且分析、设计和实现系统的观点与人们认识客观世界的自然思维方式极为一致,因而增加了仿真研究的直观性和易理解性。

面向对象的仿真具有内在的可扩充性和可重用性,因而为仿真大规模复杂系统提供了极为方便的手段。面向对象的仿真容易实现与计算机图形学、人工智能/专家系统和管理决策科学的结合,从而可以形成新一代面向对象的仿真建模环境,更便于在决策支持和辅助管理中推广与普及仿真决策技术。

Agent 是指能够在复杂环境中执行任务的自治实体,Multi-Agent 技术是当今计算机科学领域、信息工程领域和网络通信领域十分活跃的前沿方向之一,以其高度的并发性、强大的建模能力、对分布环境的适应性和良好的可扩充体系结构,得到了广泛的应用。Agent 是并发执行的自治实体,具有逻辑规范的推理系统,采用语言行为协议和消息传递方式相互通信。Agent 成员之间相互协同,相互服务,共同完成一个任务,其自身的目标和行为不受其他 Agent 成员的限制,它通过竞争或协商手段协调解决各 Agent 成员的目标和行为之间的矛盾与冲突。

2. 可视化、多媒体与虚拟现实仿真技术

面向对象的仿真方法,提供了更为自然、直观的系统仿真框架,实施框架则要进一步探

讨创建高逼真感觉的信息环境,使真实化环境、模型化的物理环境与用户融为一体,使研究主体的人产生身临其境的感觉,这种人机和谐的仿真环境的探索已广泛展开,诸如可视化仿真(Visual Simulation)、多媒体仿真(Multimedia Simulation)、虚拟现实(Virtual Reality)等。

(1) 可视化仿真

数据及其他形式的信息通过可视化处理创建仿真分析环境或仿真虚拟环境,可使参与人员通过视觉信息掌握系统中变量之间、变量与参数之间、变量与外部作用之间的变化关系,直接了解系统的静态与动态特性,通过信息系统展示的动态变化规律,深化对系统模型概念化和形象化的理解,而且还能获得启发和灵感,发现数据信息不能显示的现象。

(2) 多媒体仿真

利用系统分析的原理和信息技术,以更接近自然的多媒体形式建立描述系统内在变化规律的模型,并在计算机上以多媒体的形式再现系统动态演变过程,从而获取有关系统的感性和理性认识。

(3) 虚拟现实

由计算机全部或部分生成的多维感觉环境,使参与者获得各种感官信号的刺激,如视觉、听觉、触觉等,产生身临其境的感觉,能体验、感受和认识客观世界中的事物,同时,人与虚拟环境之间可以进行多维信息的交互,参与者从定性和定量综合集成的虚拟环境中可以获得对客观世界中客观事物的感性和理性认识,从而深化概念和建造新的构想与创意。

3. 分布式交互仿真

分布式交互仿真(Distributed Interactive Simulation)是计算机仿真技术与网络技术相结合的产物,是“采用协调一致的结构、标准、协议和数据库,通过局域网、广域网将分布在各地的仿真设备互联并交互作用,同时可由人参与交互作用的一种综合环境。”分布式交互仿真系统在功能上主要有分布性、实时性、集成性和开放性等特点。

4. 智能仿真

计算机仿真与人工智能和专家系统的结合,使系统仿真进入了新的领域。知识库仿真系统可以达到领域专家处理问题的水平,自动建模与仿真程序生成可以取代部分仿真建模与编程人员的工作,智能化仿真决策系统可辅助决策人员进行科学决策。人工智能在于计算机仿真的关系上表现为人工智能可以帮助人们进一步认识仿真,并利用人工智能技术实现智能建模、智能仿真和智能仿真查询应用,是计算机技术在进入一个新的智能化仿真的阶段。

随着计算机技术的迅速发展,系统仿真技术在交通运输与物流系统的规划和管理中将起到更加重要的作用。运用系统仿真技术,可以对交通运输与物流系统规划和管理进行分析及决策,特别是对存在多种随机性变量及复杂逻辑关系的交通运输离散事件系统,系统仿真技术更具有十分重要的作用。

5. Internet 网上仿真

因 Internet 技术的迅速发展,利用面向对象的互联网程序语言 Java,已经开发了多种面向 WWW(World Wide Web)的仿真系统,如美国海军研究院的 Simkit 可以在浏览器的支持下进行分布式仿真,用 Simkit 建立的仿真模型可以在世界任何地点的网络用户机上运行,使分布在各网点的用户仿真模型可在其他网点上运行或进行全球范围内总体仿真模型的分布式仿真运行。近年来,利用面向 WWW 的程序语言开发离散事件仿真系统、基于 WWW 的仿真建模以及互联网上的仿真运行已成为系统仿真中研究工作的热点。

第 4 章 MATLAB 基础

4.1 MATLAB 的基本构成与操作

4.1.1 MATLAB 主界面的结构与功能

1. MATLAB 工作环境

MATLAB 有以下两种启动方法:在安装有 MATLAB 的计算机上,双击 Windows 桌面的快捷图标;从“开始”菜单的“程序”子菜单中选择“MATLAB”。启动后的界面如图 4-1 所示(以 MATLAB R2012a 为例)。

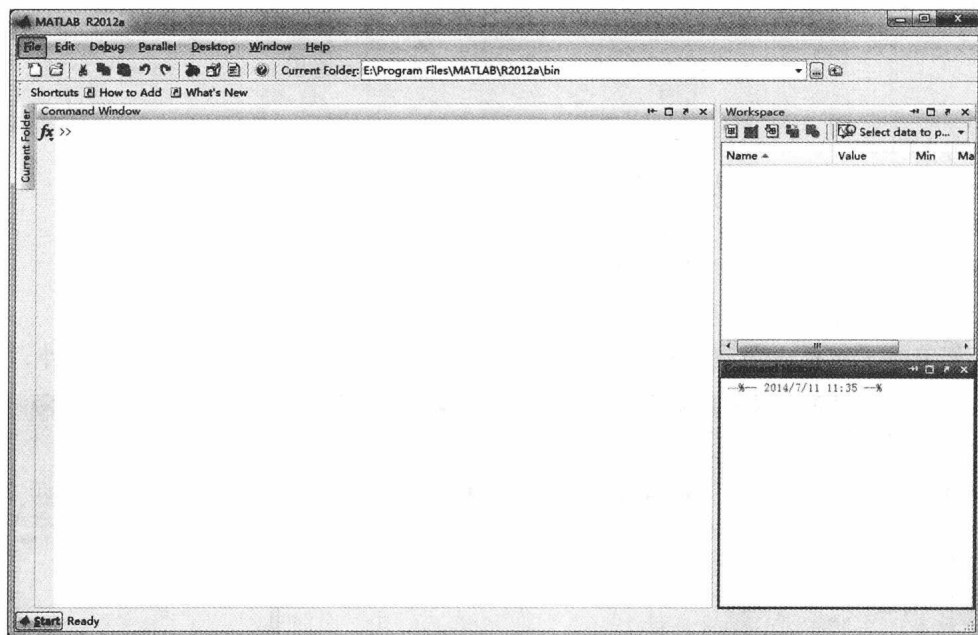


图 4-1 MATLAB 的默认操作界面示意图

图 4-1 大致包括以下几部分:菜单栏,工具栏,命令窗口“Command Window”,工作空间“Workspace”,历史记录窗口“Command History”,当前路径文件列表窗口“Current Directory”和帮助引导窗口“Launch Pad”。

主菜单包括“File”“Edit”“Debug”“Desktop”“Window”“Help”菜单。

(1)“File”(文件)下拉菜单(如图 4-2 所示)

该菜单除了包括 Windows 一般应用程序所具备的“新建”“打开”“关闭”“退出”“打印”功能外,它还包括如下菜单项:

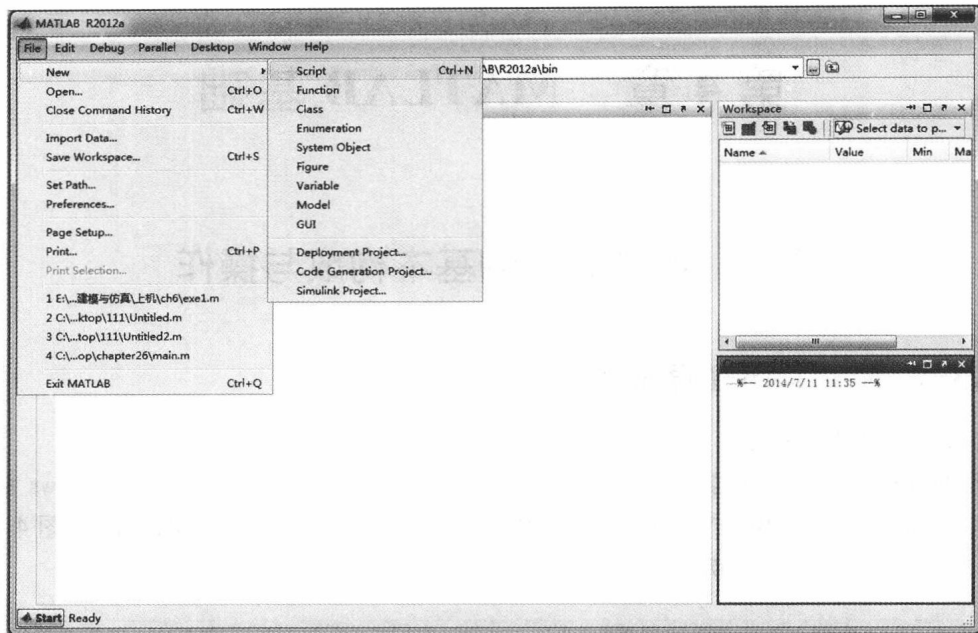


图 4-2 “File”(文件)下拉菜单示意图

“Import Data...”导入有关数据；

“Save Workspace ...”保存工作台；

“Preferences...”设置部分 MATLAB 工作环境的交互性；

“Set Path...”设置当前工作路径。

(2) “Edit”(编辑)下拉菜单(如图 4-3 所示)

该菜单除了包括一般应用程序所具备的“撤销”“重复”“粘贴”“全选”功能外,它还包

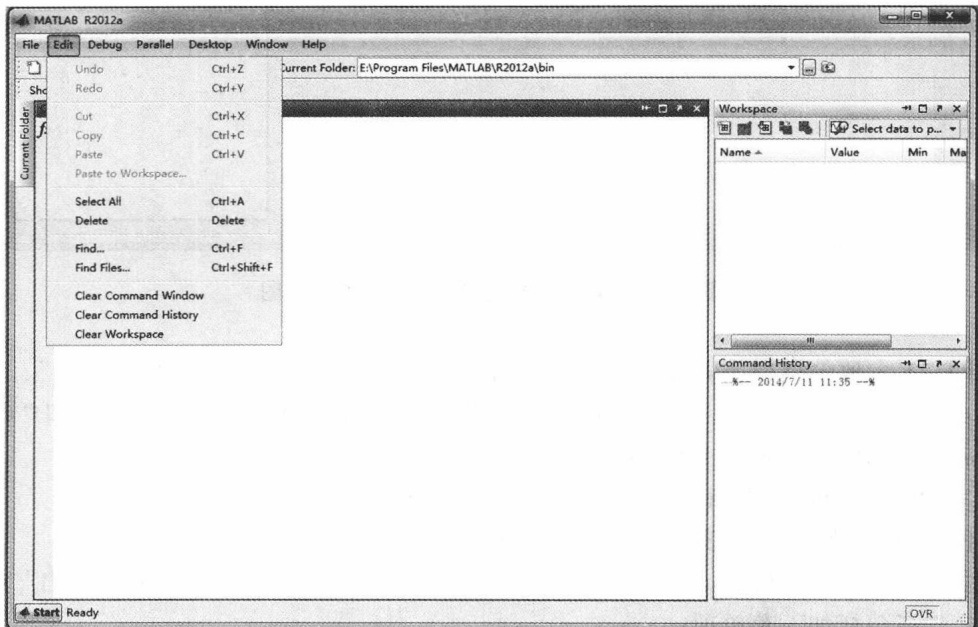


图 4-3 “Edit”(编辑)下拉菜单示意图

括如下菜单项：

- “Clear Command Window”清除命令窗口；
 - “Clear Command History”清除命令的历史记录；
 - “Clear Workspace”清除工作空间。
 - (3) “Debug”(调试)下拉菜单(如图4-4所示)
- 该菜单用于调试程序,具体功能如下：
- “Open M - File when Debugging”调试时打开文件；
 - “Step”调试时单步运行；
 - “Step In”调试时单步运行进入子函数；
 - “Step Out”调试时单步运行跳出子函数；
 - “Continue”运行程序到下一个断点或程序结束；
 - “Clear Breakpoints in All Files”清除所有的断点；
 - “Stop if Errors/Warnings”在程序出错或报警处停止；
 - “Exit Debug Mode”退出调试模式。

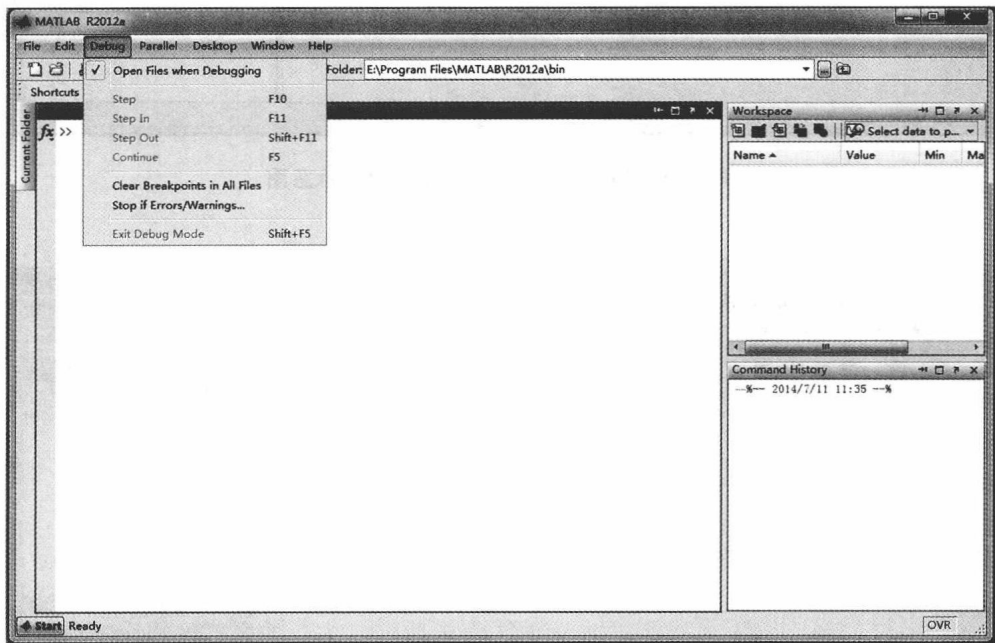


图4-4 “Debug”(调试)下拉菜单示意图

- (4) “Desktop”(桌面)下拉菜单(如图4-5所示)

为了改变 MATLAB 工作环境的外观,“桌面”菜单可以决定是否显示界面上摆布的一些窗口。

- (5) “Windows”(窗口)下拉菜单(如图4-6所示)

“窗口”菜单用于显示当前打开的 M 文件的文件名在已打开的窗口之间进行切换。

- (6) “Help”(帮助)窗口下拉菜单(如图4-7所示)

“帮助”菜单能为用户提供进入各类帮助系统的方法,通过菜单项打开帮助窗口,将显示各部分所需要的帮助信息。

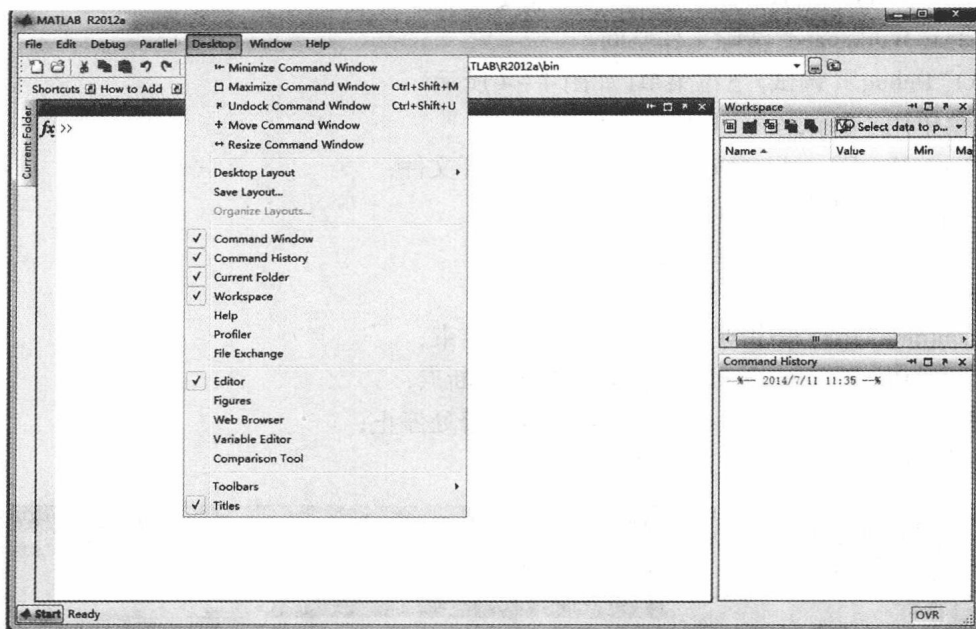


图 4-5 “Desktop”(桌面)下拉菜单示意图

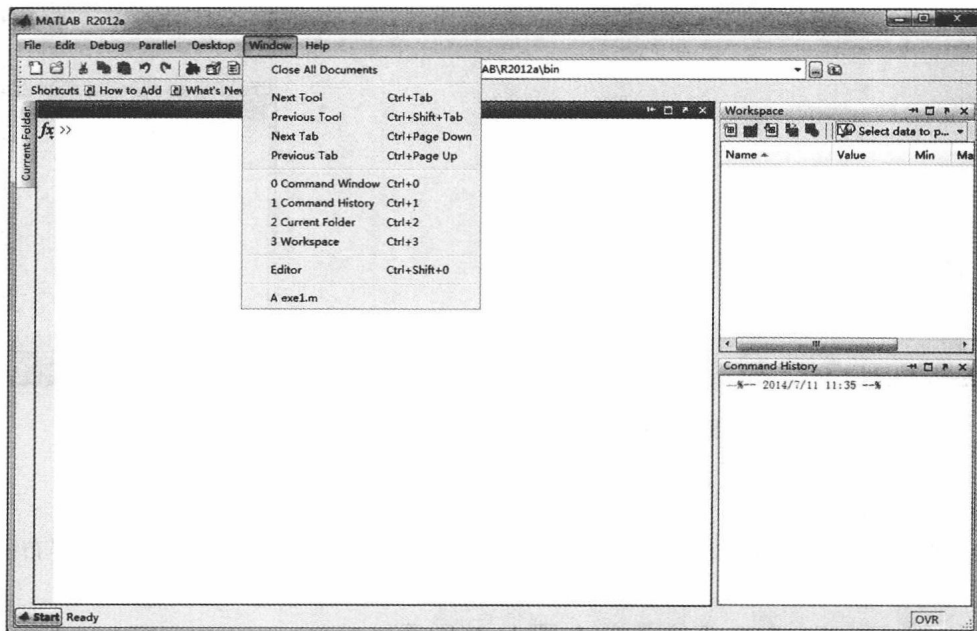


图 4-6 “Windows”(窗口)下拉菜单示意图

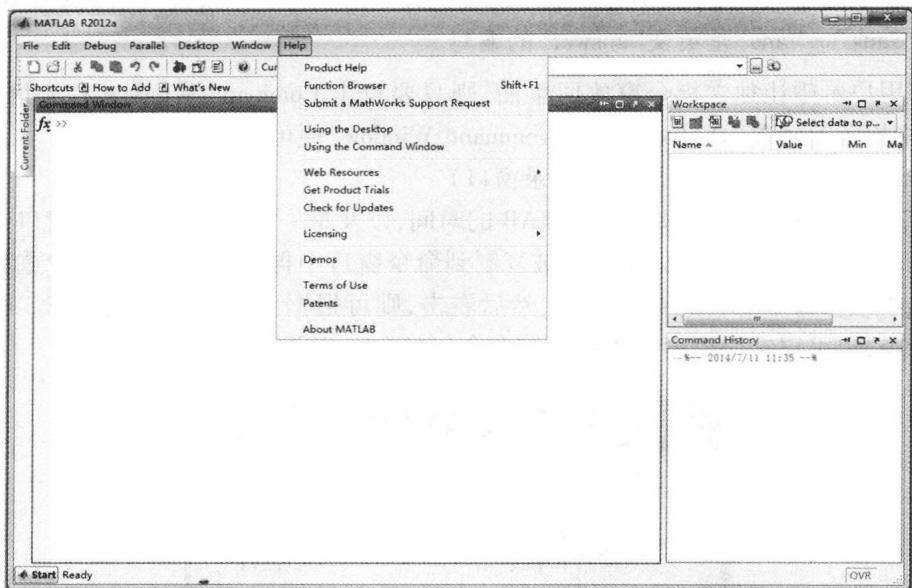


图 4-7 “Help”(帮助)下拉菜单示意图

2. 常用窗口简介

(1) Command Window (命令窗口)

MATLAB 的使用方法和界面有多种形式,但是最基本的也是入门时首先要掌握的是 MATLAB 命令窗口(Command Window)的基本表现形态和操作方式。命令窗口显示形式如图 4-8 所示。

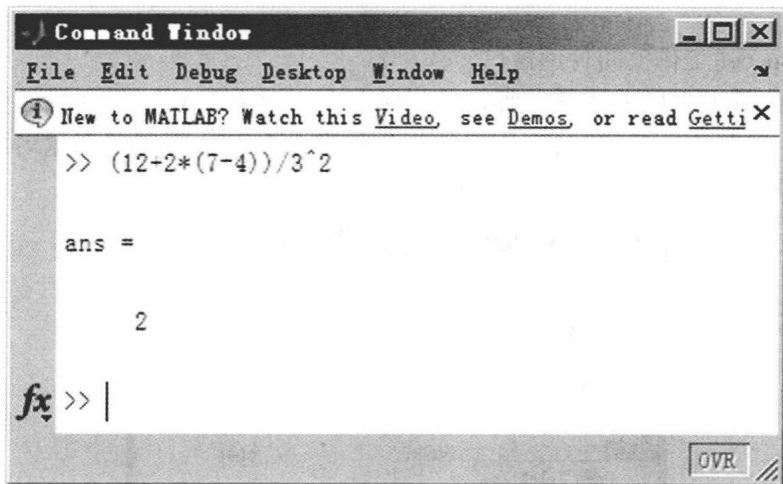


图 4-8 命令窗口示意图

【说明】

- ①指令行“头首”的“>>”是“指令输入提示符”,它是自动生成的。
- ②MATLAB 的运算符(如 +, - 等)都是各种计算程序中常见的习惯符号。
- ③一条指令输入结束后,必须按 [Enter] 键,指令才被执行。

④由于本例输入指令是“不含赋值号的表达式”，所以计算结果被赋给 MATLAB 的一个默认变量“ans”。“ans”是英文“answer”的缩写。

⑤若用户希望让独立指令窗放回桌面，则只要点击 Command Window 右上角的  按钮，或选中指令窗菜单“Desktop:Dock Command Window”便可。

(2) Command History(历史命令记录窗口)

该窗口记录着用户每次开启 MATLAB 的时间，以及每一次开启后在命令窗口中运行过的所有命令行。这些命令行记录可以被复制到命令窗口中再次运行，从而减少重新输入的麻烦。选中该窗口中的任意命令记录，然后右击，则可根据菜单运行相应操作。或者双击某一行命令，也可以在命令窗口中执行该命令(如图 4-9 所示)。

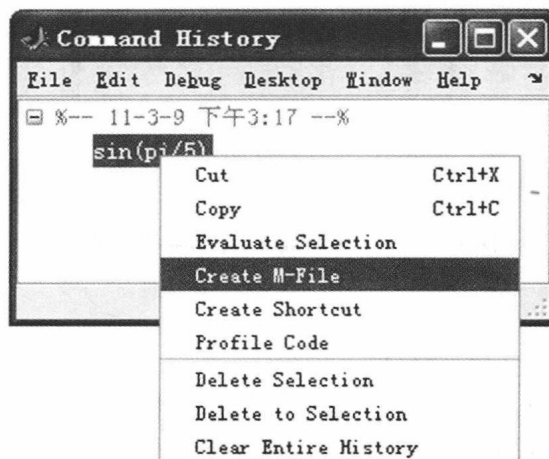


图 4-9 历史命令窗口的操作示意图

(3) Workspace(工作空间管理窗口)

在工作空间管理窗口中将显示所有目前保存在内存中 MATLAB 变量的变量名及其对应的数据结构、字节数以及类型，而不同的变量类型分别对应不同的变量名图标。选中一个变量，右击则可根据菜单运行相应的操作(如图 4-10 所示)。

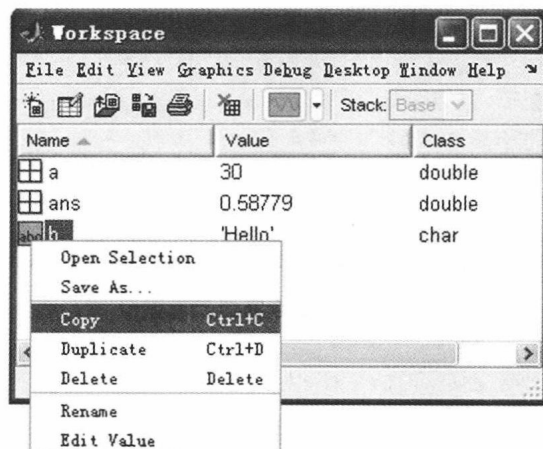


图 4-10 工作空间管理窗口的操作示意图



(4) Current Directory (当前目录窗口)

在当前目录窗口中可以显示或改变当前目录,还可以显示当前目录下的文件,包括文件名、文件类型、最后修改时间以及该文件的说明信息等,并提供搜索功能,如图 4-11 所示。

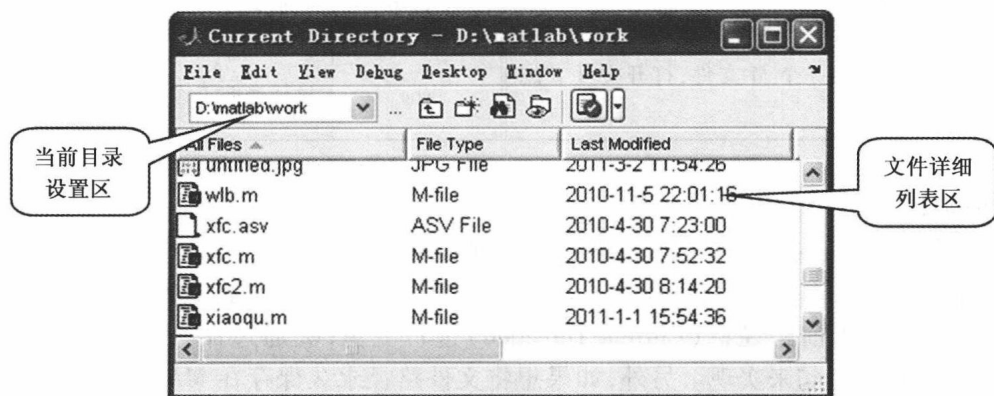


图 4-11 当前目录窗口示意图

MATLAB 只执行当前目录或搜索路径下的命令、函数及文件。当前目录是指 MATLAB 运行文件时的工作目录。用户可以将自己的工作目录列入 MATLAB 搜索路径,从而使用户目录纳入 MATLAB 系统统一管理。用对话框设置搜索路径的操作过程是:在 MATLAB 的“File”菜单中选择“Set Path”或者在命令窗口中执行“pathtool”命令,将出现搜索路径设置对话框。通过“Add Folder”或者“Add with Subfolder”按钮将指定路径添加到搜索路径列表中。在修改路径后,需要将其保存。

4.1.2 命令窗口的基本操作命令

用户除了可以利用上述介绍的菜单功能对 MATLAB 主界面进行操作外,还可以在命令窗口中输入命令语句进行相关操作。表 4-1 给出了常用的基本命令及其功能说明。

表 4-1 常用基本命令及其功能说明

命令	功能	命令	功能
cd	改变(或指定)当前路径	help fn	显示文件名为 fn 的帮助信息
clear	清除当前内存中的所有变量,恢复默认状态,同 clear all;若要清除指定变量 x,则用 clear x	hold	保持当前图形,等价 hold on
cla	清除图形坐标	hold off	结束当前图形的保持状态
clc	清除命令窗口中的所有内容	matlabroot	返回 MATLAB 的根目录
clf	清除已打开的图形	open var a	打开工作空间中的变量 a
copyfile(S,D)	复制源文件 S 至新目标文件 D	pwd	显示当前目录路径
delete	删除文件	quit	退出 MATLAB,同 exit
dir	显示当前路径下的文件目录和信息,同指令 what	support	打开 Math Works 公司的技术支持网站

表 4-1(续)

命令	功能	命令	功能
disp(' * ')	显示单引号中的内容 *	type p	显示文件名为 p 的内容
disp(b)	显示指定变量 b 的内容		
edit, edit p	创建一个 M 文件, 打开文件 p 的窗口	who	显示当前工作空间的变量

【说明】

在命令窗口中运行程序时,所需的相关文件必须在当前路径上,即 Current Direction 中显示的路径。如果不在,则要将相关文件的路径变为当前路径。有两种操作方法:①通过图标工具栏中的当前路径栏(Current Direction)进行变更;②通过命令窗口执行命令 cd(' * ')所构成的语句来实现。另外,如果想将文件路径永久保存在 MATLAB 的工作路径管理器中,可以通过主菜单的 File/Set path 弹出的对话框进行设置。

4.1.3 键盘快捷操作

按键及其组合可以实现对 MATLAB 的简便操作,表 4-2 给出了常用的快捷功能键说明。

表 4-2 快捷功能键说明

命令	功能	命令	功能
PgUp	上翻 1 页(屏)	PgDn	下翻 1 页(屏)
Home	光标返转到当前程序的开始	End	光标返转到当前程序的末尾
Ctrl + Home	光标返转到首页	Ctrl + End	光标返转到页尾
←	左移	→	右移
↑	显示上一次输入的命令	↓	显示下一次输入的命令
Ctrl + C	复制选项	Ctrl + V	粘贴
Ctrl + Q	退出 MATLAB	...Shift + Enter	命令窗口中语句未结束时的换行(续行)
Shift + Enter	命令窗口中的程序换行		

4.1.4 帮助功能操作

MATLAB 的帮助功能十分强大,是学习 MATLAB 编程方法的重要途径。

1. 由主界面主菜单进入帮助

在主菜单中选择 Help→MATLAB Help 命令,会弹出如图 4-12 所示的 MATLAB 帮助界面。

Help 界面由帮助向导窗口(Help Navigator)和帮助信息显示窗口组成。在 Help Navigator 窗口中,可以从目录(Contents)和索引(Index)中选择查找信息,也可以在搜索

(Search)中输入要查找的命令或语句并回车确认,对应的帮助说明会显现在右边的窗口中。

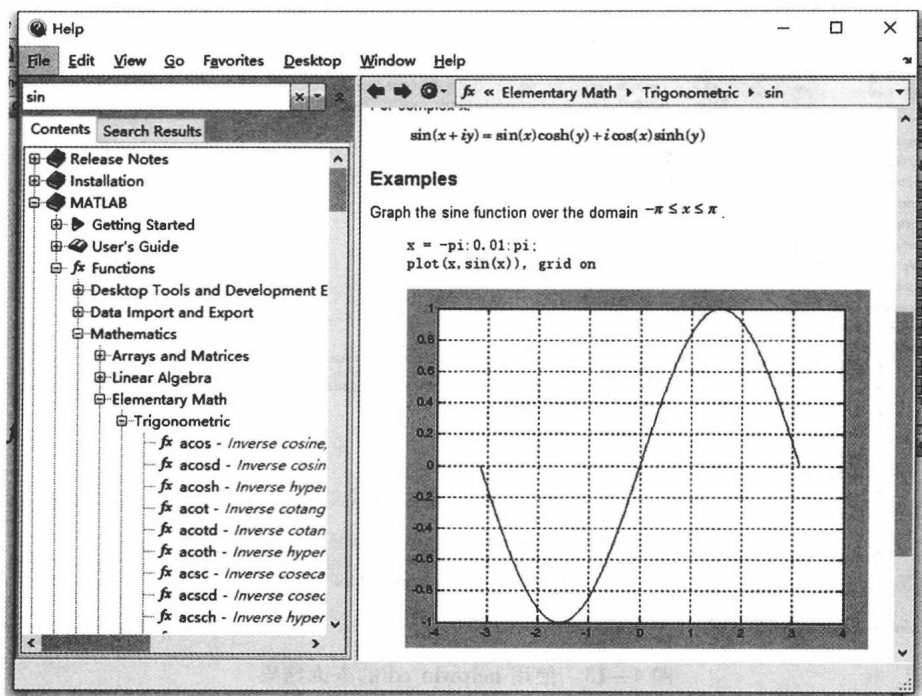


图 4-12 MATLAB 帮助界面示意图

2. 从主界面中的 Launch Pad 窗口中进行查找

在菜单“View”中选择 Launch Pad,或从 Workspace 窗口底部点击图标 Launch Pad,会弹出显示 MATLAB 目录的 Launch Pad 窗口,可以从中选择目录列表并逐级打开查阅。

3. 从图标工具栏进入帮助

选图标工具栏中的图标“?”,选中之后的操作方法与“1”相同。

4. 由命令窗口得到帮助信息

在命令窗口中键入帮助查询命令可以得到相关的帮助信息,格式及功能如下。

helpfn:在命令窗口中显示命令函数 fn 的 M 文件的功能说明。

typefn:在命令窗口中显示命令函数 fn 的 M 文件全部内容。

helpwin fn:在 Help 窗口中显示命令函数 fn 的 M 文件帮助信息。

例如,查询命令名为 cd 的使用方法,其步骤为:

①在命令窗口中键入 helpwin cd,回车后则自动弹出 Help 窗口,并显示关于 cd 的基本信息;

②在 Help 窗口上方点击“Go to online doc for cd”,则会进一步显示 cd 的详细使用说明和实例,如图 4-13 所示。

对不同的 MATLAB 版本和安装选项,上述指令所显示的结果会有所不同。

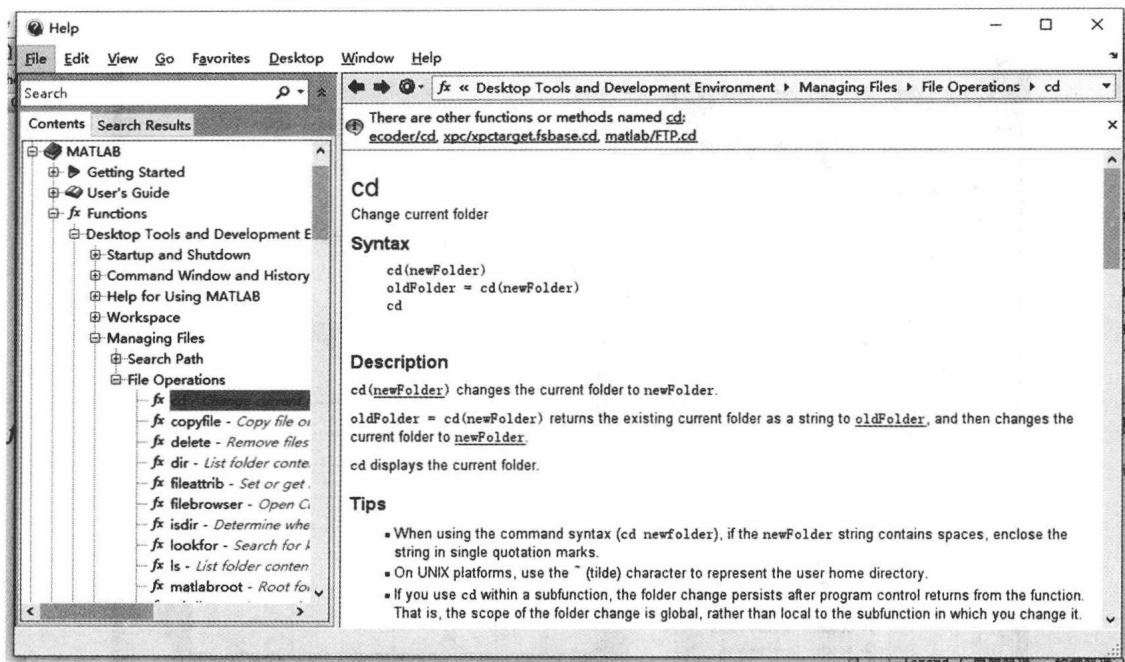


图 4-13 使用 helpwin cd 的查询结果

4.2 变量与函数

4.2.1 数值的记述

MATLAB 的数值采用习惯的十进制表示,可以带小数点或负号。以下写法都是合法的。

3 -99 0.001 9.456 1.3e-3 4.5e33

在采用 IEEE 浮点算法的计算机上,数值通常采用“占用 64 位内存的双精度”表示。其相对精度是 `eps` (MATLAB 的一个预定义变量),大约保持有效数字 16 位。数值范围大致从 10^{-308} 到 10^{308} 。

4.2.2 默认常量

为了合乎习惯且便于计算,MATLAB 默认和自定义了一些常量,但这些默认常量又可由用户作为任意变量自行定义和使用,且用户拥有对默认变量的优先定义和使用权。使用后的默认变量可由清除命令形式“`clear 变量`”来恢复为默认常量。如 `i` 和 `j` 是 MATLAB 默认的复数的虚部常量,但用户可以把它们作为任何变量来自由使用,使用后若想将其恢复为 MATLAB 默认常量,可用“`clear i`”或“`i = sqrt(-1)`”来恢复其为默认常量。通常情况下,建议尽量避免对默认常量的使用。常用的默认常量如表 4-3 所示。

表 4-3 默认常量

预定义变量	含义	预定义变量	含义
eps	浮点数相对精度 2^{-52}	NaN 或 nan	不是一个数 (Not a Number), 如 $0/0, \infty/\infty$
i 或 j	虚单元 $i=j=\sqrt{-1}$		
Inf 或 inf	无穷大, 如 $1/0$	pi	圆周率 π
intmax	可表达的最大正整数, 默认 (2147483647)	realmax	最大正实数, 默认 $1.797\ 7e+308$
intmin	可表达的最小负整数, 默认 (-2147483648)	realmin	最小正实数, 默认 $2.225\ 1e-308$

【说明】

①假如用户对表中任何一个预定义变量进行赋值, 则那个变量的默认值将被用户新赋的值“临时”覆盖。所谓“临时”是指: 假如使用 clear 指令清除 MATLAB 内存中的变量, 或 MATLAB 指令窗被关闭后重新启动, 那么所有的预定义变量将被重置为默认值, 不管这些预定义变量曾被用户赋过什么值。

②在遵循 IEEE 算法规则的机器上, 被 0 除是允许的。它不会导致程序执行的中断, 只是在给出警告信息的同时, 用一个特殊名称 (如 Inf, NaN) 记述。这个特殊名称将在以后的计算中以合理的形式发挥作用。

关于它们的更详细的帮助信息, 可在 MATLAB 帮助浏览器左侧 Contents 页的 < MATLAB/Functions/Mathematics/Math Contents > 中找到。

4.2.3 变量**1. 变量命名规则**

在 MATLAB 中, 变量 (或变量名) 是指对一个或多个数据或字符的整体表示的符号。MATLAB 对变量定义的规则是:

①变量的第一个起始符号必须是英文字母, 之后可以用任意的数字、字母、字符、下划线等符号组成;

②通常变量长度不超过 18 个符号;

③变量区分字母的大小写, 如 abc 和 Abc 是两个不同的变量。

变量中的内容称为数据或元素。如 $A=1.234$, 其中 A 是变量, 1.234 是 A 的数据; $BS=[1,a,3]$, 其中 BS 是变量, 1, a, 3 是 BS 的数据或元素。按变量的使用权限, 可划分为局部变量 (默认状态下, 变量限制在个体函数或当前工作空间内使用) 和全局变量 (global, 可以在几个函数及所有 MATLAB 函数中使用); 按变量表示的结构形式, 可划分为矩阵 (matrix) 形式、向量 (vector) 形式 (或序列) 和单元组 (cell array) 形式; 按变量所表示的内容类型, 可划分为数值 (numeric) 型、字符 (char) 型、符号 (symbol) 型和默认常量。

对任何形式的变量, 在运算之前必须对其进行赋值 (如数值变量) 或定义 (如字符、符号等变量), 否则在运算中, 会显示出错的信息提示。函数中的自变量和不确定常量需要提前定义为符号型变量使之成为常量后, 才能参与运算。

在程序运行过程中,所有经过运算的变量均会被保存并显示在 Workspace 中,其中包括变量的结构类型。可以点击变量名来查看变量的内容。可通过在 Command Window 中键入清除命令“clear”或“clear all”来清除 Workspace 中的所有变量及其内容。

2. 永久变量

永久变量是变量的一种特殊情况,它在工作空间中看不到,但是使用者可直接调用。常用的永久变量如表 4-4 所示。

表 4-4 永久变量

名称	取值
ans	用于结果的缺省变量名
pi	圆周率 π 的近似值(3.141 6)
eps	数学中无穷小(epsilon)的近似值($2.220\ 4e-016$)
inf	无穷大,如 $1/0 = \text{inf}$ (infinity)
NaN	非数,如 $0/0 = \text{NaN}$ (Not a Number), $\text{inf}/\text{inf} = \text{NaN}$
i,j	虚数单位
realmax	系统所能表示的最大数值
realmin	系统所能表示的最小数值
nargin	函数的输入参数个数
nargout	函数的输出参数个数

3. 变量符号

MATLAB 中进行符号运算时需要用 syms 命令创建符号变量和表达式,如:

```
>> syms x
```

syms 不仅可以声明一个变量,还可以指定这个变量的数学特性,比如:
声明变量 x, y 为实数类型,可用命令

```
>> syms x y real
```

声明变量 x, y 为整数类型,可用命令

```
>> syms x y positive
```

4. 变量的查询与清除

在命令窗口中,只要输入“who”,就可以看到工作空间中所有曾经设定并至今有效的变量。如果输入“whos”,不但会显示所有的变量,而且会将该变量的名称、性质等都显示出来,即显示变量的详细资料。输入“clear”,就可清除工作空间中的所有变量。如果输入“clear 变量名”,则清除工作空间中指定变量名的变量。

4.2.4 函数

MATLAB 系统提供了近 20 类基本命令函数,它们有一部分是 MATLAB 的内部命令,有一部分是以 M 文件形式出现的函数。MATLAB 中的命令函数可以通过在命令行中输入



>> Help fun

来获得有关这个命令函数的使用详细说明。式中 fun 为要查询的函数名称。表 4-5 列出了基本的数学函数。

表 4-5 基本的数学函数

函数名	中文解释	函数名	中文解释
$\sin(x)$	正弦函数	$\operatorname{asin}(x)$	反正弦函数
$\cos(x)$	余弦函数	$\operatorname{acos}(x)$	反余弦函数
$\tan(x)$	正切函数	$\operatorname{atan}(x)$	反正切函数
$\exp(x)$	以 e 为底的指数	$\log_{10}(x)$	以 10 为底数的对数
$\log(x)$	自然对数	$\operatorname{sqrt}(x)$	开平方
$\operatorname{abs}(x)$	绝对值或向量的长度	$\max(x)$	最大值
$\min(x)$	最小值	$\operatorname{sum}(x)$	元素求和
$\operatorname{sign}(x)$	符号函数	$\operatorname{round}(x)$	四舍五入到最近的整数
$\operatorname{ceil}(x)$	朝正无穷方向取整	$\operatorname{floor}(x)$	朝负无穷方向取整
$\operatorname{fix}(x)$	朝零方向取整	$\operatorname{gcd}(x,y)$	求两整数最大公约数

4.3 矩阵及其运算

矩阵是 MATLAB 数据存储的基本单元,矩阵的运算是 MATLAB 的核心,基于所有的运算都是以对矩阵的操作为基础的。矩阵的运算是按照一定的运算规则进行的,其规则是由运算符决定的。

4.3.1 操作符与运算符

1. 操作符

在编辑程序或命令中,当标点或其他符号表示特定的操作功能时就称其为操作符。表 4-6 列出了操作符。

表 4-6 操作符

操作符	使用说明
:	冒号。① $m:n$ 产生一个数组 $[m, m+1, \dots, n]$; ② $m:k:n$ 产生一个数组 $[m, m+k, \dots, n]$; ③ $A(:,j)$ 取矩阵 A 的第 j 列; ④ $A(k,:)$ 取矩阵 A 的第 k 行
;	分号。①在矩阵定义中表示一行的结束; ②在命令语句的结尾表示不显示这行语句的执行结果
...	连续点。一个命令语句非常长一行写不完,可以分几行写,此时在行的末尾加上连续点,表示这几行是一个命令语句
%	百分号。在编程时引导注释行,而系统解释执行程序时,%后面的内容不作处理

2. 运算符

运算符是构成运算的最基本的操作命令,可以在 MATLAB 的命令窗口直接运行。运算符分为算术运算符、关系运算符、逻辑运算符。不同的运算符及其功能说明如表 4-7、表 4-8、表 4-9 所示。

表 4-7 算术运算符

运算符	功能说明
+	加法运算。两个数相加或两个同阶矩阵相加。如果是一个矩阵和一个数字相加,则这个数字自动扩展为与矩阵同维的一个矩阵
-	减法运算。两个数相减或两个同阶矩阵相减
*	乘法运算。两个数相乘或两个可乘矩阵相乘
/	除法运算。两个数或两个可除矩阵相除(A/B 表示 A 乘以 B 的逆)
^	乘幂运算。数的方幂或一个方阵的多少次方
\	左除运算。两个数 $a \backslash b$ 表示 $b \div a$,两个可除矩阵相除($A \backslash B$ 表示 B 乘以 A 的逆)
.*	点乘运算。两个同阶矩阵对应元素相乘
./	点除运算。两个同阶矩阵对应元素相除
.^	点乘幂运算。一个矩阵中各个元素的多少次方
.\	点左除运算。两个同阶矩阵对应元素左除

表 4-8 关系运算符

运算符	功能说明	运算符	功能说明
>	判断大于关系	>=	判断大于等于关系
<	判断小于关系	<=	判断小于等于关系
==	判断等于关系	~=	判断不等于关系

表 4-9 逻辑运算符

逻辑运算符	功能说明	逻辑运算符	功能说明
&	与运算	~	非运算
	或运算	xor(a,b)	异或运算

逻辑运算符主要用于逻辑表达式和进行逻辑运算,参与运算的逻辑量以 0 代表“假”,以任意非 0 数代表“真”。逻辑表达式和逻辑函数的值以 0 表示“假”,以 1 表示“真”。

4.3.2 矩阵的输入与运算

1. 矩阵的输入

(1) 直接输入法

例如:

```
>> A = [2,3,5;1,3,5;6,9,4] % 同一行元素之间用空格或逗号,
                               行之间用分号或回车
```



A =

```
2 3 5
1 3 5
6 9 4
```

(2) 外部文件读入法

事先在记事本中编辑以下数据,保存为文件 data1.txt。

```
1 1 1
1 2 3
1 3 6
```

在 MATLAB 命令窗口中输入:

```
>> load data1.txt
>> data1           % 显示数据
```

data1 =

```
1 1 1
1 2 3
1 3 6
```

(3) 复制粘贴法

打开数据文件(如 WORD, EXCEL),复制文件中的数据,在命令窗口中输入矩阵名等于空的方括号(注意不要按回车键),如:

A = [];

将光标置于方括号内,右击选择粘贴,这样数据就输入了。

2. 特殊矩阵的建立

对于一些比较特殊的矩阵(如单位矩阵或元素中含 1 或 0 比较多的矩阵),可通过一些函数生成。MATLAB 提供了一些函数用于生成这些矩阵,如表 4-10 所示。

表 4-10 生成特殊矩阵的命令函数

命令函数	功能说明
a = []	生成空矩阵,当对一项操作无结果时,返回空矩阵,空矩阵的大小为零
b = zeros(m,n)	生成一个 m 行 n 列的零矩阵
c = ones(m,n)	生成一个 m 行 n 列的元素全为 1 的矩阵
d = eye(m,n)	生成一个 m 行 n 列的单位矩阵
rand(m)	生成 m 阶均匀分布的随机矩阵
randn(m)	生成 m 阶正态分布的随机矩阵

3. 矩阵中元素或块的操作

对矩阵中元素或块的常用操作如表 4-11 所示。

表 4-11 矩阵中元素或块的常用操作

表达式或命令函数	功能说明
$A(k,:)$	提取矩阵 A 的第 k 行
$A(:,k)$	提取矩阵 A 的第 k 列
$A(:)$	依次提取矩阵 A 的每一列,将 A 拉伸为一个列向量
$A(i1:i2, j1:j2)$	提取矩阵 A 的第 $i1 \sim i2$ 行、第 $j1 \sim j2$ 列,构成新矩阵
$A([a\ b\ c\ d],:)$	提取矩阵 A 指定的第 a, b, c, d 行,构成新矩阵
$A(:, [e\ f\ g\ h])$	提取矩阵 A 指定的第 e, f, g, h 列,构成新矩阵
$A(i2:-1:i1,:)$	以逆序提取矩阵 A 的第 $i1 \sim i2$ 行,构成新矩阵
$A(:, j2:-1:j1)$	以逆序提取矩阵 A 的第 $j1 \sim j2$ 列,构成新矩阵
$A(i1:i2,:) = []$	删除 A 的第 $i1 \sim i2$ 行,构成新矩阵
$A(:, j1:j2) = []$	删除 A 的第 $j1 \sim j2$ 列,构成新矩阵
$[A\ B]$ 或 $[A;B]$	将矩阵 A 和 B 拼接成新矩阵
$\text{diag}(A,k)$	抽取矩阵 A 的第 k 条对角线元素向量
$\text{tril}(A,k)$	抽取矩阵 A 的第 k 条对角线下面的部分
$\text{triu}(A,k)$	抽取矩阵 A 的第 k 条对角线上面的部分
$\text{flipud}(A)$	矩阵 A 进行上下翻转
$\text{fliplr}(A)$	矩阵 A 进行左右翻转
A'	矩阵 A 的转置
$\text{rot90}(A)$	矩阵 A 逆时针旋转 90°

4. 矩阵的运算

(1) 矩阵间的运算

矩阵间的运算见表 4-12。

表 4-12 矩阵间的运算

表达式	功能说明
$A + B$ ($A - B$)	A 与 B 为同型矩阵,对应元素相加(减)
$A * B$	A 的列数要等于 B 的行数,按代数学中定义的矩阵乘法法则计算
A/B	$X = A/B$ 是线性方程组 $XA = B$ 的解。当 A 是可逆的矩阵时, $A/B = A * B^{-1}$
$A \setminus B$	$X = A \setminus B$ 是线性方程组 $AX = B$ 的解。当 A 是可逆的矩阵时, $A \setminus B = A^{-1} * B$
$A . * B$	A 与 B 为同型矩阵,对应元素相乘
$A ./ B$	A 与 B 为同型矩阵,对应元素相除
$A.^B$	A 与 B 为同型矩阵, A 中元素对应 B 中元素乘方运算

(2) 矩阵与标量的运算

矩阵与标量的运算见表 4-13。

表 4-13 矩阵与标量的运算

表达式	功能说明(设 A 为矩阵, c 为标量)
$A + c$ ($A - c$)	A 中每个元素加(减)常数 c
$A * c$ ($c * A$)	A 中每个元素乘常数 c
A / c	A 中每个元素除常数 c
$c ./ A$	常数 c 分别被 A 中对应每个元素相除
$c.^A$	常数 c 与 A 中对应每个元素的乘方运算
$A.^c$	A 中每个元素对应常数的 c 次乘方运算
A^c	A 是方阵, 当 c 大于 0 时表示矩阵的方幂, 当 c 小于 0 时表示 A 逆的方幂

(3) 矩阵的基本函数运算

矩阵的函数运算是矩阵运算中最实用的部分, 常用的主要有以下几个, 见表 4-14 所示。

表 4-14 矩阵的函数运算命令

命令	功能	命令	功能
$\det(A)$	求矩阵 A 的行列式	$\text{rref}(A)$	求矩阵 A 的阶梯形的行最简形式
$\text{inv}(A)$	求方阵 A 的逆矩阵	$\text{rank}(A)$	求矩阵 A 的秩
$\text{size}(A)$	求矩阵 A 的阶数	$\text{trace}(A)$	求矩阵 A 的迹
$\text{eig}(A)$	求矩阵 A 的特征值及特征向量	$\text{orth}(A)$	将非奇异矩阵 A 正交规范化
$[Q, R] = \text{qr}(A)$	求正交矩阵 Q 和上三角阵 R 满足 $A = QR$		

(4) 矩阵的数据处理

MATLAB 具有强大的数据处理能力, 如数据的排序、求最大值、求和等。常用的数据处理命令见表 4-15 所示。

表 4-15 常用的数据处理命令

命令	功能	命令	功能
$\max(A)$	求向量或矩阵列的最大值	$\min(A)$	求向量或矩阵列的最小值
$\text{mean}(A)$	求向量或矩阵列的平均值	$\text{median}(A)$	求向量或矩阵列的中间值
$\text{sum}(A)$	求向量或矩阵列的元素和	$\text{prod}(A)$	求向量或矩阵列的元素积
$\text{var}(A)$	求向量或矩阵列的方差	$\text{std}(A)$	求向量或矩阵列的标准差
$\text{cov}(A)$	矩阵列向量之间的协方差矩阵	$\text{corrcoef}(A)$	矩阵列向量之间的相关系数矩阵
$\text{Length}(A)$	求向量所含元素个数	$\text{find}(A)$	求向量所满足条件的元素

4.4 M 文件与编程

4.4.1 M 文件编辑/调试器窗口 (Editor/Debugger)

在默认状态下, M 文件编辑/调试器窗口不随 MATLAB 界面的出现而启动, 只有需要编写 M 文件时, 才启动该窗口, 其窗口如图 4-14 所示。

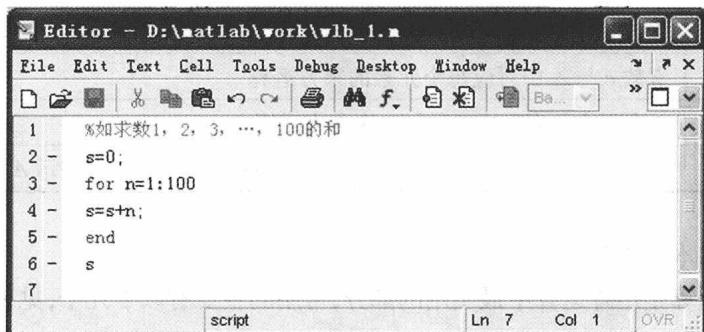


图 4-14 M 文件编辑/调试器窗口

4.4.2 M 文件

M 文件是由 MATLAB 语句(命令或函数)构成的 ASCII 码文本文件, 文件名必须以“.m”为扩展名。M 文件通过 M 文件编辑/调试器生成。在命令窗口调用 M 文件, 可实现一次执行多条 MATLAB 语句的功能。M 文件有以下两种形式。

1. 命令文件

命令文件是 MATLAB 命令或函数的组合, 没有输入输出参数, 执行命令文件只需在命令窗口中键入文件名按回车或在 M 文件编辑/调试器窗口激活状态下按 F5 键。

2. 函数文件

MATLAB 函数文件的组成:

文件第一行是函数定义行, 格式为

```
function [返回参数 1, 参数 2, ...] = 函数名(输入参数 1, 参数 2, ...)  
函数体
```

注意: 第一行的有无, 是区分命令文件与函数文件的重要标志; 函数体包含所有函数程序代码, 是函数的主体部分; 函数文件保存的文件名应与用户定义的函数名一致。

例如, 定义函数 $f(x, y) = x^3 + y^3 - 3xy$, 并计算 $f(2, 3)$ 。

在编辑器中写出如下程序:

```
function f = wlb_2(x, y)           % 函数名为 wlb_2, 返回值为 f  
f = x.^3 + y.^3 - 3*x.*y;         % 这是函数主体
```


保存为 wlb_2(这是文件名,与函数名一致),然后在命令窗口中执行

```
>> wlb_2(2,3)
ans = 17
```

4.5 基本绘图函数及其操作

4.5.1 引导

1. 离散数据和离散函数的可视化

【例 4-1】

```
n = (-10:10)';
y = abs(n);
plot(n,y,'r.','MarkerSize',20)
axis equal
grid on
xlabel('n')
```

例 4-1 中离散函数的图形如图 4-15 所示。

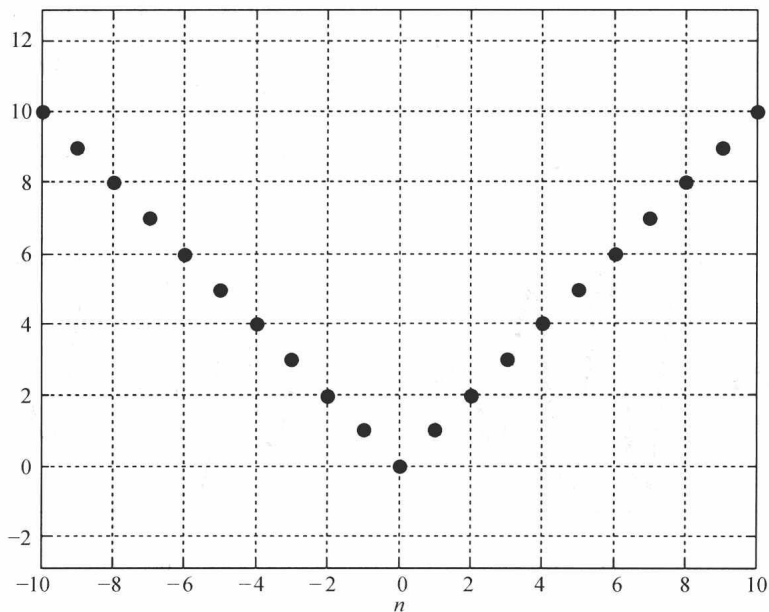


图 4-15 离散函数的图形

2. 连续函数的可视化

【例 4-2】

```

t1 = (0:11)/11*pi;
t2 = (0:400)/400*pi;
t3 = (0:50)/50*pi;
y1 = sin(t1). *sin(9*t1);
y2 = sin(t2). *sin(9*t2);
y3 = sin(t3). *sin(9*t3);
subplot(2,2,1), plot(t1, y1, 'r. ')
axis([0, pi, -1, 1]), title('点过少的离散图形')
subplot(2,2,2), plot(t1, y1, t1, y1, 'r. ')
axis([0, pi, -1, 1]), title('点过少的连续图形')
subplot(2,2,3), plot(t2, y2, 'r. ')
axis([0, pi, -1, 1]), title('点密集的离散图形')
subplot(2,2,4), plot(t3, y3)
axis([0, pi, -1, 1]), title('点足够的连续图形')

```

例 4-2 中连续函数的图形如图 4-16 所示。

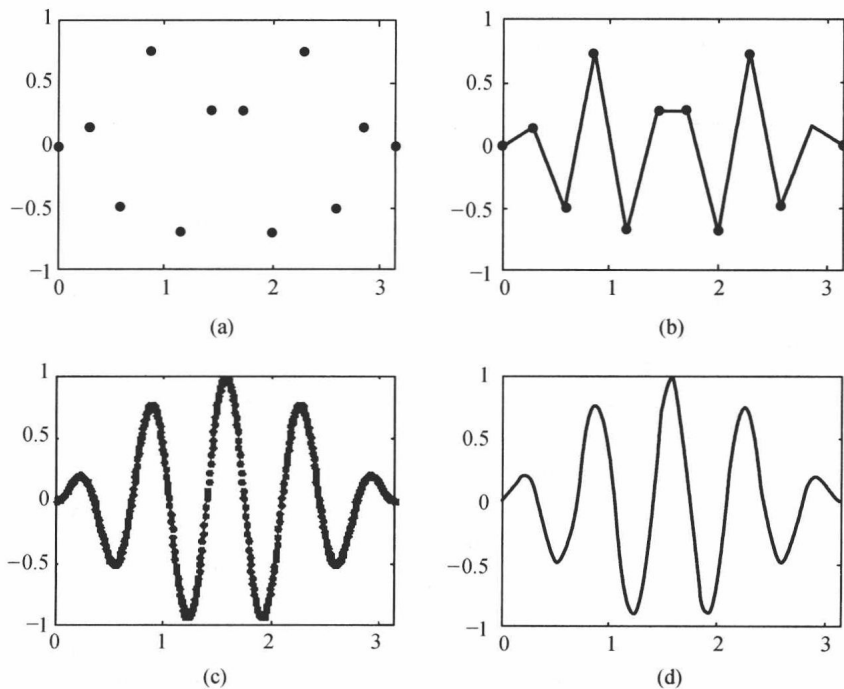


图 4-16 连续函数的图形

(a) 点过少的离散图形; (b) 点过少的连续图形;
(c) 点密集的离散图形; (d) 点足够的连续图形

【例 4-3】

```
N = 9;
t = 0:2*pi/N:2*pi;
x = sin(t); y = cos(t);
tt = reshape(t, 2, (N + 1)/2);
tt = flipud(tt);
tt = tt(:);
xx = sin(tt); yy = cos(tt);
subplot(1, 2, 1), plot(x, y)
title('正常排序图形'), axis equal off, shg
subplot(1, 2, 2), plot(xx, yy)
title('非正常排序图形'), axis equal off, shg
```

例 4-3 中自变量排列次序对连续曲线图形的影响如图 4-17 所示。

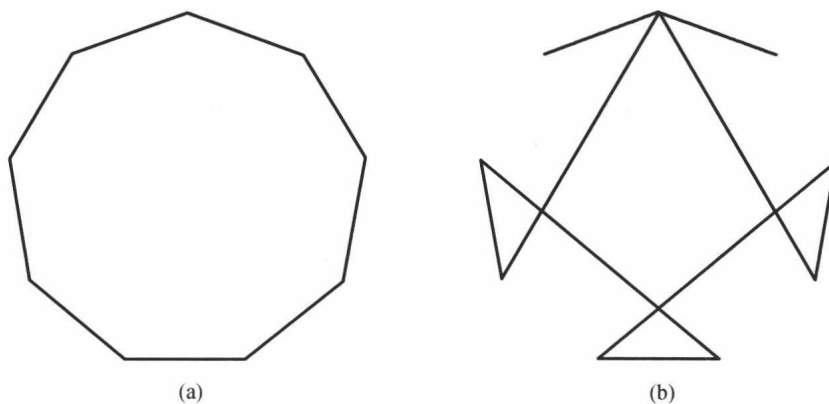


图 4-17 自变量排列次序对连续曲线图形的影响

(a) 正常排序图形; (b) 非正常排序图形

4.5.2 二维曲线和图形

1. 二维曲线绘制的基本指令 plot

【例 4-4】

```
clf
t = (0:pi/50:2*pi)';
k = 0.4:0.1:1;
Y = cos(t)*k;
subplot(1,2,1)
plot(t,Y,'LineWidth',1.5)
title('By plot(t,Y)')
xlabel('t')
subplot(1,2,2)
plot(Y,'LineWidth',1.5)
title('By plot(Y)')
xlabel('row subscript of Y')
```

例 4-4 中 $\text{plot}(t,Y)$ 与 $\text{plot}(Y)$ 所绘曲线的区别如图 4-18 所示。

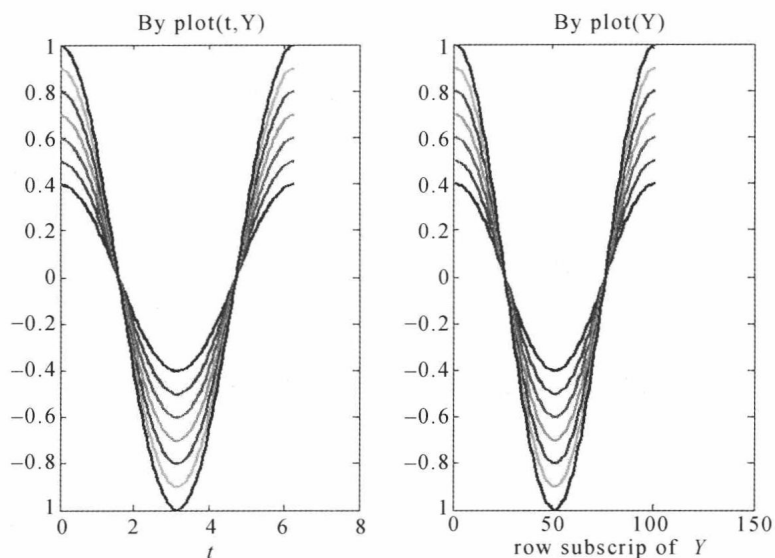


图 4-18 $\text{plot}(t,Y)$ 与 $\text{plot}(Y)$ 所绘曲线的区别

【例 4-5】

```
t = (0:pi/100:pi)';  
y1 = sin(t)*[1, -1];  
y2 = sin(t). *sin(9*t);  
t3 = pi*(0:9)/9;  
y3 = sin(t3). *sin(9*t3);  
plot(t,y1,'r:',t,y2,'-bo')  
hold on  
plot(t3,y3,'s','MarkerSize',10,'MarkerEdgeColor',[0,1,0],'MarkerFaceColor',[1,0.8,0])  
axis([0,pi,-1,1])  
hold off
```

例 4-5 中属性控制下所绘曲线如图 4-19 所示。

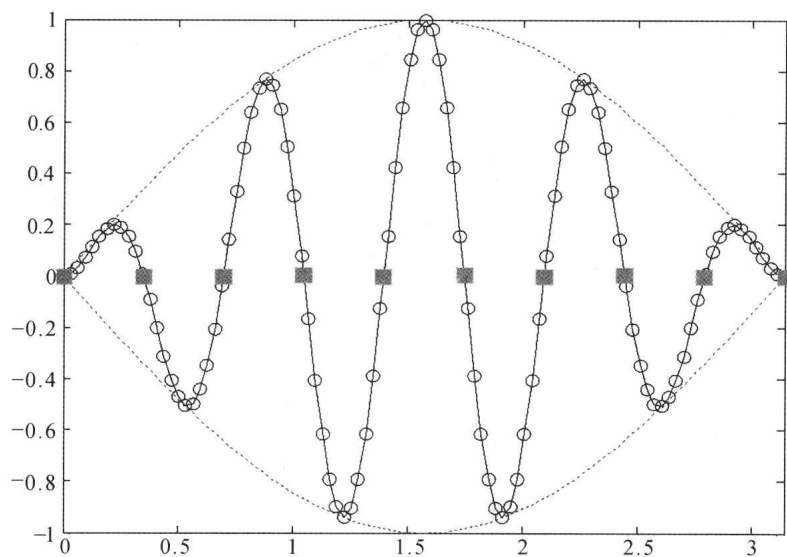


图 4-19 属性控制下所绘曲线

2. 坐标控制和图形标识

【例 4-6】

```

t = 0:2*pi/99:2*pi;
x = 1.15*cos(t); y = 3.25*sin(t);
subplot(2,3,1), plot(x,y), axis normal, grid on,
title('Normal and Grid on')
subplot(2,3,2), plot(x,y), axis equal, grid on, title('Equal')
subplot(2,3,3), plot(x,y), axis square, grid on, title('Square')
subplot(2,3,4), plot(x,y), axis image, box off, title('Image and Box off')
subplot(2,3,5), plot(x,y), axis image fill, box off
title('Image and Fill')
subplot(2,3,6), plot(x,y), axis tight, box off, title('Tight')

```

例 4-6 中各种轴控制指令的不同影响如图 4-20 所示。

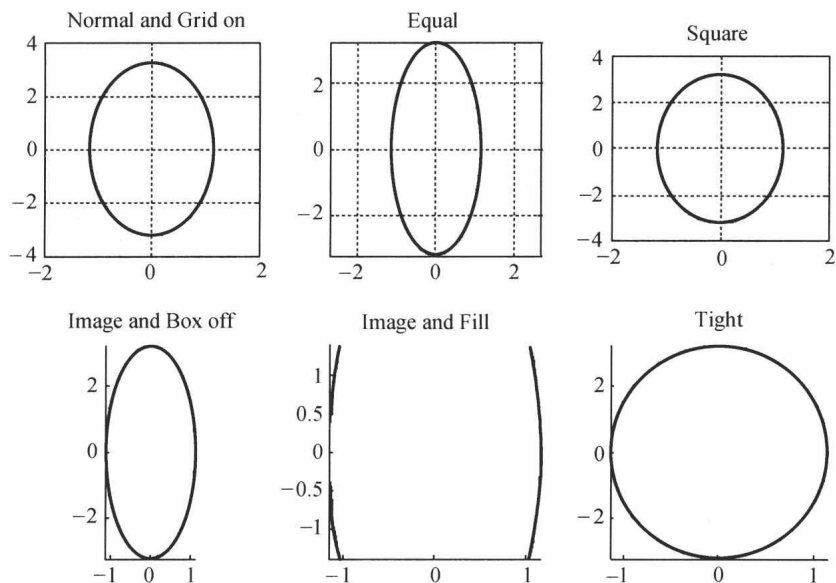


图 4-20 各种轴控制指令的不同影响

【例 4-7】

```
clf;t=0:pi/50:2*pi;  
y=sin(t);  
plot(t,y)  
axis([0,2*pi,-1.2,1.2])  
text(pi/2,1,'\fontsize{16}\leftarrow\itsin(t)\fontname{隶书}极大值')  
title('y=sin(t)')  
xlabel('t');    ylabel('y')
```

例 4-7 中实验标识的图形如图 4-21 所示。

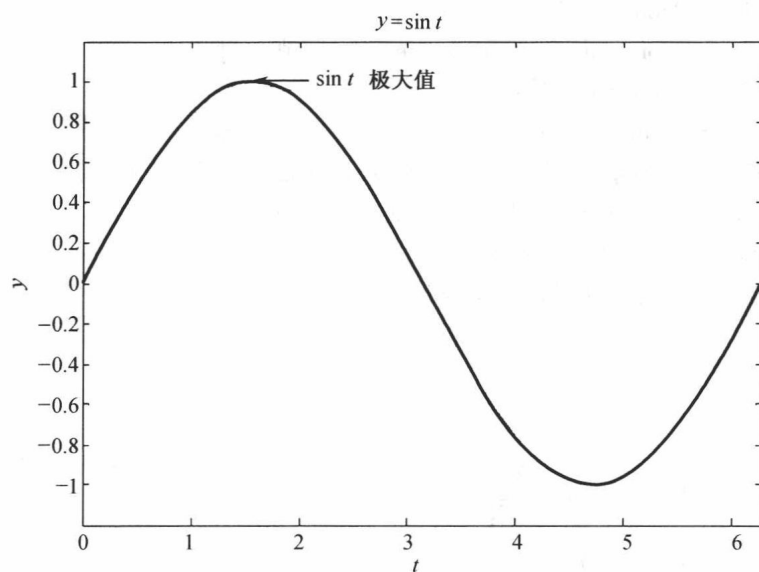


图 4-21 实验标识的图形

【例 4-8】

```

clf; t = 6*pi*(0:100)/100;
y = 1 - exp(-0.3*t) .* cos(0.7*t);
plot(t, y, 'r-', 'LineWidth', 3)
hold on
tt = t(find(abs(y-1) > 0.05)); ts = max(tt);
plot(ts, 0.95, 'bo', 'MarkerSize', 10)
hold off
axis([-inf, 6*pi, 0.6, inf])
set(gca, 'Xtick', [2*pi, 4*pi, 6*pi], 'Ytick', [0.95, 1, 1.05, max(y)])
set(gca, 'XtickLabel', {'2*pi'; '4*pi'; '6*pi'})
set(gca, 'YtickLabel', {'0.95'; '1'; '1.05'; 'max(y)'})
grid on
text(13.5, 1.2, '\fontsize{12} {\alpha} = 0.3')
text(13.5, 1.1, '\fontsize{12} {\omega} = 0.7')
cell_string{1} = '\fontsize{12} \uparrow';
cell_string{2} = '\fontsize{16} \fontname{隶书} 镇定时间';
cell_string{3} = '\fontsize{6} ';
cell_string{4} = ['\fontsize{14} \rmt_{s} = ' num2str(ts)];
text(ts, 0.85, cell_string, 'Color', 'b', 'HorizontalAlignment', 'Center')
title('\fontsize{14} \it y = 1 - e^{-\alpha t} \cos{\omega t}')
xlabel('\fontsize{14} \bft \rightarrow')
ylabel('\fontsize{14} \bfy \rightarrow')

```

例 4-8 中二阶阶跃响应图如图 4-22 所示。

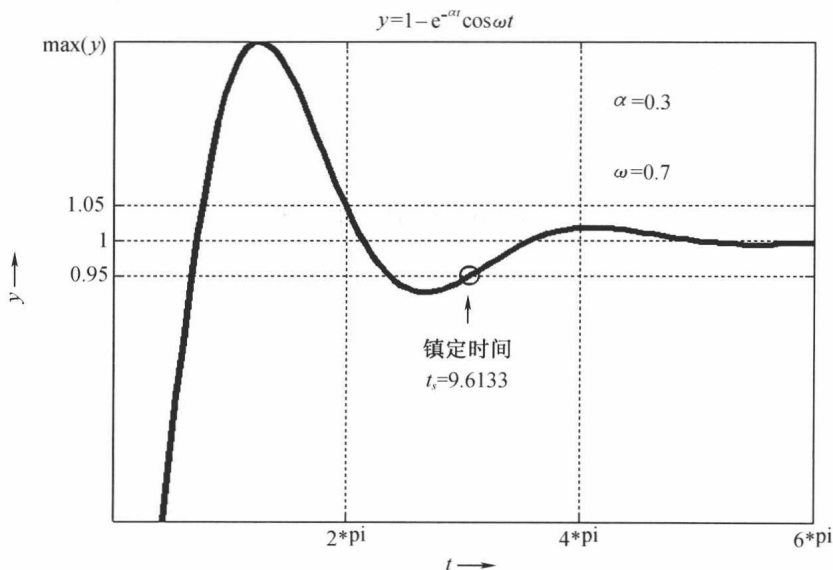


图 4-22 二阶阶跃响应图

3. 多次叠绘、双纵坐标和多子图

【例 4-9】

```
t = 2*pi*(0:20)/20;  
y = cos(t). *exp( -0.4*t);  
stem(t,y,'g','Color','k');  
hold on;  
stairs(t,y,':r','LineWidth',3)  
hold off;  
legend(' \fontsize{14} \it stem', ' \fontsize{14} \it stairs')  
box on
```

例 4-9 中离散信号的重构图如图 4-23 所示。

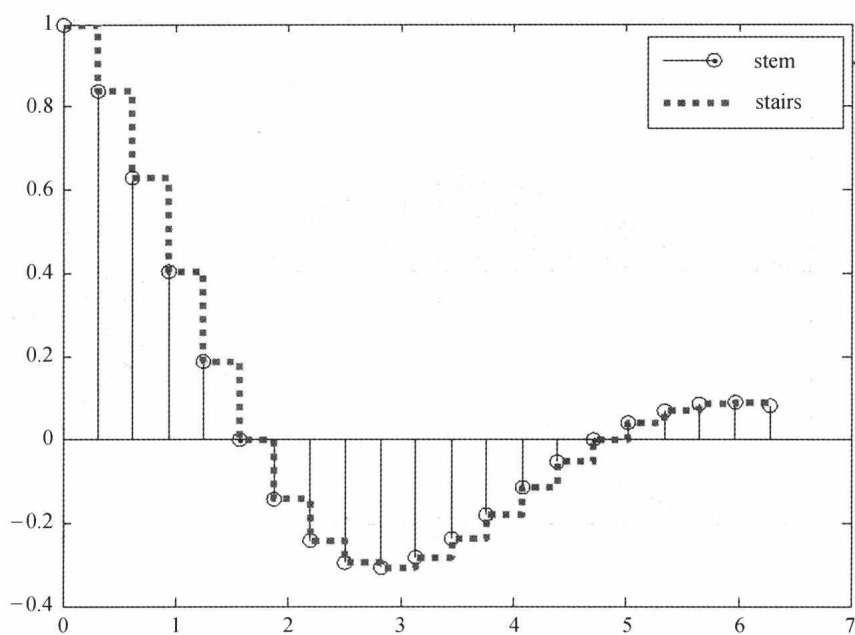


图 4-23 离散信号的重构图

【例 4-10】

```

clf;dx=0.1;    x=0:dx:4;y=x.*sin(x);
s=cumtrapz(y)*dx;    a=plotyy(x,y,x,s,'stem','plot');
text(0.5,1.5,'\fontsize{14}\ity=xsinx')
sint='\fontsize{16}\int_{\fontsize{8}0}^{\quad x}\quad';
ss=['\fontsize{14}\its=',sint,'\fontsize{14}\itxsinxdx'];
text(2.5,3.5,ss);
set(get(a(1),'Ylabel'),'String','被积函数 \ity=xsinx')
set(get(a(2),'Ylabel'),'String',ss);    xlabel('x')

```

例 4-10 中函数和积分图如图 4-24 所示。

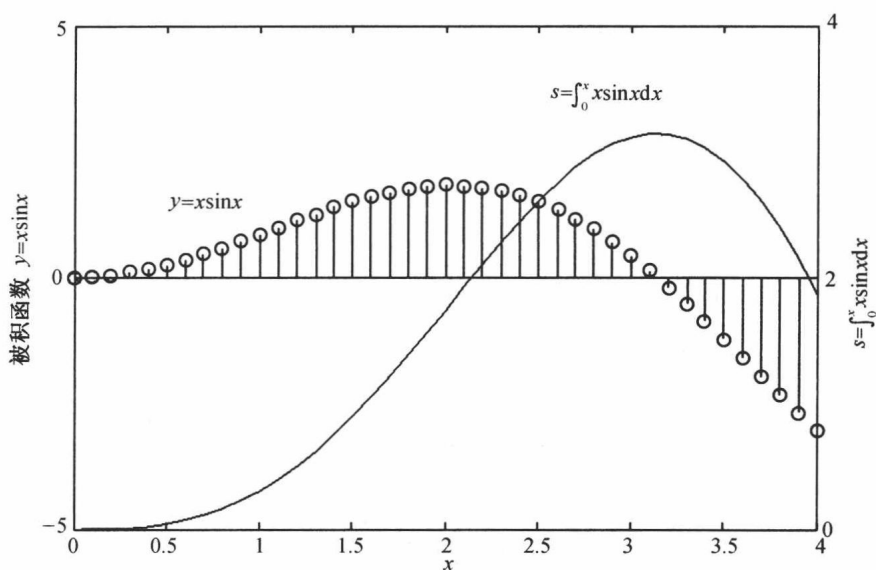


图 4-24 函数和积分图

【例 4-11】

```

clf;t = (pi*(0:1000)/1000)';
y1 = sin(t);y2 = sin(10*t);y12 = sin(t).*sin(10*t);
subplot(2,2,1),plot(t,y1);axis([0,pi,-1,1])
subplot(2,2,2),plot(t,y2);axis([0,pi,-1,1])
subplot('position',[0.2,0.1,0.6,0.4])
plot(t,y12,'b- ',t,[y1,-y1],'r:')
axis([0,pi,-1,1])

```

例 4-11 中多子图的布置图如图 4-25 所示。

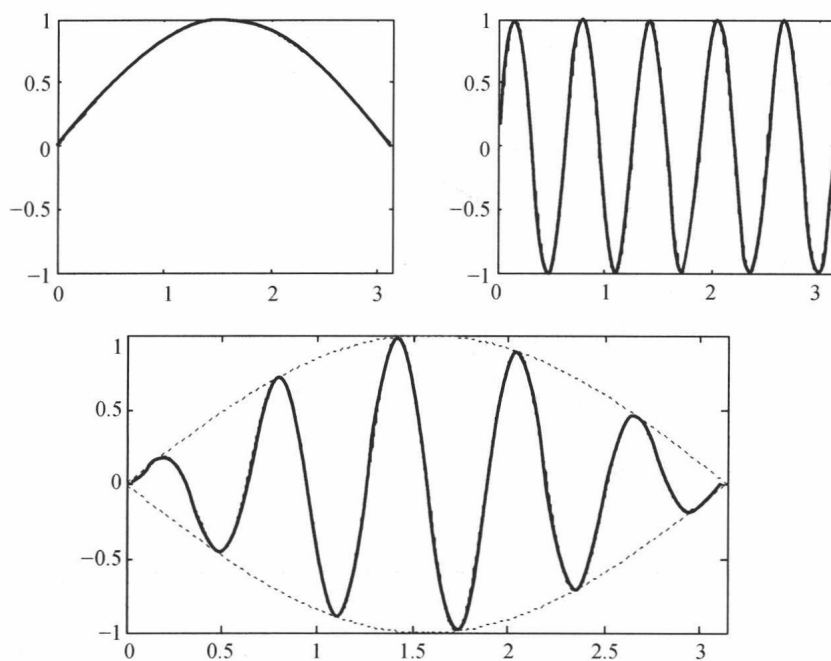


图 4-25 多子图的布置图

4. 获取二维图形数据的指令 ginput

【例 4-12】

```
clf  
x=0:0.01:1;  
y=(x+2).^x-2;  
plot(x,y);  
grid on
```

例 4-12 中经多次试探后画出的图形如图 4-26 所示。

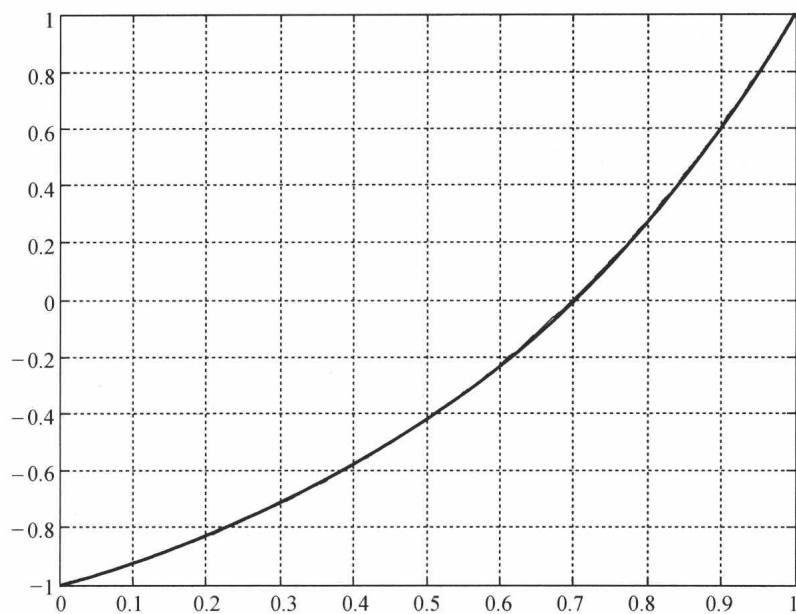


图 4-26 经多次试探后画出的图形

例 4-12 的局部放大图如图 4-27 所示。

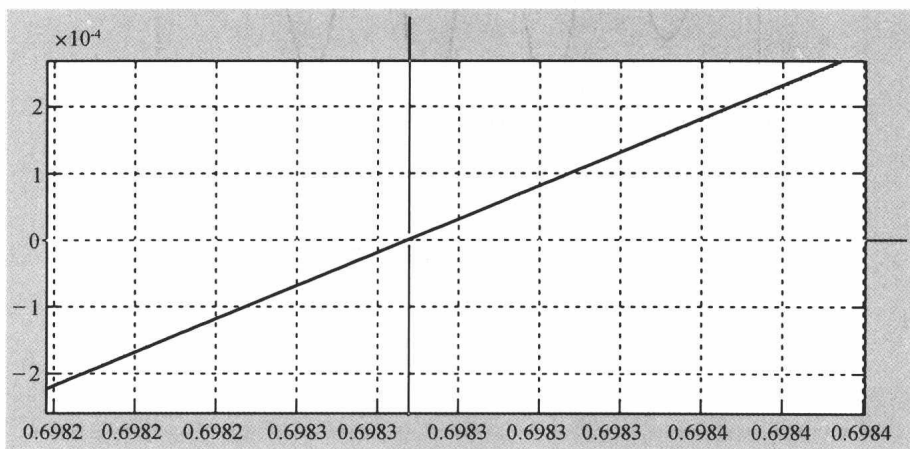


图 4-27 局部放大图


```
[x,y] = ginput(1);
```

```
format long
```

```
x,y
```

```
x =
```

```
0.698 286 929 035 37
```

```
y =
```

```
-5.884 401 711 382 421e-010
```

4.5.3 三维曲线和曲面

1. 三维线图指令 plot3

【例 4-13】

```
t = (0:0.02:2)*pi;
```

```
x = sin(t); y = cos(t); z = cos(2*t);
```

```
plot3(x,y,z,'b-','x,y,z','bd')
```

```
view([-82,58]), box on
```

```
xlabel('x'), ylabel('y'), zlabel('z')
```

```
legend('链','宝石','Location','best')
```

例 4-13 中宝石项链图如图 4-28 所示。

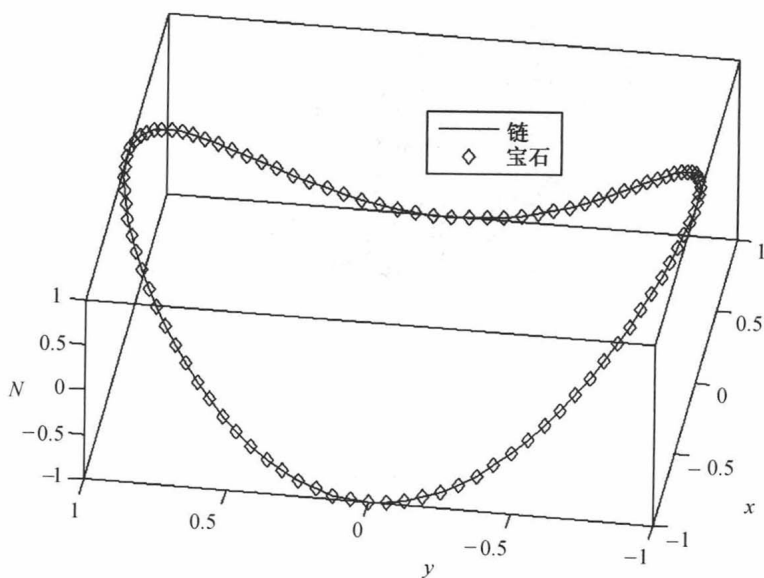


图 4-28 宝石项链图

2. 三维曲面/网线图

【例 4-14】

```

clf;      x = -4:4;  y = x;
[X,Y] = meshgrid(x,y);    Z = X.^2 + Y.^2;
surf(X,Y,Z);
colormap(hot)
hold on
stem3(X,Y,Z,'bo')
hold off
xlabel('x'),ylabel('y'),zlabel('z')
axis([-5,5,-5,5,0,inf])
view([-84,21])

```

例 4-14 中的曲面图和格点如图 4-29 所示。

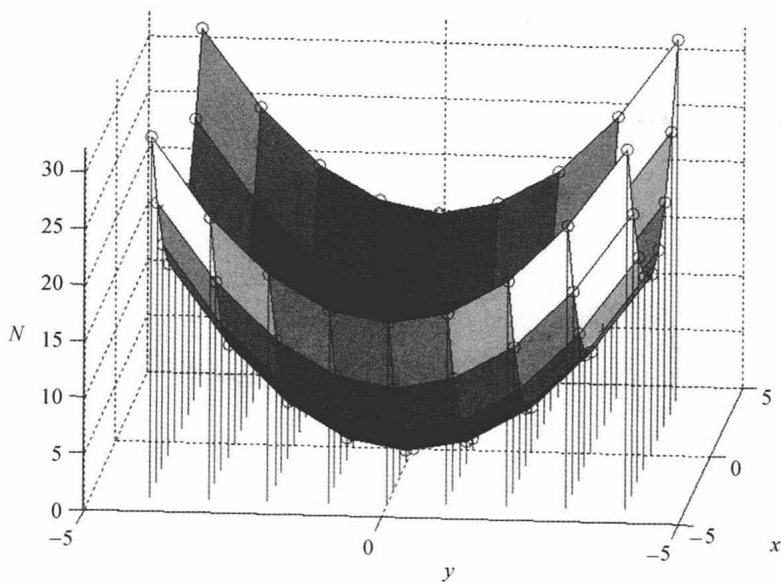


图 4-29 曲面图和格点

3. 曲面/网线图的精细修饰

【例 4-15】

```
clf;      x = -4:4;y = x;  
[X,Y] = meshgrid(x,y);    Z = X.^2 + Y.^2;  
surf(X,Y,Z);    colormap(jet)  
subplot(1,3,1),surf(Z),axis off  
subplot(1,3,2),surf(Z),axis off,shading flat  
subplot(1,3,3),surf(Z),axis off,shading interp  
set(gcf,'Color','w')
```

例 4-15 中三种浓淡处理方式的效果比较如图 4-30 所示。

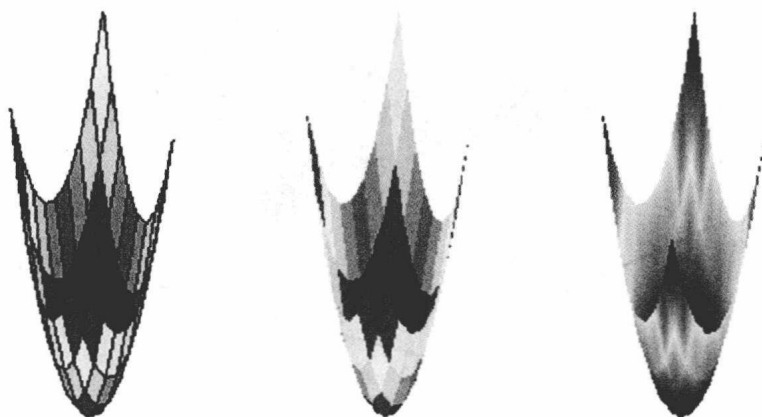


图 4-30 三种浓淡处理方式 faceted/flat/interp 的效果比较图

【例 4-16】

```
clf
surf(peaks)
shading interp
alpha(0.5)
colormap(summer)
```

例 4-16 中半透明薄膜图如图 4-31 所示。

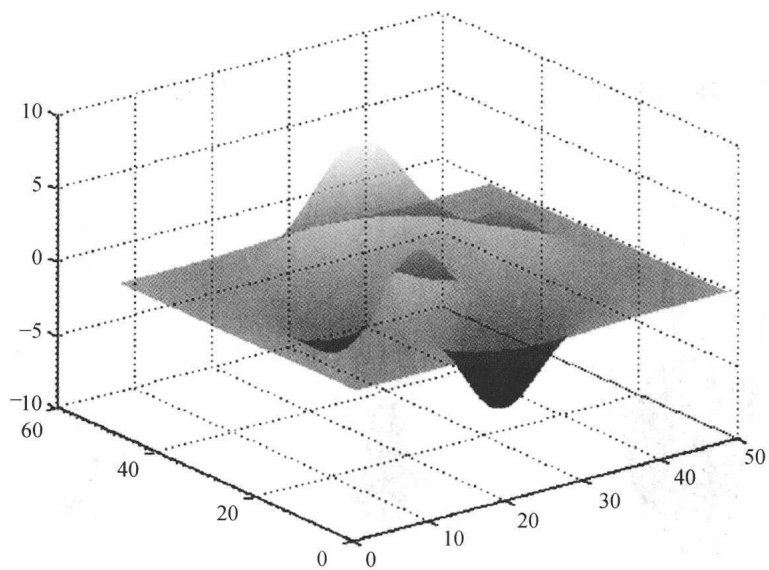


图 4-31 半透明薄膜图

【例 4 - 17】

```
clf;      [X,Y,Z] = sphere(40);  
colormap(jet)  
subplot(1,2,1),surf(X,Y,Z),axis equal off,shading interp  
light('position',[0 -10 1.5],'style','infinite')  
lighting phong  
material shiny  
subplot(1,2,2),surf(X,Y,Z,-Z),axis equal off,shading flat  
lighting flat  
light('position',[-1,-1,-2],'color','y')  
light('position',[-1,0.5,1],'style','local','color','w')  
set(gcf,'Color','w')
```

例 4 - 17 中灯光、照明、材质指令所表现的图形如图 4 - 32 所示。

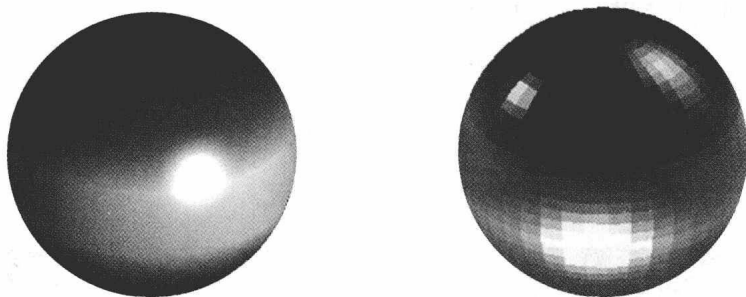


图 4 - 32 灯光、照明、材质指令所表现的图形

4.5.4 高维可视化

1. 二维半图指令 pcolor, contour, contourf

【例 4-18】

```
clf;clear;
[X,Y,Z] = peaks(40);
n = 6;
subplot(1,2,1),
pcolor(X,Y,Z)
shading interp
zmax = max(max(Z));
zmin = min(min(Z));
caxis([zmin,zmax])
colorbar
hold on
C = contour(X,Y,Z,n,'k:');
clabel(C)
hold off
subplot(1,2,2)
[C,h] = contourf(X,Y,Z,n,'k:');
clabel(C,h)
colormap(cool)
set(gcf,'Color','w')
```

例 4-18 中“二维半”指令的演示如图 4-33 所示。

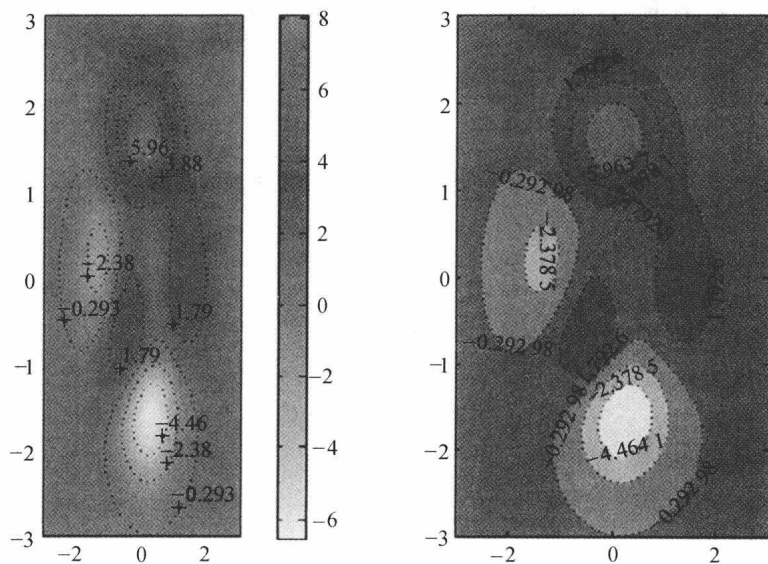


图 4-33 “二维半”指令的演示

2. 四维表现

【例 4-19】

```

clf
x = 3*pi * ( -1:1/15:1 ); y = x; [ X, Y ] = meshgrid( x, y );
R = sqrt( X.^2 + Y.^2 ) + eps; Z = sin( R ) ./ R;
[ dzdx, dzdy ] = gradient( Z );
dzdr = sqrt( dzdx.^2 + dzdy.^2 );
dz2 = del2( Z );
subplot( 1,2,1 ), surf( X, Y, Z, abs( dzdr ) )
shading faceted;
colorbar( 'SouthOutside' )
brighten( 0.6 );
colormap hsv
title( 'No. 1      surf( X, Y, Z, abs( dzdr ) )' )
subplot( 1,2,2 ); surf( X, Y, Z, abs( dz2 ) )
shading faceted
colorbar( 'NorthOutside' )
title( 'No. 2      surf( X, Y, Z, abs( dz2 ) )' )

```

例 4-19 中色彩分别表现函数的径向导数和曲率特征的图如图 4-34 所示。

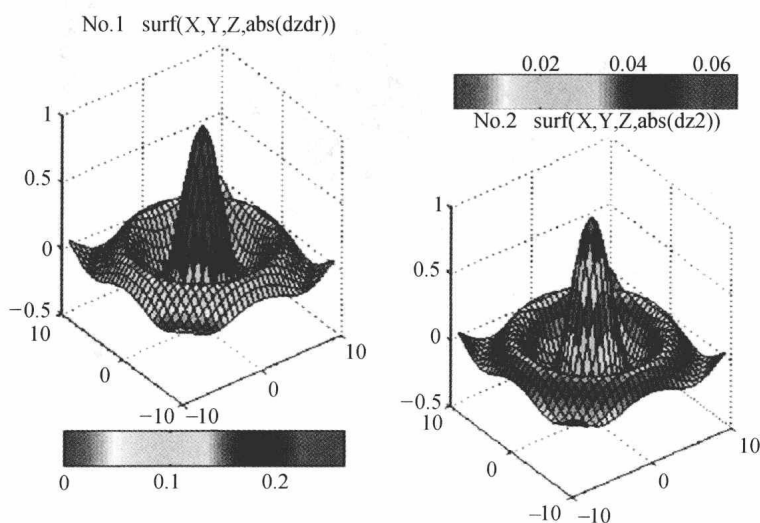


图 4-34 色彩分别表现函数的径向导数和曲率特征的图

【例 4-20】

```

clf
[x,y,z] = meshgrid(-2:.2:2,-2:.25:2,-2:.16:2);
v = x.*exp(-x.^2-y.^2-z.^2);
xs = [-0.7,0.7]; ys = 0; zs = 0;
slice(x,y,z,v,xs,ys,zs)
colorbar
shading interp
colormap hsv
xlabel('x'),ylabel('y'),zlabel('z')
title('The color-to-v(x,y,z) mapping')
view([-22,39])
alpha(0.3)

```

例 4-20 的切片图如图 4-35 所示。

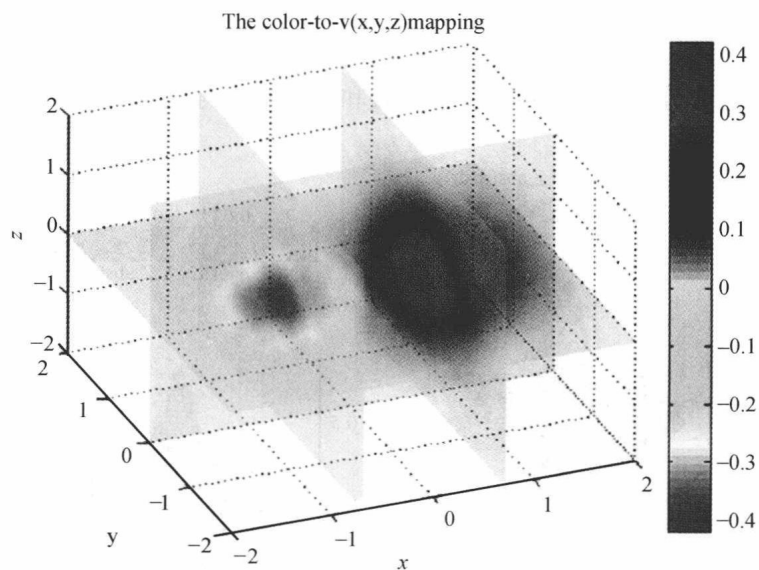


图 4-35 切片图

第 5 章 神经网络模型及仿真

5.1 人工神经网络基本理论

5.1.1 生物神经元

人脑大约由 1 000 多亿个神经元组成,神经元互相连接成神经网络。神经元是大脑处理信息的基本单元,以细胞体为主体,由许多向周围延伸的不规则树枝状纤维构成的神经细胞,其形状很像一棵枯树的枝干。它主要由细胞体、树突、轴突和突触(Synapse,又称神经键)组成,如图 5-1 所示。

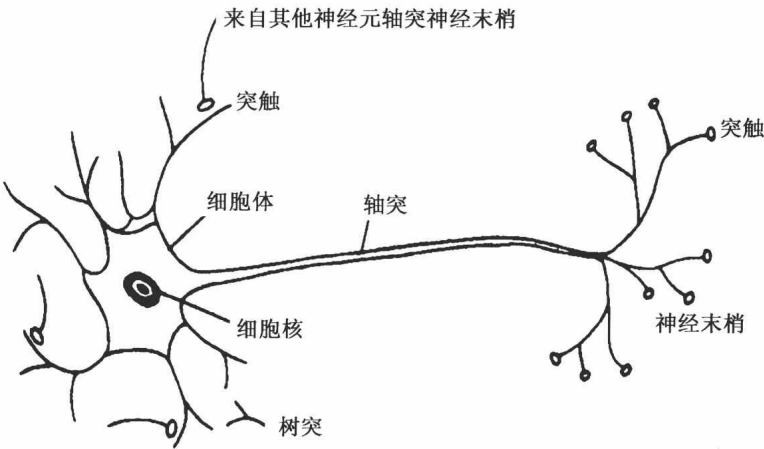


图 5-1 生物神经元示意图

从神经元各组成部分的功能来看,生物神经元中信息的处理与传递主要发生在突触附近。当神经元细胞体通过轴突传到突触前膜的脉冲幅度达到一定强度,即超过其阈值电位后,突触前膜将向突触间隙释放神经元传递的化学物质。

5.1.2 人工神经网络基本概念

人工神经网络(Artificial Neutral Network, ANN)是由大量简单的基本单元——神经元相互连接,通过模拟人的大脑神经处理信息的方式,进行信息并行处理和非线性转换的复杂网络系统。人工神经元模型如图 5-2 所示。

1. 人工神经元模型三个基本构成要素

(1) 连接权重 w_i

(2) 求和单元 $\sum (u = \sum_{i=1}^n w_i x_i, v = \sum_{i=1}^n w_i x_i - \theta)$

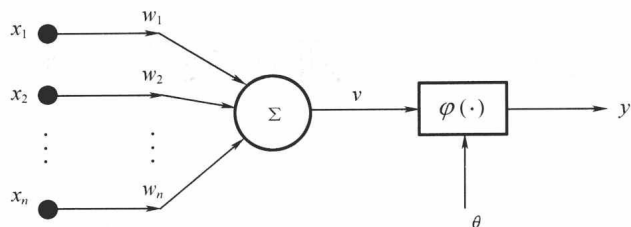


图 5-2 人工神经元模型示意图

(3) 激励函数(或传递函数) $\varphi(\cdot)$

2. 人工神经元模型的数学表达式

模型中 $x_1, x_2, \dots, x_n$ 为输入信号, $w_1, w_2, \dots, w_n$ 为连接权重, y 为输出信号, θ 为神经元的阈值, 则人工神经元模型的数学表达式为

$$u = \sum_{i=1}^n w_i x_i \quad (5-1)$$

$$v = \sum_{i=1}^n w_i x_i - \theta \quad (5-2)$$

$$y = \varphi(v) \quad (5-3)$$

如果将神经元的阈值视为一种特殊连接权重, 则人工神经元模型的数学表达式还可以表示为

$$\begin{cases} x_0 = \pm 1 \\ w_0 = \mp \theta \\ v = \sum_{i=0}^n w_i x_i \\ y = \varphi(v) \end{cases} \quad (5-4)$$

3. 常用的激励函数

激励函数又称为激活函数或传递函数, 常用的激励函数有三种形态。

(1) 阈值型(一般只用于简单分类的 MP 模型中)

$$\varphi(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (5-5)$$

(2) 线性型(一般只用于输入神经元和输出神经元)

$$\varphi(x) = x \quad (5-6)$$

(3) S 型(常用于隐含层神经元)

$$\varphi(x) = \frac{1}{1 + e^{-x}} \quad \text{或者} \quad \varphi(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (5-7)$$

通常还会在机理函数的变量中耦合一个常数以便调整激励网络的输出幅度。例如 S 型激励函数可以改写为

$$\varphi(x) = \frac{1}{1 + e^{-x}} \Rightarrow \varphi(x) = \frac{1}{1 + e^{-cx}} \quad (5-8)$$

其中, c 为常数。

5.1.3 人工神经网络结构

单个神经元对信息的处理非常有限,只有将多个神经元连接起来,构成一个神经网络体系,才能对复杂的信息进行识别处理。人工神经网络模型拓扑结构可分为前馈网络和反馈网络。

1. 前馈网络

前馈网络(Feedforward Network)的信息流由输入层逐渐向下传递,没有反馈,信息流经网络处理后由输出层输出。单层前馈网络是最简单的前馈式网络,只包含输入层和输出层;对于多层前馈网络,输入层和输出层之间加一个中间层,称为隐含层,多层前馈神经网络的结构如图 5-3 所示。

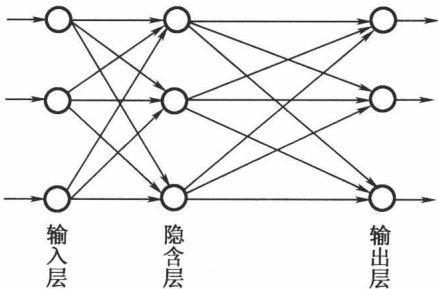


图 5-3 多层前馈神经网络

2. 反馈网络

反馈网络(Recurrent Network)是指某一层通过连接重新作为输入反馈到同一层或者前一层,反馈网络的结构如图 5-4 所示。

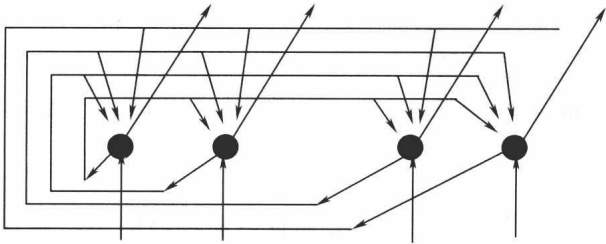


图 5-4 多层反馈神经网络

5.1.4 人工神经网络的学习

人工神经网络的学习也称为训练,目的是对网络的权值进行调整,使得在应用一个输入时,网络能产生一个期望的输出矢量。

1. 学习方式

学习方式可分为有导师学习(Learning With A Teacher)和无导师学习(Learning Without A Teacher)。

(1) 有导师学习

有导师学习又称有监督学习(Supervised Learning),在学习时需要给出导师信号(或称

为期望输出)。神经网络对外部环境是未知的,但可以将导师看作对外部环境的了解,由输入-输出样本集合来表示,如图 5-5 所示。导师信号或期望响应代表了神经网络执行情况的最佳结果,即对于网络输入调整网络参数,使得网络输出逼近导师信号或期望响应。

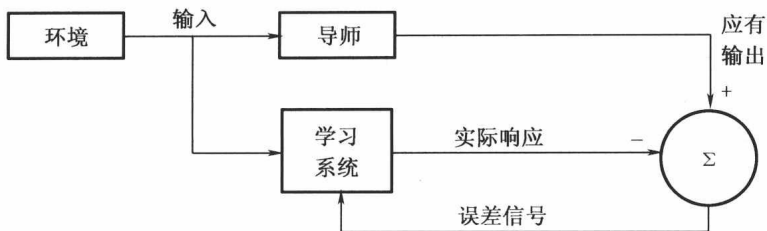


图 5-5 有导师学习示意图

(2) 无导师学习

无导师学习的样本模式中不包含期望输出,网络完全按照样本所提供的某些规律来调整网络自身,以获得输入信息的某些固有特征。无导师学习包括无监督学习(Unsupervised Learning)或称为自组织学习(Self-organized Learning)与强化学习(Reinforcement Learning),其中无监督学习中学习系统按照环境提供数据的某些统计规律来调节自身参数,如图 5-6 所示;强化学习中环境对系统输出结果只给出评价信息(奖或惩),系统通过强化受奖动作来改善自身性能,如图 5-7 所示。



图 5-6 无监督学习示意图

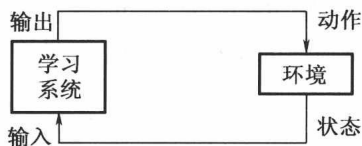


图 5-7 强化学习示意图

2. 学习规则

一个神经网络要具有任何智能特征,必须有一套完整的学习规则与之配合,学习规则就是修正神经元之间连接强度或加权系数的算法,使获得的知识结构能适应周围环境的变化。

(1) Hebb 规则

1949 年,加拿大心理学家 Donald. O. Hebb 提出了最早的关于神经网络学习机理的“突触修正”假设。该假设提出,当神经元的突触前膜电位与突触后膜电位同时为正时,突触传导增强;当前膜电位与后膜电位正负电位相反时,突触传导减弱。也就是说,当神经元 i 与神经元 j 同时处于兴奋状态时,两者之间的连接强度应增强。根据该假设定义的权值调整方法,称为 Hebb 学习规则。

在 Hebb 学习规则中,学习信号简单地等于神经元的输入,即

$$y_j = f \left(\sum_{i=1}^n w_{ij} x_i \right) \quad (5-9)$$

权向量的调整公式为

$$w_{ij}(t+1) = w_{ij}(t) + \eta y_j x_i \quad (5-10)$$

式中, η 为学习速度, x_i 为节点 i 的输入, y_j 为节点 j 对应的输出。

上式表明, 权向量调整量与输入输出的乘积成正比。显然, 经常出现的输入模式将对权向量有着较大的影响。在这种情况下, Hebb 学习规则需预先设置权饱和值, 以防止输入和输出正负始终一致时出现权值无约束的增长。此外, 要求权值初始化, 即在学习开始 ($t=0$) 时, 先对权值赋予零附近的小随机数。

Hebb 学习规则是一种纯前馈、无导师的学习, 该学习规则至今在各种神经网络模型中起着重要的作用, 是许多学习规则的假设基础。

(2) 离散感知器学习规则

1958 年, 美国计算机科学家 Frank Rosenblatt 首次定义了一个具有单纯计算单元的神经网络结构, 称为感知器 (Perceptron)。感知器的学习规则权值修正关系为

$$w_{ij}(t+1) = w_{ij}(t) + \eta(d_j - y_j)x_i \quad (5-11)$$

或者

$$\Delta w_{ij} = w_{ij}(t+1) - w_{ij}(t) = \eta(d_j - y_j)x_i \quad (5-12)$$

对于离散感知器, 有

$$y_j = f(\mathbf{W}_j^T \mathbf{X} - \theta) = \text{sgn}(\mathbf{W}_j^T \mathbf{X} - \theta) = \begin{cases} 1, & \mathbf{W}_j^T \mathbf{X} - \theta \geq 0 \\ -1, & \mathbf{W}_j^T \mathbf{X} - \theta < 0 \end{cases} \quad (5-13)$$

式中, η 为调整步幅系数, $\eta > 0$, 离散感知器学习规则只适合用于二进制神经元, 初始权值可取任意值。

(3) 连续感知器学习规则

与离散感知器相对应的是连续感知器学习规则, 也称为 δ 规则。其权值向量调整为

$$w_{ij}(t+1) = w_{ij}(t) + \eta(d_j - y_j)f'(\mathbf{W}_j^T \mathbf{X})x_i \quad (5-14)$$

或者

$$\Delta w_{ij} = w_{ij}(t+1) - w_{ij}(t) = \eta(d_j - y_j)f'(\mathbf{W}_j^T \mathbf{X})x_i \quad (5-15)$$

对于 δ 规则, 其中 $y_i = f(\mathbf{W}_j^T \mathbf{X})$, $f'(\mathbf{W}_j^T \mathbf{X})$ 是功能函数 $f(\mathbf{W}_j^T \mathbf{X})$ 的导数, 要求功能函数可导, 它只适用于有监督学习算法。

(4) 最小均方差学习规则

1962 年, Bernard Widrow 和他的学生 Marcian Hoff 提出 Widrow-Hoff 学习规则, 其权值修正关系为

$$w_{ij}(t+1) = w_{ij}(t) + \eta(d_j - \mathbf{W}_j^T \mathbf{X})x_i \quad (5-16)$$

或者

$$\Delta w_{ij} = w_{ij}(t+1) - w_{ij}(t) = \eta(d_j - \mathbf{W}_j^T \mathbf{X})x_i \quad (5-17)$$

式中, η 为学习速度, 与算法的稳定性有关, 从数学上可以证明二次函数的最大稳定学习速度为 $\eta < \frac{1}{\lambda_{\max}}$, 其中 $\lambda_{\max}$ 是有关输入向量的矩阵特征值。

(5) 相关学习规则

相关学习规则规定学习信号为 $r = d_j$, 易得出

$$\Delta \mathbf{W}_j = \eta d_j \mathbf{X} \quad (5-18)$$

$$\Delta w_{ij} = \eta d_j x_i, \quad i = 0, 1, \dots, n \quad (5-19)$$

该规则表明, 当 d_j 是 x_i 的期望输出时, 相应的权值增量 Δw_{ij} 与两者的乘积 $d_j x_i$ 成正比。

当相关学习规则中的变换函数为二进制函数, $o_j = d_j$ 时, 该学习规则可以看成是 Hebb 规则的一种特殊情况。应当注意的是, Hebb 学习规则是无导师学习, 而相关学习规则是有导师学习, 相关学习规则要求权值初始化为零。

(6) 胜者为王学习规则

对于无导师学习的情况, 事先不给定标准样本, 直接将网络置于“应用环境”中, 学习(训练)阶段与应用阶段成为一体。此时, 学习规律服从连接权值 w 的演变方程, 即

$$\frac{dw}{dt} = f[w, x] \quad (5-20)$$

初始值 w_0 选定之后, 由于环境不断改变, x, w 随之逐渐改变, 对于平稳环境, w 可达到稳定状态。如果环境发生变化, w 也随之改变。这种边学习边工作的特征与人脑学习过程更近似。胜者为王学习规则, 一般将网络的某一层定义为竞争层, 对于一个待定的输入 X , 竞争层的所有 P 个神经元均有输出响应, 其中响应值最大的神经元为竞争中获胜的神经元, 即

$$W_m^T X = \max_{i=1,2,\dots,p} (W_i^T X) \quad (5-21)$$

只有获胜神经元才有权调整其权值向量 W_m , 调整量为

$$\Delta W_m = \alpha(X - W_m) \quad (5-22)$$

式中, $\alpha \in (0, 1]$, 是学习常数, 一般其值随着学习的进展而减小。由于两个向量的点积越大, 表明两种向量越相似, 所以调整获胜神经元权值的结果是使 W_m 进一步接近当前输入 X 。显然, 当下次出现与 X 相似的输入模式时, 上次获胜的神经元更易获胜。在反复竞争学习过程中, 竞争层中的神经元所对应的权向量被逐渐调整为输入样本空间的聚类中心。

5.2 BP 神经网络算法

BP (Back Propagation) 网络是 1986 年由 Rumelhart 和 McClelland 为首的科学家小组提出的, 是一种按误差逆传播算法训练的多层前馈网络, 是目前应用最广泛的神经网络模型之一。BP 网络能学习和存贮大量的输入-输出模式映射关系, 而无需事前揭示描述这种映射关系的数学方程。它的学习规则是使用最速下降法, 通过反向传播来不断调整网络的权值和阈值, 使网络的误差平方和最小。

BP 神经网络模型拓扑结构包括输入层、隐含层(中间层)和输出层。

BP 算法由数据流的前向计算(正向传播)和误差信号的反向传播两个过程构成。正向传播时, 传播方向为输入层→隐含层→输出层, 每层神经元的状态只影响下一层神经元。若在输出层得不到期望的输出, 则转向误差信号的反向传播流程。通过这两个过程的交替进行, 在权向量空间执行误差函数梯度下降策略, 动态迭代搜索一组权向量, 使网络误差函数达到最小值, 从而完成信息提取和记忆过程。

1. 正向传播

设 BP 网络的输入层有 n 个节点, 隐含层有 q 个节点, 输出层有 m 个节点; 输入的一个样本 $X = [x_1, x_2, \dots, x_n]'$, 期望输出 $Y = [y_1, y_2, \dots, y_m]'$; 输入层与隐含层之间的权值为 v_{ik} ($i=1, 2, \dots, n; k=1, 2, \dots, q$); 隐含层与输出层之间的权值为 w_{kj} ($k=1, 2, \dots, q; j=1, 2, \dots, m$); 隐含层节点的阈值为 $\theta_k^{(1)}$ ($k=1, 2, \dots, q$), 输出层节点的阈值为 $\theta_j^{(2)}$ ($j=1, 2, \dots, m$); 隐含层



的传递函数为 $f_1(\cdot)$, 输出层的传递函数为 $f_2(\cdot)$ 。

隐含层节点的输出为

$$O_k = f_1\left(\sum_{i=1}^n v_{ik}x_i - \theta_k^{(1)}\right) = f_1(\text{net}_k^{(1)}), \quad k = 1, 2, \dots, q \quad (5-23)$$

其中, $\text{net}_k^{(1)} = \sum_{i=1}^n v_{ik}x_i - \theta_k^{(1)}, k = 1, 2, \dots, q$ 。

输出层节点的输出为

$$z_j = f_2\left(\sum_{k=1}^q w_{kj}O_k - \theta_j^{(2)}\right) = f_2(\text{net}_j^{(2)}), \quad j = 1, 2, \dots, m \quad (5-24)$$

其中, $\text{net}_j^{(2)} = \sum_{k=1}^q w_{kj}O_k - \theta_j^{(2)}, j = 1, 2, \dots, m$ 。

2. 反向传播

网络输出与期望输出的误差为

$$\begin{aligned} E &= \frac{1}{2} \sum_{j=1}^m (y_j - z_j)^2 = \frac{1}{2} \sum_{j=1}^m [y_j - f_2(\sum_{k=1}^q w_{kj}O_k - \theta_j^{(2)})]^2 \\ &= \frac{1}{2} \sum_{j=1}^m \{y_j - f_2[\sum_{k=1}^q w_{kj}f_1(\sum_{i=1}^n v_{ik}x_i - \theta_k^{(1)}) - \theta_j^{(2)}]\}^2 \end{aligned} \quad (5-25)$$

误差 E 对隐含层与输出层间的权值 w_{kj} 的偏导数为

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_{kj}} = -(y_j - z_j) f'_2(\text{net}_j^{(2)}) O_k = -\delta_j^{(2)} O_k \quad (5-26)$$

其中, $\delta_j^{(2)} = (y_j - z_j) f'_2(\text{net}_j^{(2)})$ 。

误差 E 对输入层与隐含层间的权值 v_{ik} 的偏导数为

$$\begin{aligned} \frac{\partial E}{\partial v_{ik}} &= \sum_{j=1}^m \sum_{k=1}^q \frac{\partial E}{\partial z_j} \cdot \frac{\partial z_j}{\partial O_k} \cdot \frac{\partial O_k}{\partial v_{ik}} \\ &= - \sum_{j=1}^m (y_j - z_j) f'_2(\text{net}_j^{(2)}) w_{kj} f'_1(\text{net}_k^{(1)}) x_i \\ &= -\delta_k^{(1)} x_i \end{aligned} \quad (5-27)$$

其中, $\delta_k^{(1)} = \sum_{j=1}^m (y_j - z_j) f'_2(\text{net}_j^{(2)}) w_{kj} f'_1(\text{net}_k^{(1)}) = \sum_{j=1}^m \delta_j^{(2)} w_{kj} f'_1(\text{net}_k^{(1)})$ 。

由此可得权值的调整公式为

$$\begin{cases} v_{ik}(t+1) = v_{ik}(t) + \Delta v_{ik} = v_{ik}(t) - \eta^{(1)} \frac{\partial E}{\partial v_{ik}} = v_{ik}(t) + \eta^{(1)} \delta_k^{(1)} x_i \\ w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj} = w_{kj} - \eta^{(2)} \frac{\partial E}{\partial w_{kj}} = w_{kj} + \eta^{(2)} \delta_j^{(2)} O_k \end{cases} \quad (5-28)$$

其中, $\eta^{(1)}, \eta^{(2)}$ 分别为隐含层、输出层的学习步长。

同理, 误差 E 对输出层节点的阈值 $\theta_j^{(2)}$ 的偏导数为

$$\begin{aligned} \frac{\partial E}{\partial \theta_j^{(2)}} &= \frac{\partial E}{\partial z_j} \cdot \frac{\partial z_j}{\partial \theta_j^{(2)}} = -(y_j - z_j) f'_2(\text{net}_j^{(2)}) (-1) \\ &= (y_j - z_j) f'_2(\text{net}_j^{(2)}) = \delta_j^{(2)} \end{aligned} \quad (5-29)$$

误差 E 对隐含层节点的阈值 $\theta_k^{(1)}$ 的偏导数为

$$\begin{aligned}
 \frac{\partial E}{\partial \theta_k^{(1)}} &= \sum_{j=1}^m \frac{\partial E}{\partial z_j} \cdot \frac{\partial z_j}{\partial O_k} \cdot \frac{\partial O_k}{\partial \theta_k^{(1)}} = - \sum_{j=1}^m (y_j - z_j) f'_2(\text{net}_j^{(2)}) w_{kj} f'_1(\text{net}_k^{(1)}) (-1) \\
 &= \sum_{j=1}^m (y_j - z_j) f'_2(\text{net}_j^{(2)}) w_{kj} f'_1(\text{net}_k^{(1)}) = \delta_k^{(1)}
 \end{aligned} \quad (5-30)$$

由此可得阈值的调整公式为

$$\begin{cases} \theta_k^{(1)}(t+1) = \theta_k^{(1)}(t) + \Delta \theta_k^{(1)} = \theta_k^{(1)}(t) + \eta^{(1)} \frac{\partial E}{\partial \theta_k^{(1)}} = \theta_k^{(1)}(t) + \eta^{(1)} \delta_k^{(1)} \\ \theta_j^{(2)}(t+1) = \theta_j^{(2)}(t) + \Delta \theta_j^{(2)} = \theta_j^{(2)}(t) + \eta^{(2)} \frac{\partial E}{\partial \theta_j^{(2)}} = \theta_j^{(2)}(t) + \eta^{(2)} \delta_j^{(2)} \end{cases} \quad (5-31)$$

5.3 BP 神经网络的 MATLAB 工具箱

MathWorks 公司为了让用户使用方便,特意开发了神经网络工具箱。工具箱的使用的确非常便捷,节省了用户编辑的时间。MATLAB 神经网络工具箱包含许多用于 BP 网络分析与设计的函数,这里主要介绍以下几类函数:

- ①前向网络创建函数 newcf, newff 和 newfftd。
- ②激励函数 logsig, dlogsig, tansig, dtansig, purelin, dpurelin。
- ③学习函数 learnbd, learnbmd。
- ④性能函数 mse, msereg。

5.3.1 BP 网络创建函数

在介绍这类函数之前,先给出在 BP 网络创建函数中可能用得到的变量及其含义。

PR:由每组输入元素的最大值和最小值组成的 $R \times 2$ 的矩阵。

Si:第 i 层网络的神经元个数,共计 N 层。

TFi:第 i 层的激励函数,默认为“tansig”。

BTF:网络的训练函数,默认为“trainlm”。

BLF:权值与阈值的学习算法,默认为“learnbmd”。

PF:网络的性能函数,默认为“mse”。

1. 函数 newcf

这个函数用于创建级联前向 BP 网络,其使用格式为

```
net = newcf
net = newcf(PR, [S1 S2 ...SN], {TF1 TF2...TFN}, BTF, BLF, PF)
```

其中,net = newcf 用于在对话框中创建一个 BP 网络。

2. 函数 newff

这个函数用于创建一个 BP 网络,其使用格式为

```
net = newff
net = newff(PR, [S1 S2 ...SN], {TF1 TF2...TFN}, BTF, BLF, PF)
```

其中,net = newff 用于在对话框中创建一个 BP 网络。



3. 函数 newfftd

这个函数用于创建一个存在输入延迟的前向网络,其使用格式为

```
net = newfftd
net = newfftd( PR,[ S1 S2 ...SN ],{ TF1 TF2...TFN },BTF,BLF,PF)
```

其中,net = newfftd 用于在对话框中创建一个 BP 网络。

5.3.2 神经元激励函数

激励函数是 BP 神经网络的重要组成部分,它必须是连续可微的。BP 网络经常采用 S 型的对数或者正切函数和线性函数。

1. 函数 logsig

激励函数 logsig 为 S 型的对数函数,其使用格式为

```
A = logsig( N)
info = logsig( code)
```

其中,N 为 Q 个 S 维的输入列向量;A 为函数返回值,位于区间(0,1)中;info = logsig(code) 根据 code 值不同返回不同的信息。code 取值及其对应的返回信息如表 5 - 1 所示。

表 5 - 1 code 取值及其对应的返回信息

code 的取值	返回的信息	code 的取值	返回的信息
‘deriv’	微分函数的名称	‘output’	输出值域
‘name’	函数全称	‘active’	有效的输入区间

该函数使用的算法为 $\text{logsig}(n) = 1/[1 + \exp(-n)]$ 。

下面给出一个简单的例子:

```
n = -8:0.1:8;
a = logsig( n );
plot( n,a );
grid on;
```

运行结果如图 5 - 8 所示,可以看出函数 logsig 将输入映射到(0,1)区间。

2. 函数 dlogsig

函数 dlogsig 为 logsig 的导函数,其使用格式为

```
dA_dN = dlogsig( N,A)
```

其中,N 为 $S \times Q$ 维网络输入;A 为 $S \times Q$ 维网络输出;dA_dN 为函数的返回值,输出对输入的导数。该函数使用的算法为 $d = a(1 - a)$ 。

3. 函数 tansig

函数 tansig 为双曲正切 S 型激励函数,其使用格式为

```
A = tansig( N)
info = tansig( code)
```

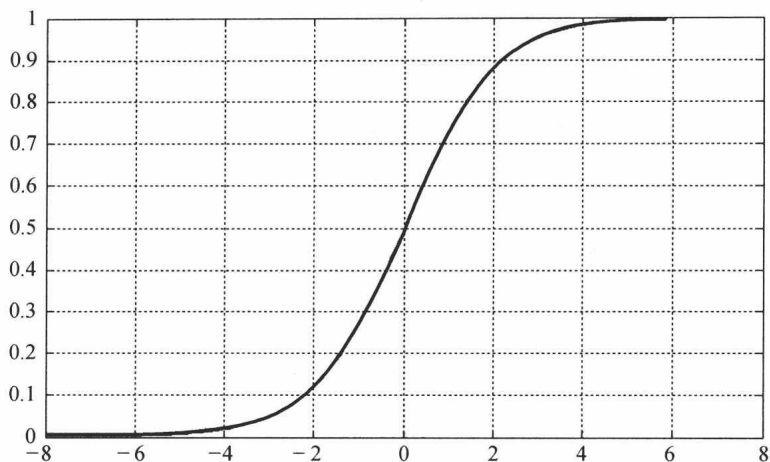


图 5-8 函数 logsig 实例运行结果

其中, N 为 Q 个 S 维的输入列向量; A 为函数返回值, 位于区间 $(-1, 1)$ 中; $\text{info} = \text{tansig}(\text{code})$ 根据 code 值不同返回不同的信息(同函数 logsig)。该函数使用的算法为

$$\text{tansig}(n) = \frac{2}{[1 + \exp(-2n)] - 1} \quad (5-32)$$

以下是一个绘制双曲正切 S 型激励函数的例子:

```
n = -5:0.1:5;
a = tansig(n);
plot(n,a);
grid on;
```

运行结果如图 5-9 所示, 可以看出函数 tansig 将输入映射到 $(-1, 1)$ 区间。

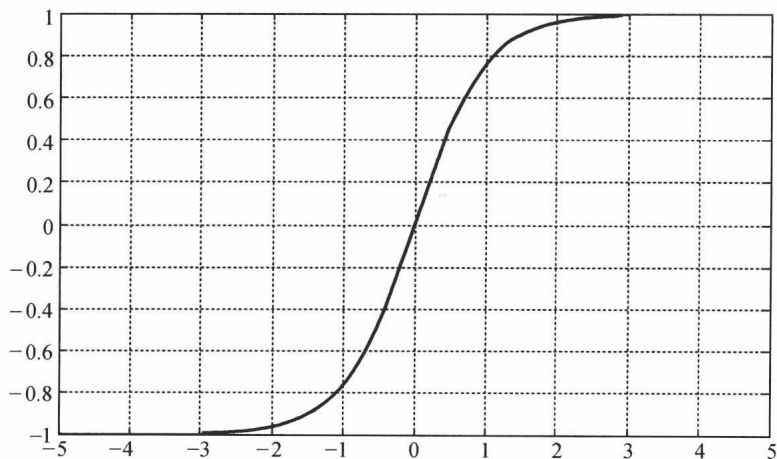


图 5-9 函数 tansig 实例运行结果



由图 5-8 和图 5-9 可以看出, BP 网络的输出量为 $0 \sim 1$ 或 $-1 \sim 1$ 之间的连续量。如果在网络学习(训练)过程中, 实际输出样本值远远超出 $0 \sim 1$ 或 $-1 \sim 1$ 区间, 则需要对训练样本进行数据预处理, 否则 BP 神经网络将无法收敛或学习速度慢甚至无效。

4. 函数 dtansig

函数 dtansig 是 tansig 的导函数, 其使用格式为

```
dA_dN = dtansig(N, A)
```

其中, 参数的含义与函数 dlogsig 相同。该函数使用的算法为 $d = 1 - a^2$ 。

5. 函数 purelin

函数 purelin 为线性激励函数, 其使用格式为

```
A = purelin(N)
```

```
info = purelin(code)
```

其中, N 为 Q 个 S 维的输入列向量; A 为函数返回值, $A = N$; $\text{info} = \text{tansig}(\text{code})$ 根据 code 值不同返回不同的信息(同函数 logsig)。

该函数使用的算法为 $\text{purelin}(n) = n$ 。

以下是一个绘制线性 S 型传递函数的例子:

```
n = -6:0.1:6;
a = purelin(n);
plot(n,a);
grid on;
```

运行结果如图 5-10 所示。

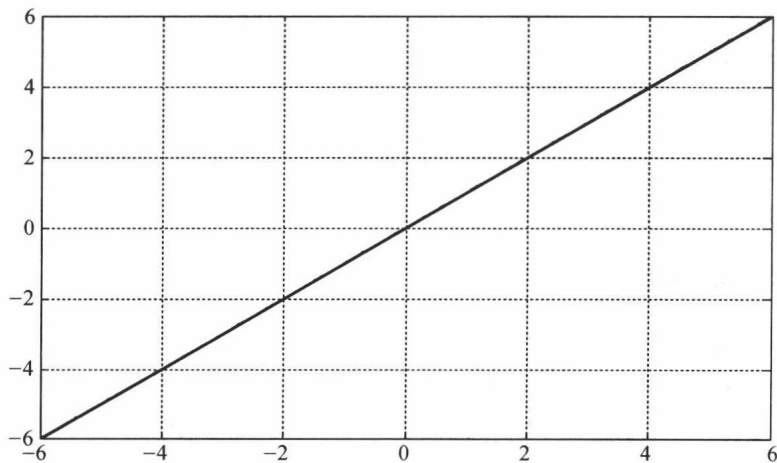


图 5-10 函数 purelin 实例运行结果示意图

6. 函数 dpurelin

函数 dpurelin 是 purelin 的导函数, 其使用格式为

$$dA_dN = dpurelin(N, A)$$

其中,参数的含义与函数 `dlogsig` 的相同,该函数使用的算法为 $D(i, q) = 1$ 。

5.3.3 BP 网络学习函数

1. 函数 `learngd`

函数 `learngd` 是梯度下降权值/阈值学习函数,它通过神经元的输入和误差,以及权值和阈值的学习速率计算权值或阈值的变化率。其使用格式为

```
[dW, LS] = learngd(W, P, Z, N, A, T, E, gW, gA, D, LP, LS)
[db, LS] = learngd(b, ones(1, Q), Z, N, A, T, E, gW, gA, D, LP, LS)
info = learngd(code)
```

其中, W 为 $S \times R$ 的权值矩阵; b 为 S 维的阈值向量; P 为 Q 组 R 维的输入向量; `ones(1, Q)` 为产生一个 Q 维的输入向量; Z 为 Q 组 S 维的加权输入向量; N, A 分别为 Q 组 S 维的输入、输出向量; T, E 分别为 Q 组 S 维的层目标向量、层误差向量; gW, gA 分别为与性能相关的 $S \times R$ 的维梯度、维输出梯度; D 为 $S \times S$ 维的神经元距离矩阵; LP 为学习参数,通过其设置学习速率,设置格式为 `LP.lr = 0.001`; LS 为学习状态,初始状态下为空; dW 为 $S \times R$ 维的权值或阈值变化率矩阵; db 为 S 维的阈值变化率向量; `learngd(code)` 依据不同的 `code` 值返回有关函数的不同信息,具体有 ‘`pnames`’——返回设置的学习参数, ‘`pdefaults`’——返回默认的学习参数, ‘`needg`’——如果函数使用了 gW 或者 gA , 则返回 1。

2. 函数 `learngdm`

函数 `learngdm` 为梯度下降动量学习函数,它利用神经元的输入和误差、权值或阈值的学习速率和动量常数计算权值或阈值的变化率,其使用格式为

```
[dW, LS] = learngdm(W, P, Z, N, A, T, E, gW, gA, D, LP, LS)
[db, LS] = learngdm(b, ones(1, Q), Z, N, A, T, E, gW, gA, D, LP, LS)
info = learngdm(code)
```

其中,各参数的含义与函数 `learngd` 相同。

动量常数 `mc` 是通过学习参数 `LP` 设置的,格式为 `LP.mc = 0.8`。

5.3.4 BP 网络训练函数

1. 函数 `trainbfg`

函数 `trainbfg` 为 BFGS 准牛顿 BP 算法函数。除 BP 网络外,该函数也可以训练任意形式的神经网络,要求它的激励函数对于权值和输入存在导数,其使用格式为

```
[net, TR, Ac, El] = trainbfg(NET, Pd, Tl, Ai, Q, TS, VV, TV)
info = trainbfg(code)
```

其中, NET 为待训练的神经网络; Pd 为有延迟的输入向量; Tl 为层次目标向量; Ai 为初始的延迟条件; Q 为批量; TS 为时间步长; VV 用于确认向量结构或者为空; TV 用于检验向量结构



或者为空; *net* 为训练后的神经网络; *TR* 为每步训练的有信息记录, 包括 *TR. epoch*——时刻点, *TR. perf*——训练性能, *TR. vperf*——确认性能, *TR. tperf*——检验性能; *Ac* 为上一步训练中聚合层的输出; *E1* 为上一步训练中的层次误差; *info = trainbfg (code)* 依据不同的 *code* 值返回不同的信息, 具体有 ‘*pnames*’——返回设置的训练参数, ‘*pdefaults*’——返回默认的训练参数。

在利用该函数进行 BP 网络训练时, MATLAB 已经默认了一些训练参数, 具体如表 5-2 所列。

表 5-2 BP 网络默认训练参数

参数名称	功能	缺省值
<i>net. trainParam. goal</i>	目标函数设定值	0
<i>net. trainParam. epochs</i>	最大迭代次数	100
<i>net. trainParam. show</i>	显示中间结果的周期	25
<i>net. trainParam. time</i>	训练时间, <i>inf</i> 表示训练时间不限	<i>inf</i>
<i>net. trainParam. lr</i>	整批学习的学习效率	0.01
<i>net. trainParam. mc</i>	势能学习规则 <i>traingdm</i> 的势能率	0.9
<i>net. trainParam. min_grad</i>	最小性能梯度	$1e-6$
<i>net. trainParam. max_fail</i>	最大确认失败次数	5
<i>net. trainParam. searchFcn</i>	所用的线性搜索路径	‘ <i>srchcha</i> ’

2. 函数 *traingd*

函数 *traingd* 是梯度下降 BP 算法训练函数, 其使用格式为

```
[net, TR, Ac, E1] = traingd( NET, Pd, Tl, Ai, Q, TS, VV, TV)
info = traingb( code)
```

以上程序中各参数含义、设置格式和使用范围与函数 *trainbfg* 相同。

3. 函数 *traingdm*

函数 *traingdm* 是梯度下降动量 BP 算法训练函数, 其使用格式为

```
[net, TR, Ac, E1] = traingdm( NET, Pd, Tl, Ai, Q, TS, VV, TV)
info = traingdm( code)
```

以上程序中各参数含义、设置格式和使用范围与函数 *trainbfg* 相同。

除了上述三种训练函数外, 还有一系列可用于 BP 网络训练的函数, 可以利用 MATLAB 的帮助系统了解这些函数的功能, 这里只列出函数名, 不再对其功能进行介绍。其他训练函数有 *trainbr*, *trainc*, *traingb*, *traingf*, *traingp*, *traingda*, *traingdx*, *trainlm*, *trainoss*, *trainr*, *trainrp*, *trains*, *trainseg*。只要这些函数对于权值和阈值存在导数, 它们也适用于其他任何神经网络。

5.3.5 性能函数

1. 函数 mse

函数 mse 为均方差性能函数,其使用格式为

```
perf = mse(e,x,pp)
perf = mse(e,net,pp)
info = mse(code)
```

其中, e 为误差向量矩阵; x 为所有的权值和阈值向量,可忽略; pp 为性能参数,可忽略; net 为待评定的神经网络; $perf$ 为函数的返回值,为平均绝对误差; $mse(code)$ 为根据 $code$ 值的不同,返回不同的信息,包括‘deriv’——返回导数函数的名称,‘name’——返回函数全称,‘pname’——返回训练函数的名称,‘pdefault’——返回默认的训练函数。

为了对该函数的用法有直观的认识,下面给出了一个该函数的应用实例,并将函数 mse 的用途作为注释插在程序中间:

```
clc; clear;    % 清屏,清除内存
% 构建网络结构
net = newff([-10 10],[4 1],{'logsig','purelin'});
p = [-10 -5 0 5 10];
t = [0 0 1 1 1];
y = sim(net,p);
e = t - y
% 对于 newff 函数产生的网络,其缺省的性能函数是网络输出和输出样本间
% 的均方差 mse
% 网络在训练过程中的默认训练过程显示的“performance is……”便是 mse
% 所以关于网络输出与输出样本间的均方误差不必刻意计算
% 网络学习时会自动显示
% 显然,mse 是评价网络学习精度的一个基本指标
perf = mse(e)
```

2. 函数 msereg

```
perf = msereg(e,x,pp)
perf = msereg(e,net)
info = msereg(code)
```

以上程序中各参数的含义与函数 mse 相同。函数 msereg 是在函数 mse 基础上增加一项网络权值和阈值的均方值,目的是使网络获得较小的权值和阈值,从而迫使网络的响应变得更平滑。在 MATLAB 程序中,可以通过 `net.performFcn = 'msereg'` 语句将网络在训练过程中默认实时显示的性能评价函数 mse 切换到 msereg,其调整幅度可以通过形如“`net.performparam. ratio = xxx`”的命令来设置。



在使用该函数前,需要设定性能参数 pp,如 pp.ration=0.6,该参数的意义是表达了误差对于权值和阈值的重要性。此时,函数返回值=返回误差×pp.ration+权值或阈值均方值×(1−pp.ration)。

下面给出了一个创建 BP 网络并评估其性能的实例。

```
clc; clear;    % 清屏,清除内存
% 构建网络结构
net = newff([-2,2],[4,1],{'tansig','purelin'},'trainlm','learnngdm','msereg');
p = [-2 -1 0 1 2];
t = [1 1 0 1 0];
y = sim(net,p);
e = t - y;
net.performparam.ratio = 0.92;
perf = msereg(e,net)
```

运行结果为

```
y =
    1.0127    1.2238    1.9337    0.5996    0.6478
perf =
    1.5773
```

5.4 组建神经网络的注意事项

5.4.1 神经元节点数

网络的输入与输出节点数是由实际问题的维数决定的,与网络性能无关。而隐含层节点数 l 目前还没有统一的规范来解决,一般可利用下面两个经验公式之一来确定:

$$l = \sqrt{m + n} + a \quad (5-33)$$

$$l = \sqrt{0.43mn + 0.12n^2 + 2.54m + 0.77n + 0.35} + 0.51 \quad (5-34)$$

其中, m,n 分别为输入、输出节点数目; a 为 1~10 之间的常数。

隐含层节点数可以根据上面两个公式得出一个初始值,然后利用逐步增长或逐步修剪法最终确定神经元的个数。逐步增长是从一个较简单的网络开始,若训练结果不符合要求,则逐步增加隐含层神经元个数直到合适为止。逐步修剪则从一个较复杂的网络开始逐步删除隐含层神经元。

需要说明的是:经验公式只是确定隐含层节点数的参考方法,并不是在任何情况下都是有效的。对于一个没有任何规律、杂乱无序、拐点特别多的训练样本,经验公式有时完全失效,原因如下:

其一,神经网络的激励函数都是连续光滑的函数,无法拟合振荡特别强烈的样本;

其二,激励函数除了小波函数,常规的 S 函数都是比较平滑的,无法精确快速响应那些从“波峰”迅速衰减到“波谷”的学习样本。

所以当出现随机程度比较大的样本时,通常会一味增加隐含层神经元以便减小学习误差,但是这样做的代价是计算机硬件开销大,网络泛化能力也随之降低,并且单纯增加隐含层神经元并不能完全消除学习误差。网络的性能与数据品质有直接的关系。因此,为了尽量减少使用不必要的隐含层神经元,可以通过以下步骤筛选学习数据:

①清理样本中过大或过小的“野点”数据;

②采样间隔需疏密有度;

③训练样本容量大小适宜,在尽量全面覆盖知识的前提下选择少一点的数据;

④输入、输出的样本选择要配合机理分析。只有输入和输出之间存在必然的因果关系时,使用神经网络才有意义。

如果样本训练的先后顺序没有差异,那么还可以按照以下步骤进一步筛选数据:通过对所有输入因子与输出因子之间的关联分析,找出各个输入因子相对输出的重要程度,依照重要程度的顺序,优先对最重要输入因子进行有规律的排序(当然输出也随之排序)。如果最重要的输入因子找不到比较好的排序方案,不妨就用单调增或单调减顺序排列。最重要输入因子排列完毕以后,在最重要因子的基础上,依次类推,排列次重要的输入因子。

5.4.2 数据预处理和后期处理

1. MATLAB 中提供的预处理方法

(1) 归一化处理

将每组数据都变为 $-1 \sim 1$ 之间的数,相关的函数有 `mapminmax`, `premnmx`, `postmnmx`, `trmnmx`。当然,也可以利用公式 $x_k = \frac{x_k - x_{\min}}{x_{\max} - x_{\min}}$ 进行编程实现归一化。

(2) 标准化处理

将每组数据都化为均值为 0、方差为 1 的一组数据,所涉及的函数有 `prestd`, `poststd`, `trastd`。

(3) 主成分分析

运用正交处理,可减少数据的维数,所涉及的函数有 `prepca` 和 `trapca`。

(4) 回归分析与相关性分析

相关的函数为 `postrg` 等。

2. 数据预处理的常用函数

(1) `premnmx` 函数

`premnmx` 函数的使用格式为

```
[pn,minp,maxp,tn,mint,maxt] = premnmx(p,t)
```

```
[pn,minp,maxp] = premnmx(p)
```

其中, p , t 分别为原始输入和输出数据; $\min p$ 和 $\max p$ 分别为 p 中的最小值和最大值; $\min t$ 和



$\max t$ 分别为 t 的最小值和最大值; pn, tn 分别为归一化后的输入、输出数据。

这个函数使用的算法为: $pn = 2 * (p - \min p) / (\max p - \min p) - 1$ 。

`premnmx` 函数用于将网络的输入数据或输出数据进行归一化, 归一化后的数据将分布在 $[-1, 1]$ 区间内。

(2) `tramnmx` 函数

在训练网络时如果所用的是经过归一化的样本数据, 那么以后使用网络时所用的新数据也应该和样本数据接受相同的预处理, 这就要用到 `tramnmx` 函数。 `tramnmx` 函数格式为

```
[pn] = tramnmx(p, minp, maxp)
```

其中, p 和 pn 分别为变换前、后的输入数据; $\max p$ 和 $\min p$ 分别为 `premnmx` 函数找到的最大值和最小值。

这个函数使用的算法为: $pn = 2 * (p - \min p) / (\max p - \min p) - 1$ 。

(3) `postmnmx` 函数

`postmnmx` 函数是将已被 `premnmx` 归一化的数据进行反归一化处理。 `postmnmx` 函数语法格式为

```
[p, t] = postmnmx(pn, minp, maxp, tn, mint, maxt)
```

其中, pn, tn 分别为归一化的输入、输出数据; $\min p$ 和 $\max p$ 分别为 p 中的最小值和最大值; $\min t$ 和 $\max t$ 分别为 t 的最小值和最大值; p, t 分别为还原的输入和输出数据。

`postmnmx` 函数主要作用是将输出的结果映射回归一化之前的数据范围。

(4) `prestd`, `poststd`, `trastd` 函数

`prestd` 函数的作用是数据前处理, 使数据转换为平均值为 0, 平均偏差为 1 的标准数据。

`poststd` 函数是与 `prestd` 函数配套使用的, 其作用是将数据进行反归一化处理。

`trastd` 函数也是与 `prestd` 函数配套使用的, 其作用是将网络中使用的其他新数据进行相同的归一化处理, 此时将会使用 `prestd` 函数中传送过来的平均值和标准方差。

`prestd` 函数的语法格式为

```
[pn, meanp, stdp, tn, meant, stdt] = prestd(p, t)
```

```
[pn, meanp, stdp] = prestd(p)
```

其中, p 和 pn 分别为变换前、后的输入数据; meanp 是 p 对应数据的平均值; stdp 是 p 对应数据的标准方差, t 和 tn 分别是目标数据和转换成标准数据的目标数据, meant 是 t 对应数据的平均值, stdt 是 t 对应数据的标准方差。

`poststd` 与 `trastd` 函数的语法格式分别为

```
[p] = poststd(pn, meanp, stdp)
```

```
[pn] = trastd(p, meanp, stdp)
```

5.4.3 学习速率的选定

学习速率参数 `net.trainparam.lr` 不能选择得太大,否则会出现算法不收敛的情况;也不能太小,否则会使训练过程时间太长。一般选择 0.01 ~ 0.1 之间的值,再根据训练过程中梯度变化和均方误差变化值来确定。

5.5 基于 MATLAB 工具箱的神经网络建模仿真实例

5.5.1 公路运量预测

1. 问题的描述

公路运量主要包括客运量和货运量两方面。据研究,某地区的公路运量主要与该地区的人数、机动车数量和公路面积有关,表 5-3 给出了某地区 20 年的公路运量相关数据,表中人数和公路客运量的单位为万人,机动车数量的单位为万辆,公路面积的单位为万平方米,公路货运量单位为万吨。

表 5-3 某地区 20 年公路运量数据

年份	人口数量	机动车数量	公路面积	公路客运量	公路货运量
1990	20.55	0.60	0.09	5 126	1 237
1991	22.44	0.75	0.11	6 217	1 379
1992	25.37	0.85	0.11	7 730	1 385
1993	27.13	0.90	0.14	9 145	1 399
1994	29.45	1.05	0.20	10 460	1 663
1995	30.10	1.35	0.23	11 387	1 714
1996	30.96	1.45	0.23	12 353	1 834
1997	34.06	1.60	0.32	15 750	4 322
1998	36.42	1.70	0.32	18 304	8 132
1999	38.09	1.85	0.34	19 836	8 936
2000	39.13	2.15	0.36	21 024	11 099
2001	39.99	2.20	0.36	19 490	11 203
2002	41.43	2.25	0.38	20 433	10 524
2003	44.59	2.35	0.49	22 598	11 115
2004	47.30	2.50	0.56	25 107	13 320
2005	52.89	2.60	0.59	33 442	16 762
2006	55.73	2.70	0.59	36 836	18 673
2007	56.76	2.85	0.67	40 548	20 724
2008	59.17	2.95	0.69	42 927	20 803
2009	60.63	3.10	0.79	43 462	21 804

根据统计,该地区2010年、2011年的人口数量分别为73.39万人和75.55万人,机动车数量分别为3.9635万辆和4.0975万辆,公路面积分别为0.9880万平方千米和1.0268万平方千米。请利用BP网络预测该地区2010、2011两年的公路客运量和货运量。

2. BP网络求解过程

为了了解利用BP网络求解问题的过程,可把问题分成以下六个模块进行处理:原始数据的输入;数据归一化;网络训练;对原始数据进行仿真;将原始数据仿真的结果与已知样本进行对比;对预测数据进行仿真。

注意:在进行数据归一化和把仿真得到的数据还原为原始数据的数量级时,在下面的程序中,都是利用系统函数进行的,大家在自己处理数据时可以利用这些函数,也可以自己编制程序进行。另外,步骤(4)和(5)是对训练过的神经网络进行测试,不经过测试或测试不合格的网络是不能使用的。

(1)原始数据的输入

```
clc
%原始数据
%人数(单位:万人)
sqrs = [20.55 22.44 25.37 27.13 29.45 30.10 30.96 34.06 36.42 38.09 ...
        39.13 39.99 41.93 44.59 47.30 52.89 55.73 56.76 59.17 60.63];
%机动车数(单位:万辆)
sqjdc = [0.6 0.75 0.85 0.9 1.05 1.35 1.45 1.6 1.7 1.85 2.15 2.2 2.25 2.35 ...
        2.5 2.6 2.7 2.85 2.95 3.1];
%公路面积(单位:万平方千米)
sqglmj = [0.09 0.11 0.11 0.14 0.20 0.23 0.23 0.32 0.32 0.34 0.36 0.36 ...
        0.38 0.49 0.56 0.59 0.59 0.67 0.69 0.79];
%公路客运量(单位:万人)
glkyl = [5126 6217 7730 9145 10460 11387 12353 15750 18304 19836 21024 ...
        19490 20433 22598 25107 33442 36836 40548 42927 43462];
%公路货运量(单位:万吨)
glhyl = [1237 1379 1385 1399 1663 1714 1834 4322 8132 8936 11099 11203 ...
        10524 11115 13320 16762 18673 20724 20803 21804];
p = [sqrs;sqjdc;sqglmj]; %输入数据矩阵
t = [glkyl;glhyl]; %目标数据矩阵
```

(2)对输入数据矩阵和目标矩阵的数据进行归一化

```
%利用 premnmx 函数对数据进行归一化
[pn,minp,maxp,tn,mint,maxt] = premnmx(p,t);
%对于输入矩阵 p 和输出矩阵 t 进行归一化处理
dx = [-1,1;-1,1;-1,1]; %归一化处理后最小值为-1,最大值为1
```

(3) 网络训练

```
% BP 网络训练
net = newff( dx,[3,7,2],{'tansig','tansig','purelin'},'traingdx');
% 建立模型,并用梯度下降法训练
net.trainParam.show = 1000;           % 1 000 轮回显示一次结果
net.trainParam.Lr = 0.025;           % 学习速度为 0.025
net.trainParam.epochs = 10000;       % 最大训练轮回为 10 000 次
net.trainParam.goal = 0.5*10^(-3);   % 均方误差
net = train( net,pn,tn);             % 开始训练,其中 pn,tn 分别为输入输出样本
```

(4) 对原始数据进行仿真

```
% 利用原始数据对 BP 网络仿真
an = sim( net,pn);                   % 用训练好的模型进行仿真
a = postmnmx( an,mint,maxt);
% 把仿真得到的数据还原为原始的数量级
```

(5) 原始数据仿真的结果与已知数据进行对比测试

```
x = 1990:2009;
newk = a(1,:);
newh = a(2,:);
figure(2);
subplot(2,1,1);plot(x,newk,'r-o',x,glkyl,'b--+') % 绘制公路客运量对比图
legend('网络输出客运量','实际客运量');
xlabel('年份');ylabel('客运量/万人');
subplot(2,1,2);plot(x,newh,'r-o',x,glhyl,'b--+') % 绘制公路货运量对比图
legend('网络输出货运量','实际货运量');
xlabel('年份');ylabel('货运量/万吨');
title('运用工具箱货运量学习和测试对比图');
```

(6) 对预测数据仿真

```
% 利用训练好的网络进行预测
% 当用训练好的网络对新数据 pnew 进行预测时,也应作相应的处理:
pnew = [73.39 75.55
        3.9635 4.0975
        0.9880 1.0268];           % 2010 年和 2011 年的相关数据;
pnewn = tramnmx( pnew,minp,maxp);
% 利用原始输入数据的归一化参数对新数据进行归一化
anewn = sim( net,pnewn);          % 利用归一化后的数据进行仿真
anew = postmnmx( anewn,mint,maxt)
% 把仿真得到的数据还原为原始的数量级
```


运行程序后,可以得到 anew 的值为 $\begin{bmatrix} 44 & 165 & 44 & 171 \\ 21 & 992 & 21 & 995 \end{bmatrix}$,也就是说,2010 年和 2011 年的公路客运量分别为 44 165 万人和 44 171 万人,公路货运量分别为 21 992 万吨和 21 995 万吨。程序运行的界面如图 5-11 所示。

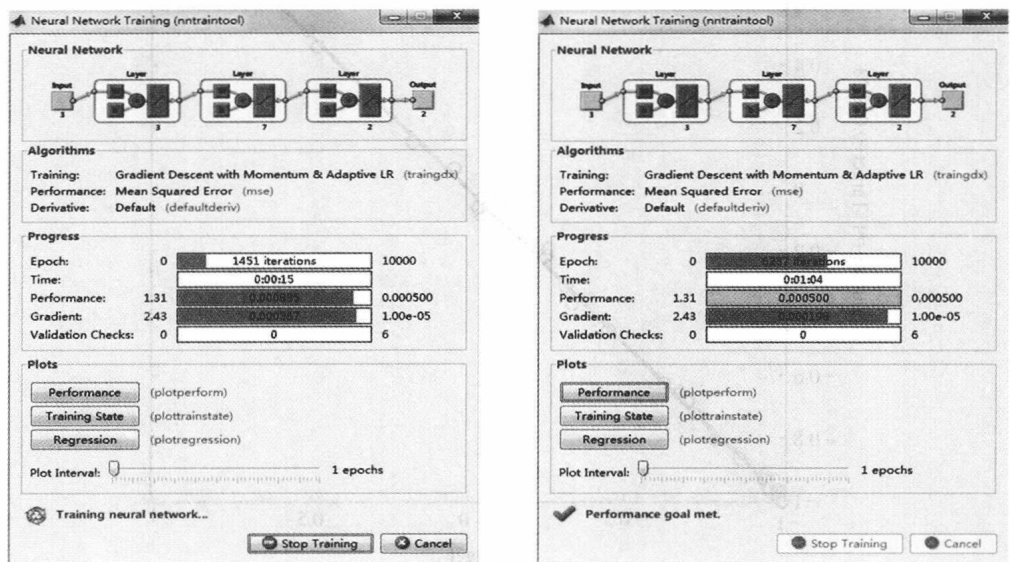


图 5-11 MATLAB 中神经网络运行界面示意图

训练结束后,点击“Performance”按钮查看网络训练结果曲线,如图 5-12 所示,可以看出网络训练所得误差满足目标要求。

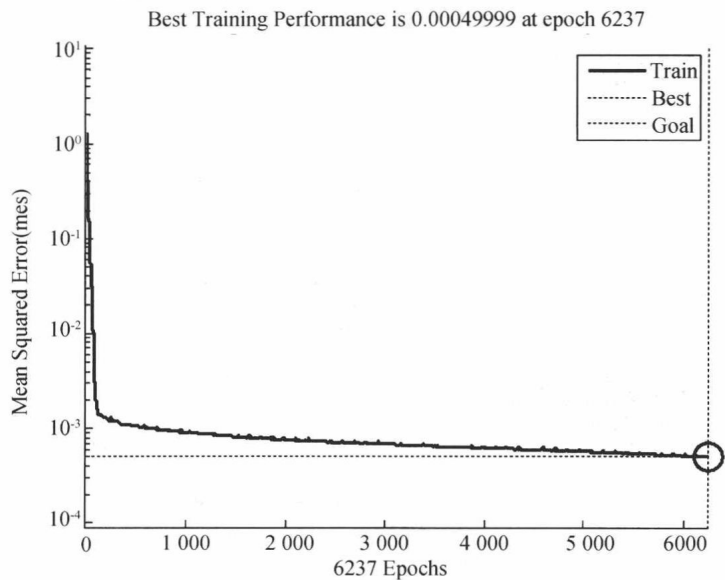


图 5-12 网络训练结果曲线图

点击“Regression”按钮,查看训练结果的回归分析,如图 5-13 所示。

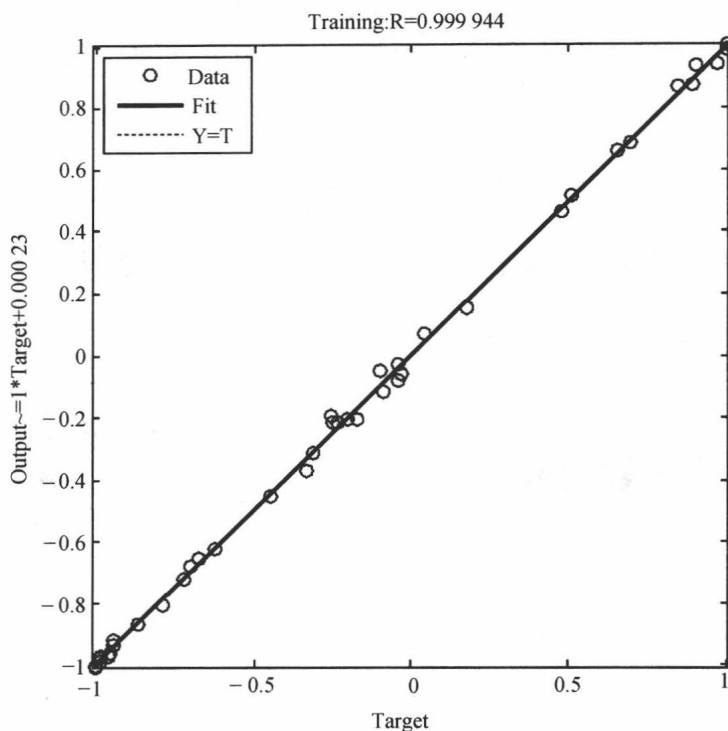


图 5-13 网络训练结果的回归分析图

图 5-14 是实际样本与网络输出值之间的训练和测试的对比图,从图中可以看出两者之间非常接近,误差极小,因此 BP 神经网络求解结果能够用于预测。这里需要说明的是,

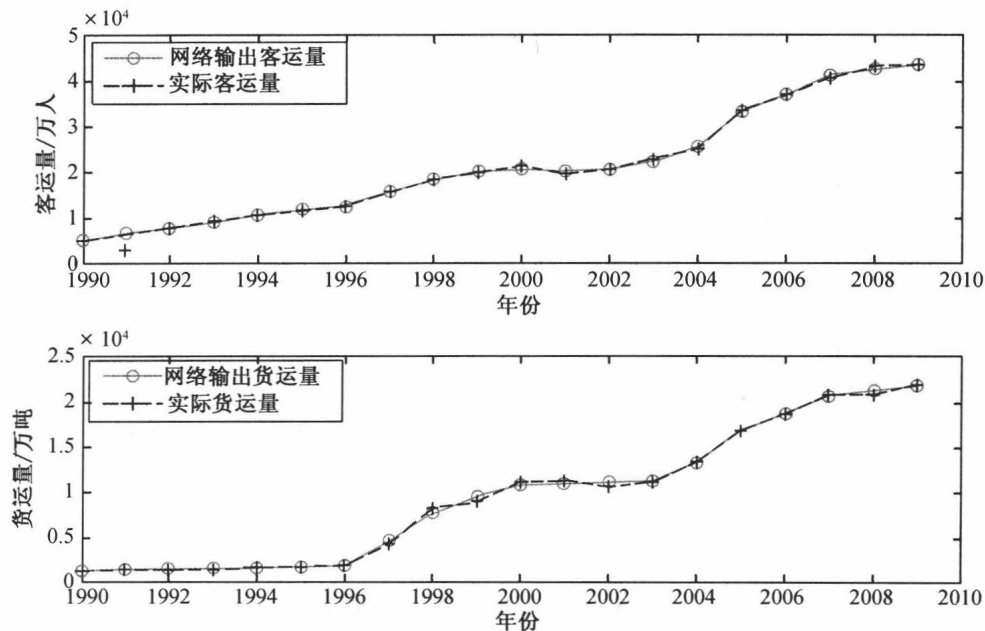


图 5-14 运用 MATLAB 的 BP 神经网络工具箱生成的数据对比图



本案例因为样本数量比较少,测试阶段使用了训练样本。通常情况下,为了测试网络的推理能力,是不宜用训练样本去测试神经网络性能的,应该选用没有用于训练的数据进行测试。

5.5.2 上市公司信用分类

1. 问题的描述

我国沪、深两市上市公司中有非 ST 公司和 ST 公司,一般而言非 ST 公司的信用等级较高,ST 公司的信用等级较差。为有效评价上市公司信用,建立了上市公司信用评价指标体系,如表 5-4 所示。

表 5-4 上市公司信用评价指标体系

一级指标	二级指标
偿债能力	流动比率 x_1
	负债比率 x_2
运营能力	存货周转率 x_3
	总资产周转率 x_4
盈利能力	净资产收益率 x_5
	每股收益 x_6
成长能力	总利润增长率 x_7
现金流量指标	每股经营现金流量 x_8

选取 60 家上市公司(其中 30 家非 ST 公司,30 家 ST 公司)作为训练样本,如表 5-5 所示。数据保存为 traindata.txt 文件。

表 5-5 上市公司信用分类训练样本数据

序号	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	非 ST	ST
1	0.404	39.15	15.55	10.75	0.524	3.645	2.395	0.31	1	0
2	1.263	54.17	6.13	19.47	2.198	10.336	0.495	0.118	1	0
3	0.871	11.88	6.98	-18.22	0.481	16.146	6.385	0.624	1	0
4	1.317	20.38	13.13	57.79	0.299	19.396	1.937	0.673	1	0
5	0.722	9.33	10.09	10.22	0.444	2.515	17.564	0.3	1	0
6	0.278	47.96	3.27	75.92	0.51	4.025	0.978	0.069	1	0
7	-0.403	36.09	8.17	-8.84	0.801	5.471	2.791	0.357	1	0
8	1.122	13.94	22.09	8.04	0.621	37.413	4.381	0.394	1	0
9	0.571	11.59	7.94	34.01	0.492	12.531	1.734	0.307	1	0
10	0.835	19.77	11.88	-16.17	0.689	3.29	4.179	0.722	1	0
11	0.281	61.76	4.49	110.01	0.458	14.23	0.535	0.029	1	0
12	0.917	37.79	19.19	14.54	0.711	78.512	0.905	0.423	1	0

表 5-5(续)

序号	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	非 ST	ST
13	1.86	26.11	21.85	40.8	0.569	18.858	2.256	0.594	1	0
14	1.141	26.63	36.07	58.37	1.177	5.847	1.32	1.232	1	0
15	0.402	32.01	7.45	6.64	0.744	5.187	1.849	0.304	1	0
16	1.169	41.41	21.92	63.29	0.726	7.692	1.164	0.558	1	0
17	1.983	57.22	18.95	180.4	0.512	8.685	0.645	0.589	1	0
18	3.424	29.98	18.65	55.72	0.54	0.417	2.528	2.134	1	0
19	1.218	29.59	17.02	31.44	2.507	13.071	1.503	0.77	1	0
20	1.277	55.43	23.63	5.95	1.993	5.796	1.315	0.61	1	0
21	0.231	54.3	2.26	36.61	0.563	2.914	1.128	0.054	1	0
22	0.687	48.21	15.88	65.01	0.615	5.305	0.912	0.7	1	0
23	1.16	54.22	15.33	30.72	0.472	6.191	1.899	0.543	1	0
24	-0.593	55.23	17.71	-14.39	1.328	6.584	1.094	0.911	1	0
25	0.722	41.09	16.42	24.81	1.87	13.26	1.378	0.434	1	0
26	0.1	43.16	8.91	60.73	0.476	4.544	0.753	0.161	1	0
27	1.714	30.43	14.34	48.08	1.013	36.002	1.083	0.714	1	0
28	1.565	48.54	10.68	-13.61	0.235	14.717	3.958	0.581	1	0
29	-0.097	18.51	6.1	-38.35	0.595	11.944	3.639	0.329	1	0
30	0.1	63.54	10.18	9.27	0.262	2.663	1.615	0.227	1	0
31	0.027	94.34	-270.5	-317.1	0.34	1.954	0.424	-1.361	0	1
32	-0.085	105.62	-304.68	-305.57	0.194	0.599	0.703	-1.448	0	1
33	0.292	65.46	-9.15	65.3	0.116	2.497	1.173	-0.139	0	1
34	-0.084	74.52	-78.12	-158.52	0.014	0.1	0.931	-0.763	0	1
35	0.607	45.86	-10.9	-579.26	0.093	0.824	1.367	-0.488	0	1
36	-0.122	51.84	-14.74	-988.67	0.143	0.184	1.939	-0.222	0	1
37	0.268	49.51	-22.43	19.16	0.066	1.589	1.365	-0.451	0	1
38	0.046	144.82	-3 339.72	-704.55	0.794	3.934	0.437	-1.941	0	1
39	0.031	94.14	-130.78	-456.48	0.217	1.008	0.887	-0.846	0	1
40	-0.007	131.05	-1 287.61	-17.58	0.005	0.583	0.025	-0.998	0	1
41	0.043	82.73	-83.93	59.53	0.356	1.134	0.659	-0.409	0	1
42	0.08	54.89	2.83	-48.37	0.247	1.631	0.811	0.051	0	1
43	-1.724	45.17	-24.42	-364.52	0.357	3.921	1.57	-0.839	0	1
44	-0.257	60.2	-14.01	-502.27	0.061	2.355	1.418	-0.437	0	1



表 5-5(续)

序号	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	非 ST	ST
45	0.135	95.06	-193.72	-160.73	0.468	5.608	0.407	-2.71	0	1
46	-0.1	43.02	-16.8	-940.87	0.243	1.179	1.524	-0.44	0	1
47	-0.081	48.21	-3.65	-387.15	0.173	0.83	1.395	-0.122	0	1
48	-0.212	56.68	-6.18	-1 511.14	0.365	2.137	1.098	-0.165	0	1
49	0.506	50.09	-3.11	44.06	0.203	2.594	0.945	0.02	0	1
50	0.003	38.26	-10.08	-3 083.81	0.177	2.446	1.124	-0.183	0	1
51	-0.031	24.27	-10.27	-146.6	0.002	0.021	0.779	-0.235	0	1
52	0.004	72.63	-39.26	-297.95	0.018	0.74	0.509	-0.34	0	1
53	0.227	63.65	7.33	5.58	0.39	2.465	1.15	0.298	0	1
54	0.285	64.04	3.3	-48.77	0.328	5.388	0.792	0.072	0	1
55	-0.013	82.08	-72.64	-122.43	0.012	0.447	0.264	-1.27	0	1
56	-0.195	26.28	0.95	-7.59	0.292	0.596	4.78	0.015	0	1
57	0.329	22.76	1.74	-56.57	0.357	0.543	3.238	0.087	0	1
58	-0.001	269.39	-20.85	44.49	0.093	3.466	0.123	-0.329	0	1
59	-0.222	73.68	2.04	106.73	0.654	3.157	0.841	0.021	0	1
60	0.005	42.77	-4.15	-205.21	0.472	2.622	1.882	-0.048	0	1

选取 10 家上市公司(其中 5 家非 ST 公司,5 家 ST 公司)作为测试数据,如表 5-6 所示。数据保存为 testdata.txt 文件。

表 5-6 上市公司信用分类测试样本数据

序号	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	0.74	13.27	6.14	-7.3	0.554	18.406	5.631	0.482
2	1.312	64.79	14.98	61.17	0.597	4.193	1.057	0.368
3	0.538	33.95	18.62	24	1.53	10.496	1.832	0.6
4	0.853	13.07	24.75	40.17	0.514	35.555	5.673	0.368
5	1.254	67.6	19.73	530.4	1.815	31.609	0.712	0.183
6	0.064	50.94	0.19	-21.99	0.027	0.197	0.625	0.002
7	0.197	90.93	-91.49	-1 045.94	0.125	0.194	0.581	-0.986
8	-0.126	19.76	74.63	3 809.21	0.386	5.422	2.725	0.524
9	1.564	59.86	-9.22	-313.31	0.284	1.565	1.444	-0.102
10	-0.012	636.06	-5.34	-184.11	0.352	2.091	0.096	-0.195

2. BP 网络求解过程

(1) 数据的输入

```

clc; clear;
load traindata.txt           % 从 traindata.txt 中读取训练的输入数据
p = traindata(:, :);
p = p';                      % 输入数据矩阵
t = [ones(1,30), zeros(1,30); zeros(1,30), ones(1,30)]; % 目标数据矩阵

```

(2) 对输入数据矩阵和目标矩阵的数据进行归一化

```

[pn, minp, maxp] = premnmx(p); % 对输入数据矩阵 p 进行归一化处理
tn = t;
dx = minmax(pn);              % 确定每组输入元素的最小值和最大值

```

(3) 网络训练

```

net = newff(dx, [8, 10, 2], {'tansig', 'tansig', 'purelin'}, 'traingdx');
% 建立模型, 并用梯度下降法训练
net.trainParam.show = 1000; % 1 000 轮回显示一次结果
net.trainParam.Lr = 0.025; % 学习速度为 0.025
net.trainParam.epochs = 10000; % 最大训练轮回为 10 000 次
net.trainParam.goal = 0.5 * 10^(-4); % 均方误差
net = train(net, pn, tn); % 开始训练, 其中 pn, tn 分别为输入输出样本

```

(4) 对训练数据进行仿真

```

% 利用原始数据对 BP 网络仿真
an = sim(net, pn); % 用训练好的模型进行仿真

```

(5) 对测试数据进行仿真

```

% 利用训练好的网络进行测试
load testdata.txt           % 从 testdata.txt 中读取测试数据
p_test = testdata(:, :);
p_test = p_test';
p_testn = trmnmx(p_test, minp, maxp); % 训练数据归一化
t_test = sim(net, p_testn) % 用训练好的模型进行仿真
plot(an(1, 1:30), an(2, 1:30), 'h', an(1, 31:60), an(2, 31:60), 'o', t_test(1, :), ...
t_test(2, :), '*') % 绘制测试数据分类图
legend('非 ST', 'ST', '测试数据');
title('上市公司信用分类图');

```

运行程序, 可以得到 t_test 结果为



$$\begin{bmatrix} 1.0015 & 0.9976 & 1.0000 & 1.0025 & 0.9966 & -0.0108 & 0.0045 & 0.9995 & 0.9802 & -0.0963 \\ 0.0006 & 0.0005 & 0.0020 & 0.0006 & 0.0036 & 1.0087 & 0.9949 & 0.0018 & 0.0196 & 1.1592 \end{bmatrix}$$

从而得到 10 组测试的上市公司的信用分类,其中第 1,2,3,4,5,8,9 家公司的信用属于非 ST 类型,第 6,7,10 家公司的信用属于 ST 类型。

程序运行的界面如图 5-15 所示。

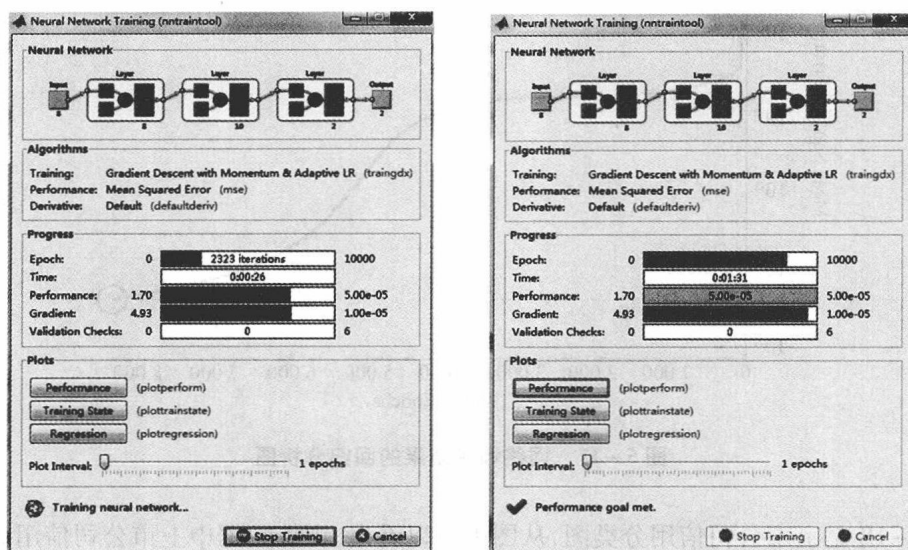


图 5-15 MATLAB 中神经网络运行界面示意图

训练结束后,点击“Performance”按钮查看网络训练结果曲线,如图 5-16 所示,可以看出网络训练所得误差满足目标要求。

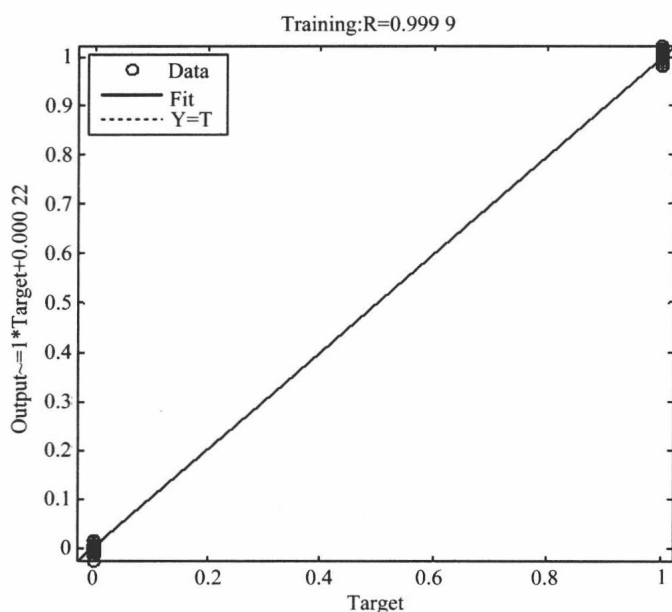


图 5-16 网络训练结果曲线图



上市公司信用分类图



第6章 灰色预测建模仿真

6.1 预测概述

6.1.1 预测的基本概念

预测是一个古老的话题。自从人类诞生以来,预测活动就已经存在了。人类的祖先由于不能理解风雨雷电、陨石流星、潮汐海啸等自然现象,而赋予它们以神秘的气息,并逐渐把这些自然现象超自然化,将自己的命运寄托于主宰这些自然现象的所谓的神的身上。远古的人们利用龟甲或兽骨去占卜(预测)战争的胜负、年成的好坏,并据此决定本部落的行动。

随着人类社会和科学技术的发展,预测的技术不断发展,预测工作逐渐褪去了神秘的色彩,进入科学预测的阶段。科学的预测能够正确地向人们展现未来,使人们不再盲目地行动,使人类可以有计划地发展自己。

科学技术作用于社会的效果,不仅体现在它给社会带来的巨大利益,而且也体现在它给社会带来的一些令人担忧的不良后果。从这个意义上来讲,预测研究更引起了人们的关注。预测研究的领域在不断扩大,研究方法也在逐渐完善。直到今天,预测科学已经成为一门发展迅速、应用广泛的新学科。

1. 预测的定义

预测,是指根据客观事物的发展趋势和变化规律对特定的对象未来发展的趋势或状态做出科学的推测与判断。换言之,预测是根据对事物的已有认识,做出对未知事物的预估。预测是一种行为,表现为一个过程;同时,它也表现为行为的某种结果。

作为探索客观事物未来发展的趋势或状态的预测活动,是人类“鉴往知来”智慧的表现,是科学实践活动的构成部分。预测之所以是一种科学活动,是由预测前提的科学性、预测方法的科学性和预测结果的科学性决定的。预测前提的科学性包括三层含义:一是预测必须以客观事实为依据,即以反映这些事实的历史与现实的资料和数据为依据进行推断;二是作为预测依据的事实资料与数据,还必须通过抽象上升到规律性的认识,并以这种规律性的认识作为预测的指导;三是预测必须以正确反映客观规律的某些成熟的科学理论作指导。预测方法的科学性包含两层含义:一是各种预测方法是在预测实践经验的基础上总结出来,并获得理论证明与实践检验的科学方法,包括预测对象所处学科领域的方法以及数学的、统计学的方法;二是预测方法的应用不是随意的,它必须依据预测对象的特点合理选择和正确运用。预测结果的科学性包含两层含义:一是预测结果是由已认识的客观对象发展的规律性和事实资料为依据,采用定性与定量相结合的科学方法做出的科学推断,并用科学的方式加以表述;二是预测结果在允许的误差范围内可以验证预测对象已经发生的事实,同时在条件不变的情况下,预测结果能够经受实践的检验。

2. 预测的基本功能

预测的基本功能就是为决策系统提供制定决策所必需的未來信息。为提高未來信息的可靠性,必须深入研究为获取这些信息所使用的各种方法与手段。对于工业企业而言,预测则贯穿于企业的经营活动之中。在销售方面,需要适合市场规模和市场特点的可靠预测,如顾客类型、市场占有率、物价变动趋势、新产品开发等方面的预测资料都对销售起促进作用。在生产方面,需要预测产品的销售规模、原材料需求量、材料成本及劳动力成本的变动趋势、材料与劳动力的可用量的变动趋势等,以便企业对生产和库存进行计划,并在合理的成本上满足销售的需求。在会计方面,预测现金流出量和各种收支项目的比率,使企业资金周转灵活,经营卓有成效。还需要预测可收取的应收账款、实际业务状况等。在人事部门方面,也需要预测每一类员工所需人数、员工流动趋势等。

3. 预测的不准确性

①预测的准确性与预测对象变化的速度及其复杂性成反向变化。只有在一个静止的系统中,一个规则不变的状态下,才能准确地预测未来。随着科学技术的发展,各种因素、现象之间的联系越来越复杂,变化的速度越来越快,准确地预测未来的难度也越来越大。

②人们的认识能力是有限的,人们的理性还不能看清楚其行为的所有结果,对很多事物还不能既知其然,又知其所以然。在这种情况下,人们想要把握其变化规律几乎是不可能的。预测要求人们能超越现实,理解未来,然而人们的理解力又局限于他们的经历,这是一个难以解决的矛盾。因而,人们很难得出准确的预测结论。

③虽然可以采用概率统计的方法来研究偶然事件,但是人们并不能消除这些事件的偶然性。预测不准确来源于未来所具有的偶然性。“有心栽花花不开,无心插柳柳成荫”正反映了这一点。

④预测活动本身也在“干扰”未来。当人们预感前景不妙时,便会设法阻止其出现;当前景不错时,人们会努力促使它尽快实现。对于前者,有学者称其为自失败预测,并将后者称为自成功预测。如科学家们指出物种的多样性是保证食物链稳定的基础,人类若乱捕滥杀,破坏生态环境,就会加速物种消亡,导致食物链崩溃。

既然如此,如何评价预测的结果?预测结果的评价主要看其是否可信、有效。是否可信,至少要考虑如下几个方面:

①预测结果应该是历史与现实的合理延伸。

②预测结果应具有可检验性。它隐含着预测资料的来源及其真实性、预测模型的合理性、预测结果的逻辑性都可检验。无论由谁来做预测,结果都应该一样。

③可信程度还与预测的时间跨度、预测对象的复杂程度、预测结果的详细程度等有关,同时还与预测机构或预测者的权威性有关。

预测的有效性是指预测结果能否为决策者提供可靠的未來信息,以使决策者做出正确决策。现实中对预测结果的评价通常是以有效性为标准的,而有效性暗含着决策者认为预测结果是可信的,同时也暗含着决策者主观上认为预测是准确的。

6.1.2 预测的基本原理与步骤

1. 预测的基本原理

(1) 系统性原理

预测的系统性原理,是指预测必须坚持以系统的观点为指导,采用系统分析方法,实现



预测的系统目标。系统是相互联系、相互依存、相互制约、相互作用的诸事物及其发展过程所形成的统一体。预测工作中体现系统本质特性的观点应包括以下三方面。一是全面地、整体地看问题,而不是片面地、局部地看问题。二是联系地、连贯地看问题,而不是孤立地、分割地看问题。在预测中,必须注意预测对象各层次之间的联系,预测对象与环境之间的联系,预测对象内部与外部各要素之间的彼此联系,预测对象各发展阶段之间的联系等;三是发展地、动态地看问题,而不是静止地、凝固地看问题。预测是对预测对象未来发展趋势的预测,没有发展变化,就不需要预测。预测必须根据预测对象的过去、现在推断未来,从而正确地反映发展观与动态观。

系统都有结构、有层次。预测对象系统的内部结构与层次及其相互关系,是系统按照一定规律运动的内在根据;外部环境因素与系统的相互关系,则是决定系统按照一定规律运动的外在条件。在预测工作中,通过对内在根据与外在条件的分析,便能较好地认识和把握预测对象的运动规律,进而依据这种规律性的认识对预测对象系统的未来状态和趋势做出科学的推测与判断。在预测工作中采用系统分析方法时要求做到:一是通过对预测对象的系统分析,确定影响其变化的变量及其关系,建立符合实际的逻辑模型与数学模型;二是通过对预测对象的系统分析,系统地提出预测问题,确定预测的目标体系;三是通过对预测对象的系统分析,正确地选择预测方法,并通过各种预测方法的综合运用,使预测尽可能地符合实际;四是通过对预测对象的系统分析,按照预测对象的特点组织预测工作,并对预测方案进行验证和跟踪研究,为经营决策的实施提供及时的反馈。

(2) 连贯性原理

事物的发展变化与其过去的行为总有或大或小的联系,过去的行为影响现在,也影响未来,这种现象称为“连贯现象”。所谓连贯性原理,就是研究对象的过去和现在,依据其惯性,预测其未来状态。

连贯性的强弱取决于事物本身的动力和外界因素的强度。连贯性越强,越不易受外界因素的干扰,其延续性越强。如属于生产资料的产品,一般对其品种、质量、产量的需求较稳定,表现出来的连贯性较强。而属于消费资料的产品,由于顾客的兴趣爱好容易变动,其连贯性就较小,尤其是流行服装,几乎没有连贯性。

在实际的运用过程中,应注意以下两方面的问题:一是连贯性的形成需要有足够长的历史,且历史发展数据所显示的变动趋势具有规律性;二是对预测对象演变规律起作用的客观条件必须保持在适度的变动范围之内,否则该规律的作用将随条件变化而中断,连贯性失效。

(3) 类推原理

许多特性相近的客观事物,它们的变化亦有相似之处。通过寻找并分析类似事物相似的规律,根据已知的某事物的发展变化特征,推断具有近似特性的预测对象的未来状态,就是所谓的类推原理。

利用事物之间的相似性进行类推预测时,两事物的发展过程间必定有一个时间差,时间会使许多条件发生变化,也给了人们总结经验和教训的机会,使人们有可能根据变化了的条件去探索后发展事物在哪些方面还保持着与先发展事物相似的特征,在哪些方面已不再相似,等等,做出较为准确的预测。

(4) 相关性原理

任何事物的发展变化都不是孤立的,都是在与其他事物的发展变化相互联系、相互影

响的过程中确定其轨迹的。

相关性最主要的表现形式是因果关系。因果关系是存在于客观事物之间的一种普遍联系。因果关系具有时间上的相随性:作为原因的某一现象发生,作为结果的另一现象必然发生;原因在前,结果在后。因果关系往往呈现出多种多样的情况,有单因单果、单因多果、多因单果、多因多果,还有互为因果以及因果链等。在预测中运用因果性原理,必须科学分析,确定相关事物之间因果联系的具体形式,找出其关键因素,适当进行简化,据此建立合适的预测模型。

(5) 概率推断原理

由于受到社会、经济、科技等因素的影响,预测对象的未来状态带有随机性。例如,某商品下个月的销售情况,可能畅销,可能销路一般,也可能滞销,事前难以确定。预测对象的未来状态如何,这实际上是一个随机事件,可以用概率来表示这一事件发生的可能性大小。

掌握预测的基本原理,可以建立正确的思维程序。这对于预测人员开拓思路,合理选择和灵活运用预测方法都是十分必要的。然而,世界上没有一成不变的事物。预测对象的发展不可能是过去状态的简单延续,预测事件也不可能是已知的类似事件的机械再现。相似不等于相同。因此,在预测过程中,还应对客观情况进行具体细致的分析,以求提高预测结果的准确程度。

2. 预测的一般步骤

预测作为一个过程,一般包括以下几个步骤。

(1) 确定预测目标

预测是为决策服务的,所以要根据决策的需要来确定预测对象、预测结果达到的精确度,确定是定性预测还是定量预测以及完成预测的期限等。如当决策只需知道产品销售发展的趋势时,能够预测出销售量是增加、减少,还是不变的就可以了。而当决策需要了解产品销售量能达到什么样的水平时,则必须对销售量增加或减少的具体数值进行预测,预测也就从定性变为定量了。总之,预测一个事物的发展变化时,首先要了解决策的要求并据此确定属于哪类预测,应满足哪些标准,等等。

(2) 收集、整理有关资料

预测是根据有关历史资料去推测未来的,资料是预测的依据。应根据预测目标的具体要求去收集资料。预测中所需的资料通常包括以下三项:

- ① 预测对象本身发展的历史资料;
- ② 对预测对象发展变化有影响作用的各相关因素的历史资料(包括因素现在的资料);
- ③ 形成上述资料的历史背景,其将影响因素在预测期间内可能表现的状况。

(3) 选择预测方法

预测方法种类很多,不同的方法有不同的适用范围、不同的前提条件和不同的要求。对于特定的预测对象很可能有多种方法可用,而人、财、物、时间等因素的限制使得只能用一种或少数几种方法。实际中应根据计划、决策的需要,结合预测工作的条件、环境,依经济、方便、精度足够好为原则去选择预测方法。

(4) 建立预测模型

预测模型是对预测对象发展变化的客观规律的近似模拟,预测结果是否有效取决于模型对预测对象未来发展规律近似的真实程度。对于数学模型,要求出其模型形式和参数



值。如用趋势外推法,则要求出反映发展趋势的公式;如用类推法,则要寻求与预测对象发展类似的事物在历史上所呈现的发展规律,等等。

(5) 评价预测模型

由于预测模型是用历史资料建立的,它们能否比较真实地反映预测对象未来发展的规律是需要讨论的。评价预测模型就是评价模型能否真实地反映预测对象的未来发展规律。例如,预测对象是否仍按原趋势发展下去,即事物发展是否会产生突变?如无突变,所建立的模型能否反映它的趋势?如果评价结果的模型不能真实地反映预测对象的未来发展状况,则需要重建模型;如能真实地反映,则可进入下一步。

(6) 利用模型进行预测

根据收集到的有关资料,利用经过评价的模型,计算或推测出预测对象的未来结果。

(7) 分析预测结果

利用模型得到的预测结果有时并不一定与事物发展的实际结果相符。这是由于所建立的模型是对实际情况的近似模拟,有的模型模拟效果可能好些,有的可能会差些;同时,在计算和推测过程中也难免会产生误差,再加上预测是在前述的假设条件下进行的,所以,预测结果与实际结果难免会发生偏差。

因此,每次得到预测结果之后,都应对其加以分析和评价。通常是根据常识和经验,检查、判断预测结果是否合理,与实际的结果之间是否存在较大的偏差,以及未来条件的变化会对实际结果产生多大的影响,等等,以确定预测结果是否可信,并想出一些办法对预测结果加以修正,使之更接近于实际。此外,在条件允许的情况下,可以采用多种方法进行预测,再经过比较或综合,确定出可信的预测结果。

在整个预测过程中,对预测成败影响最大的是两个“分析和处理”:一个是对收集到的资料进行分析和处理,资料是基础,如果基础质量不好,建立在这个基础之上的大厦(预测模型)质量也差,预测结果的质量也必定差;另一个是对预测结果的分析和处理,这是对预测效果的最后一次检查,它直接决定预测的质量。这两个分析和处理最能体现预测者的水平,预测的质量完全取决于预测者对预测对象及客观条件的熟悉程度、知识面的广度、对事物的观察能力以及逻辑推理与分析判断的能力,等等。

就像使用相同原料、相同工具进行生产的工人生产出不同质量的产品一样,不同的预测者在运用相同的资料和相同的预测技术对同一预测对象进行预测时,也可能会得到质量相差很大的预测结果。这种差别常常产生在这两个分析和处理上。

从上述基本步骤也可以看出,预测是一项“技艺”性的工作,它既需要科学的方法,又需要进行艺术的处理。由于预测对象的发展变化规律要比自然科学所研究的对象的发展变化规律复杂得多,所处的环境也复杂得多,预测工作者的这种“技艺”也就显得愈加重要。

6.1.3 预测资料的收集与预处理

预测是利用历史的和现在的资料来对事物的未来发展前景进行探索的过程。各种定量预测方法能被用来进行预测之前,都需要收集相当数量的数据。同时,由于实际收集到的数据可能不足以反映出预测对象变化发展的规律,需要对收集到的数据进行分析 and 处理。

1. 变量的确定与说明

进行预测时,首先要确定被预测的变量。其次,要确定对预测变量的发展有影响的其

他变量(这对于预测者打算用回归方法时尤其重要)。在确定有关变量时,应注意以下几点。

(1) 每一数据值涉及的时期

在实际中,很多因素可视为连续发生的,如大型商场的商品销售量。但出于核算的目的,必须规定某个时期并按该时期将每个变量的值加起来,如商场每天、每周、每月的销售总量。

在确定变量的每个观察值所涉及的时期时,还必须考虑预测的具体应用。为长期决策服务的预测,一般以相当长时期(季、年)观察的数据为基础;而为日常管理服务的预测,则常以一天甚至一小时为单位的数据值为基础。

(2) 要求的详细程度

要求的详细程度是指一个变量所要综合历史资料的总量。例如,是预测某给定时期整个公司的销售量,还是按产品类别甚至在每个地区按产品类别来预测销售量。如果最初的预测过于笼统,就必须回头收集更详细的数据。因此,在预测开始时,按时间确定数据到底需要怎样的详细程度是能够节约大量费用的。总之,按尽可能的详细程度来收集数据并加以汇总,比先收集综合数据而后才发现需要进一步细分时要有效得多。

(3) 计量单位

通常的核算体系是按价值来编制报表的,大多数数据已将其单位换算为“元”,而这实际上是资料的损失。如洗衣机的销售量,其单位已换算为“元”,当洗衣机的价格变化时就很难再进行估算了。因此,预测中的一个重要步骤是确定各变量的合适单位,而且应在原始数据已经存储的条件下来进行单位的换算。

(4) 要求的准确程度

预测的不同用途要求有不同的准确程度。如果某项预测对某些重要管理工作来说是辅助性的,这时所要求的准确程度就不会太高。另一方面,在中等重要程度的管理工作中,有可能利用个别变量的预测作为决策的基础,在这种情况下则希望有高度的准确性。由于提高预测准确性一般会带来费用的增加,因此,在确定合适的准确程度时需要恰当地权衡利弊。

在确定数据的最初阶段,还需要对预测的价值有一个粗略的估计,以便能够把数据收集过程的费用控制在预测结果价值规定的上限之内。

2. 数据的收集与整理

确定了预测中的变量之后,就要考察和了解变量的特性及其有关情况,亦即收集和获取所需的数据(或称资料)。首先,要注意数据的客观性和准确性,即要求数据如实地反映实际情况。其次,要求数据具有及时性、完整性、经济性。这些要求有时会自相矛盾。例如,完整的数据资料受时间、经费等的限制而难以收集到。实际预测时,应根据具体情况兼顾各项要求。

按调查资料的来源不同,一般将资料分为原始资料和第二手资料。原始资料是指直接对调查对象进行观察、登记搜集到的未经整理的^①第一手资料。例如,工业企业的原始记录,产品需求的调查问卷等。原始记录可如实地反映有关情况,但要反映预测对象的全面情况,还须对其进行整理和汇总。第二手资料是指已经加工整理过的资料。例如,统计年鉴、报刊和杂志上发表的调查统计资料等均属此类。由于第二手资料已经过整理,也称为间接资料。搜集原始资料要耗费较多的人力、物力和时间,同时第二手资料又未必能满足预测



研究的要求,因此常要将两种资料结合起来运用。一般来说,第二手资料便于了解研究对象的环境条件。在工商企业经营预测中,借助于国家及有关部门公布的第二手资料来研究经营环境,是简便有效的做法,往往可收到事半功倍的效果。

按照调查的组织形式区分,有统计报表和专门调查两种方式。统计报表是我国目前主要的调查方式,是了解总体情况的重要资料来源。它属于第二手资料,是以原始记录和基层核算资料为依据,按规定程序报送组织的。专门调查是指专门组织的有针对性的调查,如人口普查、全国经济普查、某地区某类产品需求调查等。统计调查按其所包括的调查单位是否完全,可分为全面调查和抽样调查。按照调查的时间连续性区分,统计调查又分为经常性调查和一次性调查。采取何种调查方式,应根据预测对资料的要求而定。

在多数情况下,收集到的资料还需经过整理才能用于预测。资料整理也就是对资料进行加工使之系统化的工作。资料整理包括以下三个环节。

(1) 对资料的校核

为了保证资料的准确性,必须进行校核以去伪存真。对资料的校核包括逻辑性校核和计算性校核。逻辑性校核是指检查搜集到的资料是否符合预测对象变动的逻辑发展,以排除明显的偶发性因素的影响;计算性校核是指检查搜集到的各种指标数据是否有计算错误,或统计与计算口径是否一致等。

(2) 对资料的分类

按搜集资料所表征的经济社会现象的特征、结构、性质、规模等方面的差异对资料进行分类,是资料整理工作的主要环节。按特征分类,通常是指按资料所显示的变动规律分类,例如,直线型变动形态、曲线型变动形态、季节型变动形态等;按结构分类,一般指按不同的市场结构层次、商品结构层次等分类,例如,国际市场容量、全国市场容量、各区域市场容量、各目标市场容量等;按性质分类,多指按不同的社会性质、经济性质进行分类,例如,人口资料、购买力资料、商品销售资料、商品供应资料等;按规模分类,是指按市场的容量规模、企业产品的生产规模、销售的赢利规模等进行分类。对资料分类取何种标准,决定于预测的任务与目标,也决定于预测方法的选择。

(3) 对变量序列的编制

经过分类整理的资料用数值表示,按不同的变量排序,形成某变量的大小序列。但对于不同的事物,我们能够予以计量或测度的程度是不同的,有些事物只能对它的属性进行分类,比如人口的性别和文化程度、产品的型号及质量等级,等等;有些则可以用比较精确的数字加以计量,比如物体长度、产品的重量和价值,等等。根据我们对事物计量的精确程度和结果来看,可以将统计数据分为以下三种。

第一种是分类数据,这类数据是对事物进行分类的结果,数据表现为类别。例如,人口按性别分为男、女两类;企业按照经济性质分为国有、集体、民营、合资和独资企业等。虽然这些数据只是表现为某种类别,但为了便于统计处理,特别是为了便于计算机识别,我们可以对不同类别用不同的数字或编码来表示。例如,用“1”表示男性人口,“0”表示女性人口;用“1”表示国有企业,“2”表示集体企业,“3”表示民营企业,等等。这些数字只是不同类别的一个代码,并不意味着这些数字可以区分大小或进行任何数学运算。对于类别数据,我们通常是通过计算出每一类别中各元素或个体出现的频数或频率来进行分析的。

第二种是顺序数据,对有些现象的计量不仅可将事物分成不同的类别,而且还可以确定这些类别的优劣或顺序。对于顺序数据,其结果虽然也表现为类别,但这些类别之间是

有顺序的。例如,可以将产品分为一等品、二等品、三等品、次品等;考试成绩可以分为优、良、中、及格、不及格等;受教育水平可以分为小学、初中、高中、大学及以上。

第三种是数值型数据,对有些事物通常使用自然或度量衡单位进行计量,其结果表现为具体的数值。例如,用人民币来度量的收入、用百分制度量的考试成绩、用摄氏度或华氏度度量的温度、用克度量的物体质量等。这些数据都是数值型数据。

区分数据的类型是很重要的,因为对不同类型的数据,将采用不同的统计方法来处理和分析。比如,对分类数据通常计算出各组的频数或频率,计算其众数;对顺序的数据,可以计算其中位数;对数值型数据还可以用更多的统计方法进行处理,如计算各种平均数、进行参数估计和检验等。

3. 数据的分析与预处理

在定量预测中,数据是建立预测模型的基础,缺少数据或数据异常都会导致所建立模型的不准确。所以需要对数据的异常情况进行鉴别与分析。

(1) 数据的分析与鉴别

首先,应对得到的数据作大体的估计,去掉与问题无关或不能说明问题的数据。

其次,对怀疑和需要探讨的数据(如有大起大落的情况)进行研究,调查其产生的背景,鉴别其真实程度,分析原因,以便判断这些受怀疑的数据是否异常或能否反映预测对象的正常情况。

异常数据的鉴别可采用图形观察法和统计滤波法等方法。

【例 6-1】 某企业 2013 年的产品销售量如表 6-1 所示,其图形如图 6-1 所示。

表 6-1 某企业 2013 年产品销售量

单位:吨

月份	1	2	3	4	5	6	7	8	9	10	11	12
销售量	320	330	180	350	340	340	360	350	380	470	350	370

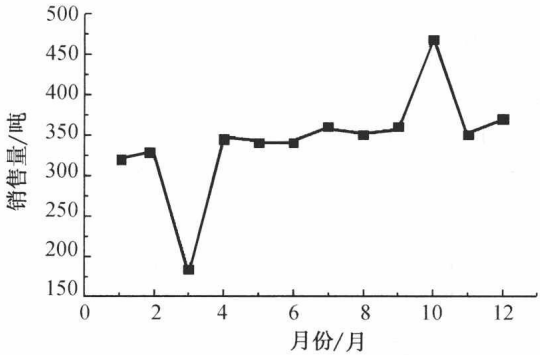


图 6-1 某企业的产品销售量图

观察图 6-1,可见该企业 3 月份的产品销售量严重下降,原因是主要设备出了故障,使产量下降而影响了销售量。10 月份的销售量明显增加,原因是外贸部门根据国际市场行情变化订购了 100 吨外销。通过对这两个数据的背景调查,并对比其他数据,显然这两个数据不能反映该厂产品销售量的正常情况,应视为异常数据。

【例6-2】某商店连续30个月的零售额(单位:万元)如下:12.43, 12.36, 12.30, 12.40, 12.51, 12.66, 12.73, 12.52, 13.01, 12.74, 12.45, 12.75, 12.64, 12.82, 13.10, 13.04, 13.23, 13.52, 18.45, 16.28, 13.50, 13.47, 13.28, 13.62, 13.33, 13.71, 13.44, 13.20, 13.83, 13.60。

可求得 $\bar{y} = 13.30$, $\bar{s} = 1.23$, 若取 $p_1 = 0.95$, $p_2 = 0.99$, 则由 $N = 30$ 查表可得 $k = 3.35$, 从而上、下限分别为: $y_{\pm} = 17.42$ 万元, $y_{\mp} = 9.18$ 万元。

逐项检查原始数据可以看出

$$y_{19} = 18.45 > y_{\pm}$$

因此, 可以认为 y_{19} 为异常数据。

应该引起读者注意的是, 统计滤波法只适用于 $y_1, y_2, \dots, y_N$ 是从一正态分布总体中抽取出来的, 即 N 个样本数据在均值上、下波动这一情况。

(2) 数据的预处理

常用的预处理方法有四种: 剔除法、还原法、拉平法、比例法。

① 剔除法

剔除法就是去掉那些不能如实反映预测对象正常发展趋势的数据。这是最简单的一种方法。如去掉例6-1中某企业2013年产品销售量中3月份和10月份的数据。

对于时间序列数据, 剔除其中某些异常数据建立的预测模型将能较好地描述研究对象的发展规律, 减小误差。剔除法的缺点是破坏了时间序列数据的连续性, 但对因果系统型的横断面数据则没有这种影响。

② 还原法

当数据比较少时, 采用剔除法会使数据变得更少, 而给建立预测模型造成不便, 这时, 可采用还原法。还原意指把数据处理成没有突变因素影响时本应表现出的数值, 当然这只是估计值。

在利用时间序列外推法求趋势方程时, 可用异常数据 y_t 前后两期数据 y_{t-1} 和 y_{t+1} 的算术平均值或几何平均值作为还原值, 即

$$y'_t = (y_{t-1} + y_{t+1})/2 \quad (6-3)$$

或

$$y'_t = \sqrt{y_{t-1} y_{t+1}} \quad (6-4)$$

例如, 利用算术平均值还原例6-1中3月份的数据 y_3 为

$$y'_3 = (y_{t-1} + y_{t+1})/2 = (330 + 350)/2 = 340 \text{ 吨}$$

通常, 当历史数据的发展趋势呈线性时, 取算术平均值较好; 当发展趋势呈非线性时, 取几何平均值较好。

在利用因果关系建立数学模型时, 设有自变量 x 和因变量 y 。为去掉偶然因素对建立模型的影响, 可采用下面的计算方法对数据中的异常数据 y_k 加以还原。

当 x 与 y 之间为线性关系时, 取

$$y'_k = \frac{y_n x_n + y_m x_m}{2x_k} \quad (6-5)$$

当 x 与 y 之间为非线性关系时, 取

$$y''_k = \sqrt{\frac{y_n x_n y_m x_m}{x_k}} \quad (6-6)$$



式中, y'_k 为有偶然因素影响时因变量的估计值; x_k 是与异常数据 y_k 同一时期对应的自变量; x_n, x_m 是与 x_k 在数值上相差最小的两个自变量, 并且有 $x_n \leq x_k \leq x_m$; y_n, y_m 分别是与 x_n, x_m 相对应的因变量值。

【例 6-3】 某日用化工厂生产的洗衣粉, 其主要销售市场是本厂所在市, 这个厂的洗衣粉在所在市的销售量 y 主要受所在市常住人口数量 x 的影响。表 6-3 所列的是所在市历年人口数 x 与洗衣粉销售量 y 的统计数据。

表 6-3 某厂洗衣粉在所在市的历年销售量与所在市的历年人口数

年份/年	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
人口 x /万人	71.5	76.8	80.3	82.6	83.1	83.7	84.5	87.2	88.4	90.5	92.0
销量 y /万千克	42.7	44.9	45.85	45.95	45.15	48.75	32.6	51.65	54.75	55.2	57.3

从表 6-3 中可以看出, 洗衣粉的销售量基本上是随着人口数量的增加而增加的, 但在 2006 年出现了异常情况, 这是因为, 2006 年该厂主要设备出了事故而停产大修以及其他一些原因影响了产量, 投放到市场上的洗衣粉数量随之减少造成的。分别利用式 (6-5) 和式 (6-6) 对 2006 年的销售量进行还原处理, 有

$$y'_{06} = (y_{05}x_{05} + y_{07}x_{07}) / (2x_{06}) = 50.8 \text{ 万千克}$$

$$y''_{06} = \sqrt{(y_{05}x_{05} + y_{07}x_{07}) / x_{06}} = 50.75 \text{ 万千克}$$

可以看出, 二者相差很小, 这是因为 x 与 y 呈线性关系。

在未对 2006 年的销售量进行还原处理时, x 与 y 的相关系数 $r = 0.6097$; 利用处理后的 y'_{06} 代替 y_{06} , 得到的相关系数 $r = 0.9205$ 。显然, 经过还原处理后再建立模型, 效果会更好些。

③拉平法

由于环境条件的变化, 常使一些历史数据不能反映现时的情况。如果数据较多, 则去掉它们即可; 如果数据较少, 去掉它们则会使建立模型增加困难。这时可应用合适的方法来处理这些历史数据, 使之成为对预测有用的数据。拉平法是一个较好的方法, 其原理是: 通过分析造成数据过时的原因, 对数据加以适当的处理, 使其符合现时的实际发展情况。

【例 6-4】 某厂历年的电视机产量如表 6-4 所示。其中 2007 年到 2012 年的产量成台阶形增长。这是因为该厂有一条年生产能力为 8 万台的流水线正式投产, 当年这条新生产线生产了 6 万多台电视机。

表 6-4 某厂电视机的产量

年份/年	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
产量/万台	5.1	5.6	5.7	5.5	11.8	12.3	12.8	13.2	13.8	14.3
处理后产量/万台	11.1	11.6	11.7	11.5	11.8	12.3	12.8	13.2	13.8	14.3

如果去掉 2006 年以前的数据, 剩下数据太少, 不符合建立预测模型的要求。将 2006 年以前的数据都分别加上 2007 年的新增产量 6 万台, 就将 2006 年前后的生产能力拉平了。这些数据就可以共同表现出工人技术水平提高和熟练程度增加等因素造成的生产能力变

化趋势。利用处理后的数据建立预测模型,用以研究该厂生产能力的变化趋势,效果无疑会比未经处理过的数据好。

④比例法

销售条件与环境条件的变化会引起企业产品市场占有率(企业在该地区的产品销售量/该地区的总销售量)的变化。当变化很大时,说明环境条件与销售条件的变化已超过其他因素对销售量的影响,也说明以前销售量统计数据所表现的发展规律不再适用于现在和将来。因此,如果不能去掉以前的数据,就需要进行修改。比例法是一种比较有效的修改方法。

【例 6-5】 某厂牙膏在本地区市场的销售量和市场占有率见表 6-5 所示。

表 6-5 某厂牙膏在本地区的销量情况

年份/年	2003	2004	2005	2006	2007	2008	2009	2010
销售量/万支	653.6	643.8	802.2	629.6	703.2	863.4	895.4	700.8
市场占有率/%	0.65	0.58	0.63	0.57	0.63	0.57	0.59	0.43
处理后销售量/万支	460.7	477.3	547.5	475.0	480.1	651.3	652.6	700.8

从市场占有率来看,2009 年以前保持在 60% 左右,2010 年下降到 43%。原因是 2009 年以前商业部门控制外地牙膏的输入量,从 2010 年开始,国家保护竞争,外地牙膏输入不受控制,由于上海牙膏的进入,故而本厂牙膏销售量下降。为了预测 2011 年以后的牙膏销售量,就需要修改 2009 年以前的数据,修改方法如下:

$$\text{某年处理后的数据} = \frac{\text{该年实际数据} \times \text{2010 年市场占有率}}{\text{该年市场占有率}} \quad (6-7)$$

6.1.4 预测方法的分类

1. 预测方法的分类体系

预测方法可按不同标准进行分类,从而形成了预测方法的分类体系。

(1) 按预测技术的差异性分类

可分为定性预测技术、定量预测技术、定时预测技术、定比预测技术和评价预测技术,共五类。

(2) 按预测方法的客观性分类

可分为主观预测方法和客观预测方法两类。前者主要依靠经验判断,后者主要借助数学模型。

(3) 按预测分析的途径分类

可分为直观型预测方法、时间序列预测方法、计量经济模型预测方法、因果分析预测方法等四类。

(4) 按采用模型的特点分类

可分为经验预测模型和正规预测模型。后者包括时间关系模型、因果关系模型、结构关系模型等。

2. 常用的方法

预测常用方法通常分为定性分析预测法与定量分析预测法两大类。

(1) 定性分析预测法

定性分析预测法亦称为经验判断预测法。它是指预测者根据历史与现实的观察资料, 依赖个人或集体的经验与智慧, 对未来的发展状态和变化趋势作出判断的预测方法。常用的有专家意见法、个人判断法、专家会议法、头脑风暴法、Delphi 法、相关类推法、对比类推法、比例类推法等。

(2) 定量分析预测法

定量分析预测法是依据调查研究所获得的数据资料, 运用统计方法和数学模型, 近似地揭示预测对象及其影响因素的数量变动关系, 建立对应的预测模型, 据此对预测目标作出定量测算的预测方法。通常有时间序列分析预测法和因果分析预测法。

① 时间序列分析预测法。这是以连续性预测原理作指导, 利用历史观察值形成的时间数列, 对预测目标未来状态和发展趋势作出定量判断的预测方法。主要有移动平均法、指数平滑法、趋势外推法、季节指数预测法、ARMA 模型预测法、马尔可夫预测法, 等等。

② 因果分析预测法。这是以因果性预测原理作指导, 以分析预测目标同其他相关事件及现象之间的因果联系, 对市场未来状态与发展趋势作出预测的定量分析方法。主要有回归分析预测法、经济计量模型预测法、投入产出分析预测法、灰色系统模型预测法等。

6.2 灰色系统概论

6.2.1 灰色系统理论的产生

灰色系统理论, 是在一般系统理论的基础上产生的, 它是系统科学思想发展的必然产物, 是社会经济深入发展对科学刺激和需要的产物。当我们认识与研究自然和社会时, 要从系统的角度出发, 从宏观上对其进行深入的剖析和整体把握。在实际中, 我们首先要对事物进行系统性认识, 进而对已有的系统进行有效控制以及设计一些最优系统。对系统进行控制就要通过系统内部和外部的信息和信息流来加以实施, 通过对信息的控制进而达到对系统本身的控制。

但是无论是现代控制理论还是经典控制理论, 它们都要依赖正确而精确的数学模型, 否则, 一切都很难取得满意的结果。然而, 在现实生活中, 有许多情况不大可能求得精确的数学模型, 如工业系统、生物系统、经济系统、社会系统等。若得不出精确的数学模型, 现代控制理论的方法和手段就无法施行, 因而现代控制理论对一些研究对象也鞭长莫及。

当人们对这些问题进行潜心研究时, 查德于 1965 年首创模糊理论, 第一次用精确的数学方式来分析和研究模糊量, 取得了新的突破。随后, 模糊集合论迅速应用于控制领域, 收到了良好的效果。模糊控制能够对一些无法构造数学模型的系统进行控制, 但模糊控制也表现出固有的弱点, 即信息利用率低、控制粗糙、精度低等。因而, 在要求高精度的情况下, 这种控制难以胜任, 并且它也未能对被控对象的运动规律作深刻的阐明, 故模糊控制有它的局限性, 只适用于一些特有的模糊系统。

经典控制理论、现代控制理论和模糊控制理论都有一个共同点, 那就是它们所研究的对象系统必须是白色系统(信息完全确知的系统), 而事实上, 无论是自然系统还是社会系统, 宏观系统还是微观系统, 无生命系统还是有生命系统, 对我们认识的主体来说, 总是信

息不完全的,很难完全地说明一个系统的内部参数。毫无疑问,内部参数不完全的系统具有极为普遍的意义。就像模糊理论的诞生一样,灰色系统理论也应运而生了。

灰色系统理论是我国学者邓聚龙教授于 20 世纪 80 年代初创立并发展的理论,它把一般系统论、信息论和控制论的观点和方法延伸到社会、经济、生态等抽象系统,结合运用数学方法发展的一套解决灰色系统的理论和方法,30 多年来,灰色系统理论引起了国内外学者的广泛关注。灰色系统理论已成功应用到工业、农业、社会、经济等众多领域,解决了生产、生活和科学研究中的大量实际问题。

6.2.2 灰色系统概念及其特点

灰色系统是指“部分信息已知,部分信息未知”的“小样本”,“贫信息”的不确定性系统,它通过对“部分”已知信息的生成、开发去了解、认识现实世界,实现对系统运行行为和演化规律的正确把握与描述。灰色系统具有以下特点。

1. 用灰色数学来处理不确定量并使之量化

在数学发展史上,最早研究的是确定型的微分方程,即在拉普拉斯决定论框架内的数学。它认为一旦有了描写事物的微分方程及初值,就能确知事物任何时候的运动。随后发展了概率论与数理统计,用随机变量和随机过程来研究事物的状态和运动。模糊数学则研究没有清晰界限的事物,如儿童和少年之间没有确定的年龄界限加以截然划分等,它通过隶属函数来使模糊概念量化,因此能用模糊数学来描述如语言、不精确推理以及若干人文科学。灰色系统理论则认为不确定量是灰数,用灰色数学来处理不确定量,同样能使不确定量予以量化。

$$\text{不确定量} \xrightarrow{1,2,3} \text{量化(用确定量的方法研究)}$$

1—概率论与数理统计;2—模糊数学;3—灰色数学(灰色系统理论)

2. 充分利用已知信息寻求系统的运动规律

研究灰色系统的关键是如何使灰色系统白化、模型化、优化。

灰色系统视不确定量为灰色量,提出了灰色系统建模的具体数学方法,它能利用时间序列来确定微分方程的参数。灰色预测不是把观测到的数据序列视为一个随机过程,而是看作随时间变化的灰色量或灰色过程,通过累加生成和累减生成逐步使灰色量白化,从而建立相应于微分方程解的模型并做出预报。这样,对某些大系统和长期预测问题,就可以发挥作用。

3. 灰色系统理论能处理贫信息系统

灰色预测模型只要求较短的观测资料即可,这和时间序列分析、多元分析等概率统计模型要求较长资料很不一样。因此,对于某些只有少量观测数据的项目来说,灰色预测是一种有用的工具。

灰色系统理论经过 30 年的发展,已基本建立起一门新兴的结构体系,其研究内容主要包括灰色系统建模理论、灰色系统控制理论、灰色关联分析方法、灰色预测方法、灰色规划方法、灰色决策方法等。

6.2.3 灰色预测概念

灰色系统分析方法是通过鉴别系统因素之间发展趋势的相似或相异程度,即进行关联



度分析,并通过对原始数据的生成处理来寻求系统变动的规律。生成数据序列有较强的规律性,可以用它来建立相应的微分方程模型,从而预测事物未来的发展趋势和未来状态。

灰色预测是指利用动态 GM 模型,对系统的时间序列进行数量大小的预测,即对系统的主行为特征量或某项指标,发展变化到未来特定时刻出现的数值进行预测。

灰色预测通常分为以下几类。

1. 灰色时间序列预测

用等时距观测到的反映预测对象特征的一系列数量(如产量、销量、人口数量、存款数量、利率等)构造灰色预测模型,预测未来某一时刻的特征量,或者达到某特征量的时间。

2. 畸变预测(灾变预测)

通过模型预测异常值出现的时刻,预测异常值什么时候出现在特定时区内。

3. 波形预测或称为拓扑预测

它是通过灰色模型预测事物未来变动的轨迹。

4. 系统预测

它是对系统行为特征指标建立一组相互关联的灰色预测理论模型,在预测系统整体变化的同时,预测系统各个环节的变化。

6.3 灰色预测模型

6.3.1 灰色生成

将原始数据列中的数据,按某种要求作数据处理称为生成。客观世界尽管复杂,表述其行为的数据可能是杂乱无章的,然而它必然是有序的,都存在着某种内在规律,不过这些规律被纷繁复杂的现象所掩盖,人们很难直接从原始数据中找到某种内在的规律。对原始数据的生成就是企图从杂乱无章的现象中去发现内在规律。

常用的灰色系统生成方式有累加生成、累减生成、均值生成、级比生成等。

1. 累加生成

累加生成,即通过数列间各时刻数据的依次累加以得到新的数据与数列,累加前的数列称为原始数列,累加后的数列称为生成数列。

累加生成是使灰色过程由灰变白的一种方法,它在灰色系统理论中占有极其重要地位,通过累加生成可以看出灰量积累过程的发展态势,使离散的原始数据中蕴含的积分特性或规律加以显化。累加生成是对原始数据列中各时刻的数据依次累加,从而生成新的序列的一种手段。

令 $X^{(0)}$ 为原始序列, $X^{(0)} = [x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n)]$, 记生成数列为 $X^{(1)}$, $X^{(1)} = [x^{(1)}(1), x^{(1)}(2), \dots, x^{(1)}(n)]$, 如果 $X^{(0)}$ 与 $X^{(1)}$ 之间满足如下关系:

$$x^{(1)}(k) = \sum_{i=1}^k x^{(0)}(i), k = 1, 2, \dots, n \quad (6-8)$$

则称为一次累加生成,记为 1-AGO (Accumulating Generation Operator)。则 r 次累加生成满足如下关系:

$$x^{(r)}(k) = \sum_{i=1}^k x^{(r-1)}(i), k = 1, 2, \dots, n \quad (6-9)$$

从公式(6-9)又可以得出从 $r-1$ 次到 r 次的累加为

$$x^{(r)}(k) = \sum_{i=1}^{k-1} x^{(r-1)}(i) + x^{(r-1)}(k) = x^{(r-1)}(k-1) + x^{(r-1)}(k), k = 1, 2, \dots, n$$

$$x^{(r)}(k) = \sum_{i=1}^k x^{(r-1)}(i) = \sum_{i=1}^k \left[\sum_{j=1}^i x^{(r-2)}(j) \right], k = 1, 2, \dots, n$$

累加生成在灰色系统理论中有着非常重要的地位,它能使任意非负数列、摆动的或非摆动的转化为非减的或递增的数列。

2. 累减生成

累减生成,即对数列求相邻两数据的差,累减生成是累加生成的逆运算,常简记为 IAGO (Inverse Accumulated Generating Operation),累减生成可将累加生成还原为非生成数列,在建模过程中用来获得增量信息,其运算符号为 Δ 。

令 $X^{(r)}$ 为 r 次生成数列,对 $X^{(r)}$ 作 r 次累减生成,记为 $\Delta^{(i)}$,其基本关系式为

$$\begin{aligned} \Delta^{(0)}[x^{(r)}(k)] &= x^{(r)}(k) \\ \Delta^{(1)}[x^{(r)}(k)] &= \Delta^{(0)}[x^{(r)}(k)] - \Delta^{(0)}[x^{(r)}(k-1)] \\ \Delta^{(2)}[x^{(r)}(k)] &= \Delta^{(1)}[x^{(r)}(k)] - \Delta^{(1)}[x^{(r)}(k-1)] \\ &\vdots \\ \Delta^{(i)}[x^{(r)}(k)] &= \Delta^{(i-1)}[x^{(r)}(k)] - \Delta^{(i-1)}[x^{(r)}(k-1)] \end{aligned} \quad (6-10)$$

式中, $\Delta^{(0)}(0)$ 为0次累减,即无累减; $\Delta^{(1)}(0)$ 为1次累减,即 $k-1$ 与 k 时刻两个零次累减量求差; $\Delta^{(i)}(0)$ 为 i 次累减,即 $k-1$ 与 k 时刻两个 $i-1$ 次累减量求差。

由公式(6-10)可知

$$\begin{aligned} \Delta^{(1)}[x^{(r)}(k)] &= \Delta^{(0)}[x^{(r)}(k)] - \Delta^{(0)}[x^{(r)}(k-1)] = x^{(r)}(k) - x^{(r)}(k-1) \\ &= \sum_{i=1}^k x^{(r-1)}(i) - \sum_{i=1}^{k-1} x^{(r-1)}(i) = x^{(r-1)}(k) \end{aligned} \quad (6-11)$$

$$\begin{aligned} \Delta^{(2)}[x^{(r)}(k)] &= \Delta^{(1)}[x^{(r)}(k)] - \Delta^{(1)}[x^{(r)}(k-1)] = x^{(r-1)}(k) - x^{(r-1)}(k-1) \\ &= \sum_{i=1}^k x^{(r-2)}(i) - \sum_{i=1}^{k-1} x^{(r-2)}(i) = x^{(r-2)}(k) \end{aligned} \quad (6-12)$$

同理可得

$$\Delta^{(i)}[x^{(r)}(k)] = x^{(r-i)}(k) \quad (6-13)$$

$$\Delta^{(r)}[x^{(r)}(k)] = x^{(0)}(k) \quad (6-14)$$

从公式(6-14)中可以看出,对 r 次生成数列作 r 次累减,即还原为非生成数列。事实上,累加中包含着累减,累减中也包含着累加。比如当 $r=1$ 时,有

$$\begin{aligned} x^{(1)}(k) &= \sum_{i=1}^k x^{(0)}(i) = \sum_{i=1}^{k-1} x^{(0)}(i) + x^{(0)}(k) = x^{(1)}(k-1) + x^{(0)}(k) \\ x^{(0)}(k) &= x^{(1)}(k) - x^{(1)}(k-1) \end{aligned} \quad (6-15)$$

进一步有

$$x^{(r-1)}(k) = x^{(r)}(k) - x^{(r)}(k-1) \quad (6-16)$$

上述关系式经常被用于从生成数列求还原数列中。

3. 均值生成

均值生成分为邻均值生成与非邻均值生成两种。

所谓邻均值生成,是指对于等时距的数列,用相邻数据的平均值构造新的数据,即若有



原始数列 $X = [x(1), x(2), \dots, x(n)]$, 记 k 点的生成值为 $z(k)$, 且 $z(k) = \frac{1}{2}[x(k) + x(k-1)]$, 则称 $z(k)$ 为邻均值生成数, 显然, 这种生成是相邻值的等权生成。

所谓非邻均值生成, 是指对于非等时距的数列, 或虽为等时距数列, 但剔除异常值之后出现空穴的数列, 用空穴两边的数据求平均值构造新的数据以填补空穴, 即若有原始数据 $X = [x(1), x(2), \dots, \varphi(k), x(k+1), \dots, x(n)]$, 这里 $\varphi(k)$ 为空穴, 记 k 点的生成值为 $z(k)$, 且 $z(k) = \frac{1}{2}[x(k-1) + x(k+1)]$, 则称 $z(k)$ 为非邻均值生成数, 显然, 这种生成也是相邻值的等权生成。

4. 级比生成

级比生成在建模中可以获得较好的灰指数率, 级比生成是级比 $\sigma(k)$ 与光滑比 $\rho(k)$ 生成的总称。

设数列 $X^{(0)} = [x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n)]$ 为原始数列, 则其级比 $\sigma(k)$ 与光滑比 $\rho(k)$ 的表达式分别为

$$\sigma(k) = x^{(0)}(k)/x^{(0)}(k-1) \quad (6-17)$$

$$\rho(k) = x^{(0)}(k)/x^{(1)}(k-1) \quad (6-18)$$

设数列 $X^{(0)} = [\varphi(1), x^{(0)}(2), \dots, x^{(0)}(n-1), \varphi(n)]$ 为端点是空穴的数列, 若用 $\varphi(1)$ 右邻的级比生成 $x^{(0)}(1)$, 用 $\varphi(n)$ 左邻的级比生成 $x^{(0)}(n)$, 则称 $x^{(0)}(1)$ 和 $x^{(0)}(n)$ 为级比生成。

6.3.2 灰色预测模型 GM

灰色系统理论是基于关联空间、光滑离散函数等概念定义灰导数与灰微分方程, 进而用离散数列建立微分方程形式的动态模型。由于这是表征灰色系统的基本模型, 而且模型是近似的、非唯一的, 故称这种模型为灰色模型, 记为 GM (Grey Model), 即灰色模型是利用离散随机数经过生成变为随机性被显著削弱而且较有规律的生成数, 建立起的微分方程形式的模型, 这样便于对其变化过程进行研究和描述。

灰色预测通过鉴别系统因素之间发展趋势的相异程度, 并对原始数据进行生成处理来寻找系统变动的规律, 生成有较强规律性的数据序列, 然后建立相应的微分方程模型, 从而预测事物的未来发展趋势。灰色预测的数据是通过生成数据的模型所得到的预测值的逆处理结果。灰色预测是以灰色模型为基础的, 在诸多的灰色模型中, 以灰色系统中单序列一阶线性微分方程模型 GM(1,1) 最为常用。下面简要地介绍 GM(1,1) 模型。

设有非负原始数据列 $X^{(0)} = [x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n)]$, 其中 n 为数据个数。如果根据 $X^{(0)}$ 数据列建立 GM(1,1) 来实现预测功能, 则基本步骤如下。

(1) 原始数据累加以便弱化随机序列的波动性和随机性, 得到新数据序列为

$$X^{(1)} = [x^{(1)}(1), x^{(1)}(2), \dots, x^{(1)}(n)]$$

其中, $x^{(1)}(t)$ 中各数据表示对应前几项数据的累加。

$$x^{(1)}(t) = \sum_{k=1}^t x^{(0)}(k), \quad t = 1, 2, \dots, n \quad (6-19)$$

或

$$x^{(1)}(t+1) = \sum_{k=1}^{t+1} x^{(0)}(k), \quad t = 1, 2, \dots, n \quad (6-20)$$

(2) 对数据 $x^{(1)}(t)$ 建立一阶线性微分方程得

$$\frac{dx^{(1)}(t)}{dt} + ax^{(1)}(t) = u \quad (6-21)$$

其中, a, u 为待定系数, 分别称为发展系数和作用量, a 的有效区间是 $(-2, 2)$, 并记 a, u 构成的矩阵为 $\hat{a} = \begin{pmatrix} a \\ u \end{pmatrix}$ 。只要求出参数 a, u , 就能求出 $x^{(1)}(t)$, 进而求出 $X^{(0)}$ 的未来预测值。

(3) 对累加生成数据做均值生成 B 与常数项向量 Y_n , 即

$$\left\{ \begin{array}{l} B = \begin{bmatrix} \frac{1}{2}(x^{(1)}(1) + x^{(1)}(2)) \\ \frac{1}{2}(x^{(1)}(2) + x^{(1)}(3)) \\ \vdots \\ \frac{1}{2}(x^{(1)}(n-1) + x^{(1)}(n)) \end{bmatrix} \\ Y_n = [x^{(0)}(2), x^{(0)}(3), \dots, x^{(0)}(n)]^T \end{array} \right. \quad (6-22)$$

(4) 用最小二乘法求解灰参数 $\hat{a}$, 则

$$\hat{a} = \begin{pmatrix} a \\ u \end{pmatrix} = (B^T B)^{-1} B^T Y_n \quad (6-23)$$

(5) 将灰参数 $\hat{a}$ 代入 $\frac{dx^{(1)}(t)}{dt} + ax^{(1)}(t) = u$, 并对 $\frac{dx^{(1)}(t)}{dt} + ax^{(1)}(t) = u$ 进行求解, 得

$$\hat{x}^{(1)}(t+1) = \left(x^{(0)}(1) - \frac{u}{a} \right) e^{-at} + \frac{u}{a} \quad (6-24)$$

由于 $\hat{a}$ 是通过最小二乘法求出的近似值, 所以 $\hat{x}^{(1)}(t+1)$ 是一个近似表达式, 为了与原序列 $x^{(1)}(t+1)$ 区分开, 故记为 $\hat{x}^{(1)}(t+1)$ 。

(6) 对函数表达式 $\hat{x}^{(1)}(t+1)$ 及 $\hat{x}^{(1)}(t)$ 进行离散, 并将二者做差以便还原 $X^{(0)}$ 序列, 得到近似数据序列 $\hat{x}^{(0)}(t+1)$ 如下

$$\hat{x}^{(0)}(t+1) = \hat{x}^{(1)}(t+1) - \hat{x}^{(1)}(t) \quad (6-25)$$

(7) 对建立的灰色模型进行检验, 检验步骤如下

① 计算 $x^{(0)}(t)$ 与 $\hat{x}^{(0)}(t)$ 之间的残差 $e^{(0)}(t)$ 和相对误差 $q(x)$:

$$e^{(0)}(t) = x^{(0)}(t) - \hat{x}^{(0)}(t) \quad (6-26)$$

$$q(x) = \frac{e^{(0)}(t)}{x^{(0)}(t)} \quad (6-27)$$

② 求原始数据 $x^{(0)}(t)$ 的均值以及方差 s_1 。

③ 求 $e^{(0)}(t)$ 的平均值 $\bar{q}$ 以及残差的方差 s_2 。

④ 计算方差比 $C = \frac{s_2}{s_1}$ 。

⑤ 求小误差概率 $P = p\{|e(t)| < 0.6745s_1\}$ 。

⑥ 灰色模型精度检验如表 6-6 所示。



表 6-6 灰色模型精度检验对照表

等级	相对误差 q	方差比 C	小误差概率 P
I 级	< 0.01	< 0.35	> 0.95
II 级	< 0.05	< 0.50	< 0.80
III 级	< 0.10	< 0.65	< 0.70
IV 级	> 0.20	> 0.80	< 0.60

在实际应用过程中,检验模型精度的方法并不唯一,可利用上述方法进行模型的检验,也可以根据 $q(x)$ 的误差百分比并结合预测数据与实际数据之间的测试结果酌情认定模型是否合理。

(8) 利用模型进行预测

$$\hat{x}^{(0)} = \left[\underbrace{\hat{x}^{(0)}(1), \hat{x}^{(0)}(2), \dots, \hat{x}^{(0)}(n)}_{\text{原数列的模拟}}, \underbrace{\hat{x}^{(0)}(n+1), \dots, \hat{x}^{(0)}(n+m)}_{\text{未来数列的预测}} \right] \quad (6-28)$$

6.4 灰色预测的 MATLAB 建模仿真实例

灰色预测中有很多关于矩阵的运算,这是 MATLAB 的特长,所以 MATLAB 是实现灰色预测过程的首选。用 MATLAB 编写灰色预测程序时,可按照预测模型的求解步骤,即

- (1) 对原始数据进行累计;
- (2) 构造累加矩阵 B 与常数向量;
- (3) 求解灰参数;
- (4) 将参数带入预测模型进行数据预测。

6.4.1 公司收入预测

已知某公司 1999~2008 年的利润为(单位:元/年) [89 677, 99 215, 109 655, 120 333, 135 823, 159 878, 182 321, 209 407, 246 619, 300 670], 现在预测该公司未来几年的利润情况。

具体的程序如下:

```
clear, clc
syms a b;
c = [ a b ]';
A = [ 89677, 99215, 109655, 120333, 135823, 159878, 182321, 209407, 246619, 300670 ];
B = cumsum( A );           % 原始数据累加
n = length( A );
for i = 1 : ( n - 1 )
    C(i) = ( B(i) + B(i+1) ) / 2;           % 生成累加矩阵
```



```

end
% 计算待定参数的值
D = A; D(1) = [];
D = D';
E = [-C; ones(1, n-1)];
c = inv(E * E') * E * D;
c = c';
a = c(1); b = c(2);
% 预测后续数据
F = []; F(1) = A(1);
for i = 2:(n+10)
    F(i) = (A(1) - b/a) / exp(a * (i - 1)) + b/a;
end
G = []; G(1) = A(1);
for i = 2:(n+10)
    G(i) = F(i) - F(i-1);           % 得到预测出来的数据
end
t1 = 1999:2008;
t2 = 1999:2018;
G
plot(t1, A, 'o', t2, G)           % 原始数据与预测数据的比较
legend('原始数据', '预测数据')
xlabel('年份/年')
ylabel('利润/元')

```

运行该程序,得到的预测数据如下:

```

G =
    1.0e + 06 *
Columns 1 through 12
    0.0897    0.0893    0.1034    0.1196    0.1385    0.1602    0.1854
    0.2146    0.2483    0.2873    0.3325    0.3847
Columns 13 through 20
    0.4452    0.5152    0.5962    0.6899    0.7984    0.9239    1.0691
    1.2371

```

预测数据与原始数据的比较如图 6-2 所示。

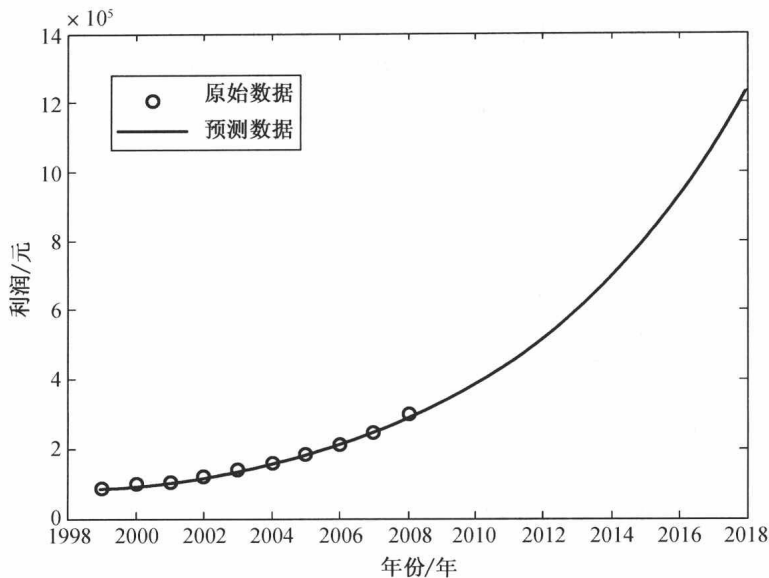


图 6-2 某公司利润预测数据与原始数据的比较示意图

6.4.2 与会代表人数预测

全国大学生数学建模大赛 2009 年赛题(CUMCM 2009 D)要求为会议筹备组制定一个预订宾馆客房、租借会议室、租用客车的合理方案。为了解决这个问题,需要先预测与会代表的人数。预测的依据是代表回执数量及往届的与会人员数据。已知第五届会议的回执情况(见表 6-7)及以往几届会议代表回执和与会情况(见表 6-8)。要解决的问题是:根据这些数据预测本届与会代表人数。

表 6-7 第五届会议的代表回执中有关住房要求的信息 单位:人

	合住 1	合住 2	合住 3	独住 1	独住 2	独住 3
男	154	104	32	107	68	41
女	78	48	17	59	28	19

说明:表头第一行中的数字 1,2,3 分别指每天每间 120 ~ 160 元,161 ~ 200 元,201 ~ 300 元三种不同价格的房间。合住是指要求两人合住一间。独住是指可安排单人间,或一人单独住一个双人间。

表 6-8 往届会议代表回执和与会情况 单位:人

	第一届	第二届	第三届	第四届
发来回执的代表数量	315	356	408	711
发来回执但未与会的代表数量	89	115	121	213
未发回执而与会的代表数量	57	69	75	104

根据表 6-7 的数据,可知本届发来回执的数量为 755。根据表 6-8 的数据,可以知道

发来回执但未与会的代表数和未发回执但与会的代表与发来回执数量间的关系。

定义 1: 缺席率 = 发来回执但未与会的代表数 / 发来回执的代表数量。

定义 2: 未知与会率 = 未发回执但与会的代表的数量 / 发来回执的代表数量。

根据以上定义, 可以得到往届的缺席率和未知与会率, 如表 6-9 所列。

表 6-9 往届的缺席率和未知与会率

	第一届	第二届	第三届	第四届
缺席率	0.282 540	0.323 034	0.296 569	0.299 578
未知与会率	0.180 952	0.193 820	0.183 824	0.146 273

从表 6-9 可以看出, 缺席率一直保持在 0.3 左右, 而未知与会率却变化较快。为此, 认为第五届的缺席率仍为 0.3。这样, 缺席的人数为 $755 \times 0.3 = 226.5$ 人, 为了保守起见, 对 226.5 人进行向下取整, 即缺席的人数为 226 人。

未知与会率变化相对剧烈, 不适合应用比例方法确定, 同时由于数据有限, 所以应用灰色预测方法比较合适。从实际问题的角度, 认为以未知与会率为研究对象较为合适。将往届的未知与会率数据带入下面程序, 对第五届会议的未知进行预测。

```

clear, clc
syms a b;
c = [ a b ]';
A = [ 0.180952 0.193820 0.183824 0.146273 ];
B = cumsum(A);           % 原始数据累加
n = length(A);
for i = 1:(n-1)
    C(i) = (B(i) + B(i+1))/2;   % 生成累加矩阵
end
% 计算待定参数的值
D = A; D(1) = [];
D = D';
E = [ -C; ones(1, n-1) ];
c = inv(E * E') * E * D;
c = c';
a = c(1); b = c(2);
% 预测数据
F = []; F(1) = A(1);
for i = 2:(n+1)
    F(i) = (A(1) - b/a) / exp(a * (i-1)) + b/a;
end
G = []; G(1) = A(1);

```

```
for i=2:(n+1)
    G(i) = F(i) - F(i-1);           %得到预测出来的数据
end
t1 = [1:4];
t2 = [1:5];
G
plot(t1,A,'o',t2,G)                %原始数据与预测数据的比较
legend('原始数据','预测数据')
xlabel('会议届数/届')
ylabel('未知与会率')
```

得到第五届会议的未知与会率为 0.1331, 预测数据与原始数据的比较如图 6-3 所示。

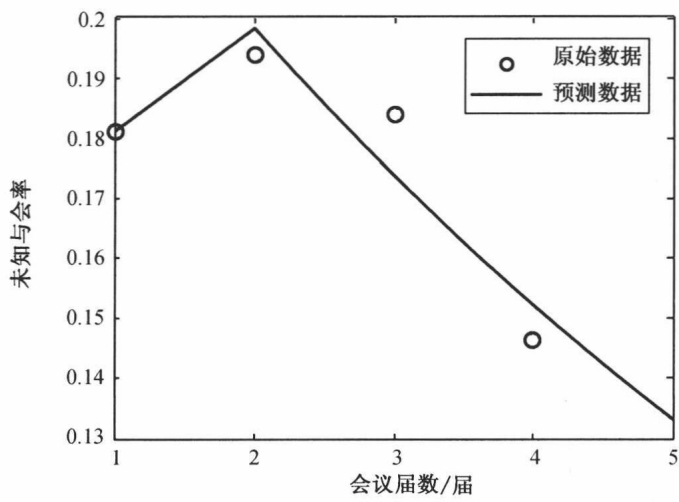


图 6-3 未知与会率预测数据与原始数据对比图

所以第五届会议的未发回执但与会的代表数量为: $755 \times 0.1331 = 100.4905$ 人。同样保守考虑, 向上取整为 101 人。这样就可以预测本届与会代表的数量为 $755 + 101 - 226 = 630$ 人。

6.4.3 SARS 疫情对经济指标影响问题的预测

1. 问题的提出

2003 年的 SARS 疫情对中国部分行业的经济发展产生了一定影响, 特别是对部分疫情严重的省市的相关行业所造成的影响更是显著, 经济影响主要分为直接经济影响和间接经济影响。直接经济影响涉及商品零售业、旅游业等行业。很多方面难以进行定量地评估, 现仅就 SARS 疫情较重的某市的商品零售业、旅游业的影响进行定量的评估分析。已知某市从 1997 年 1 月到 2003 年 12 月的商品零售额、接待旅游人数的统计数据如表 6-10、表 6-11 所示。

表 6-10 商品的零售额

单位:亿元

月份 年份	1 月	2 月	3 月	4 月	5 月	6 月	7 月	8 月	9 月	10 月	11 月	12 月
1997 年	83.0	79.8	78.1	85.1	86.6	88.2	90.3	86.7	93.3	92.5	90.9	96.9
1998 年	101.7	85.1	87.8	91.6	93.4	94.5	97.4	99.5	104.2	102.3	101.0	123.5
1999 年	92.2	114.0	93.3	101.0	103.5	105.2	109.5	109.2	109.6	111.2	121.7	131.3
2000 年	105.0	125.7	106.6	116.0	117.6	118.0	121.7	118.7	120.2	127.8	121.8	121.9
2001 年	139.3	129.5	122.5	124.5	135.7	130.8	138.7	133.7	136.8	138.9	129.6	133.7
2002 年	137.5	135.3	133.0	133.4	142.8	141.6	142.9	147.3	159.6	162.1	153.5	155.9
2003 年	163.2	159.7	158.4	145.2	124.0	144.1	157.0	162.6	171.8	180.7	173.5	176.5

表 6-11 接待海外旅游的人数

单位:万人

月份 年份	1 月	2 月	3 月	4 月	5 月	6 月	7 月	8 月	9 月	10 月	11 月	12 月
1997 年	9.4	11.3	16.8	19.8	20.3	18.8	20.9	24.9	24.7	24.3	19.4	18.6
1998 年	9.6	11.7	15.8	19.9	19.5	17.8	17.8	23.3	21.4	24.5	20.1	15.9
1999 年	10.1	12.9	17.7	21.0	21.0	20.4	21.9	25.8	29.3	29.8	23.6	16.5
2000 年	11.4	26.0	19.6	25.9	27.6	24.3	23.0	27.8	27.3	28.5	32.8	18.5
2001 年	11.5	26.4	20.4	26.1	28.9	28.0	25.2	30.8	28.7	28.1	22.2	20.7
2002 年	13.7	29.7	23.1	28.9	29.0	27.4	26.0	32.2	31.4	32.6	29.2	22.9
2003 年	15.4	17.1	23.5	11.6	1.78	2.61	8.8	16.2	20.1	24.9	26.5	21.8

试根据这些历史数据建立预测评估模型,评估 2003 年 SARS 疫情给该市的商品零售业、旅游业所造成的影响。

2. 模型的分析与假设

根据所掌握的历史统计数据可以看出,在正常情况下,全年的平均值较好地反映了相关指标的变化规律,这样可以把预测评估分成两部分:

(1) 利用灰色理论建立 GM(1,1) 模型,由 1997 ~ 2002 年的平均值预测 2003 年的平均值;

(2) 通过历史数据计算每个月的指标值与全年总值的关系,从而可预测出正常情况下 2003 年每个月的指标值,再与实际值比较就可以估算出 SARS 疫情实际造成的影响。

给出下面两条假设:

(1) 假设该市的统计数据都是可靠准确的;

(2) 假设该市在 SARS 疫情流行期间和结束之后,数据的变化只与 SARS 疫情的影响有关,不考虑其他随机因素的影响。

3. 灰色预测模型求解

由已知数据对于 1997 ~ 2002 年某项指标记为矩阵 $A = (a_{ij})_{6 \times 12}$, 计算每年的年平均值, 记为 $X^{(0)} = (x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(6))$, 并求级比 $\lambda(i) = x^{(0)}(i-1)/x^{(0)}(i) \in (0.7515,$



1.330 7)($i = 2, 3, \cdots, 6$)。对 $X^{(0)}$ 进行一次累加生成, 记 $X^{(1)} = (x^{(1)}(1), x^{(1)}(2), \cdots, x^{(1)}(6))$ 。按照灰色预测模型 GM(1,1) 对 2003 年 1~12 月指标进行预测, 其中商品零售额的预测程序如下:

```
clc,clear
load sars.txt      %把原始数据保存在文件 sars.txt 中
pn = sars(end,:);
sars(end,:) = [ ];
m = size(sars,2);
x0 = mean(sars,2);
x1 = cumsum(x0)
alpha = 0.4;n = length(x0);
z1 = alpha*x1(2:n) + (1 - alpha) * x1(1:n - 1)
Y = x0(2:n);
B = [ -z1,ones(n - 1,1) ];
ab = B\Y
k = 6;
x7 = (x0(1) - ab(2)/ab(1))*(exp(-ab(1)*k) - exp(-ab(1)*(k - 1)))
z = m*x7
u = sum(sars)/sum(sum(sars))
v = z*u
x1 = 1:12;
plot(x1,pn,'o',x1,v)
legend('实际值','预测值');
xlabel('月份')
ylabel('商品零售额/亿元')
```

预测值与实际值的对比数据如表 6-12 所示。

表 6-12 2003 年商品的零售额 单位:亿元

月份	1 月	2 月	3 月	4 月	5 月	6 月	7 月	8 月	9 月	10 月	11 月	12 月
预测值	155.2	157.7	146.4	153.5	160.1	159.8	165.1	163.8	170.5	173.1	169.3	179.8
实际值	163.2	159.7	158.4	145.2	124.0	144.1	157.0	162.6	171.8	180.7	173.5	176.5

同样,对接待海外旅游人数进行预测的预测值与实际值的对比数据如表 6-13 所示。

表 6-13 2003 年接待海外旅游人数 单位:万人

月份	1 月	2 月	3 月	4 月	5 月	6 月	7 月	8 月	9 月	10 月	11 月	12 月
预测值	14.8	26.2	25.5	31.9	33.0	30.8	30.4	37.1	36.7	37.8	33.2	25.5
实际值	15.4	17.1	23.5	11.6	1.78	2.61	8.8	16.2	20.1	24.9	26.5	21.8

商品的零售额、接待海外旅游人数的预测值与实际值的对比情况如图 6-4 和 6-5 所示。

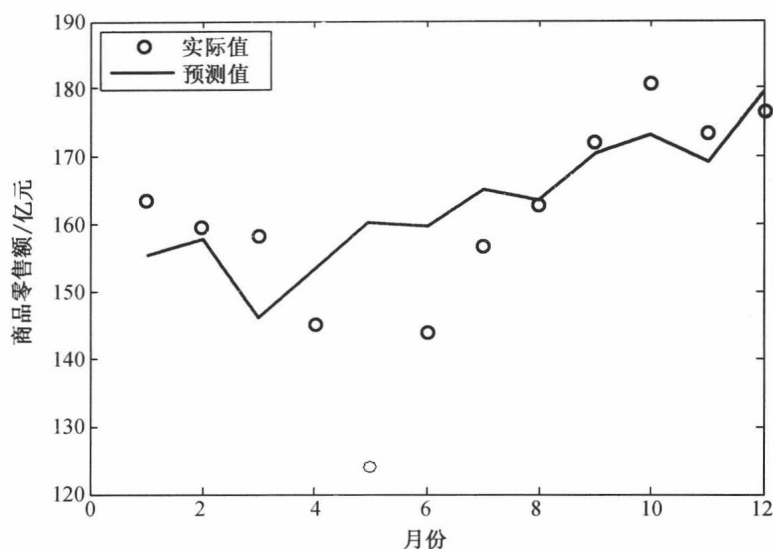


图 6-4 2003 年商品的零售额预测值与实际值对比图

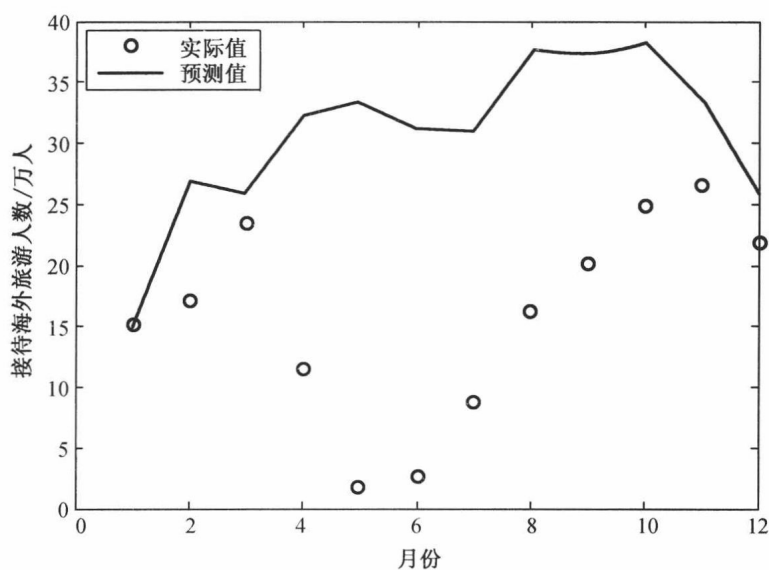


图 6-5 2003 年接待海外旅游人数预测值与实际值对比图

4. 预测结果的分析

根据该市的统计报告显示,2003 年 4、5、6 三个月的实际商品零售额分别为 145.2, 124.0, 144.1 亿元。在这之前,根据统计部门的估计 4、5、6 三个月 SARS 疫情对该市的商品零售业影响最为严重,这三个月的损失估计在 62 亿元左右。从灰色预测结果可以看出,4、5、6 三个月相比实际值损失约为 60.1 亿元,这个数基本与专家的估计值一致,8 月基本恢复



正常,这也说明了模型的正确性和可靠性。

旅游业是受 SARS 疫情影响最严重的行业之一,最严重的 4~7 月四个月就流失 100 多万人,全年大约流失 162 万人。SARS 疫情对旅游业的影响到 2003 年年底基本恢复正常。

该模型虽是就一些经济指标的发展规律进行评估预测而建立的,但类似的预测分析也适用于其他方面的一些数据规律的评估预测问题,即该模型具有很广泛的应用性。

第7章 层次分析法建模仿真

7.1 评价概述

7.1.1 评价概念

评价指评定价值的高低,即评价主体根据一定的评价标准对评价客体做出有无价值及价值大小判断的活动。评价有时是指某人对某事的认同程度,如有时我们说“你对这件事怎么评价”;也可指某人或某事的社会影响和反响,如我们也常说“大家对某某评价不错”,“大家对这件事评价很好”。一般而言,评价就是按照确定的目标,在对被评对象进行系统分析的基础上,测定被评对象的有关属性并将其转变为主观效用的过程,即明确价值的过程。所以,评价包括两个基本点:①在评价目标的指导下,对被评对象进行系统分析,确定评价指标体系及相应的权重体系;②对被评对象的有关属性进行测定,并将其转化为评价者的主观效用。因此,评价是主观、客观相结合的过程,是决策者进行决策的基础。

评价是决策过程中的一项重要活动,通过修正偏离价值标准的行为,使其按预定的目标和方向进行。在一定意义上讲,管理即是在评价中不断修正与完善的过程。评价的功能主要体现在以下几方面。

1. 判断功能

判断包含鉴别与判定,主要是对评价对象的归属进行区分,用以判定对象的某种属性对特定评价目标的从属程度。表现方式可以是评定、认定、评语等。

2. 选择功能

通过事物之间的对比,形成对优劣程度的认识,进而选择价值最大的对象作为最优的选择对象。

3. 导向功能

通过评价找出评价对象的不足之处,引导事物向有利于目标实现的方向发展,促使评价对象在今后不断地完善。

4. 调控功能

持续的评价可以形成对事物发展过程的跟踪认识,通过对事物发展过程偏离目标程度的分析,进而对事物进行调控。

5. 激励功能

评价形成的定性与定量评价结论是对被评价对象实现目标程度的判定,通过对比,可以鞭策后进、鼓励先进,起到对不同层次群体的激励。

6. 探索功能

人们对事物的认识总是由浅到深,由外及里的。评价过程对资料的收集与分析,是对事物深层次再认识的过程。通过评价,可以逐步发现事物发展的规律,进而形成科学的认识。

7.1.2 评价尺度

在进行评价时,首先需要引进和确定评价尺度(标准),然后再通过评价尺度,对评价对象进行测定,并确定其价值。评价尺度类型主要有以下几种。

绝对尺度:规定其原点尺度不变。

间隔尺度:只测得数值差才有意义。

顺序尺度:用数字或反映顺序的字符来表示。

名义尺度:仅仅是为了识别或分类需要而用数字和对象相对应。

7.1.3 评价的原则

原则体现了主体在实施评价活动过程中的基本思想、精神和理念,或者说侧重点和中心。在不同的指导原则下进行评价,会得到不同的评价结果。评价的原则一般包括以下几方面。

1. 客观性原则

在评价过程中,要以评价标准为依据和准绳,基于评价对象的基本特征,实事求是,客观、公正地评价对象,而不掺杂个人的主观臆断。评价活动必须以客观存在的大量事实为依据,一切结论都产生于分析的结尾,而不是其开头。反对先有结论,而后再去搜集罗列个别事实来“论证”,评价一定要恰如其分,不夸大、不缩小。

2. 系统性原则

系统性原则是指在评价中应从系统的整体性、有机联系性、动态性和有序性等特点出发,遵循全面的观点(以防止片面)、相互联系的观点(以防止孤立)、发展的观点(以防止静止僵化)来进行评价,使评价更准确、更概括、更深化。

3. 分类与可比性原则

评价活动是一个复杂的过程,包含多种因素、多重关系,因此,评价过程中应该根据评价对象的不同属性和特点确定相应的评价程序、评价标准和方法,进行分类评价,而不是将不同特点和不同属性的所有对象放在同一标准下评价,采用相同的程序和方法评价。由于被评价对象具有不同“质”性,因而不具有可比性。所以,在评价过程中,必须对评价对象进行科学、准确的类别划分,这样才能最大限度地使评价对象具有可比性,从而保证评价结果的科学性。

4. 科学性原则

科学性原则是指评价的方法、标准、程序和结果的科学性和可靠性,评价过程和结果的可重复性。可重复性是指按照相同的评价过程、相同的评价方法得出相同评价结果的概率。如果得到同一结果的概率大,那么评价结果的科学性、可靠性就高。由于受各种因素的影响,评价结果往往是一种概率事件。但是,评价过程越公正、方法越科学,结果的趋同性必然越强,这样的评价结果也就越科学可靠。

5. 实用和可操作性原则

评价的目的在于指导决策实践,因此,在一定条件下进行的评价既要兼顾全面,又要适当舍弃影响不太大的次级效应,简化评价过程,尽量选择有效指标,迅速得出评价结果。因为实践活动涉及的因素众多、关系复杂,加之评价的方法和评价指标也十分庞杂,不可能面面俱到,所以必须以实用和可操作性原则为基础挑选合适的评价方法和适量的评价指标,

简化评价指标和评价过程,使评价过程具有易操作性和可操作性。

7.1.4 评价的步骤

评价是一项复杂的工作,为了保证整个过程的高效性和有效性,必须遵照一定的程序或步骤来进行。评价的步骤如图 7-1 所示。一个较完整的系统评价步骤一般包括评价系统分析(前提条件探讨)、评价资料的收集、评价指标选择、评价函数确定、价值计算和综合评价等几个阶段。

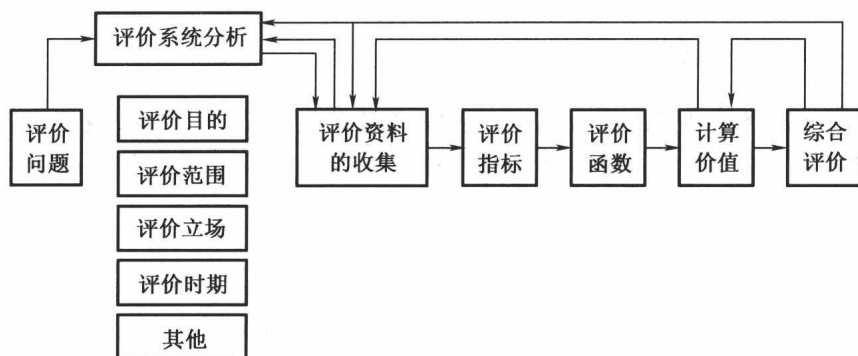


图 7-1 评价的步骤示意图

1. 评价系统分析

在正式进行评价前,有必要对评价系统进行分析,探讨和明确一系列前提条件,这是做好评价的首要工作,评价系统分析主要包括以下几方面。

(1) 明确评价的目的

评价的目的是为了更好地决策,即为选择最优的决策方案。评价的目的大致可分为以下四个方面:

①使评价系统达到最优。为了使系统技术参数达到最优,就有必要量化评价系统各种替代方案的价值。

②对决策的支持。当评价者或决策者在选择最优方案的过程中,对替代方案的各自价值感到迷惑不解时,评价者提供的信息可供决策者参考。

③决定行为的说明。对于复杂的问题即使做出合理的决定,如果没有评价或评价过程模糊不清,也会遭到人们的怀疑、误解甚至抵制,所以为了形成统一意志,需要有某种程度的客观评价。

④问题的分析。评价的过程往往是问题分析的过程。有许多问题利用相关的评价方法可以把复杂的问题分解成简单的小问题,再通过对这些小问题的分析和评价,才能获得系统的综合评价。

(2) 评价系统范围的界定

它主要是确定系统的边界,即评价对象涉及多大范围。评价系统的范围不应过小,以免忽略重要影响部门而有失系统性;同时也不应过大,以免使评价问题过度复杂化。

(3) 评价的立场

在进行评价时必须明确评价主体的立场,这对于以后评价方案的确定、评价项目的选



择等都有直接的影响和关系。

(4) 评价的时期

即评价处于实际问题所处的哪个时期,一般可分为事前评价、事中评价和事后评价。

(5) 评价系统环境的分析

系统环境的分析是指对存在于系统外的物质的、经济的、信息的影响因素进行分析,以了解这些因素对评价系统的影响。系统环境的影响可分为三大类:技术的、经济的、社会的影响。

2. 评价资料的搜集

评价资料的搜集就是对评价系统的功能、效益、耗费、时间及使用寿命进行预测和估计,为设定评价指标、确定评价方法等搜集所需资料。搜集和占有资料是进行系统评价的起点和基础,只有以准确和完整的资料为基础,才能保证整个评价工作的顺利进行。因此,在搜集资料时,要有明确的目的和计划,确保资料的可靠性、代表性、完整性和连续性。

3. 评价指标的选择

评价指标的选择是由评价目标与实际情况共同决定的,具体选择时应注意以下几点:

- ①评价指标必须与评价目的和目标密切相关;
- ②评价指标应当构成一个完整的体系,即全面地反映所要评价对象的各个方面;
- ③评价指标总数应当尽可能地少,以降低评价负担;
- ④科学地、客观地、尽可能全面地考虑各种因素。

4. 评价函数的确定

评价函数是使评价定量化的一种数学模型。不同问题使用的评价函数可能不同,同一个评价问题也可以使用不同的评价函数,因此,对选用什么样的评价函数本身也必须做出评价。一般应选用能更好地达到评价目的的评价函数或其他适应的评价函数。评价函数本身是多属性、多目标的。尤其当评价目的在于形成统一意见或进行群体决策时,则在确定评价函数时会产生不同的看法。因此,在对系统进行评价之前,有关人员应该进行充分的无拘束的讨论,否则难以获得有效的评价。

5. 评价指标价值的计算

从经济学角度讲,价值被理解为根据评价主体的效用观点对于评价对象能满足某种需求的认识或估计。进行评价指标价值的计算,首先要根据前面选定的评价函数对单项指标进行价值评价,查明方案在各项评价指标上的实现程度,然后进行单项评价指标的综合,得出某一大类指标的价值。在指标价值计算之前,一般还需要确定各评价指标的权重。总之,评价尺度和评价项目的权重应保证评价的客观正确和有效。

6. 综合评价

综合评价就是对系统进行技术、经济、社会等方面的全面评价。但综合评价的各个方面和评价项目不能一概而论,应根据具体评价对象而定。如以一个企业开发新产品为例,一个完整的综合评价体系大致包括以下几个方面:经营管理方面、技术方面、市场方面、时间方面、经济方面、体制方面和社会方面等。

7.2 层次分析法的基本原理

层次分析法 (Analytic Hierarchy Process, 简称 AHP) 是对一些较为复杂、较为模糊的问题作出决策的简易方法, 它特别适用于那些难于完全定量分析的问题。它是美国运筹学家 T. L. Saaty 教授于 20 世纪 70 年代初期提出的一种简便、灵活而又实用的多准则决策方法。

运用层次分析法建模, 大体上可按下面四个步骤进行:

- ①建立递阶层次结构模型;
- ②构造出各层次中的所有判断矩阵;
- ③层次单排序及一致性检验;
- ④层次总排序及一致性检验。

7.2.1 递阶层次结构的建立与特点

应用 AHP 分析决策问题时, 首先要把问题条理化、层次化, 构造出一个有层次的结构模型。在这个模型下, 复杂问题被分解为元素的组成部分。这些元素又按其属性及关系形成若干层次。上一层次的元素作为准则对下一层次有关元素起支配作用。

这些层次可以分为三类。

1. 目标层

这一层次中只有一个元素, 一般它是分析问题的预定目标或理想结果, 因此也称为目标层。

2. 准则层

这一层次中包含了为实现目标所涉及的中间环节, 它可以由若干个层次组成, 包括所需考虑的准则、子准则。

3. 方案层

这一层次包括了为实现目标可供选择的各种措施、决策方案等。

递阶层次结构中的层次数与问题的复杂程度及需要分析的详尽程度有关, 一般情况层次数不受限制。每一层次中各元素所支配的元素一般不要超过 9 个。这是因为支配的元素过多会给两两比较判断带来困难。

下面结合一个实例来说明递阶层次结构的建立。

【例 7-1】 假期旅游有 P1, P2, P3 三个旅游胜地供你选择, 试确定一个最佳地点。在此问题中, 你会根据诸如景色、费用、居住、饮食和旅途条件等一些准则去反复比较三个候选地点。可以建立如图 7-2 所示的层次结构模型。

7.2.2 构造判断矩阵

层次结构反映了因素之间的关系, 但准则层中的各准则在目标衡量中所占的比重并不一定相同, 在决策者的心目中, 它们各占有一定的比例。

在确定影响某因素的诸因子在该因素中所占的比重时, 遇到的主要困难是这些比重常常不易量化。此外, 当影响某因素的因子较多时, 直接考虑各因子对该因素有多大程度的影响时, 常常会因考虑不周全、顾此失彼而使决策者提出与他实际认为的重要性程度不

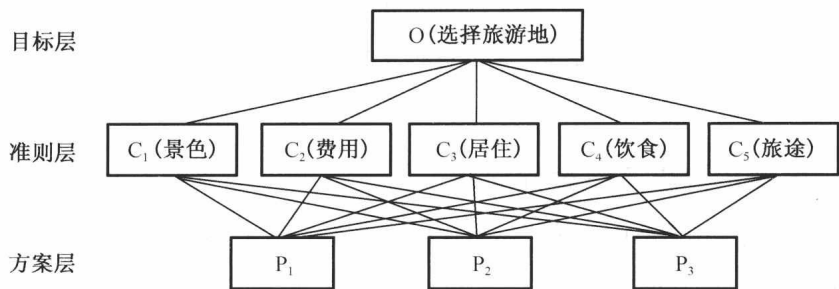


图 7-2 旅游地评价的层次结构模型图

相一致的数据,甚至有可能提出一组隐含矛盾的数据。

假设要比较 n 个因子 $X = \{x_1, x_2, \dots, x_n\}$ 对某因素 Z 的影响大小,怎样比较才能提供可信的数据呢? Saaty 等人建议可以采取对因子进行两两比较建立成对比较矩阵的办法。即每次取两个因子 x_i 和 $x_j (i, j = 1, 2, \dots, n)$, 以 a_{ij} 表示 x_i 和 x_j 对 Z 的影响大小之比,全部比较结果用矩阵 $A = (a_{ij})_{n \times n}$ 表示,称 A 为 $Z-X$ 之间的成对比较判断矩阵(简称判断矩阵)。容易看出,若 x_i 与 x_j 对 Z 的影响之比为 a_{ij} , 则 x_j 与 x_i 对 Z 的影响之比应为 $a_{ji} = \frac{1}{a_{ij}}$ 。

定义 1 若矩阵 $A = (a_{ij})_{n \times n}$ 满足 $a_{ij} > 0$ 和 $a_{ji} = \frac{1}{a_{ij}} (i, j = 1, 2, \dots, n)$, 则称之为正互反矩阵(易见 $a_{ii} = 1, i = 1, 2, \dots, n$)。

关于如何确定 a_{ij} 的值, Saaty 等建议引用数字 1~9 及其倒数作为标度。表 7-1 列出了 1~9 标度的含义。

表 7-1 标度的含义

标度	含义
1	表示两个因素相比,具有相同重要性
3	表示两个因素相比,前者稍微重要
5	表示两个因素相比,前者明显重要
7	表示两个因素相比,前者强烈重要
9	表示两个因素相比,前者极端重要
2,4,6,8	表示上述相邻判断的中间值
倒数	若因素 i 与因素 j 的重要性之比是 a_{ij} , 则因素 j 与因素 i 的重要性之比是 $a_{ji} = 1/a_{ij}$

从心理学观点来看,分级太多会超越人们的判断能力,既增加了判断的难度,又容易因此而提供虚假数据。Saaty 等人还用实验方法比较了在各种不同标度下人们判断结果的正确性,实验结果也表明,采用 1~9 标度最为合适。

最后应该指出,一般地作 $\frac{n(n-1)}{2}$ 次两两判断是必要的。有人认为把所有元素都和某个元素比较,即只作 $n-1$ 次比较就可以了。这种作法的弊病在于,任何一个判断的失误均

可导致不合理的排序,而个别判断的失误对于难以定量的系统往往是难以避免的。进行 $\frac{n(n-1)}{2}$ 次比较可以提供更多的信息,通过各种不同角度的反复比较,从而得出一个合理的排序。

7.2.3 层次单排序及一致性检验

判断矩阵 A 对应于最大特征值 $\lambda_{\max}$ 的特征向量 W ,经归一化后即为一层次相应因素对于上一层次某因素相对重要性的排序权值,这一过程称为层次单排序。

上述构造成对比较判断矩阵的办法虽能减少其他因素的干扰,较客观地反映出一对因子影响力的差别,但综合全部比较结果时,其中难免包含一定程度的非一致性。如果比较结果是前后完全一致的,则矩阵 A 的元素还应当满足:

$$a_{ij}a_{jk} = a_{ik}, \forall i, j, k = 1, 2, \dots, n \quad (7-1)$$

定义 2 满足关系式(7-1)的正互反矩阵称为一致矩阵。

需要检验构造出来的(正互反)判断矩阵 A 是否严重地非一致,以便确定是否接受 A 。

定理 1 正互反矩阵 A 的最大特征根 $\lambda_{\max}$ 必为正实数,其对应特征向量的所有分量均为正实数。 A 的其余特征值的模均严格小于 $\lambda_{\max}$ 。

定理 2 若 A 为一致矩阵,则

① A 必为正互反矩阵;

② A 的转置矩阵 A^T 也是一致矩阵;

③ A 的任意两行成比例,比例因子大于零,从而 $\text{rank}(A) = 1$ (同样, A 的任意两列也成比例);

④ A 的最大特征值 $\lambda_{\max} = n$, 其中 n 为矩阵 A 的阶, A 的其余特征根均为零;

⑤ 若 A 的最大特征值 $\lambda_{\max}$ 对应的特征向量为 $W = (w_1, w_2, \dots, w_n)^T$, 则 $a_{ij} = \frac{w_i}{w_j}, \forall i, j = 1, 2, \dots, n$, 即

$$A = \begin{pmatrix} w_1/w_1 & w_1/w_2 & \cdots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & \cdots & w_2/w_n \\ \vdots & \vdots & & \vdots \\ w_n/w_1 & w_n/w_2 & \cdots & w_n/w_n \end{pmatrix}$$

定理 3 n 阶正互反矩阵 A 为一致矩阵,当且仅当其最大特征根 $\lambda_{\max} = n$, 且当正互反矩阵 A 非一致时,必有 $\lambda_{\max} > n$ 。

根据定理 3,我们可以由 $\lambda_{\max}$ 是否等于 n 来检验判断矩阵 A 是否为一致矩阵。由于特征根连续地依赖于 a_{ij} ,故 $\lambda_{\max}$ 比 n 大得越多, A 的非一致性程度也就越严重, $\lambda_{\max}$ 对应的标准化特征向量也就越不能真实地反映出 $X = \{x_1, x_2, \dots, x_n\}$ 在对因素 Z 的影响中所占的比重。因此,对决策者提供的判断矩阵有必要作一次一致性检验,以决定是否能接受它。

对判断矩阵的一致性检验的步骤如下。

(1) 计算一致性指标 CI

$$CI = \frac{\lambda_{\max} - n}{n - 1} \quad (7-2)$$



(2) 查找相应的平均随机一致性指标 RI

对于 $n = 1, \dots, 9$, Saaty 给出了 RI 的值, 如表 7-2 所示。

表 7-2 RI 的值

n	1	2	3	4	5	6	7	8	9	10	11
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51

RI 的值是这样得到的, 用随机方法构造 500 个样本矩阵, 随机地从 1 ~ 9 及其倒数中抽取数字构造正互反矩阵, 求得最大特征根的平均值 $\lambda'_{\max}$, 并定义

$$RI = \frac{\lambda'_{\max} - n}{n - 1} \tag{7-3}$$

(3) 计算一致性比例 CR

$$CR = \frac{CI}{RI} \tag{7-4}$$

当 $CR < 0.10$ 时, 认为判断矩阵的一致性是可以接受的, 否则应对判断矩阵作适当修正。

7.2.4 层次总排序及一致性检验

上面我们得到的是一组元素对其上一层中某元素的权重向量。我们最终要得到各元素, 特别是最底层中各方案对于目标的排序权重, 从而进行方案选择。总排序权重要自上而下地将单准则下的权重进行合成, 如表 7-3 所示。

表 7-3 层次总排序合成表

层次 A 层次 B	A_1	A_2	$\dots$	A_m	B 层次 总排序权重
	a_1	a_2	$\dots$	a_m	
B_1	b_{11}	b_{12}	$\dots$	b_{1m}	$\sum_{j=1}^m a_j b_{1j}$
B_2	b_{21}	b_{22}	$\dots$	b_{2m}	$\sum_{j=1}^m a_j b_{2j}$
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
B_n	b_{n1}	b_{n2}	$\dots$	b_{nm}	$\sum_{j=1}^m a_j b_{nj}$
CI	$(CI)_1$	$(CI)_2$	$\dots$	$(CI)_m$	

设上一层次(A 层)包含 $A_1, A_2, \dots, A_m$ 共 m 个因素, 它们的层次总排序权重分别为 $a_1, a_2, \dots, a_m$ 。又设其后的下一层次(B 层)包含 n 个因素 $B_1, B_2, \dots, B_n$, 它们关于 A_j 的层次单排序权重分别为 $b_{1j}, b_{2j}, \dots, b_{nj}$ (当 B_i 与 A_j 无关联时, $b_{ij} = 0$)。现求 B 层中各因素关于总目标的权重, 即求 B 层各因素的层次总排序权重 $b_1, b_2, \dots, b_n$, 计算按表 7-3 所示方式进行, 即

$$b_i = \sum_{j=1}^m b_{ij} a_j, i = 1, \dots, n。$$

对层次总排序也需作一致性检验,检验仍像层次总排序那样由高层到低层逐层进行。这是因为虽然各层次均已经过层次单排序的一致性检验,各成对比较判断矩阵都已具有较为满意的一致性,但当综合考察时,各层次的非一致性仍有可能积累起来,引起最终分析结果较严重的非一致性。

设 B 层中与 A_j 相关的因素的成对比较判断矩阵在单排序中经一致性检验,求得单排序一致性指标为 $CI(j)$, $(j=1, \dots, m)$, 相应的平均随机一致性指标为 $RI(j)$ ($CI(j)$, $RI(j)$ 已在层次单排序时求得), 则 B 层总排序随机一致性比例为

$$CR = \frac{\sum_{j=1}^m CI(j) a_j}{\sum_{j=1}^m RI(j) a_j} \quad (7-5)$$

当 $CR < 0.10$ 时,认为层次总排序结果具有较满意的一致性并接受该分析结果。

7.2.5 层次分析法的优点和局限性

1. 层次分析法的优点

(1) 系统性

层次分析法把研究对象作为一个系统,按照分解、比较判断、综合的思维方式进行决策,成为继机理分析、统计分析之后发展起来的系统分析的重要工具。

(2) 实用性

层次分析法把定性和定量方法结合起来,能处理许多用传统的最优化技术无法着手的实际问题,应用范围很广,同时,这种方法使得决策者与决策分析者能够相互沟通,决策者甚至可以直接应用它,这就增加了决策的实用性。

(3) 简洁性

具有中等文化程度的人即可以了解层次分析法的基本原理并掌握该法的基本步骤,计算也非常简便,并且所得结果简单明确,容易被决策者了解和掌握。

2. 层次分析法的局限性

(1) 只能从原有的方案中优选一个出来,没有办法得出更好的新方案。

(2) 在很大程度上依赖于人们的经验,主观因素的影响较大,它至多只能排除思维过程中的严重非一致性,却无法排除决策者个人可能存在的严重片面性。

(3) 该法中的比较、判断以及结果的计算过程都是粗糙的,不适用于精度较高的问题。

7.3 层次分析法的 MATLAB 建模仿真实例

在应用层次分析法研究问题时,遇到的主要困难有两个:①如何根据实际情况抽象出较为贴切的层次结构;②如何将某些定性的量作比较接近实际的定量化处理。

在应用层次分析法时,建立层次结构模型是十分关键的一步。现在分析一个实例,以便说明如何从实际问题中抽象出相应的层次结构。

【例 7-2】挑选合适的工作。经双方恳谈,已有三个单位表示愿意录用某毕业生。该生根据已有信息建立了一个层次结构模型,如图 7-3 所示。

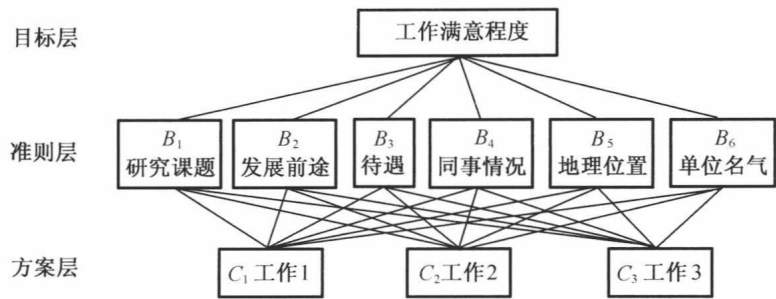


图 7-3 工作评价的层次结构模型

准则层的判断矩阵如表 7-4 所示。

表 7-4 准则层的判断矩阵

<i>A</i>	<i>B</i> ₁	<i>B</i> ₂	<i>B</i> ₃	<i>B</i> ₄	<i>B</i> ₅	<i>B</i> ₆
<i>B</i> ₁	1	1	1	4	1	1/2
<i>B</i> ₂	1	1	2	4	1	1/2
<i>B</i> ₃	1	1/2	1	5	3	1/2
<i>B</i> ₄	1/4	1/4	1/5	1	1/3	1/3
<i>B</i> ₅	1	1	1/3	3	1	1
<i>B</i> ₆	2	2	2	3	3	1

方案层的判断矩阵如表 7-5 所示。

表 7-5 方案层的判断矩阵

<i>B</i> ₁	<i>C</i> ₁	<i>C</i> ₂	<i>C</i> ₃	<i>B</i> ₂	<i>C</i> ₁	<i>C</i> ₂	<i>C</i> ₃	<i>B</i> ₃	<i>C</i> ₁	<i>C</i> ₂	<i>C</i> ₃
<i>C</i> ₁	1	1/4	1/2	<i>C</i> ₁	1	1/4	1/5	<i>C</i> ₁	1	3	1/3
<i>C</i> ₂	4	1	3	<i>C</i> ₂	4	1	1/2	<i>C</i> ₂	1/3	1	1/7
<i>C</i> ₃	2	1/3	1	<i>C</i> ₃	5	2	1	<i>C</i> ₃	3	7	1
<i>B</i> ₄	<i>C</i> ₁	<i>C</i> ₂	<i>C</i> ₃	<i>B</i> ₅	<i>C</i> ₁	<i>C</i> ₂	<i>C</i> ₃	<i>B</i> ₆	<i>C</i> ₁	<i>C</i> ₂	<i>C</i> ₃
<i>C</i> ₁	1	1/3	5	<i>C</i> ₁	1	1	7	<i>C</i> ₁	1	7	9
<i>C</i> ₂	3	1	7	<i>C</i> ₂	1	1	7	<i>C</i> ₂	1/7	1	1
<i>C</i> ₃	1/5	1/7	1	<i>C</i> ₃	1/7	1/7	1	<i>C</i> ₃	1/9	1	1

计算的 MATLAB 程序如下:

```

clc,clear
fid=fopen('txt3.txt','r');
n1=6;n2=3;
a=[];
for i=1:n1
    tmp=str2num(fgetl(fid));
    a=[a;tmp]; %读准则层的判断矩阵
end
for i=1:n1
    str1=char(['b',int2str(i),'=';]);
    str2=char(['b',int2str(i),'=';tmp;]);
eval(str1);
    for j=1:n2
        tmp=str2num(fgetl(fid));
        eval(str2); %读方案层的判断矩阵
    end
end
ri=[0,0,0.58,0.90,1.12,1.24,1.32,1.41,1.45]; %一致性指标
[x,y]=eig(a);
lamda=max(diag(y));
num=find(diag(y)==lamda);
W0=x(:,num)/sum(x(:,num));
cr0=(lamda-n1)/(n1-1)/ri(n1);
for i=1:n1
    [x,y]=eig(eval(char(['b',int2str(i)])));
    lamda=max(diag(y));
    num=find(diag(y)==lamda);
    W1(:,i)=x(:,num)/sum(x(:,num));
    cr1(i)=(lamda-n2)/(n2-1)/ri(n2);
end
cr1,ts=W1*W0,cr=cr1*W0

```

纯文本文件 txt3. txt 中的数据格式如下：

1 1/2 1 5 3 1
1/4 1/4 1/5 1 1/3 1/3
1 1 1/3 3 1 1
2 2 2 3 3 1
1 1/4 1/2
4 1 3
2 1/3 1
1 1/4 1/5
4 1 1/2
5 2 1
1 3 1/3
1/3 1 1/7
3 7 1
1 1/3 5
3 1 7
1/5 1/7 1
1 7 9
1/7 1 1
1/9 1 1

运行得到方案层单排序一致性比例：

cr1 =
0.0158 0.0212 0.0061 0.0559 -0.0000 0.0061

总排序权值：

ts =
0.3952 0.2996 0.3052

方案层总排序一致性比例:cr=0.011 7。

层次总排序的结果如表 7-6 所示。

表 7-6 层次总排序结果

准则		研究 课题	发展 前途	待遇	同事 情况	地理 位置	单位 名气	总排序 权值
准则层权值		0.150 7	0.179 2	0.188 6	0.047 2	0.146 4	0.287 9	
方案层 单排序 权值	工作 1	0.136 5	0.097 4	0.242 6	0.279 0	0.466 7	0.798 6	0.395 2
	工作 2	0.625 0	0.333 1	0.087 9	0.649 1	0.466 7	0.104 9	0.299 6
	工作 3	0.238 5	0.569 5	0.669 4	0.071 9	0.066 7	0.096 5	0.305 2

根据总排序权值,该毕业生最满意的工作应该是工作 1。

第 8 章 现代优化模型与仿真

8.1 最优化问题概述

人类所从事的一切生产或社会活动均是有目的的,其行为总是在特定的价值观念或审美取向的支配下进行的,经常面临求解一个可行的甚至是最优的方案的决策问题。

在实际工作中,人们常常会遇到下面这样一些问题:

①在安排生产计划方面,如何在现有的人力、物力等条件下,合理安排生产,使总产值或总利润最高。

②在生产工艺确定方面,如何在保证产品质量的前提下,选择合理的操作方式,使操作费用最低。

③在产品设计方面,如何选择参数使设计既满足要求,成本又最低。

④在配料方面,如何合理配料,在保证质量要求的前提下使成本最低。

⑤在资源分配中,如何使分配的方案既能满足要求,又能取得较好的经济效益。

上述这些问题都可以转化为最优化问题,需要寻求最优策略。

8.1.1 最优化问题涉及的基本概念

1. 最优化问题

在工业、农业、交通运输、商业、国防、建筑、通信、政府机关等各部门各领域的实际工作中,我们经常会遇到求函数的极值或最大值最小值问题,这一类问题我们称之为最优化问题。而求解最优化问题的数学方法被称为最优化方法。它主要解决最优生产计划、最优分配、最佳设计、最优决策、最优管理等最大值最小值问题。

最优化问题的目的有两个:①求出满足一定条件下,函数的极值或最大值最小值;②求出取得极值时变量的取值。

最优化问题所涉及的内容种类繁多,有的十分复杂,但是它们都有共同的关键因素:变量、约束条件和目标函数。

2. 变量

变量是指最优化问题中所涉及的与约束条件和目标函数有关的待确定的量。一般来说,它们都有一些限制条件(约束条件),与目标函数紧密关联。

设问题中涉及的变量为 $x_1, x_2, \dots, x_n$, 常常也用 $X = (x_1, x_2, \dots, x_n)$ 表示。

3. 约束条件

在最优化问题中,求目标函数的极值时,变量必须满足的限制称为约束条件。

例如,许多实际问题中变量要求必须非负,这是一种限制;在研究电路优化设计问题时,变量必须服从电路基本定律,这也是一种限制,等等。在研究问题时,这些限制我们必须用数学表达式准确地描述它们。

用数学语言描述约束条件一般来说有两种。

(1) 等式约束条件

$$g_i(X) = 0, \quad i = 1, 2, \dots, m \quad (8-1)$$

(2) 不等式约束条件

$$h_i(X) \geq 0, \quad i = 1, 2, \dots, r$$

或

$$h_i(X) < 0, \quad i = 1, 2, \dots, r \quad (8-2)$$

注:在最优化问题研究中,由于解的存在性十分复杂,一般来说,我们不考虑不等式约束条件 $h(X) > 0$ 或 $h(X) < 0$ 。这两种约束条件最优化问题最优解的存在性较复杂。

4. 目标函数

在最优化问题中,与变量有关的待求其极值(或最大值、最小值)的函数称为目标函数。

目标函数常用 $f(X) = f(x_1, x_2, \dots, x_n)$ 表示。当目标函数为某问题的效益函数时,问题即为求极大值;当目标函数为某问题的费用函数时,问题即为求极小值,等等。

实际上求极大值和极小值问题没有原则上的区别,因为求 $f(X)$ 的极小值,也就是要求 $-f(X)$ 的极大值,两者的最优值在同一点取到。

8.1.2 最优化问题分类

最优化问题种类繁多,因而分类的方法也有许多。可以按变量的性质分类,按有无约束条件分类,按目标函数的个数分类,等等。

一般来说,变量可以分为确定性变量、随机变量和系统变量等,相对应的最优化问题分别称为普通最优化问题、统计最优化问题和系统最优化问题。

1. 按有无约束条件分类

分为无约束最优化问题和有约束最优化问题。

2. 按目标函数的个数分类

分为单目标最优化问题和多目标最优化问题。

3. 按约束条件和目标函数是否是线性函数分类

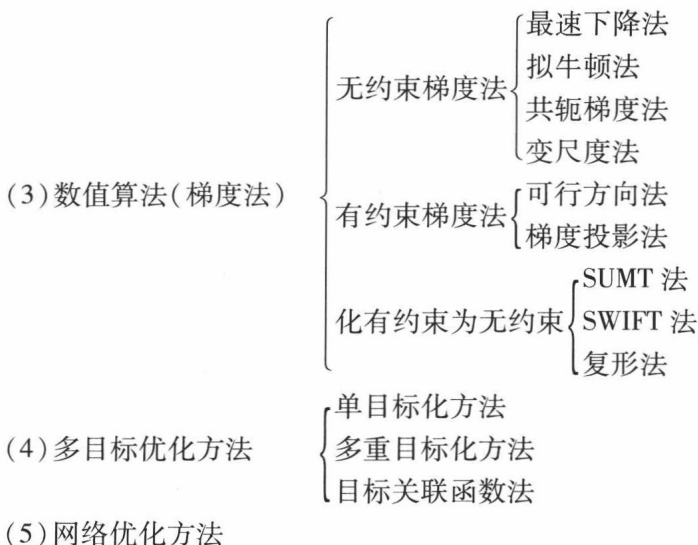
分为线性最优化问题(线性规划)和非线性最优化问题(非线性规划)。

4. 按约束条件和目标函数是否是时间的函数分类

分为静态最优化问题和动态最优化问题(动态规划)。

5. 按最优化问题求解方法分类

$$\begin{aligned}
 (1) \text{解析法(间接法)} & \begin{cases} \text{无约束} \begin{cases} \text{古典微分法} \\ \text{古典变分法} \end{cases} \\ \text{有约束} \begin{cases} \text{极大值原理} \\ \text{库恩-图克定理} \end{cases} \end{cases} \\
 (2) \text{数值算法(直接法)} & \begin{cases} \text{一维搜索法} \begin{cases} \text{斐波那西法} \\ \text{黄金分割法} \\ \text{插值法} \end{cases} \\ \text{多维搜索法} \begin{cases} \text{坐标轮换法} \\ \text{步长加速法} \\ \text{方向加速法} \\ \text{单纯形法} \\ \text{随机搜索法} \end{cases} \end{cases}
 \end{aligned}$$



8.1.3 最优化问题的求解步骤

最优化问题的求解涉及应用数学、计算机科学以及各专业领域,是一个十分复杂的问题,然而它却是需要我们重点关注的问题之一。怎样研究分析求解这类问题呢?其中最关键的是建立数学模型和求解数学模型。一般来说,应用最优化方法解决实际问题可分为以下四个步骤进行。

步骤 1:建立模型

提出最优化问题,包括:变量是什么,约束条件有哪些,目标函数是什么。建立最优化问题数学模型:确定变量,建立目标函数,列出约束条件。

步骤 2:确定求解方法

分析模型,根据数学模型的性质,选择优化求解方法。

步骤 3:计算机求解

编程序(或使用数学计算软件),应用计算机求最优解。

步骤 4:结果分析

对算法的可行性、收敛性、通用性、时效性、稳定性、灵敏性和误差等作出评价。

8.1.4 现代优化算法

现代优化算法又称智能优化算法或现代启发式算法,是一种具有全局优化性能、通用性强且适合于并行处理的算法。这种算法一般具有严密的理论依据,而不是单纯凭借专家经验,理论上可以在一定的时间内找到最优解或近似最优解。常用的现代优化算法包括禁忌搜索算法、模拟退火算法、遗传算法、蚁群算法、粒子群算法等。

相对于传统优化算法,现代优化算法从任一解出发,按照某种机制,以一定的概率在整个求解空间中探索最优解。由于它们可以把搜索空间扩展到整个问题空间,因而具有全局优化性能。现代优化算法具有如下特点:

- ①不依赖于初始条件;
- ②不与求解空间有紧密关系,对解域无可微或连续的要求;
- ③算法容易实现,求解稳健;



④能获得全局最优,适合于求解空间不知的情况;

⑤可应用于大规模、多峰多态函数、含离散变量等全局优化问题,且求解速度和质量远超过常规方法。

8.2 遗传算法及其 MATLAB 仿真

8.2.1 遗传算法的基本原理

遗传算法 (Genetic Algorithm, GA) 是近几年发展起来的一种崭新的全局优化算法。1962 年霍兰 (Holland) 教授首次提出了 GA 算法的思想,遗传算法借用了仿真生物遗传学和自然选择机理,通过自然选择、遗传、变异等作用机制,实现各个个体的适应性的提高。

从某种程度上说遗传算法是对生物进化过程进行的数学方式仿真。这一点体现了自然界中“物竞天择、适者生存”进化过程。与自然界相似,遗传算法对求解问题的本身一无所知,它所需要的仅是对算法所产生的每个染色体进行评价,把问题的解表示成染色体,并基于适应值来选择染色体,使适应性好的染色体有更多的繁殖机会。遗传算法与生物进化的对应关系如表 8-1 所示。

表 8-1 遗传算法与生物进化的对应关系

生物进化	遗传算法
环境	适应函数
适者生存	适应函数值最大的解被保留的概率最大
个体	问题的一个解
染色体	解的编码
基因	编码的元素
群体	被选定的一组解
种群	根据适应函数选择的一组解
交叉	以一定的方式由双亲产生后代的过程
变异	编码的某些分量发生变化的过程

遗传算法的搜索机制是模拟自然选择和自然遗传过程中发生的繁殖、交叉和基因突变现象,在每次迭代中都保留一组候选解,并按某种指标从解群中选取较优的个体,利用遗传算子(选择、交叉和变异)对这些个体进行组合,产生新一代的候选解群,重复此过程,直到满足某种收敛指标为止。

1. 编码方法

编码在遗传算法的设计中发挥着非常重要的作用,但同时编码问题也是运用遗传算法时的一个难点,编码的好坏直接决定了遗传算法设计的好坏。遗传算法的编码方法有浮点数编码方法、二进制编码方法和符号编码方法三种。

(1) 二进制编码方法

通过二进制符号 0 和 1 构成编码符号集,其个体的基因型与一个二进制符号串相对应,它在遗传算法中是最常用的一类编码,如图 8-1 所示。

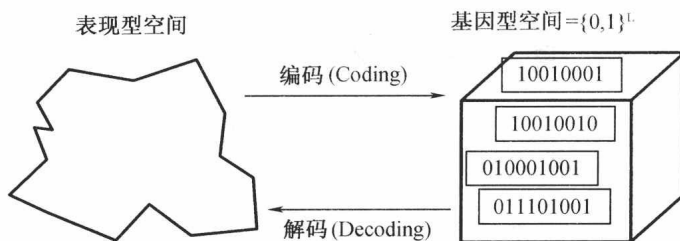


图 8-1 遗传算法二进制编码示意图

(2) 浮点数编码方法

又叫真值编码方法。在该编码方式中,在一定范围内,采用浮点数来表示个体染色体的所有基因值,那么个体染色体的决策变量的数量在大小上就是它的编码长度。

(3) 符号编码方法

将个体编码串中的基因值用一个无任何数值意义的符号集来描述,这个符号集只有代码意义,它可以是数字序号,也可以是字母表,还可以是一个代码表等。

2. 适应度函数

遗传算法中,个体遗传到下一代的概率由个体的适应度大小决定。一般情况下,为了对个体的遗传概率进行正确的计算,必须确保全部个体的适应度为非负数。

由于适应度的非负性,在遗传算法中,对目标函数 $f(x)$ 与适应度进行转换的方法主要有以下两种。

第一种方法:若优化问题为求目标函数的最大值的问题,采用的方法为

$$F(x) = \begin{cases} f(x) - C_{\min}, & \text{if } f(x) + C_{\min} > 0 \\ 0, & \text{if } f(x) + C_{\min} \leq 0 \end{cases} \quad (8-3)$$

式中, $C_{\min}$ 取当前代或最近几代群体中的最小目标函数值。

第二种方法:若优化问题是求目标函数的最小值的问题,采用的方法为

$$F(x) = \begin{cases} C_{\max} - f(x), & \text{if } f(x) - C_{\max} < 0 \\ 0, & \text{if } f(x) - C_{\max} \geq 0 \end{cases} \quad (8-4)$$

式中, $C_{\max}$ 取当前代或最近几代群体中的最大目标函数值。

3. 遗传算子

(1) 选择算子

遗传算法利用选择算子对种群中的个体进行处理,其意图主要是为了防止基因缺失、增加获得全局最优解的能力和加快运算速度。

在遗传算法中,比例选择算子是最常用的选择算子。但根据不同的情况需要不同的选择算子,因此出现了其他种类的一些选择算子,共分为三种:第一种是主要依据各个个体的适应度的具体数值,如确定式采样选择、最优保存策略、无放回余数随机选择和无放回随机选择;第二种是排序选择方法(Rank-based Model),该方法的关键是找准着眼点,即个体适



应度之间的大小关系,而个体适应度的正负以及它们之间的数据差距程度对其并无影响;第三种是随机联赛选择方法(Stochastic Tournament Model),其基本思想为,在多个个体中选取拥有最高适应度的个体,并将该个体遗传到下一代种群里,联赛选择操作只涉及适应度之间的比较操作,并无任何算术运算。

(2) 交叉算子

遗传算法中,将父代个体中个别染色体结构加以替换,并重新组合生成新个体,该过程就是交叉操作。交叉操作的运用使遗传算法的搜索能力飞速提高。同时交叉也是遗传算法获得新优良个体的最重要的途径。常用的交叉算子有一点交叉、两点交叉、多点交叉以及一致交叉。

(3) 变异算子

遗传算法中变异算子以变异率随机地改掉种群中个体的某些基因值。变异操作的基本过程为:对经过交叉操作后形成的新个体的每一个基因值,生成一个 $[0,1]$ 间的伪随机数rand,若rand小于变异概率 P_m ,则进行变异操作。二进制编码方式中的变异算子随机地把某个基因值取反,即 $0 \rightarrow 1$ 或是 $1 \rightarrow 0$ 。

变异操作主要用来防止算法早熟。针对不同的情况,人们提出几种不同的变异算子。常用的变异操作有基本位变异、均匀变异、边界变异和高斯变异。根据不同的场合,变异操作需要采用不同的算子。

4. 遗传算法的运行参数

一般情况下,在遗传算法中需要使用的运行参数主要有以下几种。

(1) 个体编码串长度 l

若个体用二进制编码表示, l 取决于问题要求的精度;若个体用浮点数编码表示, l 取决于决策变量的数量。

(2) 群体大小 M

在种群中,个体的个数用群体大小 M 表示。 M 取值较小,虽然算法的运算速度可以提高,但是却不利于种群的多样性,会增大早熟现象的产生几率; M 取值较大,则会对遗传算法的运行效率产生不利影响。所以,一般将取值控制在20~100的范围内。

(3) 交叉概率 P_c

在遗传算法中,交叉操作是产生新个体的最重要方法,通常情况下交叉概率要求取较大值。但若取值过大,它就会影响种群中优良模式的产生,不利于进化操作;但如果取值偏低,又会降低新个体的产生速度。所以,一般将取值控制在0.4~0.99范围内。除此之外,也可以根据自适应的思想方法来确定交叉概率 P_c 。

(4) 变异概率 P_m

当变异概率 P_m 取值较大时,将有利于新个体的产生,但同时却不利于种群中优良模式的产生,从而会降低遗传算法的性能;当变异概率 P_m 取值较小时,则会减弱新个体产生的能力,并会导致早熟现象的产生。一般将取值控制在0.0001~0.1范围内。当然,变异概率 P_m 还可以通过自适应的思想进行确定。

(5) 终止代数 T

遗传算法运行结束的标志可以用终止代数 T 来表示。当遗传算法运行到指定代数后,操作就会自动停止,同时就会产生出当前种群中最优个体,则这个个体就可以作为当前群体中的最优解,其取值的范围为100~500。对于遗传算法的终止条件,也能够选用某些判断准则,如

果可以判断出该种群已进化成熟并且不会有进化趋势时就能够使算法终止运行。

(6) 代沟 G

代沟 G 表示群体中的重叠程度大小,它表示在每一代群体中,被替换的个体与总个体之比。

8.2.2 遗传算法的流程

遗传算法是一个典型的迭代过程,其解决问题的基本流程如下:

- ①选择编码策略,把参数集合 X 和研究域转换为编码结构空间 S ;
- ②定义适应值函数;
- ③确定遗传策略,包括选择群体大小 M ,选择、交叉、变异方法,以及确定交叉概率 P_c 、变异概率 P_m 等遗传参数;
- ④随机初始化生成群体 P ;
- ⑤计算群体中个体的适应值;
- ⑥按照遗传策略,运用选择、交叉和变异算子作用于群体,形成下一代群体;
- ⑦判断群体性能是否满足某一指标,或者已完成预定迭代次数,不满足则返回步骤④,或者修改遗传策略再返回步骤⑥。

8.2.3 遗传算法的优缺点

1. 遗传算法的优点

遗传算法的优点主要包括:

- ①与问题领域无关且快速随机的搜索能力;
- ②搜索从群体出发,具有潜在的并行性,可以进行多个个体的同时比较;
- ③搜索使用评价函数启发,过程简单;
- ④使用概率机制进行迭代,具有随机性;
- ⑤具有可扩展性,容易与其他算法结合。

2. 遗传算法的缺点

遗传算法在具有上述优点的同时也存在一些不足,主要表现为:

- ①遗传算法的编程实现比较复杂,首先需要问题编码,找到最优解之后还需要对问题进行解码;
- ②另外三个算子的实现也有许多参数,如交叉率和变异率,并且这些参数的选择严重影响解的品质,而目前这些参数的选择大部分是依靠经验;
- ③算法对初始种群的选择有一定的依赖性,能够结合一些启发算法进行改进;
- ④算法并行机制的潜在能力没有得到充分的利用,这也是当前遗传算法的一个研究热点方向。

8.2.4 遗传算法的 MATLAB 仿真

1. 旅行商问题

旅行商问题(Traveling Salesman Problem, TSP)是诸多领域内出现的多种复杂问题的集中概括和简化,其可以描述为:一共有 n 个城市,有一位商人想要走遍每一个城市来推销自己的商品,假定任意两个城市 i 和 j 之间的距离为 d_{ij} ,在不同的路径中如何选择一个总路程最短的路径。

旅行商问题的数学描述可表示为

$$\begin{aligned} \min \quad & \sum_{i \neq j} d_{ij} x_{ij} \\ \text{s. t.} \quad & \sum_{i=1}^n x_{ij} = 1, \quad i = 1, 2, \cdots, n \\ & \sum_{j=1}^n x_{ij} = 1, \quad j = 1, 2, \cdots, n \\ & \sum_{i,j \in S} x_{ij} \leq |S| - 1 \\ & 2 \leq |S| \leq n - 2, \quad S \in \{1, 2, \cdots, n\} \\ & x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \cdots, n (i \neq j) \end{aligned} \tag{8-5}$$

式中, x_{ij} 为决策变量, 当 $x_{ij} = 1$ 时, 表示在该路线中商人会从城市 i 走到城市 j , 如果 $x_{ij} = 0$, 则表示在该路线中商人没有从城市 i 走到城市 j 。在旅行商问题中有几个要求: 首先是所求的最优线路必须是路程和最小的线路, 这是旅行商问题的根本问题; 其次是在最优路线中, 商人每个城市都要走一次, 且仅走一次, 也就是整个最优线路不包含回路。

2. 应用遗传算法求解旅行商问题的 MATLAB 仿真

【例 8-1】 已知敌方 100 个目标的经度、纬度如表 8-2 所示, 我方有一个基地, 经度和纬度为 (70, 40)。假设我方飞机速度为 1 000 千米/小时, 我方派一架飞机从基地出发, 侦查完敌方所有目标, 再返回原来的基地。在敌方每一目标点的侦查时间不计, 求该架飞机所花费的时间 (假设我方飞机巡航时间可以充分长)。

表 8-2 敌方 100 个目标的经度与纬度

经度	纬度	经度	纬度	经度	纬度	经度	纬度
53.712 1	15.304 6	51.175 8	0.032 2	46.325 3	28.275 3	30.331 3	6.934 8
56.543 2	21.418 8	10.818 9	16.252 9	22.789 1	23.104 5	10.158 4	12.481 9
20.105 0	15.456 2	1.945 1	0.205 7	26.495 1	22.122 1	31.484 7	8.964 0
26.241 8	18.176 0	44.035 6	13.540 1	28.983 6	25.987 9	38.472 2	20.173 1
28.269 4	29.001 1	32.191 0	5.869 9	36.486 3	29.728 4	0.971 8	28.147 7
8.958 6	24.663 5	16.561 8	23.614 3	10.559 7	15.117 8	50.211 1	10.294 4
8.151 9	9.532 5	22.107 5	18.556 9	0.121 5	18.872 6	48.207 7	16.888 9
31.949 9	17.630 9	0.773 2	0.465 6	47.413 4	23.778 3	41.867 1	3.566 7
43.547 4	3.906 1	53.352 4	26.725 6	30.816 5	13.459 5	27.713 3	5.070 6
23.922 2	7.630 6	51.961 2	22.851 1	12.793 8	15.730 7	4.956 8	8.366 9
21.505 1	24.090 9	15.254 8	27.211 1	6.207 0	5.144 2	49.243 0	16.704 4
17.116 8	20.035 4	34.168 8	22.757 1	9.440 2	3.920 0	11.581 2	14.567 7
52.118 1	0.408 8	9.555 9	11.421 9	24.450 9	6.563 4	26.721 3	28.566 7
37.584 8	16.847 4	35.661 9	9.933 3	24.465 4	3.164 4	0.777 5	6.957 6
14.470 3	13.636 8	19.866 0	15.122 4	3.161 6	4.242 8	18.524 5	14.359 8
58.684 9	27.148 5	39.516 8	16.937 1	56.508 9	13.709 0	52.521 1	15.795 7

表 8-2(续)

经度	纬度	经度	纬度	经度	纬度	经度	纬度
38.430 0	8.464 8	51.818 1	23.015 9	8.998 3	23.644 0	50.115 6	23.781 6
13.790 9	1.951 0	34.057 4	23.396 0	23.062 4	8.431 9	19.985 7	5.790 2
40.880 1	14.297 8	58.828 9	14.522 9	18.663 5	6.743 6	52.842 3	27.288 0
39.949 4	29.511 4	47.509 9	24.066 4	10.112 1	27.266 2	28.781 2	27.665 9
8.083 1	27.670 5	9.155 6	14.130 4	53.798 9	0.219 9	33.649 0	0.398 0
1.349 6	16.835 9	49.981 6	6.082 8	19.363 5	17.662 2	36.954 5	23.026 5
15.732 0	19.569 7	11.511 8	17.388 4	44.039 8	16.263 5	39.713 9	28.420 3
6.990 9	23.180 4	38.339 2	19.995 0	24.654 3	19.605 7	36.998 0	24.399 2
4.159 1	3.185 3	40.140 0	20.303 0	23.987 6	9.403 0	41.108 4	27.714 9

这是一个旅行商问题。依次给基地编号为 1, 敌方目标依次编号为 2, 3, ..., 101, 最后我方基地再重复编号为 102(这样便于程序中计算)。距离矩阵 $D = (d_{ij})_{102 \times 102}$, 其中 d_{ij} 表示 i, j 两点的距离, $i, j = 1, 2, \dots, 102$, 这里 D 为实对称矩阵。则问题是求一个从点 1 出发, 走遍所有中间点, 到达点 102 的一个最短路径。

问题中给定的是地理坐标(经度、纬度), 需计算出两点间的实际距离。设 A, B 两点的地理坐标分别为 $(x_1, y_1), (x_2, y_2)$, 过 A, B 两点的大圆的劣弧长即为两点的实际距离。以地心为坐标原点 O , 以赤道平面为 XOY 平面, 以 0 度经线圈所在的平面为 XOZ 平面建立三维直角坐标系。则 A, B 两点的直角坐标分别为

$$\begin{aligned} A(R \cos x_1 \cos y_1, R \sin x_1 \cos y_1, R \sin y_1) \\ B(R \cos x_2 \cos y_2, R \sin x_2 \cos y_2, R \sin y_2) \end{aligned} \quad (8-6)$$

其中, $R = 6\,370$ 为地球半径。

得到 A, B 两点的实际距离为

$$d_{ij} = R \times \arccos[\cos(x_i - x_j) \times \cos y_i \times \cos y_j + \sin y_i \times \sin y_j] \quad (8-7)$$

其中, (x_i, y_i) 和 (x_j, y_j) 分别为第 i, j 点的坐标。

(1) 遗传算法的参数设定

种群大小为 $M = 50$, 最大代数 $G = 1\,000$, 交叉率为 $P_c = 1$, 变异率为 $P_m = 0.1$ 。

(2) 编码策略

采用十进制编码, 用随机数列 $\omega_1 \omega_2 \dots \omega_{102}$ 作为染色体, 其中 $0 < \omega_i < 1, (i = 2, 3, \dots, 101), \omega_1 = 0, \omega_{102} = 1$, 每一个随机序列都和种群中的一个个体相对应。例如, 一个 9 城市问题的一个染色体为

$$[0.23, 0.82, 0.45, 0.74, 0.87, 0.11, 0.56, 0.69, 0.78]$$

其中, 编码位置 i 代表城市 i , 位置 i 的随机数表示城市 i 在巡回中的顺序, 将这些随机数按升序排列得到如下巡回顺序: 6—1—3—7—8—4—9—2—5。

(3) 初始种群

此处用经典的近似算法——改良圈算法求得一个较好的初始种群。即对于初始圈 $C = \pi_1 \dots \pi_{u-1} \pi_u \pi_{u+1} \dots \pi_{v-1} \pi_v \pi_{v+1} \dots \pi_{102}$, $2 \leq u \leq v \leq 101$, $2 \leq \pi_u \leq \pi_v \leq 101$, 交换 u 与 v 之间的顺序, 此时的新路径为 $C' = \pi_1 \dots \pi_{u-1} \pi_v \pi_{v-1} \dots \pi_{u+1} \pi_u \pi_{v+1} \dots \pi_{102}$, 记 $\Delta f = (d_{\pi_{u-1}\pi_v} + d_{\pi_u\pi_{v+1}}) -$



$(d_{\pi_u-1\pi_u} + d_{\pi_v\pi_v+1})$, 若 $\Delta f < 0$, 则以新的路径修改旧的路径, 直到不能修改为止。

(4) 目标函数

目标函数为侦查所有目标的路径长度, 适应度函数就取目标函数, 即

$$\min f(\pi_1, \pi_2, \dots, \pi_{102}) = \sum_{i=1}^{101} d_{\pi_i\pi_{i+1}} \quad (8-8)$$

(5) 交叉操作

交叉操作采取单点交叉, 设计如下, 对于选定的两个父代个体 $f_1 = \omega_1\omega_2 \cdots \omega_{102}$, $f_2 = \omega'_1\omega'_2 \cdots \omega'_{102}$, 我们随机地取第 t 个基因处为交叉点, 则经过交叉运算后得到的子代编码为 S_1 和 S_2 , S_1 的基因由 f_1 的前 t 个基因和 f_2 的后 $102-t$ 个基因构成, S_2 的基因由 f_2 的前 t 个基因和 f_1 的后 $102-t$ 个基因构成。例如:

$$f_1 = [0, 0.14, 0.25, 0.27, \mid 0.29, 0.54, \dots, 0.19, 1]$$

$$f_2 = [0, 0.23, 0.44, 0.56, \mid 0.74, 0.21, \dots, 0.24, 1]$$

设交叉点为第四个基因处, 则

$$S_1 = [0, 0.14, 0.25, 0.27, \mid 0.74, 0.21, \dots, 0.24, 1]$$

$$S_2 = [0, 0.23, 0.44, 0.56, \mid 0.29, 0.54, \dots, 0.19, 1]$$

交叉操作的方式有很多种选择, 应该尽可能选取好的交叉方式, 保证子代能继承父代的优良特性。

(6) 变异操作

变异操作按给定的变异率 P_m 对选定变异的个体随机地选取三个整数, 满足 $1 < u < v < w < 102$, 把 u, v 之间 (包括 u 和 v) 的基因段插到 w 之后。

(7) 选择

采用确定性的选择策略, 即选择适应度函数值最小的 M 个个体进化到下一代。

上述例子的 MATLAB 程序如下:

```
clc,clear
load sj.txt %加载敌方100个目标的数据
x = sj(:,1:2:8); x = x(:);
y = sj(:,2:2:8); y = y(:);
sj = [x y];
d1 = [70,40];
sj0 = [d1;sj;d1];
%距离矩阵d
sj = sj0*pi/180;
d = zeros(102);
for i = 1:101
    for j = i+1:102
        temp = cos(sj(i,1) - sj(j,1))*cos(sj(i,2))*cos(sj(j,2)) + sin(sj(i,2))*sin(sj(j,2));
```

```

        d(i,j) = 6370*acos( temp );
    end
end
d = d + d'; L = 102; w = 50; dai = 100;
% 通过改良圈算法选取优良父代 A
for k = 1:w
    c = randperm( 100 );
    c1 = [ 1, c + 1, 102 ];
    flag = 1;
    while flag > 0
        flag = 0;
        for m = 1:L - 3
            for n = m + 2:L - 1
                if d(c1(m),c1(n)) + d(c1(m + 1),c1(n + 1)) < d(c1(m),c1(m + 1)) + d
                    (c1(n),c1(n + 1))
                    flag = 1;
                    c1(m + 1:n) = c1(n - 1:m + 1);
                end
            end
        end
    end
    J(k,c1) = 1:102;
end
J = J/102;
J(:,1) = 0; J(:,102) = 1;
rand('state',sum(clock));
% 遗传算法实现过程
A = J;
for k = 1:dai % 产生 0 ~ 1 间随机数列进行编码
    B = A;
    c = randperm(w);
% 交配产生子代 B
    for i = 1:2:w
        F = 2 + floor(100*rand(1));
        temp = B(c(i),F:102);
        B(c(i),F:102) = B(c(i + 1),F:102);
        B(c(i + 1),F:102) = temp;
    end
end

```

% 变异产生子代 C

```
by = find( rand(1,w) < 0.1 );
```

```
if length( by ) == 0
```

```
    by = floor( w*rand(1) ) + 1;
```

```
end
```

```
C = A( by, : );
```

```
L3 = length( by );
```

```
for j = 1:L3
```

```
    bw = 2 + floor( 100*rand(1,3) );
```

```
    bw = sort( bw );
```

```
    C(j,:) = C(j,[1:bw(1)-1,bw(2)+1:bw(3),bw(1):bw(2),bw(3)+1:102]);
```

```
end
```

```
G = [ A;B;C ];
```

```
TL = size( G,1 );
```

% 在父代和子代中选择优良品种作为新的父代

```
[ dd,IX ] = sort( G,2 );    temp(1:TL) = 0;
```

```
for j = 1:TL
```

```
    for i = 1:101
```

```
        temp(j) = temp(j) + d( IX(j,i),IX(j,i+1) );
```

```
    end
```

```
end
```

```
[ DZ,IZ ] = sort( temp );
```

```
A = G( IZ(1:w), : );
```

```
end
```

```
path = IX( IZ(1), : );
```

```
long = DZ(1)
```

```
xx = sj0( path,1 );    yy = sj0( path,2 );
```

```
plot( xx,yy, ' - o' );
```

```
xlabel( '经度/(°)' );ylabel( '纬度/(°)' );
```

应用遗传算法计算输出的路径结果为：

path =

Columns 1 through 26

	1	17	3	45	67	2	92	87	83	82	48	72	14
27	10	84	18	40	79	77	31	97	85	65	64	11	

Columns 27 through 52

	76	69	94	70	19	63	62	66	26	29	34	90	86
8	39	78	47	23	58	81	25	68	7	22	71	37	

Columns 53 through 78

	32	13	24	49	28	57	88	61	16	91	41	4	73
33	12	53	54	75	5	60	9	15	42	20	30	74	

Columns 79 through 102

	51	80	50	98	100	38	44	55	96	89	6	56
21	99	101	52	46	59	93	43	36	35	95	102	

最优路线长度为：

long =

3.9401e+04

巡航路径如图 8-2 所示。

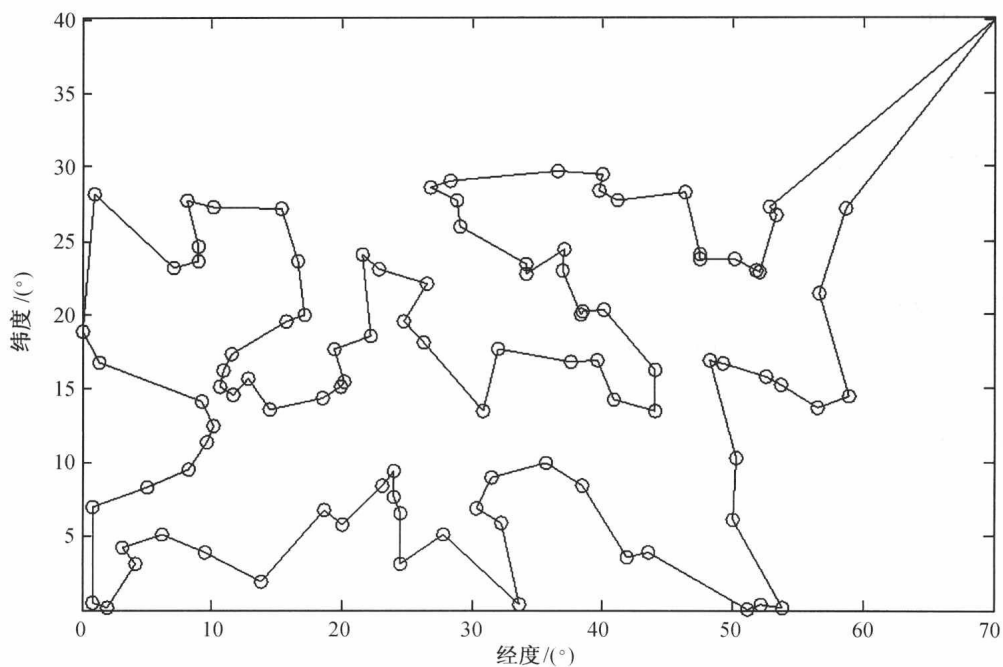


图 8-2 用遗传算法求出的巡航路径图

8.3 模拟退火算法及其 MATLAB 仿真

8.3.1 模拟退火算法的基本原理

模拟退火算法(Simulated Annealing, SA)最早的思想由 Metropolis 等于 1953 提出,1983 年 Kirkpatrick 等将其引入组合优化领域。模拟退火算法的思想来源于固体退火原理,即将固体加热至充分高的温度,再让其徐徐冷却,升温时,固体内部粒子随温度升高变为无序状,内能增大,而徐徐冷却时粒子渐趋有序,从理论上说,如果冷却过程足够缓慢,那么冷却中在每个温度都能达到热平衡态,最后在常温时达到基态,内能减为最小。

物理退火过程包括:

- ①加温过程——增强粒子的热运动,消除系统原先可能存在的非均匀态;
- ②等温过程——对于与环境换热而温度不变的封闭系统,系统状态的自发变化总是朝自由能减少的方向进行,当自由能达到最小时,系统达到平衡态;
- ③冷却过程——使粒子热运动减弱并渐趋有序,系统能量逐渐下降,从而得到低能的晶体结构。

退火过程在恒定温度下达到热平衡的过程可以用 Monte Carlo 算法进行模拟, Monte Carlo 方法的特点是算法简单,但必须大量采样才能比较精确,因此计算量很大。因而 1953 年 Metropolis 等提出了这样一个重要性采样的方法,即设从当前状态 i 生成新状态 j ,若新状态的内能小于状态 i 的内能(即 $E_j < E_i$),则接受新状态 j 作为新的当前状态;否则,以概率 $p = \exp\left[\frac{-(E_j - E_i)}{k_B T}\right]$ 接受状态 j ,其中 k_B 为 Boltzmann 常数。这就是通常所说的 Metropolis 准则。

1983 年 Kirkpatrick 把模拟退火思想与组合优化问题的相似点进行类比,将模拟退火应用到了组合优化问题中。组合优化问题解空间的每一点都代表一个具有不同目标函数的解。所谓优化,就是在解空间寻找函数最小解的过程。若把函数看成能量函数,把控制参数视为温度,解空间作为状态空间,那么模拟退火算法寻找基态的过程就是求目标函数极小值的优化过程,因此,基于 Metropolis 接受准则的最优化过程与物理退火过程存在一定的相似性,如表 8-3 所示。

表 8-3 组合优化与物理退火的相似性

组合优化	物理退火	组合优化	物理退火
解	粒子状态	Metropolis 抽样过程	等温过程
最优解	能量最低态	控制参数的下降	冷却
设定初始温度	熔解过程	目标函数	能量

8.3.2 模拟退火算法的流程

(1) 令 $T = T_0$, 即开始退火的初始温度,随机生成一个初始解 x_0 , 并计算相应的目标函数

值 $E(x_0)$ 。

(2) 令 T 等于冷却进度表中的下一个值 T_i 。

(3) 根据当前解 x_i 进行扰动,产生一个新解 x_j ,计算相应的目标函数值 $E(x_j)$,得到 $\Delta E = E(x_j) - E(x_i)$ 。

(4) 若 $\Delta E < 0$,则新解 x_j 被接受,作为新的当前解;若 $\Delta E > 0$,则新解 x_j 按概率 $\exp[-\Delta E/k_B T_i]$ 接受, T_i 为当前温度。

(5) 在温度 T_i 下,重复 L_k 次的扰动和接受过程,及执行步骤(3)和(4)。

(6) 判断 T 是否已到达 T_f ,是则终止算法;否则转到步骤(2)继续执行。

算法是指分两层循环,在任一温度随机扰动产生新解,并计算目标函数值的变化,决定新解是否被接受。由于算法初始温度比较高,这样,使 E 增大的新解在初始时也可能被接受,因而能跳出局部极小值,然后通过缓慢地降低温度,算法就最终可能收敛到全局最优解。还有一点要说明的是,虽然在低温时接受函数已经非常小了,但仍不排除有接受更差的解的可能,因此,一般都会把退火过程中碰到的最好的可行解(历史最优解)也记录下来,与终止算法前最后一个被接受解一并输出。

8.3.3 模拟退火算法的一些说明

1. 冷却进度表

退火过程由一组初始参数,即冷却进度表(Cooling Schedule)控制,它的核心是尽量使系统达到准平衡,以使算法在有限的时间内逼近最优解。冷却进度表包括:

(1) 控制参数的初始值 T_0

冷却开始的温度。

(2) 控制参数 T 的衰减函数

因计算机能够处理的都是离散数据,因此需要把连续的降温过程离散化成降温过程中的一系列温度点,衰减函数即计算这一系列温度的表达式。

(3) 控制参数 T 的终值 T_f (停止准则)

(4) Markov 链的长度 L_k

任一温度 T 的迭代次数。

2. 新解的产生

新解产生机制的基本要求是能够尽量遍及解空间的各个区域,这样,在某一恒定温度不断产生新解时,就可能跳出当前区域的极小值以搜索其他区域,这是模拟退火算法能够进行广域搜索的一个重要条件。

3. 参数的选择

(1) 控制参数 T 的初值 T_0

求解全局优化问题的随机搜索算法一般都采用大范围的粗略搜索与局部的精细搜索相结合的搜索策略。只有在初始的大范围搜索阶段找到全局最优解所在的区域,才能逐渐缩小搜索的范围,最终求出全局最优解。模拟退火算法是通过控制参数 T 的初值 T_0 和其衰减变化过程来实现大范围的粗略搜索与局部的精细搜索。一般来说,只有足够大的 T_0 才能满足算法要求。在问题规模较大时,过小的 T_0 往往导致算法难以跳出局部陷阱而达不到全局最优。但为了减少计算量, T_0 不宜取得过大,而应与其他参数折中选取。

(2) 控制参数 T 的衰减函数

衰减函数可以有多种形式,一个常用的衰减函数是

$$T_{k+1} = \alpha T_k, k = 0, 1, 2, \dots \quad (8-9)$$

其中, α 是一个常数, 可以取为 $0.5 \sim 0.99$, 它的取值决定了降温的过程。小的衰减量可能导致算法进程迭代次数的增加, 从而使算法进程接受更多的变换, 访问更多的邻域, 搜索更大范围的解空间, 返回更好的最终解。同时由于在 T_{k+1} 时只需少量的变换就可以达到准平衡。这样就可选取较短长度的 Markov 链来减少算法时间。

(3) Markov 链长度

Markov 链长度的选取原则是: 在控制参数 T 的衰减函数已选定的前提下, L_k 应能使在控制参数 T 的每一取值上达到准平衡。从经验上说, 对简单的情况可以令 $L_k = 100n$, n 为问题规模。

(4) 算法停止准则

对 Metropolis 准则中的接受函数 $\exp\left[\frac{-(E_j - E_i)}{k_B T}\right]$ 分析可知, 在 T 比较大的高温情况, 指数上的分母比较大, 而这是一个负指数, 所以整个接受函数可能会趋于 1, 即比当前解 x_i 更差的新解 x_j 也可能被接受, 因此就有可能跳出局部极小而进行广域搜索, 去搜索解空间的其他区域; 而随着冷却的进行, T 减小到一个比较小的值时, 接受函数分母小了, 整体也小了, 即难于接受比当前解更差的解, 也就不太容易跳出当前的区域。如果在高温时, 已经进行了充分的广域搜索, 找到了可能存在的最好解的区域, 而在低温再进行足够的局部搜索, 则可能最终找到全局最优解。

8.3.4 模拟退火算法的 MATLAB 仿真

选取旅行商问题来研究如何用 MATLAB 实现模拟退火算法的仿真。

1. 算法设计步骤

(1) 旅行商问题的解空间和初始解

旅行商问题的解空间 S 是遍访每个城市恰好一次的所有回路, 是所有城市排列的集合。旅行商问题的解空间 S 可表示为 $\{1, 2, \dots, n\}$ 的所有排列的集合, 即

$$S = \{(c_1, c_2, \dots, c_n) \mid (c_1, c_2, \dots, c_n) \text{ 为 } \{1, 2, \dots, n\} \text{ 的排列}\} \quad (8-10)$$

其中, 每一排列 s_i 表示遍访 n 个城市的一路径, $c_i = j$ 表示第 i 次访问城市 j 。模拟退火算法的最优解和初始状态没有强的依赖关系, 故初始解为随机函数生成一个 $\{1, 2, \dots, n\}$ 的排列作为 s_0 。

(2) 目标函数

旅行商问题的目标函数即为访问所有城市的路径总长度, 也可称为代价函数:

$$C(c_1, c_2, \dots, c_n) = \sum_{i=1}^{n-1} d(c_i, c_{i+1}) + d(c_1, c_n) \quad (8-11)$$

现在旅行商问题的求解就是通过模拟退火算法求出目标函数 $C(c_1, c_2, \dots, c_n)$ 的最小值, 相应地, $S(c_1^*, c_2^*, \dots, c_n^*)$ 即为旅行商问题的最优解。

(3) 新解产生

新解的产生对问题的求解非常重要。新解可通过分别或者交替使用以下两种方法来产生。

- ①二变换法:任选序号 u, v (设 $u < v < n$), 交换 u 和 v 之间的访问顺序。
 - ②三变换法:任选序号 u, v, w (设 $u \leq v < w$), 交换 u 和 v 之间的路径插到 w 之后访问。
- (4) 目标函数差

计算变换前和变换后目标函数的差值:

$$\Delta C' = C(s'_i) - C(s_i) \tag{8-12}$$

(5) Metropolis 接受准则

以新解与当前解的目标函数差定义接受概率, 即

$$p = \begin{cases} 1, & \Delta C' < 0 \\ \exp(-\Delta'/T), & \Delta C' > 0 \end{cases} \tag{8-13}$$

2. 典型程序结构

TSPLIB (<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95>) 是一组各类旅行商问题的实例集合。这里以 berlin52 为例进行求解, berlin52 有 52 座城市, 其坐标数据如表 8-4 所示(可从 TSPLIB 的网站下载)。

表 8-4 berlin52 中城市的坐标数据

编号	X 坐标	Y 坐标	编号	X 坐标	Y 坐标	编号	X 坐标	Y 坐标
1	565.0	575.0	19	510.0	875.0	37	770.0	610.0
2	25.0	185.0	20	560.0	365.0	38	795.0	645.0
3	345.0	750.0	21	300.0	465.0	39	720.0	635.0
4	945.0	685.0	22	520.0	585.0	40	760.0	650.0
5	845.0	655.0	23	480.0	415.0	41	475.0	960.0
6	880.0	660.0	24	835.0	625.0	42	95.0	260.0
7	25.0	230.0	25	975.0	580.0	43	875.0	920.0
8	525.0	1 000.0	26	1 215.0	245.0	44	700.0	500.0
9	580.0	1 175.0	27	1 320.0	315.0	45	555.0	815.0
10	650.0	1 130.0	28	1 250.0	400.0	46	830.0	485.0
11	1 605.0	620.0	29	660.0	180.0	47	1 170.0	65.0
12	1 220.0	580.0	30	410.0	250.0	48	830.0	610.0
13	1 465.0	200.0	31	420.0	555.0	49	605.0	625.0
14	1 530.0	5.0	32	575.0	665.0	50	595.0	360.0
15	845.0	680.0	33	1 150.0	1 160.0	51	1 340.0	725.0
16	725.0	370.0	34	700.0	580.0	52	1 740.0	245.0
17	145.0	665.0	35	685.0	595.0			
18	415.0	635.0	36	685.0	610.0			

应用模拟退火算法求解 berlin52 旅行商问题的 MATLAB 程序如下：

```
clear,clc
a = 0.99; % 温度衰减函数的参数
t0 = 97; tf = 3; t = t0;
Markov_length = 10000; % Markov 链长度
load berlin52.txt;
coordinates = berlin52;
coordinates(:,1) = [];
amount = size(coordinates,1); % 城市的数目
% 通过向量化的方法计算距离矩阵
dist_matrix = zeros(amount, amount);
coor_x_tmp1 = coordinates(:,1)*ones(1,amount);
coor_x_tmp2 = coor_x_tmp1';
coor_y_tmp1 = coordinates(:,2)*ones(1,amount);
coor_y_tmp2 = coor_y_tmp1';
dist_matrix = sqrt((coor_x_tmp1 - coor_x_tmp2).^2 + ...
    (coor_y_tmp1 - coor_y_tmp2).^2);
sol_new = 1:amount; % 产生初始解
% sol_new 是每次产生的新解;sol_current 是当前解;sol_best 是冷却
% 中的最好解;
E_current = inf;E_best = inf; % E_current 是当前解对应的回路距离
% E_new 是新解的回路距离
% E_best 是最优解的
sol_current = sol_new; sol_best = sol_new;
p = 1;
while t >= tf
    for r = 1:Markov_length % Markov 链长度
        % 产生随机扰动
        if (rand < 0.5) % 随机决定是进行两交换还是三交换
            % 两交换
            ind1 = 0; ind2 = 0;
            while (ind1 == ind2)
                ind1 = ceil(rand.*amount);
                ind2 = ceil(rand.*amount);
            end
            tmp1 = sol_new(ind1);
            sol_new(ind1) = sol_new(ind2);
            sol_new(ind2) = tmp1;
        else
```

```

% 三交换
ind1 = 0; ind2 = 0; ind3 = 0;
while (ind1 == ind2) || (ind1 == ind3) ...
    || (ind2 == ind3) || (abs(ind1 - ind2) == 1)
    ind1 = ceil(rand.*amount);
    ind2 = ceil(rand.*amount);
    ind3 = ceil(rand.*amount);
end
tmp1 = ind1; tmp2 = ind2; tmp3 = ind3;
% 确保 ind1 < ind2 < ind3
if (ind1 < ind2) && (ind2 < ind3);
elseif (ind1 < ind3) && (ind3 < ind2)
    ind2 = tmp3; ind3 = tmp2;
elseif (ind2 < ind1) && (ind1 < ind3)
    ind1 = tmp2; ind2 = tmp1;
elseif (ind2 < ind3) && (ind3 < ind1)
    ind1 = tmp2; ind2 = tmp3; ind3 = tmp1;
elseif (ind3 < ind1) && (ind1 < ind2)
    ind1 = tmp3; ind2 = tmp1; ind3 = tmp2;
elseif (ind3 < ind2) && (ind2 < ind1)
    ind1 = tmp3; ind2 = tmp2; ind3 = tmp1;
end

tmplist1 = sol_new((ind1 + 1):(ind2 - 1));
sol_new((ind1 + 1):(ind1 + ind3 - ind2 + 1)) = sol_new((ind2):(ind3));
sol_new((ind1 + ind3 - ind2 + 2):ind3) = tmplist1;
end

% 检查是否满足约束
% 计算目标函数值(即内能)
E_new = 0;
for i = 1 : (amount - 1)
    E_new = E_new + ...
        dist_matrix(sol_new(i), sol_new(i + 1));
end
% 再算上从最后一个城市到第一个城市的距离
E_new = E_new + dist_matrix(sol_new(amount), sol_new(1));
if E_new < E_current
    E_current = E_new;
    sol_current = sol_new;
end

```

```
if E_new < E_best
% 把冷却过程中最好的解保存下来
    E_best = E_new;
    sol_best = sol_new;
end
else
% 若新解的目标函数值小于当前解的,
% 则仅以一定概率接受新解
if rand < exp( -(E_new - E_current)./t)
    E_current = E_new;
    sol_current = sol_new;
else
    sol_new = sol_current;
end
end
end
t = t.*a; % 控制参数 t(温度)减少为原来的 a 倍
end
disp('最优解为:')
disp(sol_best)
disp('最短距离:')
disp(E_best)

xx = coordinates(sol_best,1);
yy = coordinates(sol_best,2);
plot(xx,yy,'- *');
xlabel('X 坐标');ylabel('Y 坐标');
```

最优解为:

Columns 1 through 26													
	31	22	1	49	32	45	19	41	8	9	10	43	33
51	11	52	14	13	47	26	27	28	12	25	4	6	
Columns 27 through 52													
	15	5	24	48	38	37	40	39	36	35	34	44	46
16	29	50	20	23	30	2	7	42	21	17	3	18	

最短距离为:

$7.5444e+03$

巡航路径如图 8-3 所示。

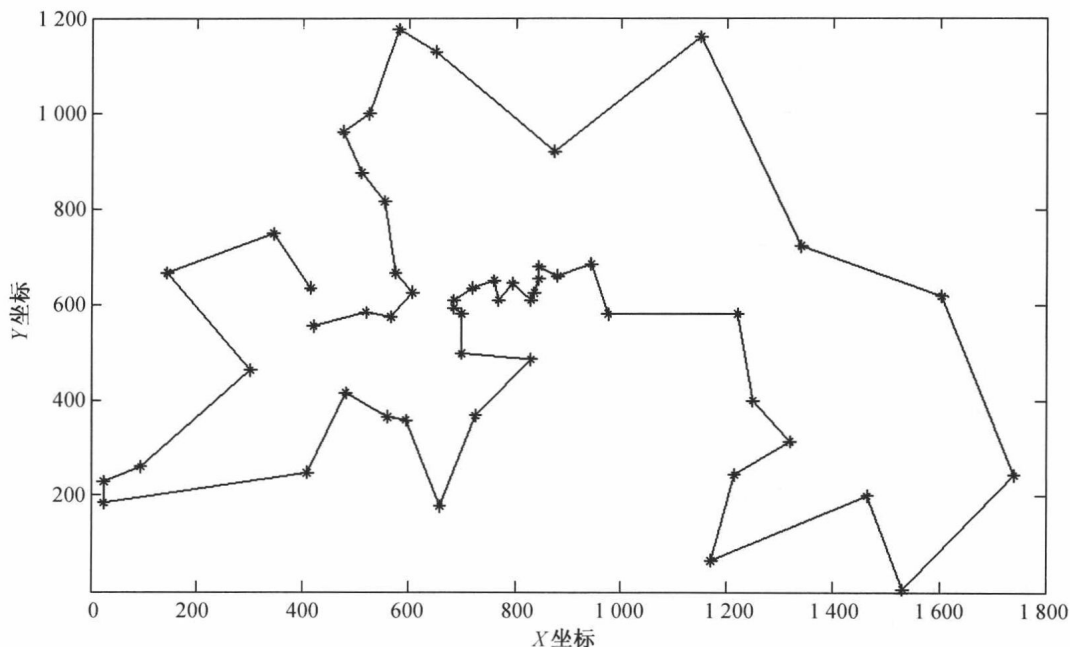


图 8-3 模拟退火算法求解旅行商问题运行结果图

8.4 粒子群算法及其 MATLAB 仿真

8.4.1 粒子群算法的基本原理

粒子群算法(Particle Swarm Optimization, PSO)是由 Kennedy 和 Eberhart 在 1995 年提出的,是模拟生物群体的社会行为来进行优化的智能优化算法,粒子群算法基本思想最早源于对鱼群、鸟群迁徙、捕食等行为的研究。在智能性生物群体中存在着个体之间的、个体与群体间以及个体与环境之间的相互作用、相互传递信息的行为,这种行为体现的是一种在于智能生物群体内部的信息共享的机制。粒子群算法最初的提出是为解决连续变量的十分复杂的优化问题的,但是随着粒子群理论的不断发展和完善,粒子群算法的应用领域现在已经扩展到求解离散问题。粒子群算法的基本思想简单,效果较好,易于编程,易于实现,虽然没有严格地证明算法的收敛性,但是优化复杂问题的效果已经赶超传统的智能优化算法。

群鸟觅食其实是一个最佳决策的过程,群鸟在觅食的过程中,每只鸟的初始状态处于随机位置,且飞翔的方向也是随机的,每只鸟都不知道食物在哪里,但是随着时间的推移,



这些初始处于随机位置的鸟通过群内相互学习、信息共享和个体不断积累自身寻觅食物的经验,自组织积聚成一个群落,并逐渐朝唯一的目标—食物—前进。每只鸟能够通过一定经验和信息估计目前所处的位置对于能寻找到食物有多大的价值,及多大的适应值;每只鸟能够记住自己所找到的最好位置,称之为局部最优。此外,还能记住群鸟中所有个体所能找到的最好位置,称为全局最优,整个鸟群的觅食中心都趋向全局最优移动,通过鸟群觅食的位置不断移动,即不断迭代,可以使鸟群向食物步步逼近。

在群鸟觅食模型中,每个个体被抽象为没有质量和体积的微粒,并延伸到 N 维空间,粒子 I 在 N 维空间的位置表示为矢量 $\mathbf{X}_i = (x_1, x_2, \dots, x_N)$, 飞行速度表示为矢量 $\mathbf{V}_i = (v_1, v_2, \dots, v_N)$ 。每个粒子都有一个由目标函数决定的适应值 (Fitness Value), 并且知道自己到目前为止发现的最好位置 ($pbest$) 和现在的位置 $\mathbf{X}_i$, 这个可以看作粒子自己的飞行经验。除此之外,每个粒子还知道到目前为止整个群体中所有粒子发现的最好位置 ($gbest$) ($gbest$ 是 $pbest$ 中的最好值)。这个可以看作粒子同伴的经验,粒子就是通过自己的经验和同伴中最好的经验来决定下一步的运动。

PSO 初始化为一群随机粒子 (随机解), 然后通过迭代找到最优解。在每一次的迭代中,粒子通过跟踪两个“极值” ($pbest, gbest$) 来更新自己。

在找到这两个最优值后,粒子通过下面的公式来更新自己的速度和位置:

$$V_i = V_i + c_1 \times \text{rand}() \times (pbest_i - x_i) + c_2 \times \text{rand}() \times (gbest_i - x_i) \quad (8-14)$$

$$x_i = x_i + V_i \quad (8-15)$$

式中, $i = 1, 2, \dots, M$, M 是该群体中粒子的总数; V_i 是粒子的速度; $\text{rand}()$ 是介于 $(0, 1)$ 之间的随机数; X_i 是粒子的当前位置; c_1 和 c_2 是学习因子, 通常取 $c_1 = c_2 = 2$ 。

在每一维,粒子都有一个最大限制速度 $V_{\max}$, 如果某一维的速度超过设定的 $V_{\max}$, 那么这一维的速度就被限定为 $V_{\max}$ ($V_{\max} > 0$)。以公式 (8-14) 和公式 (8-15) 为基础, 形成了 PSO 的标准形式。

1998 年 Shi 等人在进化计算的国际会议上发表了一篇论文 “A modified Particle Swarm Optimizer” 对前面的公式 (8-14) 进行了修正, 引入惯性权重因子 ω (ω 非负)。

$$V_i = \omega \times V_i + c_1 \times \text{rand}() \times (pbest_i - x_i) + c_2 \times \text{rand}() \times (gbest_i - x_i) \quad (8-16)$$

公式 (8-16) 和 (8-15) 被视为标准 PSO 算法。

当 ω 值较大时, PSO 算法的全局寻优能力强, 局部寻优能力弱; 当 ω 值较小反之。初始时, Shi 将 ω 取为常数, 后来实验发现, 动态 ω 能够获得比固定值更好的寻优结果。动态 ω 可以在 PSO 搜索过程中线性变化, 也可根据 PSO 性能的某个测度函数动态改变。目前, 采用较多的是 Shi 建议的线性递减权值 (Linearly Decreasing Weight, LDW) 策略。

$$\omega^{(t)} = \frac{(\omega_{\text{ini}} - \omega_{\text{end}})(G_k - g)}{G_k} + \omega_{\text{end}} \quad (8-17)$$

式中 G_k ——最大进化代数;

ω_{ini} ——初始惯性权值;

ω_{end} ——迭代至最大代数时惯性权值。

典型取值: $\omega_{\text{ini}} = 0.9$, $\omega_{\text{end}} = 0.4$ 。

ω 的引入使 PSO 算法性能有了很大提高, 针对不同的搜索问题, 可以调整全局和局部搜索能力, 也使得 PSO 算法能成功地应用于很多实际问题。

8.4.2 标准粒子群算法的流程

标准粒子群算法的流程如图 8-4 所示。

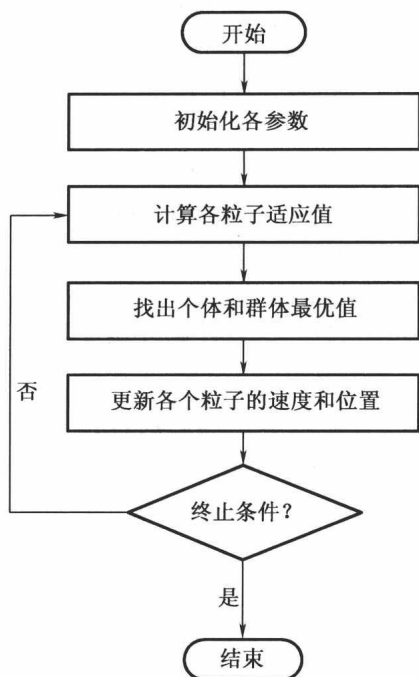


图 8-4 标准粒子群算法流程图

(1) 初始化粒子群的速度和位置、惯性因子、加速常数、最大迭代次数和算法终止的最小允许误差；

(2) 评价每个粒子的初始适应值；

(3) 将初始适应值作为当前每个粒子的局部最优值,并将各适应值对应的位置作为每个粒子的局部最优值所在的位置；

(4) 将最佳初始适应值作为当前全局最优值,并将最佳适应值对应的位置作为全局最优值所在的位置；

(5) 更新每个粒子当前的飞翔速度；

(6) 对每个粒子的飞翔速度进行限幅处理,使之不能超过设定的最大飞翔速度；

(7) 更新每个粒子当前所在的位置；

(8) 比较当前每个粒子的适应值是否比历史局部最优值好,如果好,则将当前粒子适应值作为粒子的局部最优值,其对应的位置作为每个粒子的局部最优值所在的位置；

(9) 在当前群中找出全局最优值,并将当前全局最优值对应的位置作为粒子群的全局最优值所在的位置；

(10) 重复步骤(5)~(9),直到满足设定的最小误差或者达到最大迭代次数；

(11) 输出粒子群全局最优值和其对应的位置以及每个粒子的局部最优值和其对应的位置。



8.4.3 标准粒子群算法的参数选取

在标准粒子群算法中,群体规模 m , 惯性因子 ω , 学习因子 c_1 和 c_2 , 最大速度 $V_{\max}$, 迭代次数 G_k 等参数对算法寻优性能的影响非常显著。

1. 群体规模 m

群体规模 m 一般取 20 ~ 40。实验表明,对于多数问题,30 个粒子就够用了,不过对于特殊的较难或特定类别的问题可以取到 100 ~ 200。粒子数量越多,搜索范围越大,越容易找到全局最优解,但算法运行的时间也越长。

2. 惯性因子 ω

惯性因子 ω 对于粒子群算法的收敛起到很大作用, ω 值越大,粒子飞翔速度越大,容易错失局部寻优能力,而全局搜索能力越强;反之,局部寻优能力增强,而全局寻优能力减弱。通过调整 ω 的大小来控制历史的速度对当前速度的影响程度,使其成为兼顾全局搜索和局部搜索的一个折中。惯性因子 ω 的大小决定了对粒子当前速度继承的多少。需要特别说明的是,这里关于惯性因子 ω 的理解往往有一个误区,认为惯性因子 ω 越大表示继承当前的飞翔速度也就越大,所以应该是较小程度地离开原先的寻优轨道。理解错误的原因是, ω 越大,粒子飞翔的速度和位置更新的幅度就越大,因此偏离原先寻优轨道的程度也就越大。

如果惯性因子 ω 是变量,通常在迭代开始时将惯性因子 ω 设置得较大,然后在迭代过程中逐步减小。这样可使粒子群在开始优化时搜索较大的解空间,得到合适的种子,然后在后期逐渐收缩到较好的区域进行更精细的搜索,以加快收敛速度和目标精度。 ω 也可以取 $[0, 1]$ 区间的随机数。如果惯性因子 ω 是定值,建议取 $[0.6, 0.75]$ 区间的合理值。

3. 学习因子 c_1 和 c_2

对于简单的常规问题,一般情况下取 $c_1 = c_2 = 2.0$ 。目前对于学习因子 c_1 和 c_2 的确切取值,学术界分歧很大,各学者观点还不尽一致。表 8-5 所列是各主要学者给出的参考值。

表 8-5 学习因子参考值

学者	c_1 和 c_2
Clerc	$c_1 = c_2 = 2.05$
Carlisle	$c_1 = 2.8, c_2 = 1.3$
Trelea	$\omega = 0.6, c_1 = c_2 = 1.7$
Eberhart	$\omega = 0.729, c_1 = c_2 = 1.494$

学习因子 c_1 和 c_2 是调整自身经验和社会经验在其运动中所起作用的权重。如果 $c_1 = 0$, 则粒子没有自身经验,只有社会经验,它的收敛速度可能较快,但在处理较复杂问题时,容易陷入局部最优点。如果 $c_2 = 0$, 则粒子群没有群体共享信息,只有自身经验,因为个体间没有信息共享,一个规模为 m 的粒子群等价运行了 m 个单个粒子,因而得到最优解的几率非常小。如果 $c_1 = c_2 = 0$, 则粒子将在没有任何经验和信息借鉴的情况下,一直杂乱无章地朝边界飞翔,飞翔的速度也在不断地衰减。由于只能盲目搜索到有限的区域,因此很难找到最优解。

4. 最大速度 $V_{\max}$

速度决定当前位置与最好位置之间的区域的分辨率(或精度)。如果太快,则粒子有可能越过极小点;如果太慢,则粒子不能在局部极小点之外进行足够的探索,会陷入到局部极值区域内。这种限制可以达到防止计算溢出、决定问题空间搜索的粒度的目的。 $V_{\max}$ 的选择需要对问题有一定的先验知识。为了跳出局部最优,需要较大的寻优步长,而在接近最优值时,采用更小的步长会更好。如果 $V_{\max}$ 的选择是固定不变的,通常 $V_{\max}$ 设定为每维变化范围的 10% ~ 20%。

8.4.4 标准粒子群算法的优缺点

1. 标准粒子群算法的优点

- (1) 不依赖于问题信息,采用实数求解,算法通用性强。
- (2) 需要调整的参数少,原理简单,容易实现。
- (3) 协同搜索,同时利用个体局部信息和群体全局信息指导搜索。
- (4) 收敛速度快,算法对计算机内存和 CPU 要求不高。
- (5) 容易飞跃局部最优信息。对于目标函数仅能提供极少搜索最优值的信息,在其他算法无法辨别搜索方向的情况下,粒子群算法的粒子具有飞跃性的特点使其能够跨过搜索平面上信息严重不足的障碍,飞抵全局最优目标值。

2. 标准粒子群算法的缺点

- (1) 算法局部搜索能力较差,搜索精度不够高。
- (2) 算法不能绝对保证搜索到全局最优解,主要有两方面的原因:
 - ① 有时粒子群在俯冲过程中会错失全局最优解。粒子运动过程中的俯冲动作使搜索行为不够精细,不容易发现全局最优目标值,所以对粒子的最大速度进行限制既是为了使粒子不要冲出搜索区域的边界,同时也是为了使搜索行为不至于太粗糙。
 - ② 应用粒子群算法处理高维复杂问题时,算法可能会早熟收敛,也就是粒子群在没有找到全局最优信息之前就陷入停顿状态。
- (3) 算法搜索性能对参数具有一定的依赖性。
- (4) 标准粒子群算法是一种概率算法,算法理论不完善,缺乏独特性,理论成果偏少。从数学角度严格证明算法的正确性和可靠性还比较困难;缺少算法结构设计和参数选取的实用性指导原则,特别是全局收敛研究和大型多约束非线性规划的研究成果非常少。

8.4.5 标准粒子群算法的 MATLAB 仿真

1. 背包问题

背包问题(Knapsack Problem)是在 1978 年由 Merkel 和 Hellman 提出的,是一种组合优化的 NP 完全问题。背包问题可以描述为:给定一组物品,每种物品都有自己的重量和价格,在限定的总重量内,我们如何选择才能使得物品的总价格最高。背包问题经常出现在商业、组合数学,计算复杂性理论、密码学和应用数学等领域中。

2. 标准粒子群算法的背包问题求解

应用标准粒子群算法求解背包问题的 MATLAB 程序如下:

```
% 背包问题:对于 n 个体积为 aj、价格为 cj 的物品,总体积为 b
% n = 10
% aj = [95, 4, 60, 32, 23, 72, 80, 62, 65, 46]
% cj = [55, 10, 47, 5, 4, 50, 8, 61, 85, 87]
% b = 269
a = [95 4 60 32 23 72 80 62 65 46]; % 物品的体积
c = [55 10 47 5 4 50 8 61 85 87]; % 物品的价格
b = 269; % 背包的重量限制
% 初始化
Dim = 10; % 粒子的维数
xSize = 20; % 种群数
MaxIt = 30; % 最大迭代次数
c1 = 0.7;
c2 = 0.7; % 定义加速因子
w = 0.8; % 定义惯性因子
A = repmat(a, xSize, 1); % 将 a 扩展成一个 30 * 10 的矩阵
C = repmat(c, xSize, 1); % 将 c 扩展成一个 30 * 10 的矩阵
x = round(rand(xSize, Dim)); % 随机取一个 30 * 10 的 0/1 矩阵作为初始位置
v = rand(xSize, Dim); % 粒子的初始速度
xbest = zeros(xSize, Dim); % 单个粒子的初始最佳位置
fxbest = zeros(xSize, 1); % xbest 的适应度
gbest = zeros(1, Dim); % 粒子群的初始最佳位置
fgbest = 0; % gbest 的适应度

% 粒子群最优位置和单个粒子最优位置的选定
iter = 0;
while iter < MaxIt
    iter = iter + 1;
    fx = sum((C .* x)'); % 计算粒子群的适应度,即背包内物品的价格
    sx = sum((A .* x)'); % 限制函数,背包内物品的体积
    for i = 1 : xSize
        if sx(i) > 269
            fx(i) = 0; % 当背包内物品的体积超过限制时,将其适应度设置为 1
        end
    end
end
```

```

for i = 1 : xSize
    if fxbest(i) < fx(i)
        fxbest(i) = fx(i);
        xbest(i,:) = x(i,:); % 当粒子的适应度 fx(i) 大于其最佳适应度
        % fxbest(i) 时, 用其替代原来粒子的最佳适应度, 并记下此解
    end
end
if fgbest < max( fxbest)
    [ fgbest, g] = max( fxbest);
    gbest = xbest( g, :); % 当存在粒子的最佳适应度 fxbest(g) 大于种群
    % 的最佳适应度时, 用其替代原来种群的最佳适应度, 并记下此解
end
for i = 1 : xSize
    if x(i,:) == gbest
        x(i,:) = round( rand( 1, Dim) ); % 为防止算法陷入局部最优, 若某个
        % 粒子的位置等于种群最佳位置, 将对该粒子的位置重新初始化赋值
    end
end
R1 = rand( xSize, Dim);
R2 = rand( xSize, Dim);
v = v * w + c1 * R1. * ( xbest - x) + c2 * R2. * ( repmat( gbest, xSize, 1) - x);
% 用速度迭代公式产生新的速度
x = x + v; % 更新粒子群的位置
for i = 1 : xSize
    for j = 1 : Dim
        if x(i,j) < 0.5
            x(i,j) = 0;
        else x(i,j) = 1;
        end
    end
end
% 由于粒子的位置只有 0,1 两种状态, 此处以 0.5 为分界
% 点对函数值进行离散化
end
fgbest
sgbest = sum( ( a. * gbest)')
gbest

```

输出结果为:

```
fgbest = 295
```

```
sgbest = 269
```

```
gbest = 0 1 1 1 0 0 0 1 1 1
```


第9章 复杂适应系统建模与仿真

9.1 复杂适应系统理论的基本观点与概念

复杂适应系统(Complex Adaptive System, CAS)理论是美国霍兰(John Holland)教授于1994年,在圣塔菲研究所(Sante Fe Institute, SFI)成立十周年时正式提出的。在圣塔菲研究所的乌拉姆系列讲座的首次报告会上,霍兰以“复杂创造简单”为题作了演讲,提出了关于复杂适应系统的比较完整的理论。

9.1.1 复杂适应系统理论的核心思想——适应性造就复杂性

复杂适应系统理论的提出,是从对系统演化规律的思考引起的。复杂性的产生机制是系统复杂性研究的一个重要方面,CAS就是对这个问题的一种回答。CAS理论的基本思想就是“适应性造就复杂性”。CAS理论的提出对于人们认识、理解、控制、管理复杂系统提供了新的思路。

CAS理论的最基本的概念是具有适应能力的、主动的个体,简称主体(Agent)。所谓具有适应性就是指主体与环境以及其他主体进行交互作用。主体在这种持续不断地交互作用的过程中,不断地“学习”或“积累经验”,并且根据学到的经验改变自身的结构和行为方式。宏观系统的演变或进化,包括新层次的产生、分化和多样性的出现,新的、聚合而成的、更大的主体的出现等,都是在这个基础上逐步派生出来的。

按照某些传统的看法,复杂性来自系统的外部。系统行为之复杂和不可预测,也总是归之于外部的随机性的干扰。近几百年发展起来的近代科学,回答了以前人类所不知道的许多问题,然而,对于复杂性研究是从何而来的这个基本问题,却似乎没有什么进步。而且,由于对热力学第二定律的片面理解,认为任何事物的发展都按照这样一条道路:从不对称到对称,从有结构到无结构,从有差别到无差别,一句话,从复杂到简单。显然,这种认识与丰富多彩的大千世界是完全不相容的。然而,囿于还原论的束缚,人们对于这种矛盾视而不见,仍然不得不把复杂性的来源归之于神秘的、外来的力量。

现代系统科学的发展打破了这种形而上学的看法,特别是对于自组织现象的研究,引导人们从系统内部寻找复杂性的来源。然而,最初研究的自组织现象,在某种意义上还是比较简单的。这些系统中的元素(或个体)还是被动的、“死的”,并没有自身的目的和主动性。这就使得这一阶段的理论与方法,难以有效地应用到生物、生态、经济、社会等领域。

复杂适应系统理论正是在这方面向前迈进了一大步,即把系统的成员看作具有自身目的与主动性的积极的“活的”主体。更重要的是,复杂适应系统理论认为,微观方面,主体在与环境的交互作用中遵循一般的“刺激-反应”模型,所谓适应能力表现在它能够根据行为的效果修改自己的行为规则,以便更好地在客观环境中生存;宏观方面,由这样的主体组成的系统,将在主体之间以及主体与环境的相互作用中发展,表现出宏观系统中的分化、涌现



等种种复杂的演化过程。宏观的变化和个体分化都可以从个体的行为规律中找到根源。霍兰把个体与环境之间这种主动的、反复的交互作用用“适应性”一词加以概括。这就是复杂适应系统理论的基本思想——适应性造就复杂性。

9.1.2 复杂适应系统理论的基本概念

系统中的个体一般称为元素、部分或子系统。复杂适应系统理论采用了 Adaptive Agent (具有适应能力的个体)这个词,是为了强调它的主动性,强调它具有自己的目标、内部结构和生存能力。因此,为了行文方便,在下文中简称其为主体,不在每次讲具有适应能力的个体。Agent 一词本来是经济学中的用语,表示代理人或代理商的意思。霍兰借用这个词,明显地表示经济系统是他建立复杂适应系统理论时心目中的主要背景之一。

但是单独用主体这个概念,是无法完全表达出复杂适应系统理论的丰富内容的,所以围绕主体这个最核心的概念,霍兰进一步提出了研究适应和演化过程中特别要注意的 7 个有关概念:聚集、非线性、流、多样性、标志、内部模型以及积木。在这 7 个概念中,前 4 个是复杂适应系统的通用特性,它们将在适应和进化中发挥作用,后 3 个则是个体与环境进行交流时的机制和有关概念。

1. 聚集(Aggregation)

聚集主要用于个体通过“黏着”(Adhesion)形成较大的所谓的多主体的聚集体。由于个体具有这样的属性,它们可以在一定条件下,在双方彼此接受时组成一个新的个体——聚集体,在系统中像一个单独的个体那样运动。

在复杂系统的演变过程中,较小的、较低层次的个体通过某种特定的方式结合起来,形成较大的、较高层次的个体,这是一个十分重要的关键步骤。这个过程重复几次后,就得到了复杂适应系统的层次组织。

事实上,聚集现象在许多系统中都存在。例如,在生物界中,共生现象越来越多地得到重视和研究。近年来,人们发现在一些高等生物体内存在着许多独立的低等生物。这些低等生物完全是独立的个体,按照自身的规律生存和发展。它把高等生物体内的条件作为自己的生存环境,进行着物质、能量与信息的流通与处理。如人体细胞中的线粒体,一方面,它们是完整意义下的、独立自主的生物;另一方面,在长期演化过程的作用下,它们必须也只能在人体内的这种环境中生存。如果按照传统意义下的系统元素去理解,则这两方面的矛盾将是无法解决的。因为,在传统的思维框架中,部分或元素是“死的”、被动的、没有自己的目的与意志的;如果它有自己的目的、意志、主动性,那就只会对系统起瓦解作用。这就把上述的两个方面绝对地、不可调和地对立起来、割裂开来。在社会生活中这种例证也是普遍存在的,如个人与社会,雇员与企业,以及企业集团的形成等,都反映了不同层次的主体之间的有效协调与共生。

聚集不是简单的合并,也不是消灭个体的吞并,而是新类型的、更高层次上的个体的出现;原来的个体没有消失,而是在新的更适宜自己生存的环境中得到了发展。这就是后面将要讲到的“黏着”的意义。

2. 非线性(Nonlinearity)

非线性指主体以及它们的属性在发生变化时,并非遵从简单的线性关系。特别在主体与系统或环境反复的交互作用中,这一点更为明显。复杂适应系统理论认为个体之间相互影响不是简单的、被动的、单向的因果关系,而是主动的适应关系。以往的历史会留下痕

迹,以往的经验会影响将来的行为。在这种情况下,线性的、简单的、直线式的因果链已经不复存在,实际的情况往往是各种反馈作用(包括负反馈和正反馈)交互影响的、互相缠绕的复杂关系。正因为这样,复杂系统的行为才会如此难以预测;也正因为这样,复杂系统才会经历曲折的进化过程,呈现出丰富多彩的性质和状态。

复杂适应系统理论把非线性的产生归之于内因,归之于主体的主动性和适应能力。这就进一步把非线性理解为系统行为的必然的、内在的要素,从而大大丰富和加深了对于非线性的理解。正因为如此,霍兰在提出具有适应性的主体这一概念时,特别强调其行为的非线性特征,并且认为这是复杂性产生的内在根源。

3. 流(Flow)

流可以看成是有着众多节点与连接者的某个网络上的某种资源的流动。这些流的渠道是否通畅,周转迅速到什么程度,都直接影响系统的演化过程。越复杂的系统,其中的各种交换(物质、能量、信息)就越频繁,各种流也就越错综复杂。

在复杂适应系统中,网络上的流动因时而异,节点和连接会随着主体的适应和不适应而出现或消失。因此,无论是流还是网络,皆随时间而变化,它们是随着时间的流逝和经验的积累而反映出变化适应性的模式。

4. 多样性(Diversity)

在适应过程中,由于种种原因,个体间的差别会发展与扩大,最终形成分化,这是复杂适应系统的一个显著特点。

系统复杂性的重要思想之一就是主体之间的差别,主体类型的多样性。当前复杂性研究着眼于主体类型多种多样的情况,而其中的复杂适应系统理论则进一步研究这种多样性是怎样产生的,即分化的过程。霍兰指出,正是相互作用和不断适应的过程,造成了主体向不同的方面发展变化,从而形成了主体类型的多样性。而从整个系统来看,这事实上是一种分工。如果和前面讲的聚集结合起来看,这就是系统从宏观尺度上看到的“结构”的“涌现”,即所谓“自组织现象”的出现。

5. 标识(Tagging)

为了相互识别和选择,个体的标识在个体与环境的相互作用中是非常重要的,因而无论在建模中还是实际系统中,标识的功能与效率都是必须认真考虑的因素。

标识的作用主要在于实现信息的交流。在以往的系统研究中,信息和信息交流的作用没有得到足够的重视。这是对于复杂系统行为的研究难以深入的原因之一。复杂适应系统理论在这方面的发展就在于把信息的交流和处理作为影响系统进化过程的重要因素加以考虑。强调流与标识就为把信息因素引入系统研究创造了条件。

谈到信息的作用,现在大家都承认,它是人类社会经济生活中的基本要素之一。然而,在复杂系统中,它是怎样发挥作用的呢?对此一直没有深入的研究。以前的系统研究中缺乏对于信息流的具体机制的思考。标识的意义就在于提出了个体在环境中搜索和接收信息的具体实现方法。设置良好的、基于标识的相互作用,为筛选、特化和合作提供了合理的基础,这就使聚集体和组织结构得以涌现。标识是隐含在复杂适应系统中具有共性的层次组织结构背后的机制。

6. 内部模型(Internal Model)

霍兰用内部模型来定义主体实现某项功能的机制。在复杂适应系统中,当适应性主体接收到大量涌入的输入时,就会选择相应的模式去响应这些输入,而这些模式最终会凝固



成具有某项功能的结构——内部模型。

7. 积木 (Building Blocks, 也译为构件)

复杂系统常常是在一些相对简单的部件的基础上,通过改变它们的组合方式而形成的。因此,事实上的复杂性往往不在于块的多少和大小,而在于原有积木的重新组合。

内部模型和积木的作用在于加强层次的概念。客观世界的多样性不仅表现在同一层次中个体类型的多种多样,还表现在层次之间的差别和多样性,当跨越层次时就会有新的规律与特征出现。这样一来,我们需要深入考虑的就是:怎样合理地地区分层次,不同层次的规律之间怎样相互联系和相互转化的问题。内部模型和积木的概念就是用来回答这些问题的。概括地说,它们提供了这样一个思路:把下一层次的内容和规律作为内部模型“封装”起来,作为一个整体参与上一层次的相互作用,暂时“忽略”或“搁置”其内部细节,而把注意力集中于这个积木和其他积木之间的相互作用和相互影响上,因为在上一层次中,这种相互作用和相互影响是关键性的,起决定性作用的主导因素。

9.1.3 复杂适应系统的主要特点

在复杂适应系统中,任何特定的适应性主体所处环境的主要部分都由其他适应性主体组成,复杂适应系统中的主体在与环境的交互作用中遵循一般的刺激——反应模式。霍兰把复杂适应系统看成是由相互作用的具有适应性的主体组成、以规则来进行描述的系统。主体具有的主动的、不断适应性是指系统中的每个个体能够在与别的个体或环境的相互作用过程中,不断“学习”或是“积累经验”,主体根据学到的知识和经验不断改变其规则、改变自身的结构和行为方式,具体表现在一边接收输入的不同信息,一边采取不同的行为方式和不停变更自身的结构,以提高自身生存和发展的空间,最终体现为主体适应环境变化的能力提高上。复杂适应系统的整个宏观演变和进化,包括新聚合体的生成、更大主体的出现、新结构和层次的出现、分化以及多样性的出现等,都是基于此基础逐步派生出来的。因此,任何主体不断提高自身的适应性都是为了更好地去适应其他的适应性主体,这个特性是致使复杂动态模式生成的主要根源。

虽然身边存在着具有各自独有特征的不同的复杂适应系统,但所有的复杂适应系统都有一些共同的特点,具体主要表现在以下四个方面。

1. 基于适应性主体 (Adaptive Agent)

适应性主体具有感知和效应的能力,自身有目的性、主动性和积极的“活性”,能够与环境及其他主体随机进行交互作用,自动调整自身状态以适应环境,或其他主体进行合作或竞争,争取最大的生存和延续自身的利益。但它不是全知全能的或是永远不会犯错失败的,错误的预期和判断将导致它趋向消亡。因此,也正是主体的适应性造就了纷繁复杂的系统复杂性。

一般来说,人们认为只有较高等的哺乳动物才具有这样的适应性而不是所有生物体。在实际情况下,所有的生物体都有适应性表现,例如一只老虎会根据地理标志和气味这两者的共同作用来形成一幅相对环境而言的地图,它所表现出来的行动也是受到这幅概念地图影响的;再如一个大肠杆菌能不由自主地游向周围葡萄糖梯度较大的地方。这些都表现了适应性主体具有感知和预测的能力,自身具有目的性、积极性和主动性,能够与其他主体以及环境之间进行交互作用,自觉地调整自身状态,与其他主体合作或竞争,进而去适应周围环境,为自己争取最大的生存空间和延续自身的利益。当然,这些主体不是先知先觉或

事先遇到什么样的情形就知道怎么去处理或永不犯错的,主体错误的感知、预测和判断会导致它趋向消亡。所以,要研究与探索宏观系统的演化现象必须从主体的适应性出发,系统的演化现象也正是在主体与别的主体或环境积极主动的交互作用中形成、发展起来的,并造就了系统的复杂性,主体的积极性程度决定了整个复杂系统行为的复杂程度。这正印证了复杂适应系统理论的核心思想“适应性造就复杂性”。

2. 共同演化

适应性主体通过获得正反馈从而加强其存在,也给自己带来了延续和变化的机会。主体能从一种多样性的统一形式转变成另外一种多样性的统一形式,这个具体过程就是多主体共同演化的过程。系统中无数个主体之间能如此完美地相互适应并共存于环境中就是因为多主体的共同演化结果。就像人类社会,每个个体都需要其他个体的协助来完成某一项工程一样,在接受其他个体帮助或挑战的同时,也给予其他个体帮助和压力。

复杂系统中的适应性主体不是独自演化,而是与其他主体共同演化。共同演化产生了无数能够完美地相互适应并能够适应于其生存环境的适应性主体;共同演化是任何复杂适应系统突变和自组织的强大动力,共同演化会一直趋向混沌的边缘。

3. 趋向混沌的边缘

复杂适应系统有能力将秩序和混沌融入某种特殊的平衡,其平衡点就是趋向混沌的边缘,也就是说,一个复杂系统的各类要素永远不会静止在某一个状态中,但不可能动荡到快解体的地步。一般地,每个主体都会为了提高自身的适应性去加强和其他主体的合作,能根据其他主体的行为来适时地调整自身的行为,以达到为自己争取有利的存在和连续空间,使整个系统在主体的共同演化中趋向于混沌的边缘;另外,混沌的边缘不只是简单地游走于完全有秩序系统和完全无秩序系统之间的区界,而是自我发展并进入一种特殊区界。在这个特殊区界中,系统会产生“涌现”现象。

4. 产生涌现现象

沃尔德罗普提出:“非常有趣的是,复杂行为是从极其简单的元素群中涌现出来的,而非来自复杂的基本结构。比如众所周知的分子结构是因为原子通过相互间的化学键而找到了最小的能量形式的涌现;人类社会通过相互间的买卖和交易关系来获得自身物质需要的满足,从而形成了市场这个无处不见的涌现结构;自然界中的生物体在合作和竞争的双重作用下共同进化,最终形成了和谐稳定的生态系统。”这就是涌现现象的本质特征:从小至大、从简入繁。涌现现象产生的主要根源在于适应性主体在一个或多个毫不关联的简单规则控制下相互作用。主体适应规则体现主体之间的相互作用,这种相互作用具有耦合性和前后关联性,并且这种作用是非线性的,致使涌现出来的整体行为比各部分行为的总和复杂得多。复杂适应系统是由适应性主体相互作用、共同演化并层层涌现出来的系统。在涌现形成过程中,虽然规律本身不会变化,但是这些规律所决定的事物会发生改变,所以会不断地生成无数的新结构和新模式。这些新结构和新模式,既具有动态性又具有层次性,在生成的原有结构基础上,涌现能够再生成新的结构,这些结构具有更多组织层次。换句话说,一种简单的涌现会生成更高层次的涌现,涌现是复杂适应系统层级结构间整体宏观现象的动态表现。

9.2 个体怎样适应和学习

9.2.1 刺激 - 反应模型

霍兰认为系统中的主体在与环境的交互作用中遵循一般的“刺激 - 反应”模型,描绘主体对环境的适应性规则。“刺激 - 反应”模型如图 9 - 1 所示。

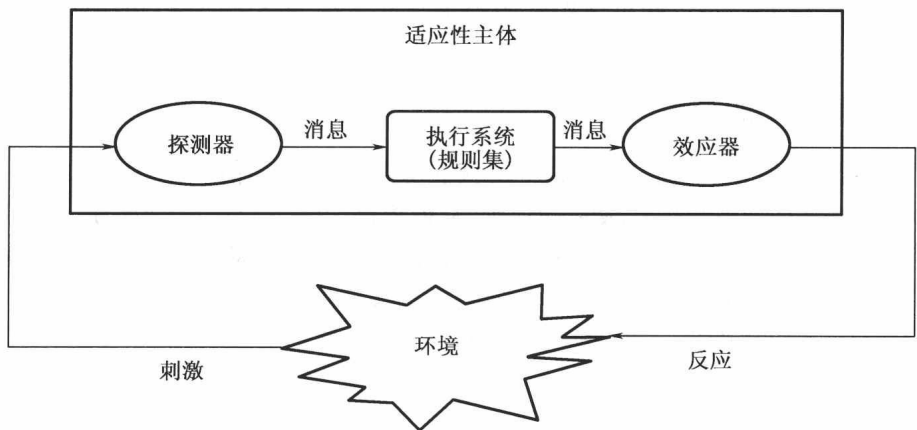


图 9 - 1 “刺激 - 反应”模型示意图

- 刺激 - 反应模型中涉及以下几个的概念：
- 输入——环境(包括其他个体)的刺激；
 - 输出——个体的反应(一般是动作)；
 - 规则——对什么样的刺激,做出怎样的反应的规则；
 - 探测器——接受刺激的器官；
 - 效应器——给出反应的器官。

探测器代表主体从环境中获取信息的能力,它决定了主体感知外部环境变化的能力。IF - THEN 规则代表了处理探测器所抽取到信息的能力,这些规则以“染色体”的方式存在于主体内部,基于一定的条件下被选中,这种选择既有确定性的因素,即按一定的条件挑选,也有随机性的因素,即按一定的概率选择。主体的适应性体现在它与外部环境的交互中,即遵循“刺激 - 反应”模型。主体不断地从外界感知并接受刺激,根据经验做出反应,效应器代表了主体的应对策略,直接作用于外部环境,不同的反应规则对外部环境的影响能力不同。

“刺激 - 反应”模型的基本原理是:主体将探测器感知到的消息与规则集进行匹配,发现匹配规则后可以直接激活效应器产生应对策略,或激活另一个与之相匹配的规则,探测—匹配—激活的反应过程也可以并行,即主体的多个相匹配规则同时参与刺激反应活动。经过多次刺激反应后,符合环境状态的规则被遗传,不符合环境状态的规则被淘汰,并随着时间推移趋于消亡。主体之间的相互作用同样遵循着“刺激 - 反应”模型,发展出交互、排斥、条件交换、复制、结合、繁殖等复杂的相互关系,进而产生分化,直到形成更高一层

的主动个体,并引起整个系统结构的变化。

9.2.2 适应度的确认和修改

为了对规则进行比较和选择,首先要把假设的信用程度定量化,为此给每一条规则一个特定的数值,称为强度(或适应度)。每次需要使用规则的时候,系统按照一定的方法加以选择,选择的基本算法是:按照一定的概率选择,具有较大强度或适应度的规则有更多的机会被选用。在这个基本算法的基础上,还可以加入并行算法和缺省层次等思想,使得规则的选择更加灵活,更加符合现实的系统行为。

当每次应用规则之后,个体将根据应用的结果修改强度或适应度,这实际就是“学习”或“经验积累”。

9.2.3 新规则的产生

经过与环境的对话和交流,已有的规则就能够得到不同的信用指数。在这个基础上,下一步的要点就是如何发现或形成新的规则,从而提高个体适应环境的能力。

新规则产生的基本思想是,在经过检验后的较成功的规则基础上,通过交换、突变等手段创造出新的规则。需要注意的是,由于在这里是基于经验来进行新规则的创造,所以比纯粹根据概率去查找和检验一切可能性要快得多,效率高得多。

9.3 从个体到全局——回声模型

9.3.1 位置和资源

现在我们从微观转向宏观,在前面所定义的主体模型的基础上,建立整个系统的宏观模型,霍兰称之为回声模型(Echo Model)。

从微观到宏观的过渡历来是一个难题。不只在系统科学中,而且在经济学中(微观经济学和宏观经济学)、生物与环境科学中(生物个体与生态系统)、社会科学中(心理学与社会学)都存在着这个问题。现在,我们在前面建立的适应性主体的模型的基础上,来建立由这样的个体组成的复杂系统的模型。

传统科学在这个问题上的偏颇主要表现在以下两个方面:一方面,往往忽视层次之间质的差别,总是试图用一个固定的、统一的模式去套用不同层次的问题,或者简单地把宏观的、系统的属性当作统计规律看待;另一方面,把两者完全割裂开,切断它们之间的有机联系,如在经济学中长期把微观经济学和宏观经济学截然分开,回避其间的矛盾和不一致。

我们仍然分三步走,首先定义资源和位置,然后提出一个基本模型,在此基础上补充一些更复杂的属性,形成最终的模型。

为了给主体一个进行活动的外部环境,我们定义了如下两个概念。

1. 资源

资源的意义十分广泛,例如现实生活中的水井或泉水,它可以不断向主体提供所需要的某种物质或能量。在经济系统中,银行则起着提供资金的作用。

2. 位置

我们可以用城市去理解它。它是一个可以容纳若干个体活动的“容器”，具有一定的环境条件与资源条件，例如，温度、服务水平等。个体可以在位置之间转移和选择。各个位置之间还有“距离”等概念。

不难体会出这两个概念的背景——生物学和经济学。生物界的土地、水分、空气、食物、养料等都是资源。在经济管理中，资金、原材料、能源、人力、技术、信息等也都是资源。然而，在复杂适应系统理论中，资源的含义更广泛，更抽象。任何系统中，为了维持这些“活的”、有主动适应能力的主体的生存与发展，必定要消耗或使用某种资源。事实上，它们不仅是生存的条件，而且是生存质量的标志。资源太少，少到一定程度，主体就会“饿死”。而资源丰富到一定程度，主体就会“繁殖”，分出或产生出新的主体。此外，主体还具备加工资源的能力。它可以用集中不同的资源生产出新的资源，例如一个工厂用不同的原材料，再加上必要的人力资源，能够生产出某种产品。所有这些方面的性质，就是复杂适应系统理论所提出的资源概念的含义。

位置的概念也是从许多实际情况中抽象出来的，它可以是某个市场、某个城市（在经济领域），也可以是某个树林，某个湖泊（在环境和生物科学领域）。对于系统中的个体来说，位置是它们活动的“场所”。不同的场所对于主体的生存和活动来说，提供了若干基础条件：资源的充裕程度，相邻的主体的数量与情况，以及发展的空间与余地等。此外，主体的主动性还表现在它们可以在位置之间移动，即选择适合于自己生存的，更为适宜的位置，正像动物的迁移、人口的迁移等。

9.3.2 回声模型的基本框架

在回声模型中，主体具有最简单的功能——寻找交换资源的其他主体，与其他主体进行资源交流，保存及加工资源。

为此，主体要有三个基本部分：

- ①进攻标识——用于主动与其他主体联系和接触；
- ②防御标识——用于其他主体与自己联系时决定应答与否；
- ③资源库——用于存储和加工资源。

这样，这个主体的基本情况如图 9-2 所示。

它的功能包括：主动与其他主体接触，同时也对其他主体的接触进行应答，如果匹配成功则进行资源交流，在自己内部存储与加工资源，如果资源足够，则繁殖新的主体。

在此基础上，整个回声模型成为如下情况：整个系统包括若干个位置，每个位置中有若干个主体，主体之间进行交往，交流资源和信息。这就是最基本的回声模型。

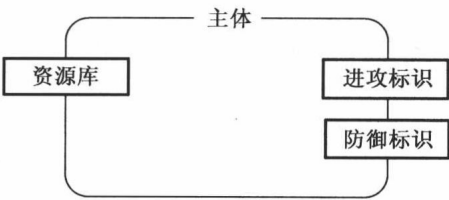


图 9-2 回声模型中主体的基本模型

9.3.3 回声模型的扩充

基本的回声模型还过于简单，无法描述复杂的系统行为，霍兰在此基础上增加了各种功能，形成了扩展的回声模型。在《隐秩序》一书中，他进行了初步的扩充，先后加入了以下

一些机制。

1. 增加“交换条件”机制

即在进攻标识与防御标识相符的条件下,还要加上某种交换条件的确认。比如,在采购原料时,并不只是有某种原料就行,还要考虑价格和质量。

2. 增加“资源转换”机制

即主体具备加工、利用和重组资源的能力。增加这一功能为主体的分工和专门化打下了基础。

3. 增加“黏着”机制

即若干主体通过建立固定的联系,成为一个多主体的聚合体在系统中一起活动。显然,它的来源是生物界的共生体以及经济活动中的企业集团。

4. 增加“选择交配”机制

即主体可以有选择地与其他主体结合,通过交换形成新的更强的主体。

5. 增加“条件复制”机制

即主体在资源充裕、条件适宜的情况下,复制增加自身的功能。

扩充的回声模型如图 9-3 所示。

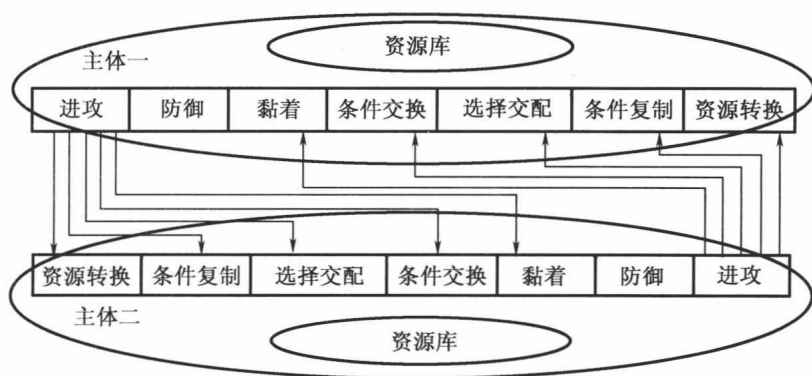


图 9-3 扩充回声模型示意图

通过这一步步扩充,回声模型的表达和描述能力不断增强,从而具备了描述和研究各种复杂系统的能力。不难看出,所有这些模型都有着明显的经济、生物、生态领域的背景和影响。所以,复杂适应系统理论的研究并不是单纯理论思考的产物,而是在大量现实的复杂适应系统的考察与研究的基础上逐步抽象出来的。

当然,扩充到这种程度的回声模型仍然是十分简单的。这几步扩充还只是初步的。不难看出,生物、经济、社会等还有许多丰富多彩的现象和规律还没有加以描述和反映。这主要表现在以下几个方面。

①主体的功能单纯,结构单纯,类型单一。也就是说,他还不具备加工资源的能力;也没有分化;也没有选择性,包括对于环境和交换资源的对象。

②主体还不具备繁殖的功能,即只能根据其资源的状况确定能否生存下去,而不能进一步产生新的主体。

③信息的收集和处理能力还比较简单,进攻标识和防御标识都只具备基本的功能,还不能进行更为复杂的信息识别、分类等活动。

④资源的种类只是一种,因此也没有加工产生新资源,以及更为复杂的资源类型。

9.4 复杂适应系统建模与仿真工具 NetLogo 简介

NetLogo 是一个多主体建模仿真集成环境,特别适合对随时间演化的复杂系统进行建模仿真。它由 Uri Wilensky 在 1999 年发起,由美国西北大学连接学习和计算机建模中心 (Center for Connected Learning and Computer – Based Modeling, CCL) 负责持续开发,目的是为科学研究和教育提供易用且强大(称为“Low threshold, no ceiling”)的计算机辅助工具。NetLogo 是免费软件,可以在不同的平台上运行,目前最新的版本是 5.2.1 (可通过 <http://ccl.northwestern.edu/netlogo/> 进行下载)。

9.4.1 NetLogo 的主要功能

1. 建模

NetLogo 模型的基本假设是:将空间划分为网格,每个网格是一个静态的主体 (Patches),多个移动主体 (Turtles) 分布在二维空间中,每个主体自主行动,所有主体并行异步更新,整个系统随着时间推进而动态变化。主体的行为用编程语言定制,NetLogo 中的编程语言是一种 Logo 方言,支持主体操作和并发运行。

2. 仿真运行控制

NetLogo 可以采用命令行方式或通过可视化控件进行仿真控制。在命令行窗口可以直接输入命令,另外还提供了可视化控件实现仿真控制,进行仿真初始化、启动、停止、调整仿真运行速度等。还提供了一组控件,如开关、滑动条、选择器等,用来修改模型中的全局变量,实现仿真参数的修改。

3. 仿真输出

提供了多种手段实现仿真运行监视和结果输出。在主界面中有一个视图 (View) 区域显示整个空间上所有主体的动态变化,可以进行 2D/3D 显示,在 3D 视图图中可以进行平移、旋转、缩放等操作。另外,可以对模型中的任何变量、表达式进行监视,可以实现曲线/直方图等图形输出,或将变量写入数据文件。

4. 实验管理

NetLogo 提供了一个实验管理工具行为空间 (BehaviorSpace),通过设定仿真参数的变化范围、步长、设定输出数据等,实现对参数空间的抽样或穷举,自动管理仿真运行,并记录结果。

5. 系统动力学仿真

系统动力学是应用广泛的一类社会经济系统仿真方法,但与多主体仿真有不同的建模思想。NetLogo 可以直接进行系统动力学建模仿真。

6. 参与式仿真

NetLogo 提供了一个分布式仿真工具,称为 HubNet,实现模型服务器和客户端之间的通信。多个参与者可以通过计算机或计算器分别控制仿真系统的一部分,实现参与式仿真 (Participatory Simulation)。

7. 模型库

NetLogo 收集了许多复杂系统经典模型,涵盖数学、物理、化学、生物、计算机、经济、社

会等许多领域。这些模型可以直接运行,例子中的文档对模型进行了解释,为可能的扩展提供了建议。建模人员可以通过阅读经典实例的程序代码,学习建模技术,或在研究相关问题时以此为基础进行扩展或修改,大大减少了技术难度和工作量。

9.4.2 NetLogo 特点

1. 系统运行环境

NetLogo 提供免费开放的软件系统,能够跨平台使用,可以在 Mac, Windows, Linux 等平台运行。

2. 系统开发语言

- ①完全可编程语言,简单语言结构;
- ②语言是由 Logo 语言扩展而来,能够支持主体建模;
- ③移动主体(Turtles)在由静态主体(Patches)组成的网格上移动;
- ④主体之间可以创建链接,形成聚集、网络和图;
- ⑤内置大量原语;
- ⑥一流的函数值;
- ⑦运行过程在不同平台上完全可复现环境。

3. 系统仿真环境

- ①有可以进行运行中(on-the-fly)交互的命令中心;
- ②界面构建,包括按钮、滑动条、开关、选择器、监视器、文本框、注解、输出区;
- ③信息页用来解释模型;
- ④HubNet,使用联网设备进行参与式仿真;
- ⑤主体监视器用来监视和控制主体;
- ⑥输出输入功能(输出数据,保存、恢复模型状态,制作电影);
- ⑦行为空间(BehaviorSpace),工具用来从多次运行中收集数据;
- ⑧系统动力学建模;
- ⑨可以用 NetLogo 3D 建造 3D 世界模型;
- ⑩任意模式可以在命令行批量运行程序。

4. 展示与可视化

- ①线图,条形图和散点图;
- ②快进滑动条使你可以对模型进行快进和慢放;
- ③用 2D 或 3D 模式查看模型;
- ④可伸缩、旋转矢量图形;
- ⑤海龟和瓦片标签。

5. 系统 Web 化

模型和 HubNet 客户可以存为 Applets 嵌入 Web 页面中(但是有些特性不能从 Applets 获得,像某些扩展功能与 3D 显示)。

6. APIs 扩展

- ①控制 API 可以在脚本或应用中嵌入 Netlogo;
- ②扩展 API 可在 Netlogo 语言中加入新的命令报告,包括开源的扩展例子。

9.4.3 NetLogo 平台功能介绍

NetLogo 仿真界面如图 9-4 所示,主窗口分为模型部分和指令中心窗口部分。在指令中心窗口可以直接输入指令直接与模型进行交互,而不需将这些指令加入模型的程序。模型部分又分为“界面”“说明”和“程序”三个标签页面。“界面”页是模型的可视化接口,包括各种参数、图形、动画输入/输出控件和仿真世界图形化显示窗口;“说明”页是对模型的说明,如解释如何使用模型、探索的主要问题以及可能的扩展等;“程序”页是模型的程序代码部分。“界面”页中的可视化控件与“程序”页中的程序存在密切的联系,是建模时必须编写的部分。

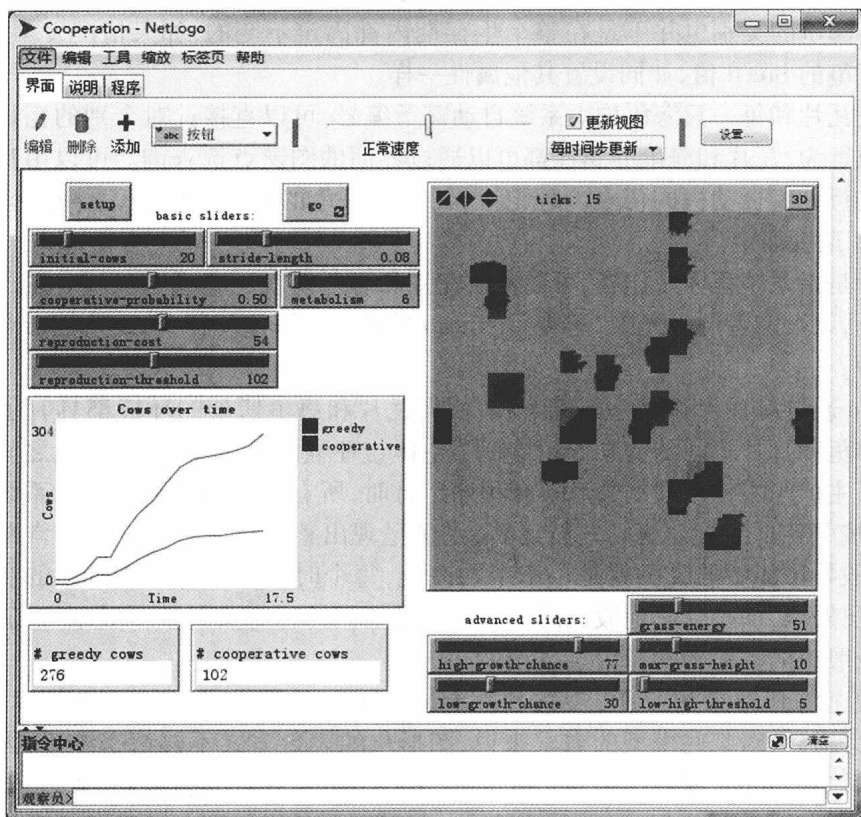


图 9-4 NetLogo 仿真界面

程序模块 (Procedures) 由一些程序及变量声明所组成。程序的始末由「to...end」所界定。界面的可视化控件主要有按钮 (Button)、滑动条 (Slider)、开关 (Switch)、监视器 (Monitor)、图形 (Plot) 等。按钮对应到某些主要的程序,滑动条是能让使用者操控的全局变量 (Global Variables),开关是 on/off 的全局变量,监视器可以显示程序内部的全局变量值的数值变化,图形输出可以用图形显示程序内部的全局变量值的动态变化历程。

NetLogo 仿真系统有四类主体,分别为海龟 (Turtles)、瓦片 (Patches)、观察者 (Observer) 和链 (Links)。用户通过建立这些主体来构建复杂系统。瓦片相当于海龟存在的外部环境,也是海龟的活动范围,为网格系统,纵横格数可以在界面上直接拉框调整 (或设定其格数),坐标原点 (0,0) 在正中央。瓦片并不是个消极的对象,它同海龟类似,也是具有自主性的主体,拥有行为、属性、类别等基本特性。瓦片主要的固有属性是颜色和位置。同样,也可以

视仿真需要给它们定义一些新的属性。值得强调的是,瓦片并不仅仅只是受海龟作用的被动的对象,它们也可以有主动的行为,也可以作用于海龟和其他的瓦片。

海龟是 NetLogo 世界里的主要活动对象,海龟可在瓦片的网格环境里自由移动。每一个海龟就是一个独立的行为主体。可以用海龟来代表现实世界中的任何一个对象,对海龟制定相应的行为规则。NetLogo 给每一个海龟赋予了一些固有属性(如位置、方向、颜色等),用于描述对象主体的状态。同时,还可以根据具体仿真主体的特殊性质,给海龟添加一些新的属性。在 NetLogo 中,可以并行控制多个海龟的个体活动以及它们之间的交互行为。NetLogo 允许对海龟进行分类,用于描述不同种类的对象。这一点对于仿真现实世界来说非常必要,因为即使在一个很小的系统内,也常常存在着多种具有不同属性和行为方式的角色。海龟的类别也并不是在整个生存期内都固定不变的,如果情况发生变化,可以重新设置海龟的 breed 值,如同设置其他属性一样。

每一块瓦片和每一只海龟均由系统自动赋予编号,可以直接针对个别瓦片和海龟改变其属性或行为,瓦片和海龟的属性都可以通过界面的图窗点选查询。可以用瓦片的指令很方便地进行元胞自动机的仿真。理论上,瓦片和海龟的数目无限制,上限视计算机内存容量及 CPU 速度而定。

观察者是指系统程序写作者,可以操控瓦片和海龟以及一些系统环境变量。它相当于一个观察员,以第三者的眼光来“透视”NetLogo 世界中的海龟和瓦片,能监控所有的海龟和瓦片的行为。

NetLogo 通过对瓦片和海龟进行编程,使得瓦片和海龟所处的环境都具有了自己的变化方式。海龟和瓦片之间的交互作用体现了主体与环境的关系,海龟与海龟之间的相互影响则体现了主体与主体之间的关系。在程序运行时,所有的海龟和瓦片都以系统规定的时间步前进,按照各自的行动规则并行运转,系统呈现出整体行为。这一点非常贴近现实世界。例如,股票市场中的股民就是一个一个的海龟,每个时刻他们都根据自己的利益采取行为,买进卖出股票,而不受其他股民的支配。但是他们的行为又与整个股票市场的状况、其他股民以及股市的控股者的行为息息相关。

为了便于仿真结果的分析、观察和控制,NetLogo 还提供了良好的操作界面。此外,NetLogo 还考虑到,对某些模型的需求来说,图形化的表现方式不够精密准确,因此它还提供了 plot 窗口和 output 窗口。在 plot 窗口中,用户可以选择创建折线图、条形统计表、柱状图等来精确地刻画某些变量。output 窗口则可以用来记录在程序运行过程中输出的变量值,便于用户对模型进行分析。

9.5 NetLogo 建模仿真框架

9.5.1 NetLogo 软件

NetLogo 软件主窗口分为模型部分和指令中心窗口部分,其中模型部分又分为“界面”“说明”和“程序”三个标签页面。其界面如图 9-4 所示。

9.5.2 模型的抽象

总体上,NetLogo 模型可抽象为大量的可移动主体在二维空间中交互作用,随着时间推



进,微观个体的属性不断发生变化,系统的宏观特征也因此而变化。

1. 主体

虚拟世界由主体构成,主体能够接受命令,进行活动,所有主体的行为并行发生。NetLogo 中共有四类主体:海龟(Turtles)、瓦片(Patches)、观察者(Observer)和链(Links)。NetLogo 的虚拟世界示意图如图 9-5 所示。

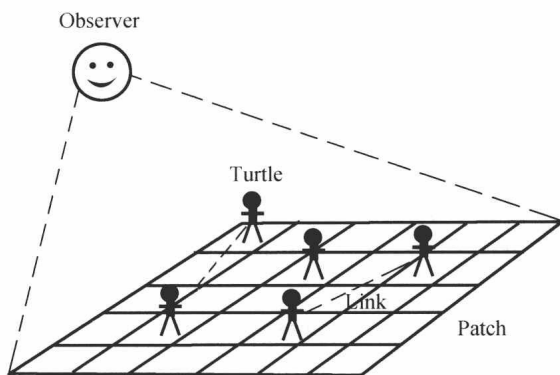


图 9-5 NetLogo 的虚拟世界示意图

2. 空间表达

每个瓦片都有二维坐标($pxcor, pycor$),坐标值为整数。默认情况下,二维世界的水平、垂直坐标范围为 $[-17, 17]$ 。每个海龟也有坐标($xcor, ycor$),海龟的坐标不必是整数,因此海龟不一定正好位于某个瓦片的中心。一个瓦片上也可以同时有多个海龟。实际上对海龟而言,NetLogo 的空间是连续的。链只能存在于海龟之间。

3. 仿真推进

仿真推进没有明确的仿真时钟变量,也没有提供特定的事件处理机制。仿真推进是通过不断重复执行某个程序实现的,模型中至少要有初始化程序和仿真执行程序。初始化程序实现对模型初始状态的设置,生成所需的海龟,设置其状态,以及其他工作。仿真的执行通过程序 go 实现,在 go 程序中编写所需执行的各种指令,完成一个仿真步的工作。在“界面”页中需要建立一个按钮与 go 程序相联系,该按钮是一个永久(Forever)按钮,点击后将不断重复执行 go 程序,直到遇到 stop 指令或用户再次点击该按钮则仿真终止。

9.5.3 建模仿真基本过程

NetLogo 模型包括可视化控件和程序两部分,二者具有紧密联系。建模时需要先在“界面”中创建可视化控件,然后在“程序”中实现相应的代码,通过设置控件的属性将二者联系起来。

“界面”中主要有三类控件:运行控制、参数控制和仿真显示。“程序”中的程序分为两类:指令(Command)程序和报告(Reporter)程序。

建模仿真基本过程如下:

- ①初始化;
- ②仿真执行程序;
- ③定义主体属性与行为;

- ④监视仿真运行;
- ⑤图形输出;
- ⑥仿真参数控制。

9.6 NetLogo 建模技术

9.6.1 主体交互

1. P-T 交互

P-T 交互即瓦片与海龟之间的交互行为。最为常见的一种方法是,海龟能够直接访问所在之处的瓦片,对该瓦片的属性进行读写,其程序语句为:

```
ask turtles [ set pcolor blue ]
```

也可以给出海龟的行动指令,对指定的瓦片的属性进行读写,其程序语句为:

```
ask turtles [ set pcolor-of patch-at 1 0 blue ]
```

同样的,瓦片也可以获取相关的海龟,利用 Turtles-here 返回当前瓦片处的海龟集合。

也可以通过空间相关操作获取相应瓦片上的海龟。

例如,“turtles at dx dy”返回与当前瓦片相对距离(dx,dy)处的海龟集合。

2. T-T 交互

T-T 交互即海龟与其他海龟之间的相互作用。实现 T-T 交互的第一步是得到目标海龟的句柄,然后进行操作。获取目标海龟句柄的常用方式有三种:随机选取、根据特定条件、空间相关。

(1) 随机选取

随机选取是指在特定主体集合中以随机方式选取一个或 n 个主体。选取一个主体,其原语为 one-of。

例如,“set color-of one-of turtles red”在所有海龟中随机选择一个海龟,将其颜色设为红色。

(2) 根据特定条件

使用 with 原语,语法为 agentset with [condition], 返回满足条件 condition 的主体集合。

例如,“turtles with [color = red]”返回红色的海龟集合。

(3) 空间相关

比较直接的一种方式获取当前瓦片上的海龟集合,原语有 Turtles-here 和 Other-turtles-here,二者的区别仅在于是否包含调用者自身。

例如,“ask turtle 0 [ask other-turtles-here [fd 10]]”表示 ID = 0 的海龟令处于同一瓦片上的其他海龟前进 10。

另一种方式是,获取特定瓦片上的所有海龟。原语有 <turtles> -at dx dy, <turtles> -on agentset 等。

例如,“turtles-at 10”返回右侧紧邻瓦片上的所有海龟,“turtles-on patch-ahead”返回前方瓦片上的所有海龟。

3. P-P 交互

海龟所处的环境由瓦片拼合而成,瓦片具有某种程度的元胞自动机特征,也具有属性和行为,通过使用 Neighbors (Moore 邻域) 及 neighbors4 (Von Neumann 邻域) 原语设定演化规则,改变临域瓦片的变量值,也可以以此来影响附着其上的海龟的行为。

9.6.2 持久关系的建立

基本方法是在个体中相互保持对方的引用,这样可以随时对对方进行操作。例如,在个体之间建立简单的伙伴关系:

①假设还没有找到伙伴的个体随机移动,直到与另一个尚没有伙伴的个体建立伙伴关系,一旦建立伙伴关系后将长期保持。

②为实现这一模型,首先为海龟增加变量 Partner,用于保持对伙伴的引用,然后在移动时从相遇的一些无伙伴的海龟中选择一个,相互建立伙伴关系。

9.6.3 多类异质主体

NetLogo 可以定义不同类别(称为 Breeds)的海龟,各类海龟可以拥有特有的属性和程序。定义关键词为 breed,例如,NetLogo 模型库的“Biology”中的“Wolf Sheep Predation”模型中可以定义两种类型:

①breed [wolves wolf];

②breed [sheep a-sheep]。

一旦定义了一类海龟,系统自动创建该类所有海龟的集合,一些相关的原语也马上可以使用了,例如,对于 sheep 类就有 create-sheep, hatch-sheep, is-a-sheep? 等。也可以指定该类具有的变量,例如,对于 sheep,“sheep-own [grabbed?]”就为 sheep 增加了一个变量。

9.7 NetLogo 建模仿真操作学习

你是否参加过聚会,注意过人们是怎样聚集成小组的吗? 你可能注意到人们并非一直停留在一个小组里,而是走来走去。当个人走来走去时,小组就发生变化。如果你长期观察这种变化,应该注意到模式的形成。

例如,在社交场合人们倾向于展示出与工作或家庭中不同的行为。那些在工作中信心满满的人可能在社交场合变得羞怯,而那些在工作中安静保守的人却可能与朋友发起聚会。

聚集模式也取决于聚会的性质。在某些场合,人们接受训练组织成混合小组,例如聚会游戏或校园活动。但在非结构化的气氛里,人们以更加随机的方式形成小组。

这种分组行为有没有什么模式呢?

让我们使用计算机对聚会中人们的行为建模,更详细地考察这个问题。NetLogo 的“Party”模型从性别这个特殊角度考察这个问题:为什么这些小组多数是男性,或多数是女性?

我们使用 NetLogo 研究这个问题。操作步骤如下：

- (1) 启动 NetLogo。
- (2) 在“文件”菜单中选择“模型库”，如图 9-6 所示。

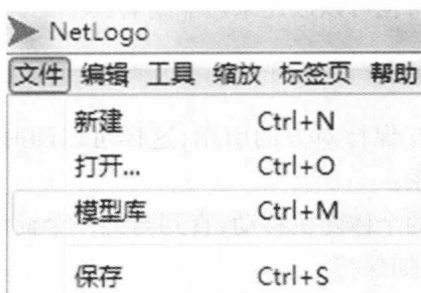


图 9-6 “文件”菜单示意图

- (3) 打开文件夹“Social Science”。
- (4) 点击名为“Party”的模型，如图 9-7 所示。



图 9-7 “Party”模型示意图

(5) 点击“打开”，进入“Party”模型界面，如图 9-8 所示。

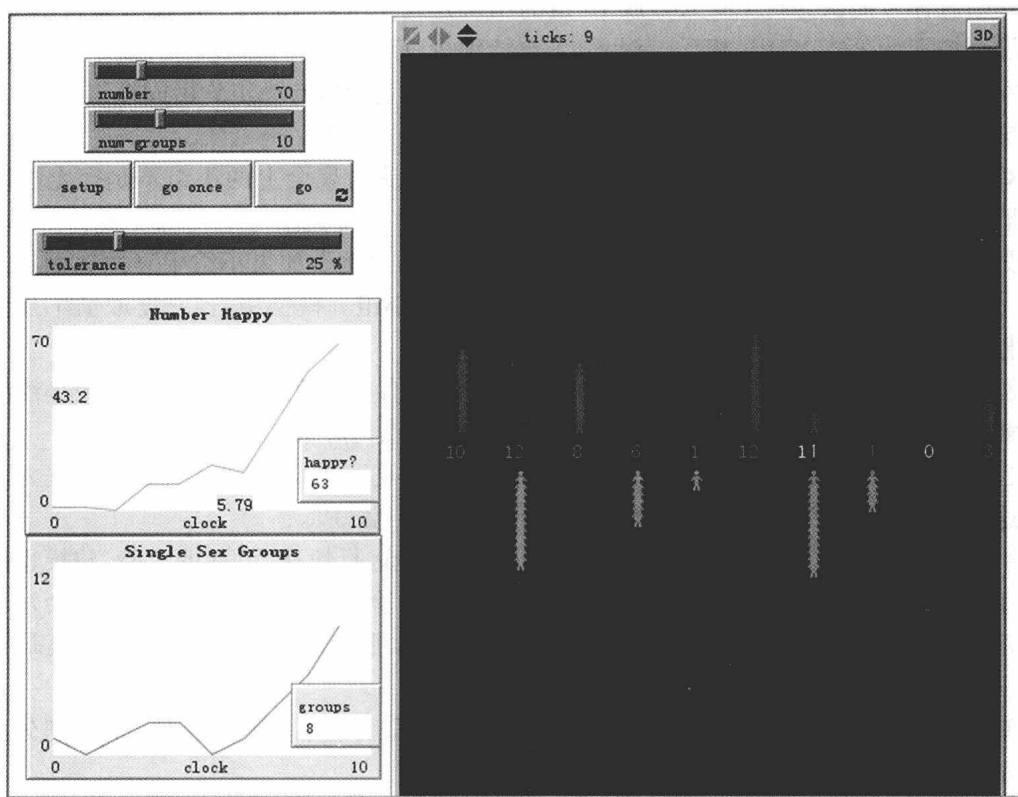


图 9-8 “Party”模型界面示意图

(6) 点击“setup”按钮，在视图中形成初始的小组分组情况，如图 9-9 所示，其中，数字表示每个小组的人数，数字上方灰暗的小人图标代表男性，下方明亮的小人图标代表女性。

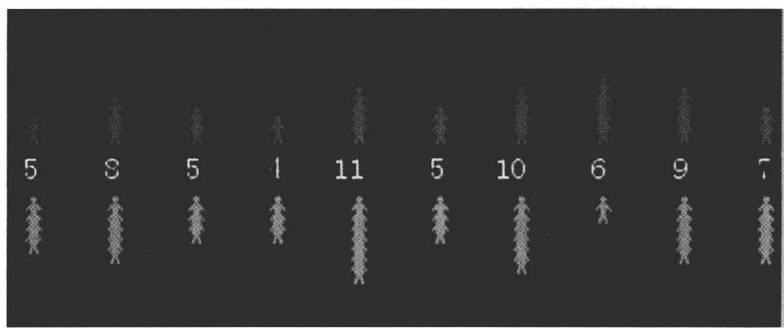


图 9-9 聚会上小组初始分组情况示意图

例如，你邀请了 150 人参加聚会，你想知道人们怎样扎堆。假设预计将人们分成 10 组。你怎么思考分组情况？这里我们使用计算机仿真。仿真操作步骤如下：

- ① 按下“go”按钮（再次按下“go”会停止模型运行）；
- ② 观察人们的移动直到模型停止；
- ③ 看图形输出了解发生了什么。

现在每组有多少人?

开始时你可能认为将 150 人分成 10 组的结果是每组大约 15 人。从模型运行得知,人们并没有均等地分成 10 组,相反,有些组人数特别少,而有些组人数却特别多。另外,随着时间发展,小组成员组成从初始的所有小组男女都有转变为所有小组均由同性组成。

这怎么解释?

对这个问题有很多可能的回答。本模型的设计者认为聚会上的小组不是完全按随机方式形成的,小组如何形成取决于个体的行为。模型设计者关注一个特殊变量“tolerance”(容忍度),这里将容忍度定义为个体感到舒服的异性的比例。如果小组中异性比例超过容忍度,他们就觉得不舒服,因此离开这一组去寻找别的小组。模型中,容忍度取值不是固定的,用户可以用滑动条进行调整。

例如,如果容忍度水平设为 25%,那么一个男性只有在女性比例少于 25% 的小组里才感到舒服。同样女性只有在男性少于 25% 的小组里才感到舒服。

当个体变得不舒服时选择离开,移动到别的小组,这可能又让这个组中的某些人不舒服。这种链式反应不断进行,直到聚会上的所有人都感到舒服。

用 NetLogo 对聚会这样的情景建模使你可以对系统进行快速、灵活的实验,建模也给了你少受偏见的影响去观察各种情景的机会,因为你可以检查系统内部的动态。你会发现随着你建模越来越多,对许多现象的原有的想法会遇到挑战,而在现实情况下建模是很困难的。

例如,聚会模型中一个令人惊讶的结果是,即使容忍度水平相对较高,大量性别分离仍然会发生。

这是关于“涌现”现象的一个经典例子,这里小组模式是许多个体交互的结果。“涌现”思想可以应用在几乎任何领域。

9.7.1 模型(Model)

通过聚会的例子我们已经简单地了解了如何与 NetLogo 模型进行交互。接下来我们深入了解一些 NetLogo 模型的功能。

我们先看一个模型实例,这是一个捕食者 - 猎物种群模型:狼吃羊(Wolf Sheep Predation)。从文件菜单打开模型库,从 Biology 中选择“Wolf Sheep Predation”,打开如图 9-10 所示的狼吃羊模型界面。

界面标签页填满了许多按钮、开关、滑动条和监视器。按钮用于设置、启动、停止模型;滑动条和开关用于修改模型配置;监视器和绘图用于显示数据。

第一次打开模型时,看到视图是空的(全黑)。要让模型开始,需要先进行初始设置,即点击“setup”按钮。点击“go”按钮,狼吃羊模型开始仿真。

1. 控制模型:按钮(Button)

按钮按下后模型就会通过执行一个动作做出响应。按钮分为“一次性”(Once)和“永久性”(Forever)两种,其中永久性按钮的右下角有两个箭头,一次性按钮和永久性按钮如图 9-11 所示。一次性按钮执行动作一次然后停止。当动作完成后,按钮弹起。永久性按钮不断地执行一个动作,当你想让动作停止时,再次按下按钮,它会完成当前动作,然后弹起。

大多数模型,包括狼吃羊模型,有一个一次性按钮称为“setup”和一个永久性按钮称为“go”。许多模型还有一个一次性按钮称作“go once”或“step once”,它们很像 go 按钮,但区

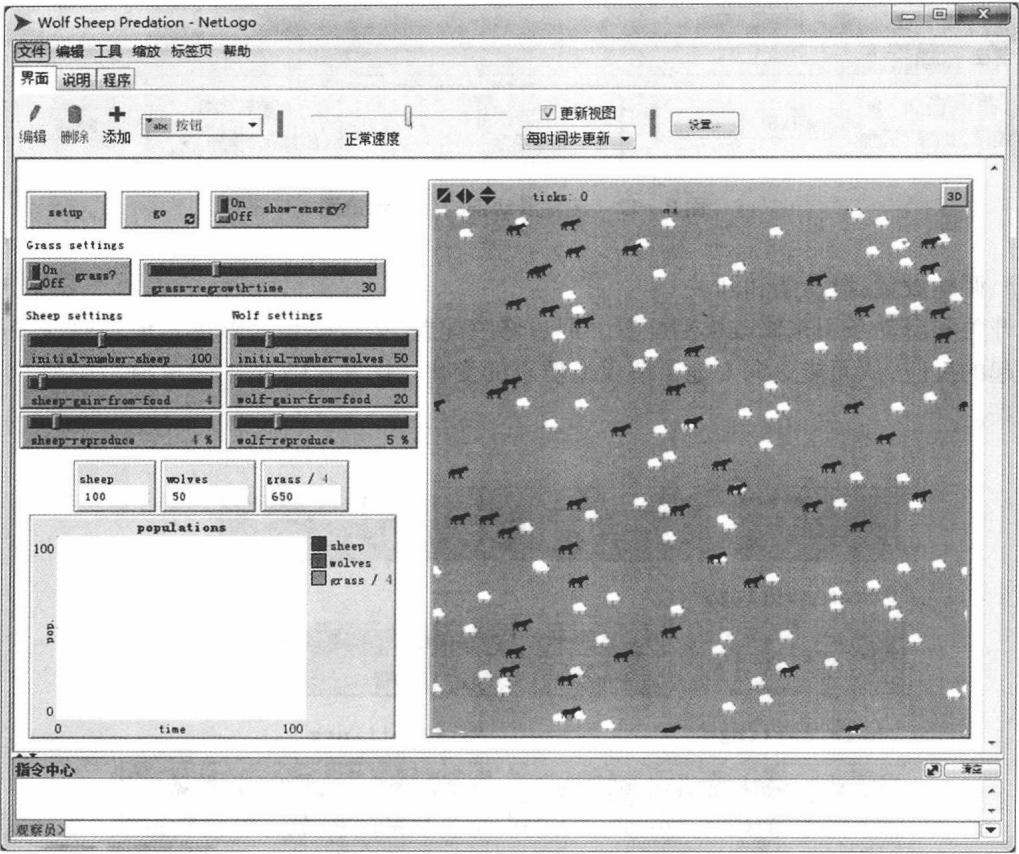


图 9-10 狼吃羊模型界面示意图



图 9-11 一次性按钮和永久性按钮示意图

别在于它们只执行一步(时间步长)。使用这样的一次性按钮能让你更仔细地查看模型的运行过程。

通过停止永久性按钮暂停模型运行,然后再次按下按钮让模型继续,这非常安全。也可以使用 Tools 菜单的“Halt”停止模型运行,但是只有当模型因某种原因卡住时才应该这样做。使用“Halt”可能会让模型在某次行动的中间停住,这可能导致模型乱套。

2. 控制速度:速度滑动条

速度滑动条控制模型运行速度,就是海龟的移动速度、瓦片颜色改变的速度,等等。速度滑动条的位置如图 9-12 所示。

滑块左移使模型速度变慢,每个时间步之间的暂停时间更长,这样更容易观察;滑块右移使模型速度变快。注意,如果滑块太靠右的话,视图更新太频繁,看起来好像变慢了,但实际并没有变慢,你可以根据时钟显示确认这一点,只是视图更新频率降低了。

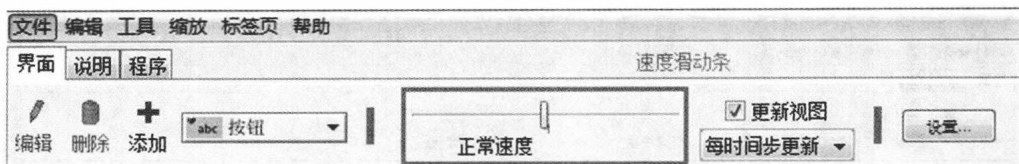


图 9-12 速度滑动条的位置示意图

3. 调整设置:滑动条和开关

开关和滑动条用来修改模型配置。通过修改配置,运行模型能够更好地观察这些改变所引起的反应,从而能更深入地了解模型仿真的变化。图 9-13 是狼吃羊模型中的开关和滑动条。

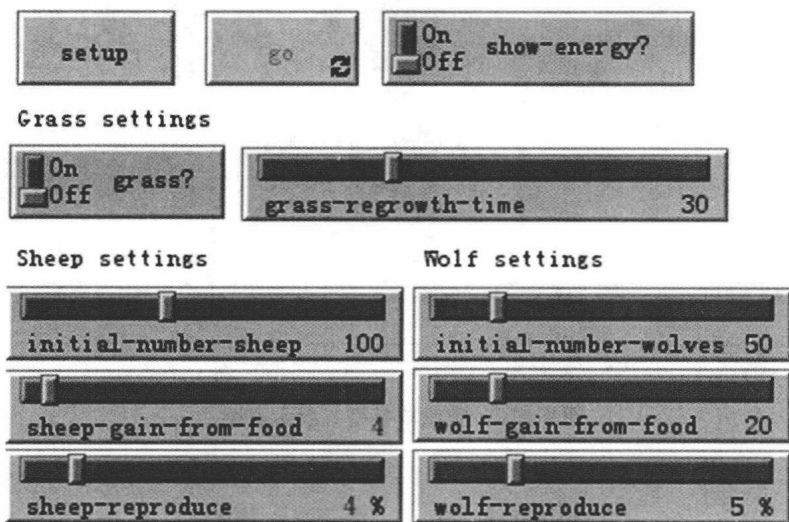


图 9-13 狼吃羊模型中的开关和滑动条示意图

(1) 开关(Switch)

- ①打开“grass?”开关;
- ②按下“setup”和“go”,运行与上次差不多相同的时间。

这个开关对模型有什么作用? 结果和上次一样吗?

像按钮一样,开关也有与它相连的信息,这些信息采用开/关格式。开关发出特别的指令,这些指令对模型并非必要,但为模型增加了附加的维度。打开“grass?”表示在模型中增加一个新因素“草”,并且可以通过“grass-regrowth-time”滑动条调整“草”的增长率。此时模型可以对羊、狼和草三个因素进行仿真分析。

(2) 滑动条(Slider)

滑动条是不同于开关的一种配置类型,开关有两个值,即开(on)或关(off),而滑动条则是一个可调的数值范围。例如,“initial-number-sheep”滑动条最小值为0,最大值为100。模型运行时可以有0只羊,也可以有100只羊,或者中间的任何一个数值。当从左到右移动滑块时,滑动条右侧的数字变化,用以显示滑块对应因素的数量变化情况。

如果仿真开始时羊更多而狼更少,会怎么样?

- ①关掉“grass?”开关;
- ②设置“initial - number - sheep”滑动条为 100;
- ③设置“initial - number - wolves”滑动条为 20;
- ④按下“setup”和“go”;
- ⑤模型运行约 100 时间步。

尝试重复运行模型几次。

羊群数量发生了什么变化?

结果令你惊讶吗? 调整哪些开关、滑动条能帮助羊群?

- ①设置“initial - number - sheep”为 80,“initial - number - wolves”为 50;
- ②设置“sheep - reproduce”为 10.0% ;
- ③按下“setup”和“go”;
- ④模型运行约 100 时间步。

本次运行狼群怎么样?

4. 收集信息:绘图和监视器

建模仿真的一个目的是对那些难以在实验室中进行研究的问题收集数据。NetLogo 主要有两个显示数据的方式:绘图和监视器。

(1) 绘图(Plots)

狼吃羊中的图有三条线:羊、狼和草/4(草除以 4 的原因是为了不使图形太高)。这些线显示了随着时间推进模型中发生的变化,如图 9-14 所示。

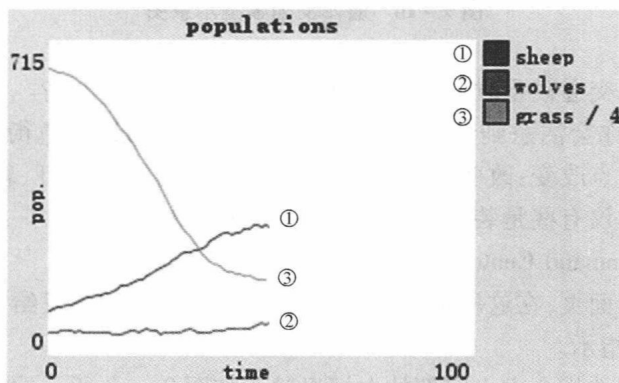


图 9-14 狼吃羊模型的绘图示意图

当图快被充满时,水平轴增加,以前的数据被压缩只占一部分空间,更多的空间用来绘制将来的图形。

如果想保存图上数据以备查看,或在另一个程序里进行分析,使用文件菜单中“导出”的“导出当前绘图的数据...”。数据格式可以被电子表格(如 Excel)或数据库程序识别。

(2) 监视器(Monitors)

监视器是模型显示信息的另一种方法。狼吃羊模型中的监视器如图 9-15 所示。

狼吃羊模型的监视器分别用来表示模型仿真过程中狼、羊、草的数量。当模型运行时监视器中的数值不断更新,而图形能显示模型整个运行过程中的数据。

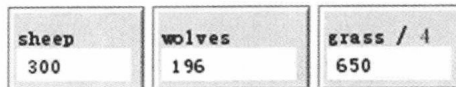


图 9-15 狼吃羊模型中的监视器示意图

9.7.2 指令 (Commands)

在这一节中,我们将重点学习如何操纵 NetLogo 模型。

1. 模型实例:基本交通模型(Traffic Basic)

打开模型库中“Social Science”部分的“Traffic Basic”。在这个模型里,你会注意到一队列蓝车里有一辆红车,车流同向移动,如图 9-16 所示。这些车时不时地会挤成一堆,无法移动。这是关于幽灵式阻塞的模型,即有时交通流会出现阻塞,但却找不到任何明显的原因(Centralized Cause),例如事故、断桥、侧翻的卡车等。

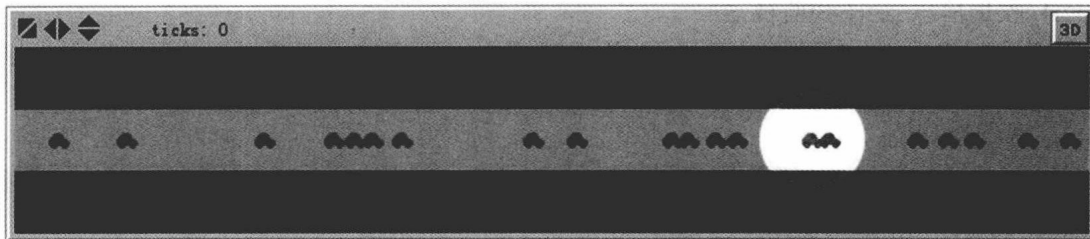


图 9-16 基本交通模型示意图

当使用这个基本交通模型时,是否注意到需要给模型增加什么?

看看这个模型,你会注意到环境太简单了,就是黑色背景、白色街道、一些蓝车和一辆红车。需要对模型做点改变:改变车的形状和颜色、加上房子或路灯、新建信号灯或者再创建一条车道。这些建议有些是装饰性的,只是改善模型的观感,另外一些是行为性的。

2. 指令中心(Command Center)

指令中心位于界面页,在这里可以向模型发出指令,指令就是发给 NetLogo 主体(海龟、瓦片、链、观察者)的指示。

在基本交通模型的指令中心底部的白框里输入如图 9-17 所示的指令。

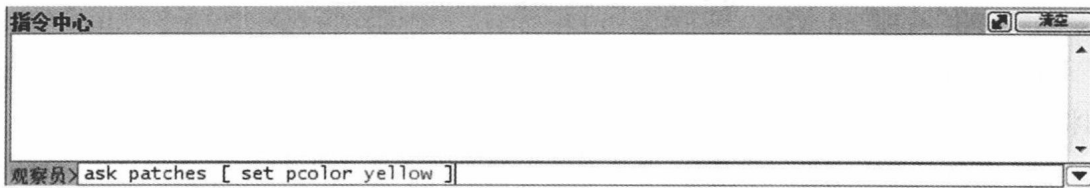


图 9-17 基本交通模型的指令中心示意图

按回车键,视图发生了相应的变化,如图 9-18 所示。视图背景变成了黄色,街道消失了。

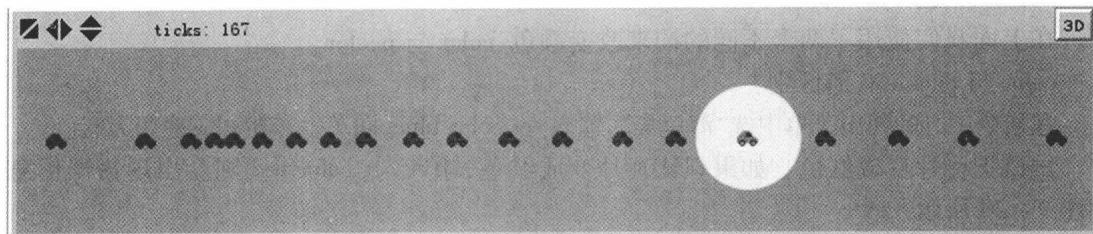


图 9-18 视图背景变化情况示意图

为什么车没有变成黄色？

回头看看刚才输入的指令,这里只是请求瓦片改变颜色。在这个模型里,车辆用“海龟”表示,因此车辆并没有接收指令,也就没有改变。

同样,在指令中心底部的白框中输入下面的指令:

```
ask turtles [ set color brown ]
```

视图变为黄色背景,中间是一串灰色的车。

NetLogo 是由海龟(Turtle)、瓦片(Patch)和观察员(Observer)组成的二维世界。瓦片构成背景,海龟在背景上移动,观察员是观察着所有事情的一个主体。在指令中心,可以给海龟、瓦片和观察员发出指令。通过指令中心左下角的弹出式菜单可以选择指令发出的主体,如图 9-19 所示。

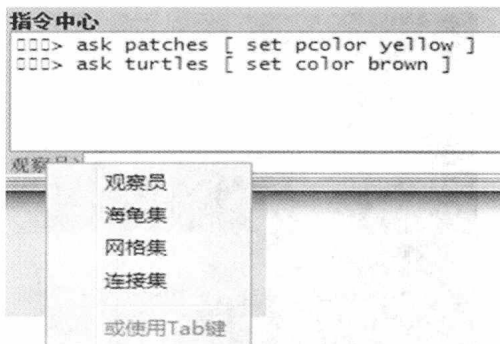


图 9-19 指令发出主体选择示意图

进行如下操作:

选择指令发出主体为“海龟集”,输入指令 `set color pink`,回车;

然后选择指令发出主体为“网格集”,输入指令 `set pcolor white`,回车。

执行指令后,视图变回了原样。

你注意到这两条命令和前面的“观察员”命令的区别了吗?

观察员俯视着世界,因此使用 `ask` 向瓦片或海龟发出指令。正如前面第一条指令例子那样(观察员 > `ask patches [set pcolor yellow]`),观察员指令执行时必须请求(`ask`)瓦片把它的颜色 `pcolor` 设为黄色。但在刚才的指令例子中,指令直接发给了一组主体(网格集 > `set pcolor white`),此时因为指令发出主体是瓦片,所以可以直接给出指令。

3. 操纵颜色

在上面我们使用了两个不同的词来改变颜色:color 与 pcolor。

color 与 pcolor 有何区别?

在指令中心的弹出菜单中选“海龟集”,输入 set color blue,回车。车辆相应地变为蓝色。

试试把瓦片变成红色。如果想让瓦片变成红色,输入 set color red 则会出现错误信息,如图 9-20 所示。

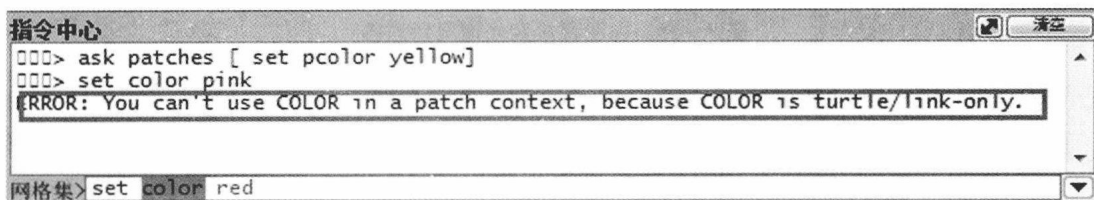


图 9-20 瓦片改变颜色过程中出现的错误信息示意图

color 和 pcolor 是变量(variables)。有些指令和变量是海龟专用的,有些是瓦片专用的。其中,color 就是一个海龟变量,而 pcolor 则是一个瓦片变量。

为了能对海龟和瓦片做更多的颜色改变,需要了解 NetLogo 如何处理颜色。在 NetLogo 中,所有颜色对应一个数值。上面的操作中,我们使用了颜色名,只是因为 NetLogo 认识 16 个不同的颜色名,但这并不意味着 NetLogo 只能分辨 16 种颜色,这些颜色之间的中间色也可以使用。图 9-21 是 NetLogo 颜色空间的色板。

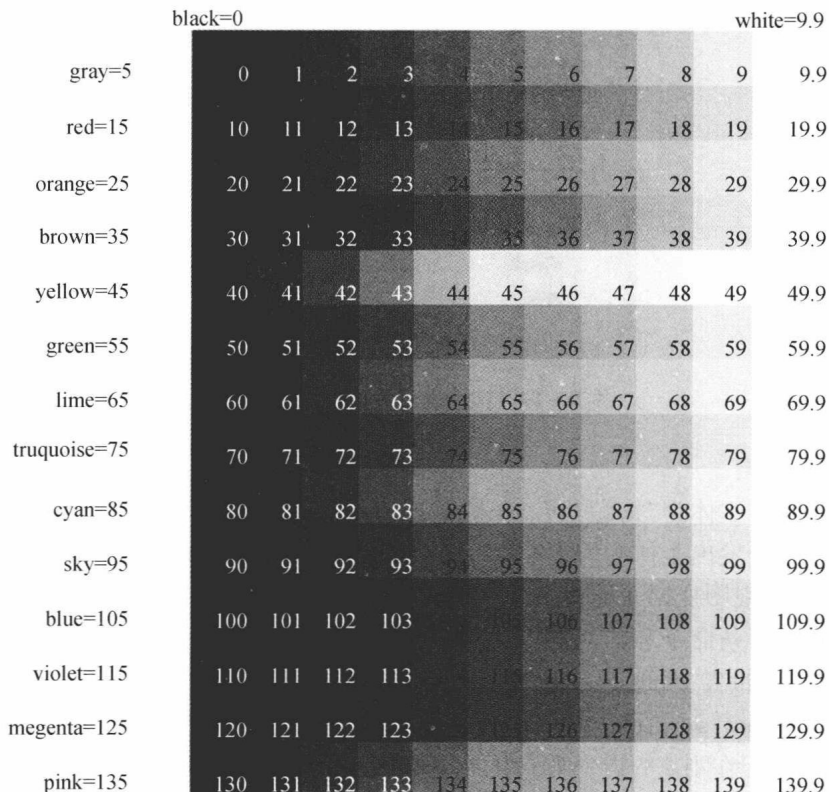


图 9-21 NetLogo 颜色空间的色板示意图

为得到一个没有名字的颜色,需要使用一个数值,或者在颜色名上“加上”或“减去”一个数值。例如,输入 `set color red` 与输入 `set color 15` 效果完全一样。要得到一个更浅或更深的颜色,只需使用一个比该颜色更小或更大的一个数值。例如,输入 `set pcolor red - 2` (注意,“-”两侧的空格很重要)得到更深的颜色;输入 `set pcolor red + 2` 得到更浅的颜色。

4. 主体监视器(Turtle Monitors)和主体命令器(Patch Monitors)

上面我们使用 `set` 命令改变所有车辆的颜色,最初的模型中一群蓝车里有一辆红车。现在看看怎样只改变一辆车的颜色。

按下“`setup`”让红车再次出现,在红车上点击鼠标右键,出现那辆红车的一个海龟监视器,如图 9-22 所示。

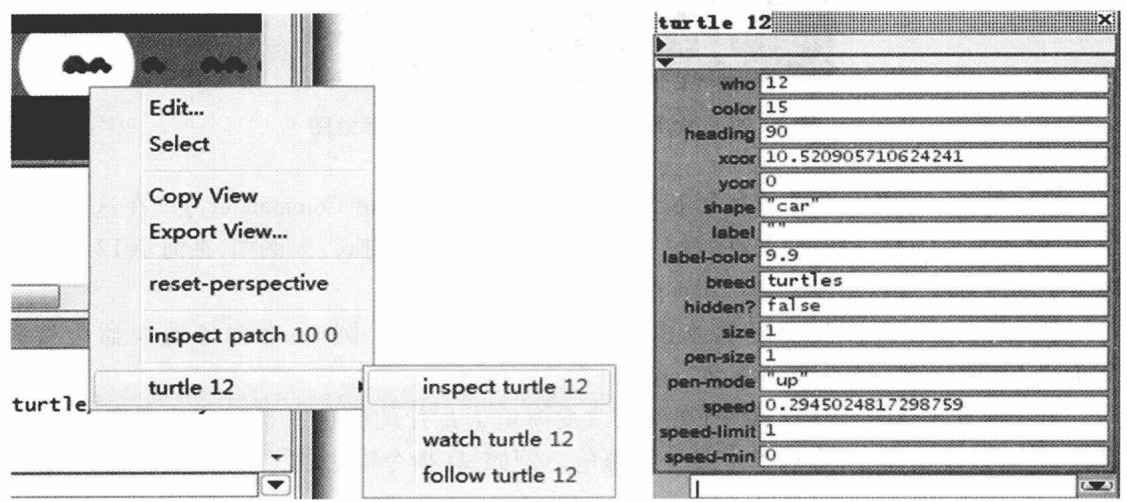


图 9-22 海龟监视器示意图

在海龟监视器中可以看到属于红车的所有变量。变量是存储数值的,可以改变。例如,每个主体都有一个 ID 号,称为“who number”。这个监视器显示该海龟的 who number 是 12,color(颜色)是 15(红色),shape(形状)是“car”。

在 NetLogo 中每个新模型包含的海龟形状可以自行设置,NetLogo 中可选择的默认形状如图 9-23 所示。

图 9-23 中形状对应值的表述如下。

- 第一行:default, airplane, arrow, box, bug, butterfly, car。
- 第二行:circle, circle 2, cow, cylinder, dot, face happy, face neutral。
- 第三行:face sad, fish, flag, flower, house, leaf, line。
- 第四行:line half, pentagon, person, plant, sheep, square, square 2。
- 第五行:star, target, tree, triangle, triangle 2, truck, turtle。
- 第六行:wheel, x。

除了点击鼠标右键外,还有两个方法可以打开海龟监视器:方法一,从工具菜单选择“海龟监视器”,然后在“who”域中输入要查看的海龟的 ID 即可;方法二,在指令中心输入“`inspect turtle 12`(或其他 ID)”。

在 NetLogo 中,改变一个海龟的颜色有三种方式。

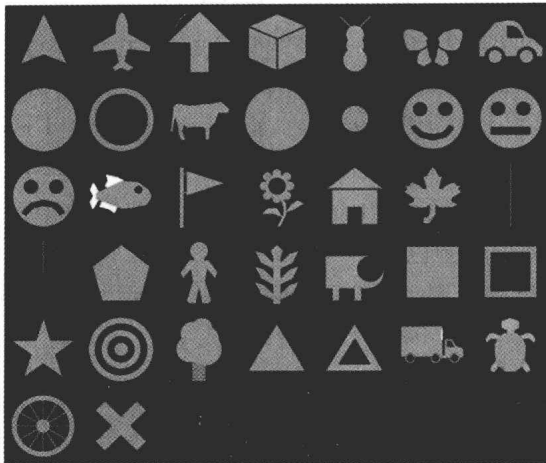


图 9-23 NetLogo 中可选择的默认形状示意图

第一种方式,使用主体监视器底部的主体指令器(Agent Commander)。在这输入指令,就像在指令中心一样,只是在这输入的指令只由这个海龟执行。例如,在海龟 12 主体监视器中的主体指令器中输入 `set color pink`。

第二种方式,直接改变海龟监视器中的“color”变量。例如,在海龟监视器中选择“color”右侧的文本,输入新颜色,如 `green + 2`。

第三种方式,在指令中心使用观察者改变海龟或瓦片的颜色。观察者俯视着 NetLogo 世界,它可以发出命令,影响单个或一组海龟。例如,在指令中心的菜单中选择“观察着”,输入 `ask turtle 12 [set color blue]`。

除了海龟监视器,也有瓦片监视器。瓦片监视器与海龟监视器很相似,如图 9-24 所示。

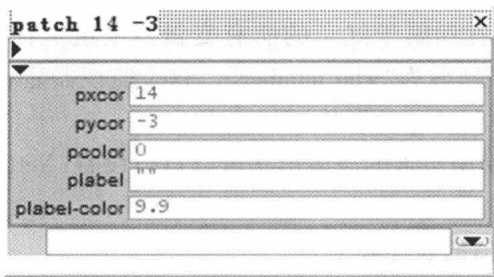


图 9-24 瓦片监视器示意图

瓦片存在于一个坐标系统中,要在图上画个点需要两个数: x 坐标和 y 坐标。瓦片的定位方式与此相同。图 9-24 中的监视器表明这个瓦片的 `pxcor` 变量是 14, `pycor` 变量是 -3,在坐标平面上,这个点处于右下象限。

使用坐标让这个特定的瓦片改变颜色的操作如下。

①在瓦片监视器的底部,输入 `set pcolor blue`,回车。

在海龟或瓦片的监视器中输入指令只对这个海龟或瓦片管用。此外在指令中心也可操作单个瓦片。

②在指令中心输入 ask patch 14 - 3 [set pcolor green],回车。

9.7.3 程序 (Procedures)

NetLogo 中可以执行命令的主体类型有:瓦片、海龟、链和观察者。瓦片是静止的,组成网格;海龟在网格上移动;链连接两个海龟。观察者俯视在进行的所有事情,做那些海龟、瓦片和链不能做的事情。这四种主体都能执行 NetLogo 指令,前三种主体还能运行程序 (Procedures)。

1. 制作 setup 按钮

要开始一个新模型,在“文件”菜单中选择“新建”。新模型一般从创建 setup 按钮开始:

- ①在界面页上部的工具条上单击“按钮”图标;
- ②在界面页的空白区域,定位到想放置按钮的地方单击;
- ③编辑按钮的对话框如图 9-25 所示,在“指令”文本域中输入 setup ;
- ④按下“确定”按钮,对话框关闭。

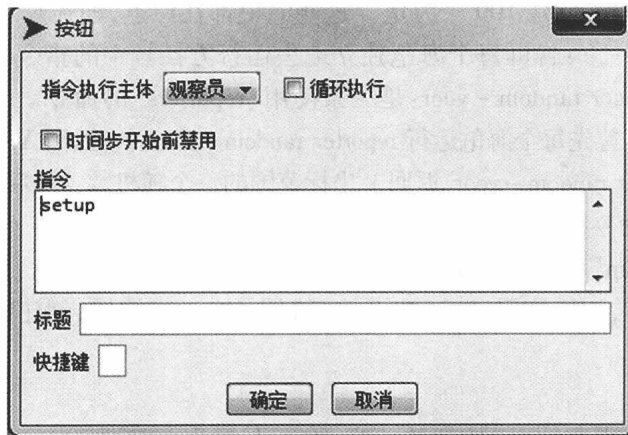


图 9-25 “setup”按钮对话框示意图

有了一个 setup 按钮后,按下按钮就执行一个名为“setup”的程序。程序就是一系列的 NetLogo 指令。此时因为还没有定义程序,所以对应的按钮呈现红色。

现在创建“setup”程序,切换到“程序”页,输入下面的代码:

```
to setup
  clear - all
  create - turtles 100 [ set shape "sheep" ]
  ask turtles [ setxy random - xcor random - ycor ]
  reset - ticks
end
```

得到的“程序”页如图 9-26 所示。

在程序中,每行缩进量不同。多数人觉得代码这样缩进很有用,但这不是强制的,只是使得代码易读易改而已。程序以 to 开始,以 end 结束。所有的程序都要用这两个词开始和结束。



图 9-26 “程序”页示意图

“setup”程序中代码的含义如下：

①setup:开始定义一个名为“setup”的程序。

②clear-all:将世界重设为初始、全空状态。

③create-turtles 100:创建 100 个海龟。这些海龟都在原点,即瓦片 0,0 的中心。

④ask turtles [...]:告诉每个海龟独立地去运行方括号中的指令。

⑤setxy random-xxcor random-yycor:是一条使用“reporters”的命令。reporter 与指令不同,它只报告一个结果。首先每个海龟运行 reporter random-xxcor,它返回 X 坐标范围内的一个随机数,然后运行 reporter random-yycor,返回 Y 坐标范围的一个随机数。最后使用前面的两个数做输入参数运行 setxy 命令,这使得海龟移动到相应的坐标处。

⑥end:结束“setup”程序的定义。

程序输入完成后,切换到界面页,按下前面制作的 setup 按钮,创建的 100 个海龟随机分散在世界内。

2. 制作 go 按钮

制作“go”按钮步骤与 setup 按钮的一样,除了下面几点差别:

①在命令部分输入 go 而非 setup。

②在编辑对话框里勾选“循环执行”,同时也选上“时间步开始前禁用”,如图 9-27 所示。

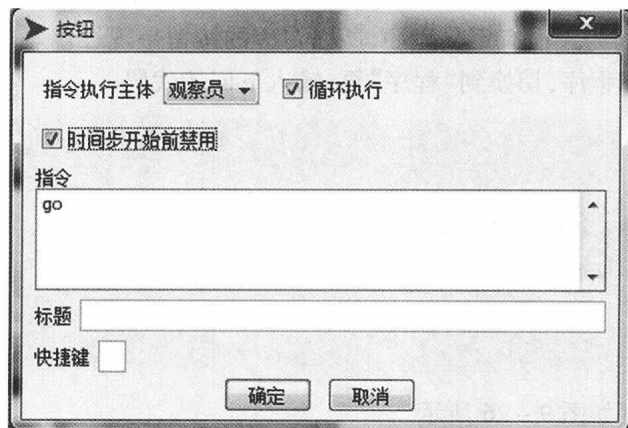


图 9-27 “go”按钮的对话框示意图

在“程序”页增加 go 程序：

```
to go
  move-turtles
  tick      ;; 是一个原语,用来使得计时器跳 1 步。
end
```

move-turtles 是一个需要在 go 程序后面添加的程序。

```
to move-turtles
  ask turtles [
    right random 360
    forward 1
  ]
end
```

注意 move-turtles 中间的连字符两侧没空格。程序 move-turtles 中的指令含义如下。

①ask turtles […]:每个海龟运行[]中的指令。

②right random 360:是使用 reporter 的指令。首先每个海龟在 0 和 359 之间随机选一个整数,然后右转这个度数。

③forward 1:让海龟前进 1 步。

为什么我们不把所有这些指令都写在 go 程序里,而是分为几个程序呢?确实可以这样做,但是在创建项目的过程中,你很可能会增加更多的东西。最好保持 go 程序尽量简单,这样更容易理解。最后还会包括许多其他的东西,比如计算、画图等。这些事情由相应的程序完成,每个程序有各自的名字。

界面页中的 go 按钮是永久性的,意味着将不断执行指令,直到关掉为止。按下 setup 按钮,创建海龟,然后按下 go 按钮,让海龟在世界中移动。注意当海龟越过世界边缘时,它要回绕(Wrap),即出现在另外一边。

3. 瓦片和变量

现在界面中有 100 个海龟,它们漫无目的地移动,对周围的事物毫无知觉。可以给海龟换一个好点的背景,让模型稍微有趣一些。相应地对 setup 程序做出如下修改:

```
to setup
  clear-all
  setup-patches
  setup-turtles
  reset-ticks
end
```

新的 setup 引用了两个新程序 setup-patches 和 setup-turtles。

其中, setup-patches 的程序为:

```
to setup-patches
  ask patches [
    set pcolor green
  ]
end
```

setup-patches 程序作用是将开始时所有瓦片颜色定义为绿色。

setup-turtles 的程序为:

```
to setup-turtles
  create - turtles 100
  ask turtles [ setxy random - xcor random - ycor ]
end
```

4. 海龟变量

目前,海龟仅仅是移动,还什么都不做。现在在海龟和瓦片之间加上一些交互:让海龟吃“草”(绿色瓦片),繁殖,死亡;草被吃掉后还会逐渐恢复。

我们需要一种控制海龟繁殖和死亡的方式,为此可以通过跟踪海龟有多少能量(Energy)来决定。因此需要为海龟增加一个新的变量。要添加新的海龟变量,需要在程序页的顶部加上 turtles-own 声明,这个声明必须在所有程序之前,变量名为 energy,具体表述为:turtles-own [energy]。使用这个新定义的变量(energy),允许海龟吃草。

重写的 go 程序如下:

```
to go
  move-turtles
  eat-grass
  tick
end
```

新加上的程序 eat-grass 为:

```
to eat-grass
  ask turtles [
    if pcolor = green [
      set pcolor black
      set energy ( energy + 10 ) ]
  ]
end
```

当每个海龟执行这些指令时,比较它所处的瓦片的颜色(pcolor)与绿色是否相同。如果瓦片是绿色则返回 true,这时才执行[]中的指令,否则跳过。这些指令让海龟将瓦片改为黑色,海龟能量值增加 10。瓦片变黑表明该处的草被吃掉,因为吃了草,海龟能量增加,

运行界面如图 9-28 所示。

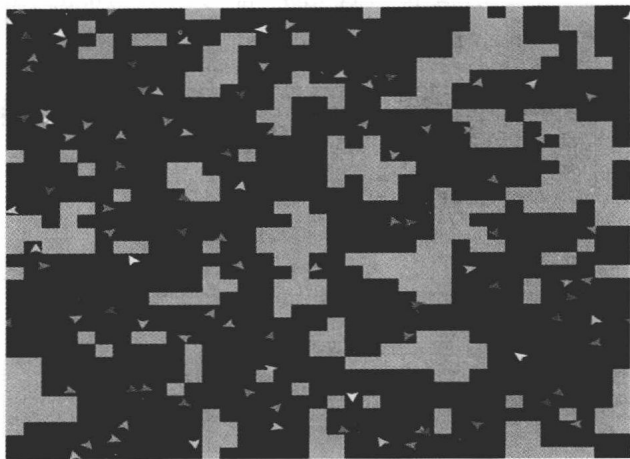


图 9-28 海龟和瓦片交互运行界面示意图

下面,让海龟移动时消耗一些能量,当海龟移动时,每步减少 1 个单位的能量。重写 move-turtles 如下:

```
to move-turtles
  ask turtles [
    right random 360
    forward 1
    set energy energy - 1
  ]
end
```

5. 监视器 (Monitors)

下面使用工具条在界面页中创建两个监视器。先做第一个监视器,使用工具条上的监视器图标,在界面页空白处创建一个监视器。出现如图 9-29 所示的对话框,在对话框的“Reporter”部分输入:count turtles,点击“确定”。

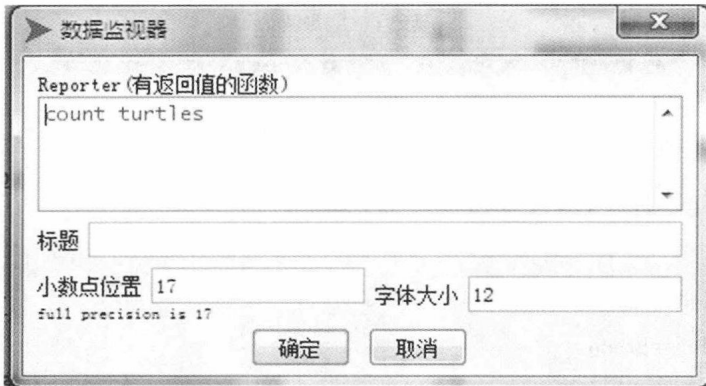


图 9-29 count turtles 监视器对话框示意图

turtles 是所有海龟的集合, count 确定海龟集合中主体数量。

制作第二个监视器, 在对话框的“Reporter”部分输入: count patches with [pcolor = green], 在“标题”部分输入: green patches, 点击“确定”, 如图 9-30 所示。

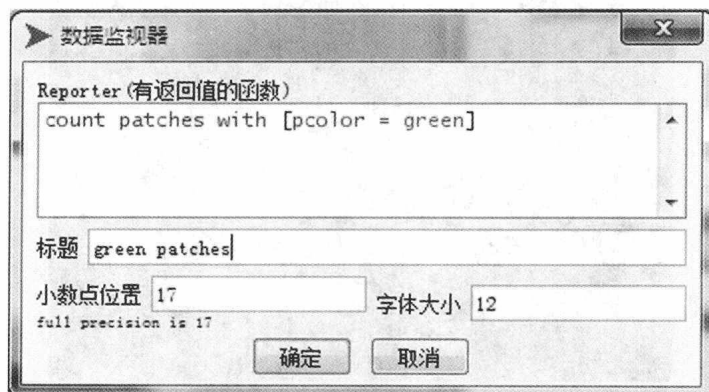


图 9-30 green patches 监视器对话框示意图

此处 patches 是所有瓦片的集合, 但我们并不想知道总共有多少瓦片, 而是想知道有多少绿色的, 这就是 with 要做的, 它创建一个较小的主体集, 只有满足 [] 中的条件的主体才会包含进来, 条件是 pcolor = green, 因此得到的是绿色瓦片的数量。

6. 开关和标签 (Switches and Labels)

如果能在任何时候都能看到所有海龟的能量, 那么将会更好地了解海龟。可以通过增加一个开关实现随时控制是否显示额外信息的功能。在界面页的工具条上选择开关图标, 在空白处单击, 创建一个开关, 出现如图 9-31 所示的对话框。在对话框的全局变量部分输入 show-energy? (别忘了包括问号)。

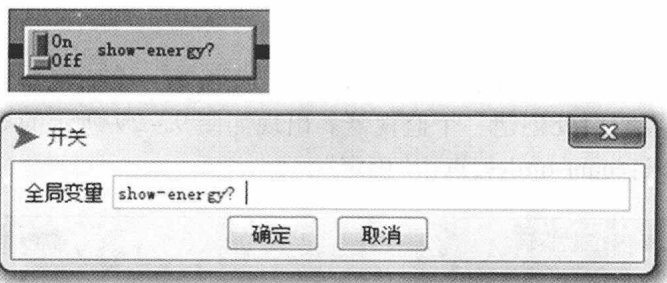


图 9-31 开关控件的对话框示意图

相应地更新“go”程序, 重写 eat-grass 程序如下:

```
to eat-grass
  ask turtles [
    if pcolor = green [
      set pcolor black
      set energy ( energy + 10 ) ]
```

```

        ifelse show-energy?
          [ set label energy ]
          [ set label " " ]
      ]
end

```

每个海龟当运行这些新命令时, show-energy? 的值(由开关决定)。如果开关打开, 比较结果为 true, 海龟执行第一个[]中的命令, 这时将能量值赋给海龟标签, 如图 9-32 所示。如果开关关闭比较结果为 false, 海龟执行第二个[]中的命令, 这时移去文本标签(通过将海龟标签设为空)。

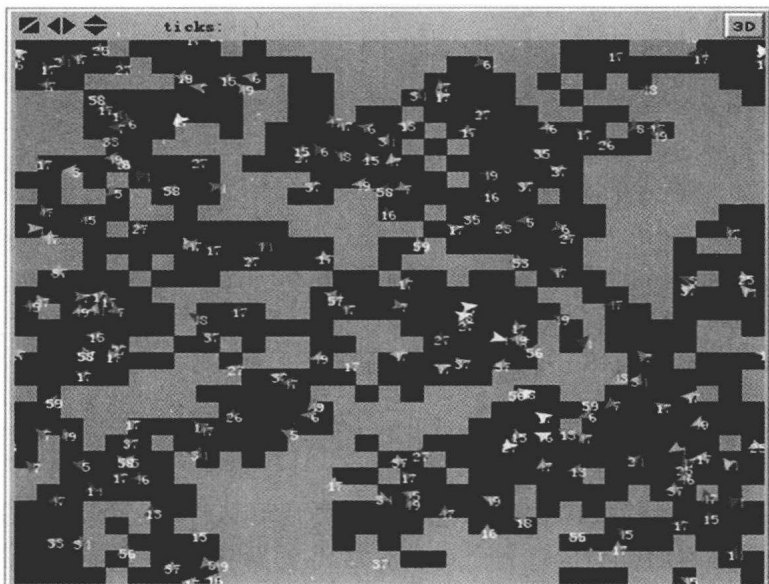


图 9-32 附带能量标签的海龟和瓦片交互运行界面示意图

在 NetLogo 中几个字符称为字符串 (String)。字符串是在双引号之间的一串字母和字符。此处两个双引号之间什么也没有,这是一个空串。如果海龟的标签是空串,就表示没有任何文本。

现在海龟在吃草,再让它们繁殖和死亡,也让草能恢复。为此需增加三个程序,分别负责这三种行为。

```
to go
  move-turtles
  eat-grass
  reproduce
  check-death
  regrow-grass
end
```

增加的程序 reproduce, check-death 和 regrow-grass 如下:

```

to reproduce
  ask turtles [
    if energy > 50 [
      set energy energy - 50
      hatch 1 [ set energy 50 ]
    ]
  ]
end

to check - death
  ask turtles [
    if energy <= 0 [ die ]
  ]
end

to regrow - grass
  ask patches [
    if random 100 < 3 [ set pcolor green ]
  ]
end

```

当繁殖时,每个海龟检查它的 energy 值,如果大于 50 执行第一个 [] 中的命令。在这里 energy 减少 50,然后孵出 (hatches) 一个 energy 为 50 的新海龟。hatch 命令是 NetLogo 中的一个原语,形如 hatch number [commands],海龟创建 number 个新海龟,每个都与母体相同,并且请求这些新海龟执行 commands,你可以使用 commands 让这些新海龟有不同的颜色、方向等。这里运行一条命令,将新海龟的 energy 设为 50。

当每个海龟运行 check-death 时,它检查 energy 是否小于等于 0。如果是真,则海龟被告知 die (这是 NetLogo 的一个原语)。

当每个海龟运行 regrow-grass 时,它检查随机产生的 0~99 之间的整数是否小于 3。如果是,瓦片颜色设为绿。对每个瓦片来说,发生的次数(平均)是 3%,因为在 100 个可能的数中,有三个数(0,1,2)小于 3。

切换到界面页,按下 setup 和 go。现在能看到模型的一些有趣行为。一些海龟死掉,一些新海龟出现(孵出),一些草恢复。如果继续看模型的监视器,会发现 count turtles 和 green patches 监视器都有振荡。

7. 画图 (Plotting)

NetLogo 可以提供画图功能,以便于更容易地跟踪模型行为。画图时需要在界面页创建一个 plot,做一些设置,然后在程序页增加一些程序用于更新绘图。

修改 setup,调用将要增加的新程序 do-plots:

```

to setup
  clear-all
  setup-patches
  setup-turtles
  do-plots
end

```

还要修改 go, 调用 do-plots:

```

to go
  move-turtles
  eat-grass
  reproduce
  check-death
  regrow-grass
  do-plots
end

```

画图用于记录海龟的数量和绿瓦片的数量。在每个时间步(go 程序的一次运行)把这些值加到图中。

```

to do-plots
  set-current-plot "Totals"
  set-current-plot-pen "turtles"
  plot count turtles
  set-current-plot-pen "grass"
  plot count patches with [ pcolor = green ]
end

```

注意使用 plot 是在图上增加新点,然而在做之前,需要告诉 NetLogo 两件事。第一,需要指明要使用哪个图(因为后面的模型有多个图);第二,要指定使用哪支笔(pen)画图。

plot 指令告诉画笔移动到新点,新点的 X 坐标是先前的 X 坐标加 1, Y 坐标就是 plot 指令中给定的值(第一种情况是海龟数量,第二种情况是绿瓦片数量)。当画笔移动时就会画出线。

为了让 set-current-plot "Totals" 工作,必须在界面页中为模型增加一个图(plot),然后进行编辑,让它的名字与程序中用到的相同。画图的过程如下。

①在界面页上使用 plot 图标创建一个 plot。

②设置图名为“Totals”,如图 9-33 所示。

③设置 X 轴标签为“time”。

④设置 Y 轴标签为“total”。

下一步创建两支笔(pen)。

①在“绘图画笔”对话框中,按下“添加画笔”按钮创建一支新笔。



图 9-33 “绘图”界面示意图

- ②输入“画笔名称”为“turtles”。
- ③在“绘图画笔”对话框中,按下“添加画笔”按钮创建第二支新笔。
- ④输入“画笔名称”为“grass”。
- ⑤选择笔的颜色,变为 green。
- ⑥选择“确定”。

注意在创建图的时候,也可设置 X,Y 轴的最大最小值,勾选“自动比例?”,如果图超过坐标轴的设定范围,坐标轴会自动增长,以便能看到所有数据。

再次按 Setup 和 go 按钮,运行模型,得到模型运行的绘图如图 9-34 所示。

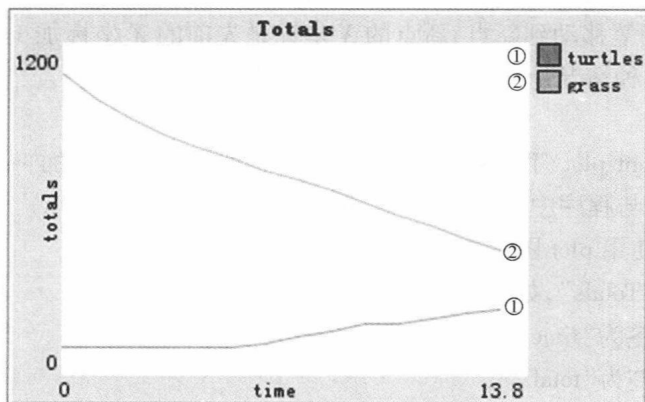


图 9-34 绘图的运行界面示意图

8. 时钟计数器(Tick Counter)

学会怎样让模型在特定的时刻停止或启动很有帮助,因为我们可以让模型在同一时刻停止。跟踪 go 程序运行了多少次是实现这一技术的关键。要跟踪这一点,需使用 NetLogo 内建的时钟计数器(Tick Counter)。

修改 go 程序:

```
to go
  if ticks >= 500 [ stop ]
  move = -turtles
  eat-grass
  reproduce
  check-death
  regrow-grass
  tick
  do-plots
end
```

按下 setup,图形和模型不会一直运行下去,当界面页工具条上的时钟计数器达到 500 时,模型自动停止。tick 指令将时钟计数器加 1,ticks 是一个报告器,返回时钟计数器当前值。每次重新运行时 clear-all 将时钟计数器清 0。

注意 tick 在 do-plots 之前。如果绘图代码使用时钟计数器就要这样,它会查看新值,而非旧值。

现在模型使用了 ticks,你可能想使用界面页顶部的菜单从“连续更新”(Continuous Updates)变为“基于时钟更新”(Tick-based Updates)。这意味着 NetLogo 仅在时钟点上更新(重画)视图(主体显示区),tick 中间不更新。这样模型运行的更快一些,并保证稳定观感(因为更新的时间间隔固定)。

9. 更多细节

首先,可以有可变数量的海龟,而非总是 100 个。为此需要作如下调整:

- ①增加一个滑动条“number”,改变最大最小值;
- ②在程序 setup-turtles 中,不使用 create-turtles 100,而是

```
to setup-turtles
  create-turtles number
  ask turtles [ setxy random-pxcor random-pycor ]
end
```

测试一下,比较初始时刻海龟更多或更少时图的变化。

其次,如果能调整海龟吃草获得的能量和移动时消耗的能量不是更好吗?

- ①增加滑动条 energy-from-grass;
- ②经增加滑动条 birth-energy;
- ③修改 eat-grass 程序为

```

to eat-grass
  ask turtles [
    if pcolor = green [
      set pcolor black
      set energy ( energy + energy-from-grass )
    ]
    ifelse show-energy?
      [ set label energy ]
      [ set label "" ]
  ]
end

```

④修改 reproduce 程序为

```

to reproduce
  ask turtles [
    if energy > birth-energy [
      set energy energy-birth-energy
      hatch 1 [ set energy birth-energy ]
    ]
  ]
end

```

9.8 实例分析:网络危机信息传播的演化仿真

9.8.1 网络危机信息仿真模型的构建

1. 模型的描述

复杂适应系统理论中研究的对象——主体,具备一定的自治性、智能性和适应性,能与环境和其他主体进行交流,并通过“学习”和“积累经验”来改变自身的结构和行为方式。在网络危机信息传播过程中,政府、媒体、网民等主体都具有这些性质,因此在复杂适应性系统理论指导下,运用多主体建模方法构建仿真模型。

本书在 SIR 病毒传播模型的基础上,结合网络传播的复杂性提出了 SRQ 网络危机信息模型,其中 S,R,Q 分别为参与危机事件的传播者(Spread)、接受危机的信息者(Receiver)、退出的关注者(Quit)。SRQ 模型主要研究网络危机事件发生后,不同时间段网民的参与人数、媒体和政府的介入程度,以及在网络危机信息传播过程中各种影响因素对各个主体行为的影响和各主体之间的相互影响情况。

2. 模型假设

由于信息传播的过程非常复杂,在建模时不能将所有因素都考虑在内,所以本书在构



建网络危机信息传播模型时提出以下几个假设:

假设 1:网络环境中参与危机信息传播的网民和媒体数量随时间变化,假设网民 t 时刻的增长数与该时刻被感染网民的能量值相等,媒体的数量与参与危机事件的网民数成正比,但是网民和媒体数量的最大规模一定,分别设为 $N_{\max}$ 和 $M_{\max}$ 。

假设 2:假设模型中只存在一类危机信息,政府、网络媒体、网民三类主体可能作为危机信息的接收者和传播者,对于网络危机信息各主体有三种可能的状态,即传播、接收和退出,网民退出后不能重新接收和传播信息。

假设 3:初始条件下信源为单传播源,初始节点可以是网民或媒体。

假设 4:由于不同网民的传播欲望随时间的变化程度是不同的,又因为无法考虑每个网民传播欲望在单位时间内的不同变化情况,因此,在模型中假设所有网民的传播欲望在单位时间内下降程度相等。

假设 5:对于不同的危机事件,政府不一定都起着主导性的作用,对于涉及个人或企业的危机事件,政府的介入度相对很小,危机事件的当事者才是处理危机的关键,因此,本书对只有政府参与事件才能平息的危机进行建模仿真。

3. 主体分类及属性描述

根据网络危机信息传播中主体之间的交互作用,设置了网民、政府、网络媒体、网络危机信息四类主体的属性。各类主体的属性如下表 9-1 所示。

表 9-1 主体属性

主体	网民	政府	网络媒体	网络危机信息
属性	①态度 ②影响力 ③可信度 ④从众性 ⑤传播欲望	①态度 ②公信力 ③信息公开速度 ④信息公开透明度 ⑤监督和引导力度	①态度 ②权威性 ③可信度 ④影响力	①重要性 ②模糊性 ③危机信息时效性价值 ④传播强度

(1) 网民主体的属性描述

本书采用函数的形式将网民主体的属性表示出来,形式如下:

$$N[A_i(t), I_i(t), Cb_i(t), Cf_i(t), D_i(t)] \quad (9-1)$$

①态度 (Attitude): 在网络危机信息传播过程中,网民有传播 (spread)、接收 (receiver)、退出 (quit) 三种态度, $A_i(t) = 1$ 表示网民 i 在 t 时刻持传播态度; $A_i(t) = 0$ 表示网民 i 在 t 时刻关注危机事件,为接收态度; $A_i(t) = -1$ 表示网民 i 在 t 时刻不再关注危机事件,为退出状态。

②影响力 (Influence): $I_i(t)$ 是指 t 时刻网民 i 传播能力的强弱,模型中按影响力的大小将网民分为小、中、大三个等级, $I_i(t)$ 取值在 $[0, 0.7]$ 之间时,表示网民 i 的影响力小; $I_i(t)$ 取值在 $[0.7, 0.9]$ 之间时,表示网民 i 的影响力中等; $I_i(t)$ 取值在 $[0.9, 1]$ 之间时,表示网民 i 的影响力大。不同类型的网民所占的比例不同,影响力小的网民所占比例最大,中的其次,大的最小,设所占比例分别为 $P(s) = 0.8$ 、 $P(m) = 0.15$ 、 $P(b) = 0.05$ 。在仿真模型中主体的初始能量大小由其影响力的等级决定,并且主体的能量随时间的变化而减小,用公式

$$Ne_i(t) = Ne_i(t_0) - c(t - t_0) \quad (9-2)$$

表示 t 时刻网民 i 的能量值,其中 $Ne_i(t_0)$ 为初始值, t_0 为网民开始参与网络危机事件的时刻, c 为单位时间内能量的递减量,它是取值可以根据实际情况调整的常数。

③可信度(Credibility): $Cb_i(t)$ 表示 t 时刻网民 i 传播的危机信息的可信度,其值在连续区间 $[0,1]$ 随机选取。 $Cb_i(t) = 0$ 表示网民完全不信任传播的信息,看到后不会去传播,退出关注的概率很大; $Cb_i(t) = 1$ 表示网民完全信任,认为信息为真,看到后有很大概率传播。

④从众性(Conformity): $Cf_i(t)$ 表示 t 时刻网民主体的从众性,其值在连续区间 $[0,1]$ 随机选取。 $Cf_i(t) = 0$ 表示网民的状态完全不会因其他网民的影响而改变, $Cf_i(t) = 1$ 表示网民没有主见,其状态完全受其他网民的影响。

⑤传播欲望(Spread Desire):因为不同网民的个性不同,其对信息的偏好程度也不同,并且网民的传播欲望会随着时间增长而下降。用 $D_i(t)$ 表示 t 时刻网民 i 的传播欲望,其初始值在连续区间 $[0,1]$ 随机选取, $D_i(t) = 0$ 表示网民对危机信息完全没有传播欲望,接收信息后不会去传播, $D_i(t) = 1$ 表示网民传播欲望最强,接收危机信息后就会去传播。用公式 $D_i(t) = d_i - \alpha(t - t_0)$ 表示网民的传播欲望随时间的变化,其中 d_i 为网民传播欲望的初始值, α 是取值可以根据实际情况调整的常数。

(2) 政府的属性描述

针对不同的危机事件,政府的介入时间和介入力度是不同的,政府所持的态度也不同。本书定义了政府的属性函数,形式如下:

$$G[T(t), G_c(t), G_v(t), G_t(t), S(t)] \quad (9-3)$$

①态度:政府在不同的时间针对不同的网络危机信息会持不同的态度,用 $T(t)$ 表示 t 时刻政府所持的态度,在模型中设政府共持有 $T(t) = 1$ (传播) 和 $T(t) = -1$ (退出) 两种态度。

②公信力:是指网民和网络媒体对政府的信任程度,用 $G_c(t) \in [0,1]$ 表示政府 t 时刻的公信力,可以根据情况给政府设置不同的初始公信力。在危机传播过程中,政府的公信力会随着危机信息的公布情况而不断变化。

③信息公开速度: $G_v(t) \in [0,1]$ 表示政府信息公开的速度,危机信息的公开速度对于扭转网络危机的局面非常重要。在模型中可以根据情况对其取值进行手动设置, $G_v(t)$ 取值越接近于 1,危机信息公开速度越快。

④信息公开透明度:随着危机事件爆发和蔓延,危机信息对社会造成的危害性也越来越大,政府会迫于舆论压力不断公开危机信息的透明度。 $G_t(t) \in [0,1]$ 表示 t 时刻政府信息公开透明度, $G_t(t) = 0$ 表示政府完全没有公开危机事件的真实情况, $G_t(t) = 1$ 时,表示政府完全公开危机事件的真实情况。政府信息透明度越高,危机信息的模糊性就会越小,网民或媒体的态度越容易改变。

⑤监督和引导力度:用 $S(t)$ 表示政府对网络媒体报道危机信息的监管和引导程度。

(3) 网络媒体主体的属性描述

由于利益背景不一样,不同媒体对危机事件会持不同的态度,并且媒体的态度决定着危机信息的传播趋势,另一方面媒体的态度会受网民和政府的影响。

本书定义了网络媒体的属性函数,形式如下:

$$M[P_n(t), U_n(t), Md_n(t), MI_n(t)] \quad (9-4)$$

①态度:在网络危机信息传播过程中,用 $P_n(t)$ 表示媒体 n 在 t 时刻的传播态度,在模型



中媒体共有 $P_n(t) = 1$ (传播) 和 $P_n(t) = -1$ (退出) 两种态度。

②可信度: $U_n(t) \in [0, 1]$ 为 t 时刻媒体 n 传播的危机信息的可信度, 不同媒体的可信度各有不同, 如果 $U_n(t) = 0$ 表示媒体 n 报道的危机信息完全不可信, 网民接收后, 传播的可能性较小; 如果 $U_n(t) = 1$ 表示媒体 n 报道的危机信息完全可信, 网民去传播的概率很大。

③权威性: $Md_n(t) \in [0, 1]$ 为 t 时刻媒体 n 的权威度, 媒体的权威度越大对网民的影响程度也就越大。

④影响力: $MI_n(t)$ 是指 t 时刻媒体 n 影响范围的大小。 $MI_n(t) \in [0, 0.4]$ 时, 表示媒体 n 的影响范围较小; $MI_n(t) \in [0.4, 0.7]$ 时, 表示媒体 n 的影响范围中等; $MI_n(t) \in [0.7, 1]$ 时, 表示媒体 n 的影响范围很大。假设媒体的影响范围越大, 其可信度和权威性也就越大。用公式 $Me_n(t) = Me_n(t_0) - d(t - t_0)$ 来表示 t 时刻媒体 n 的能量值, 其中 $Me_n(t_0)$ 为初始值, t_0 表示网民开始参与网络危机事件的时刻, d 为单位时间内能量的递减量, 它是取值可以根据实际情况调整的常数。

(4) 网络危机信息主体的属性描述

①重要性: 指危机信息对接收者的重要性, 危机信息不一定对社会造成很大的危害, 但其涉及的内容对人们容易产生影响甚至危害。危机信息对每个网民都有一定的重要性, 而且对每个人的重要性可能不同, 重要性越高网民越容易去接收和传播危机信息, 本书假设危机信息对网民、媒体、政府的重要性相同。用 $I(t) \in [0, 1]$ 表示 t 时刻危机信息的重要程度。

②模糊性: 是指信息的不确定性, 由于传播初期缺乏信息鉴别数据以及可靠的权威证据, 使得人们既不能确认它为事实也不能驳斥它的谬误。信息的不确定性越高, 对网民的影响越大, 并且信息越容易被传播。用 $\alpha(t) \in [0, 1]$ 表示危机信息在 t 时刻的模糊程度。随着政府信息公开速度和透明度的提高, 危机信息的模糊程度也随之下降。

③危机信息时效性: 由于网络信息的海量丰富性、网民关注力的非持续性等使得信息价值具有非线性和时变性, 用公式 $v(T) = a \frac{\lambda^T}{T!} e^{-\lambda T}$ 表示危机信息的时效性, λ 为目标信息的时效性特征标度因子, a 为取值可以动态调整的常数。

④传播强度: 用来表示危机信息在网络中传播速度的快慢, 信息传播强度对网民状态的变化有较大影响。根据奥尔波特提出的谣言传播强度的公式: $R = ia$, 其中 i (Importance) 指信息的重要性, a (Ambiguity) 指信息的模糊程度, 可将危机信息的传播强度用公式 $CI(t) = \beta_1 I(t) a(t) + \beta_2 v(T)$ 来表示, 其中 $T = \beta t$, β, β_1, β_2 为取值可以根据实际情况调整的常数。

4. 主体之间的交互规则构建

在仿真模型中, 网民状态的变化除受自身属性的影响外, 其他网民、网络媒体、政府以及网络危机信息也对其起着决定性的作用, 而网络媒体和政府的态度又受网民参与程度的影响。网络危机信息传播的过程中, 各主体之间相互作用和影响。

(1) 网民主体之间的交互规则

在危机信息传播过程中, 传播者会影响网民使其成为危机事件的新参与者, 而能否使新参与者去传播信息, 与危机信息传播者的可信度、网民自身的从众性、传播欲望以及危机信息的传播强度有关。

设 t 时刻传播者 i 对网民 j 的影响函数为

$$f(i, j, t) = a_1 Cb_i(t) + a_2 Cf_j(t) D_j(t) + a_3 CI(t) \quad (9-5)$$

式中, $Cb_i(t)$, $Cf_j(t)$, $D_j(t)$, $CI(t)$ 分别为 t 时刻传播者 i 的可信度、网民 j 的从众性和传播欲

望以及危机信息的传播强度。 a_1, a_2, a_3 可根据实际情况调整取值, 且 $a_1 + a_2 + a_3 = 1$ 。

①当网民 j 是受传播者 i 影响而产生的新参与者时, 可能出现两种不同的态度:

当 $f(i, j, t) \in [0, \theta]$ 时, 令 $A_j(t) = 0$, 网民 j 变为接收者;

当 $f(i, j, t) \in [\theta, 1]$ 时, 令 $A_j(t) = 1$, 网民 j 直接变为传播者, θ 为 $[0, 1]$ 区间内的常数, 其值在仿真模型中根据情况动态调整。

②当网民 j 为已经存在的接收者时, 在受到传播者 i 影响后, 可能出现三种不同的态度:

当 $f(i, j, t) \in [0, m]$ 时, $A_j(t) = -1$, 接收者 j 对传播者 i 传播的危机信息不感兴趣, 退出关注;

当 $f(i, j, t) \in [m, n]$ 时, $A_j(t) = 0$, 接收者 j 对传播者 i 传播的危机信息感兴趣, 选择持续关注;

当 $f(i, j, t) \in [n, 1]$ 时, $A_j(t) = 1$, 接收者 j 对传播者 i 传播的危机信息非常感兴趣, 选择传播信息。

其中, m 和 n 为 $[0, 1]$ 区间内的常数, 其值在仿真模型中根据情况动态调整。

③网民在不受其他网民、媒体、政府的影响情况下其态度也会随着自己传播欲望的下降以及危机信息传播强度的变化而发生转变。网民 j 自身态度转变函数为

$$f(j, t) = a_4 D_j(t) + a_5 CI(t) \quad (9-6)$$

若 $f(j, t) < \eta_1$, 则 $A_j(t) = -1$; 若 $\eta_1 \leq f(j, t) < \eta_2$, 则 $A_j(t) = 0$; 若 $f(j, t) \geq \eta_2$, 则 $A_j(t) = 1$ 。其中, a_4, a_5, η_1, η_2 为取值可以根据实际情况动态调整的常数, 且 $a_4 + a_5 = 1$ 。

(2) 网民与媒体之间的交互规则

由假设 1 可知参与危机事件的媒体数量 $M(t)$ 与网民数 $N(t)$ 成正比, 设 $M(t) = bN(t)$, b 为常数, 在仿真过程中动态调整。

网民的态度受媒体可信度和权威性的影响, 媒体 n 对网民 i 的影响函数为

$$f(n, i, t) = b_1 U_n(t) M d_n(t) + b_2 C f_i(t) D_i(t) \quad (9-7)$$

式中, $U_n(t)$, $M d_n(t)$, $C f_i(t)$, $D_i(t)$ 分别为 t 时刻媒体 n 的可信度、权威性以及网民 i 的从众性和传播欲望, b_1 和 b_2 为常数, 取值可以根据实际情况调整, 且 $b_1 + b_2 = 1$ 。

①当网民 j 是受媒体 n 影响而产生新参与者时, 可能出现两种不同的态度:

当 $f(n, i, t) \in [0, \theta']$ 时, 令 $A_i(t) = 0$, 网民 i 变为接收者;

当 $f(n, i, t) \in [\theta', 1]$ 时, 令 $A_i(t) = 1$, 网民 i 直接变为传播者。其中 θ' 为 $[0, 1]$ 区间内的常数, 其值在仿真模型中根据情况动态调整。

②当网民 i 为已经存在的接收者时, 在受到媒体 n 影响后, 可能出现三种不同的态度:

若 $f(n, i, t) \in [0, u]$, 则 $A_i(t) = -1$, 网民 i 对媒体 n 传播的危机信息不感兴趣, 退出关注;

若 $f(n, i, t) \in [u, v]$, 则 $A_i(t) = 0$, 网民 i 对媒体 n 传播的危机信息感兴趣, 选择持续关注;

若 $f(n, i, t) \in [v, 1]$, 则 $A_i(t) = 1$, 网民 i 对媒体 n 传播的危机信息非常感兴趣, 选择传播信息。

其中, u 和 v 为 $[0, 1]$ 区间内的常数, 其值在仿真模型中根据情况动态调整。

媒体 t 时刻的态度受网民中传播者所占比例和事件的传播强度的影响, 网民对媒体的影响函数为



$$f(n, t) = b_3 \frac{\mu N(t)}{N_{\max}} + b_4 CI(t) \quad (9-8)$$

式中, $N(t)$ 和 $N_{\max}$ 分别为 t 时刻参与危机事件的网民数和网民人数的最大值, μ, b_3, b_4 为常数, 取值可以根据实际情况调整, 且 $b_3 + b_4 = 1$ 。

若 $f(n, t) \in [0, u']$, 则令 $P_n(t) = -1$;

若 $f(n, t) \in [u', 1]$, 则令 $P_n(t) = 1$ 。

其中, u' 为 $[0, 1]$ 区间内的常数, 其值在仿真模型中根据情况动态调整。

(3) 网民与政府之间的交互规则

政府的介入, 对于网络危机事件的平息至关重要。政府公布信息的速度和透明度能够有效降低危机信息的模糊程度, 在一定程度上能够减少网民对危机信息的恐惧感。政府公信力越强, 网民就越信任政府, 谣言的产生和传播就会有效地被遏制。

政府对网民的影响函数为

$$f(G, i, t) = c_1 Cf_i(t) D_i(t) + c_2 (1 - G_c(t)) + c_3 (1 - G_v(t) G_t(t)) \quad (9-9)$$

式中, $Cf_i(t), D_i(t), G_c(t), G_v(t), G_t(t)$ 分别为 t 时刻网民 i 的从众性、传播欲望、政府公信力、政府公开信息的速度和透明度, c_1, c_2, c_3 为可以根据实际情况取值的常数。

若 $f(G, i, t) \in [0, x]$, 则 $A_i(t) = -1$; 若 $f(G, i, t) \in [x, y]$, 则 $A_i(t) = 0$; 若 $f(G, i, t) \in [y, 1]$, 则 $A_i(t) = 1$ 。其中 $x, y \in [0, 1]$, 可以根据实际情况调整。

网民对危机事件的关注度以及危机信息的传播强度决定了政府是否有必要介入, 网民对政府的影响函数为

$$f(G, t) = c_4 \frac{\mu N(t)}{N_{\max}} + c_5 CI(t) \quad (9-10)$$

式中, $N(t)$ 和 $N_{\max}$ 分别为 t 时刻参与危机事件的网民数和网民人数的最大值, c_4 和 c_5 为取值可以根据实际情况调整的常数。

若 $f(G, t) \in [0, x']$, 则政府的态度 $T(t) = -1$; 若 $f(G, t) \in [x', 1]$, 则政府的态度 $T(t) = 1$ 。其中, $x' \in [0, 1]$ 。

9.8.2 仿真的实现与仿真结果

1. 网民之间的交互仿真

(1) 仿真算法流程图

在网民之间的交互仿真模型中主要包括三个仿真算法, 在仿真过程的推进中三个算法是循环、并行执行的。三个算法主要包含在传播者与新参与网民、传播者与接收者以及网民自身状态变化三个过程中。具体仿真算法流程图如图 9-35 所示。

网民之间的 NetLogo 仿真界面如图 9-36 所示。

(2) 参数设置

将网民按影响力的大小分为 $[1, 10), [10, 40), [40, 50]$ 三个区间, 按照所在区间的不同, 在仿真模型中用“person”“person student”“person doctor”三种不同的形状表示, 形状大小分别为 0.5, 0.8, 1.5, 将传播源的初始影响力设置为 9。

网络危机信息的初始时效性、重要性和初始模糊程度分别设为 0.9, 0.92, 0.92; 在危机信息传播强度的计算中, 信息的重要性模糊程度与信息的信息时效性所占权重分别为 0.6 和 0.4; 模糊程度和信息时效性的递减参数分别为 0.04 和 0.06。

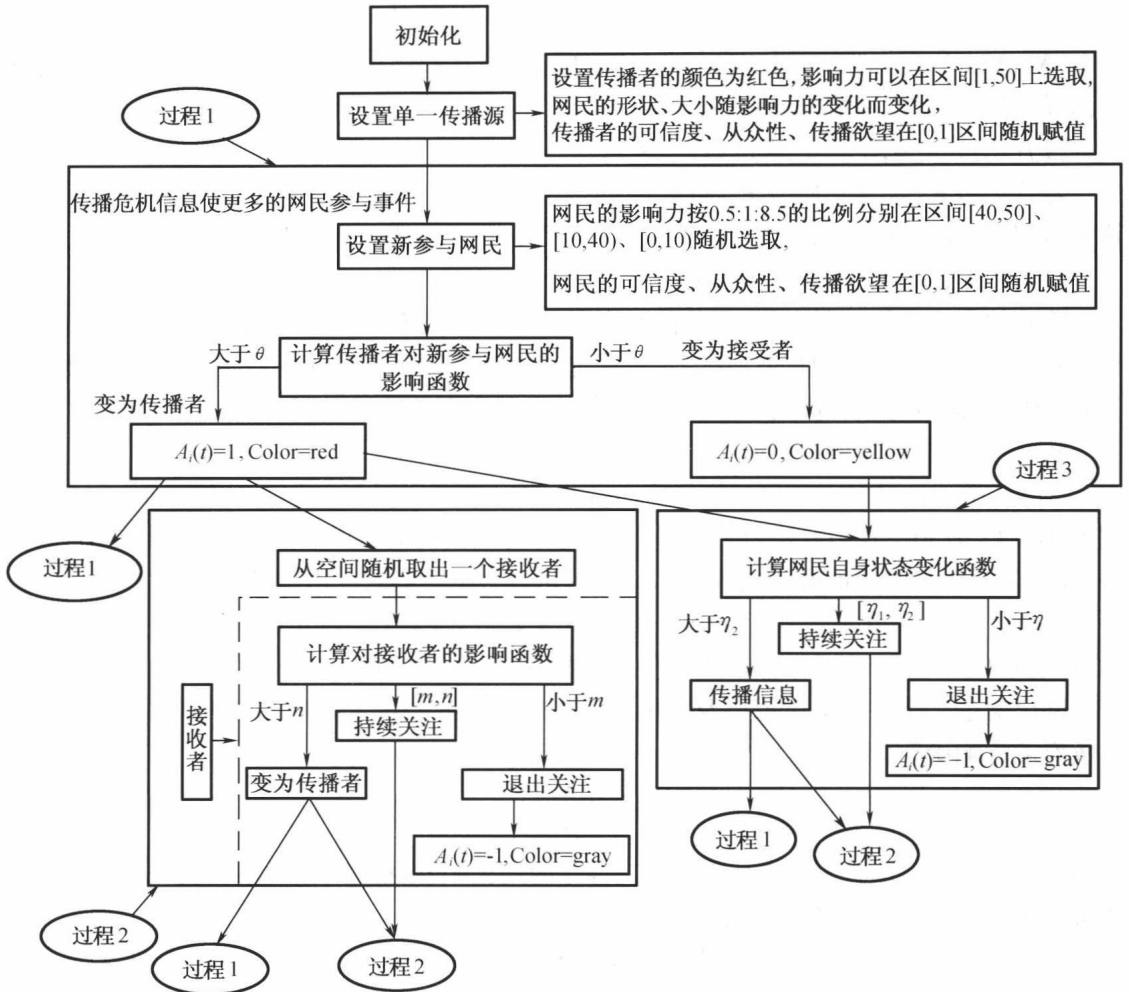


图 9-35 网民之间的交互仿真算法流程图

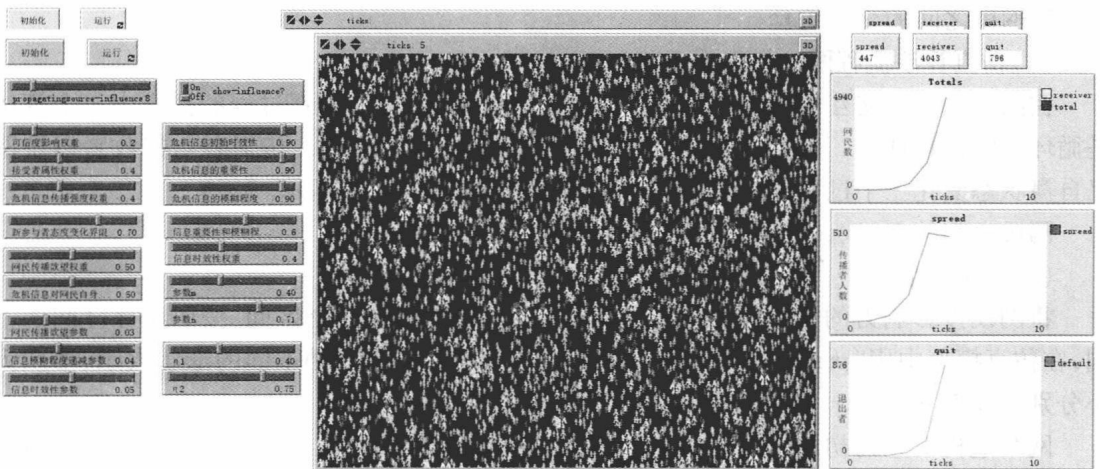


图 9-36 网民之间的 NetLogo 仿真界面示意图



在如图 9-35 的过程 1 和过程 2 中,设置传播者的可信度影响权重为 0.2,接收者属性权重为 0.4,危机信息传播强度影响权重为 0.4;在过程 3 中设置网民传播欲望权重为 0.5,危机信息对网民自身影响权重为 0.5。其余的常数 $\theta, m, n, \eta_1, \eta_2$ 分别设为 0.7, 0.4, 0.7, 0.45, 0.75。

(3) 仿真代码

网民之间的交互仿真程序如下:

```
globals
[ a b Ia d a1 a2 a3 a4 a5 It a0 v0 Iv  $\beta_1$   $\beta_2$   $\theta$  m n ]
turtles-own
[ spread?
quit?
Influence ;网民的影响力
credibility ;网民可信度
conformity ;网民的从众性
spread-desire ;网民的传播欲望
p-time ] ;参与时间
to setup
clear-all
reset-ticks
setup-constant
setup-propagatingsource ;初始化传播源
display-labels
end

to setup-constant
set a1 可信度影响权重 set a2 接收者属性权重 set a3 危机信息传播强度权重
set It 危机信息的重要性 set a0 危机信息的模糊程度 set v0 危机信息初始时效性
set  $\beta_1$  信息重要性和模糊程度权重 set  $\beta_2$  信息时效性权重 set  $\theta$  新参与者态度变化界限
set m 参数 m set n 参数 n set a4 网民传播欲望权重 set a5 危机信息对网民自身影响权重
set Ia 信息模糊程度递减参数 set Iv 信息时效性参数 set d 网民传播欲望参数
end

to setup-propagatingsource
set-default-shape turtles "person"
create-turtles 1 [
set p-time 0
set influence propagatingsource-influence
setxy (random-xxcor * 0.9) (random-yycor * 0.9)
```

```

set credibility (random 10)/10
set conformity (random 10)/10
set spread-desire (random 10)/10
get-spread
]
ask turtles[
  ifelse propagatingsource-influence < 10 ;根据影响力的不同将传播源设置为不同的形状
  [ set shape "person" set size 0.5 ]
  [ ifelse propagatingsource-influence > 20
    [ set shape "person doctor" set size 1.5 ]
    [ set shape "person student" set size 0.8 ]
  ]
]
end

to get-spread ;定义传播者
  set spread? true
  set quit? false
  set color red
end

to get-receiver ;定义接收者
  set spread? false
  set quit? false
  set color yellow
end

to become-quit ;定义退出者
  set spread? false
  set quit? true
  set color gray
end

to go
  if ticks > 20 [stop]
  ask turtles [ set p-time p-time + 1 ]
  move
  if count turtles < 20000 [spread]
  interactive ;传播者对其他参与者的影响
  netizen-oneself-change ;网民自身的状态变化
  display-labels ;设置网民影响力是否显示

```



```

mutation                                ;突变,即网民可能因为一些特殊的原因而去传播危机信息
tick
end

to spread
  ask turtles with [ spread? ]
    [ hatch influence [
      set conformity (random 10)/10
      set spread-desire (random 10)/10
      set p-time 0
      setxy random-xcor random-ycor
    ]
    ifelse random 100 < 75
      [ set influence random 10 ]
      [ ifelse random 100 > 95 [ let h1 random 10 set influence (20 + h1) set spread-desire 1 ]
        [ let h2 random 10 set influence (5 + h2) set spread-desire 0.8 ] ]
    ifelse influence < 10
      [ set shape "person" set size 0.5 ]
      [ ifelse influence > 20
        [ set shape "person doctor" set size 1.5 ]
        [ set shape "person student" set size 0.8 ] ] ;按影响力不同将网民设置不同形状
    ifelse (a1 * credibility + a2 * conformity * spread-desire + a3 * ( $\beta_1 * It * a_0 + \beta_2 * v_0$ )) < 0
      [ get-receiver set credibility (random 10)/10 ]
      [ set credibility (random 10)/10 ]
    ]
    if influence > 0 [ set influence influence - 1 ] ;网民的影响力随时间的变化而减少
  ]
end

to interactive
  ask turtles with [ spread? ] [ set a credibility ] ;将传播者的可信度赋值给 a
  ask turtles with [ spread? ]
    [ ask one-of turtles with [ not quit? ] ;传播者随机选取一个参与者进行交互
    [ ifelse ( a1 * a + a2 * conformity * (spread-desire - d * p-time) + a3 * ( $\beta_1 * It * (a_0 - Ia * ticks) + \beta_2 * (v_0 - Iv * ticks)$ )) < m
      [ become-quit ] ;如果影响函数值小于 m 则被影响的参与者变为退出者
      [ ifelse ( a1 * a + a2 * conformity * (spread-desire - d * p-time) + a3 * ( $\beta_1 * It * (a_0 - Ia * ticks) + \beta_2 * (v_0 - Iv * ticks)$ )) > n
        [ get-spread ] ;如果影响函数值大于 n 则被影响的参与者变为传播者
        [ get-receiver ] ;如果影响函数值介于 [ m, n ] 之间则变为接收者
      ]
    ]
  ]
end

```

```

    ]
end

to netizen-oneself-change
  ask turtles [
    ifelse (a4 * (spread-desire - d * p-time) + a5 * ( $\beta_1$  * It * (a0 - Ia * ticks) +  $\beta_2$  * (v0 - Iv * ticks))) <  $\eta_1$ 
      [ become-quit ]
      [ ifelse (a4 * (spread-desire - d * p-time) + a5 * ( $\beta_1$  * It * (a0 - Ia * ticks) +  $\beta_2$  * (v0 - Iv * ticks))) >  $\eta_2$ 
        [ get-spread ]
        [ get-receiver ]
      ]
  ]
end

to mutation
  ask turtles with [ color = yellow ] ;接收者有 0.01 的概率变为传播者
  [ if random 100 < 1
    [ get-spread ]
  ]
end

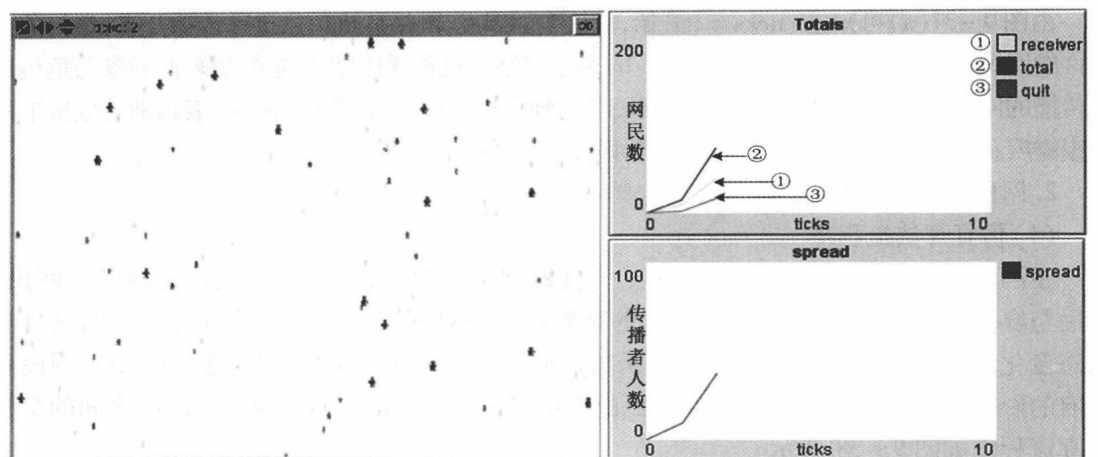
to display-labels
  ask turtles [
    ifelse show-influence?
      [ set label round influence ]
      [ set label " " ]
  ]
end

to mov
  ask turtlese
  [ rt random 100
    lt random 100
    fd 1 ]
end

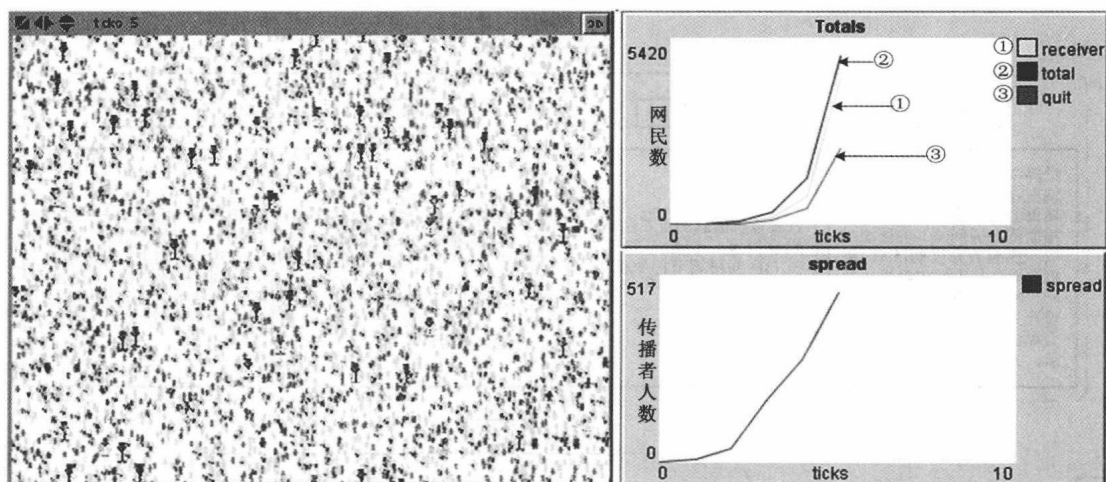
```

(4) 仿真结果

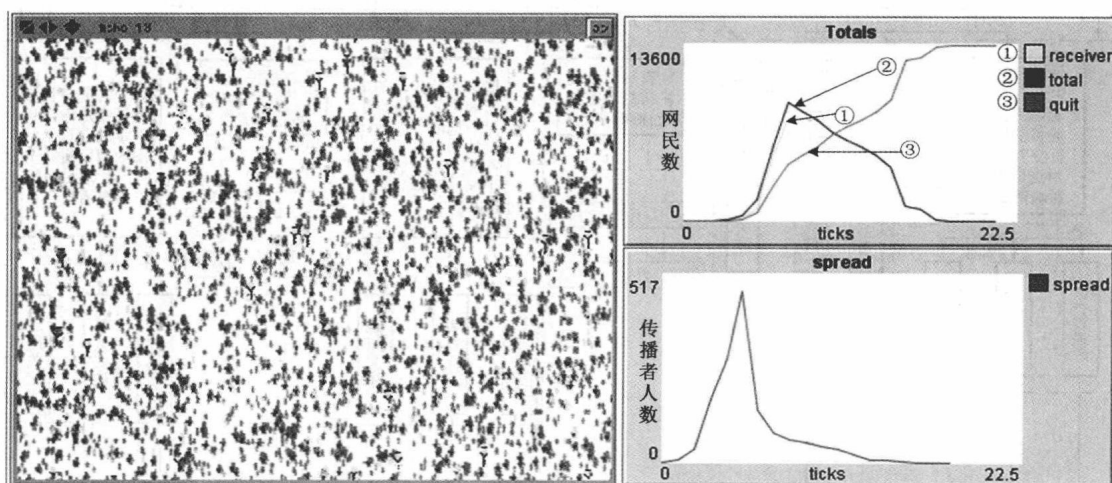
网络危机在经过短暂的潜伏期后突然爆发,图 9-37 为危机爆发后各传播阶段中持各种态度的网民数量变化仿真结果演示图,左边为仿真世界的图形化显示窗口,右边为对应时刻用绘画功能输出的各种状态下的网民数量变化曲线,从上到下分别为接收者、全体参与者(包括传播者和接收者)、退出者和传播者数量随时间演化的曲线图。



(a)



(b)



(c)

图 9-37 网络危机信息传播爆发、缓解阶段仿真结果图

(a) 网络危机爆发初期; (b) 网络危机信息传播高峰期; (c) 网络危机缓解期

由图9-37(a)可知,从 $\text{ticks} = 2$ 开始,危机全面爆发,网络危机信息迅速传播开来,参与危机事件的网民数呈几何倍数增长;图9-37(b)表示网络危机传播速度达到最高峰,此时参与危机信息传播的网民数最多,使得越来越多的网民参与到危机事件中来;图9-37(c)表示随着危机事件的模糊度及时效性的减小,网民逐渐退出对危机事件的关注。

2. 网民与媒体、政府之间的交互仿真

(1) 仿真算法流程图

在网民、媒体、政府三个主体的交互仿真模型中主要包括三个交互过程:网民与网民、网民与媒体以及网民与政府,主要涉及的函数有网民对网民的影响函数 $f(i, j, t)$ 、网民自身状态变化函数 $f(j, t)$ 、网民对媒体影响函数 $f(n, t)$ 、媒体对网民的影响函数 $f(n, i, j)$ 、网民对政府的影响函数 $f(G, t)$ 以及政府对网民的影响函数 $f(G, i, j)$ 。网民、媒体、政府之间的交互仿真流程图如图9-38所示。

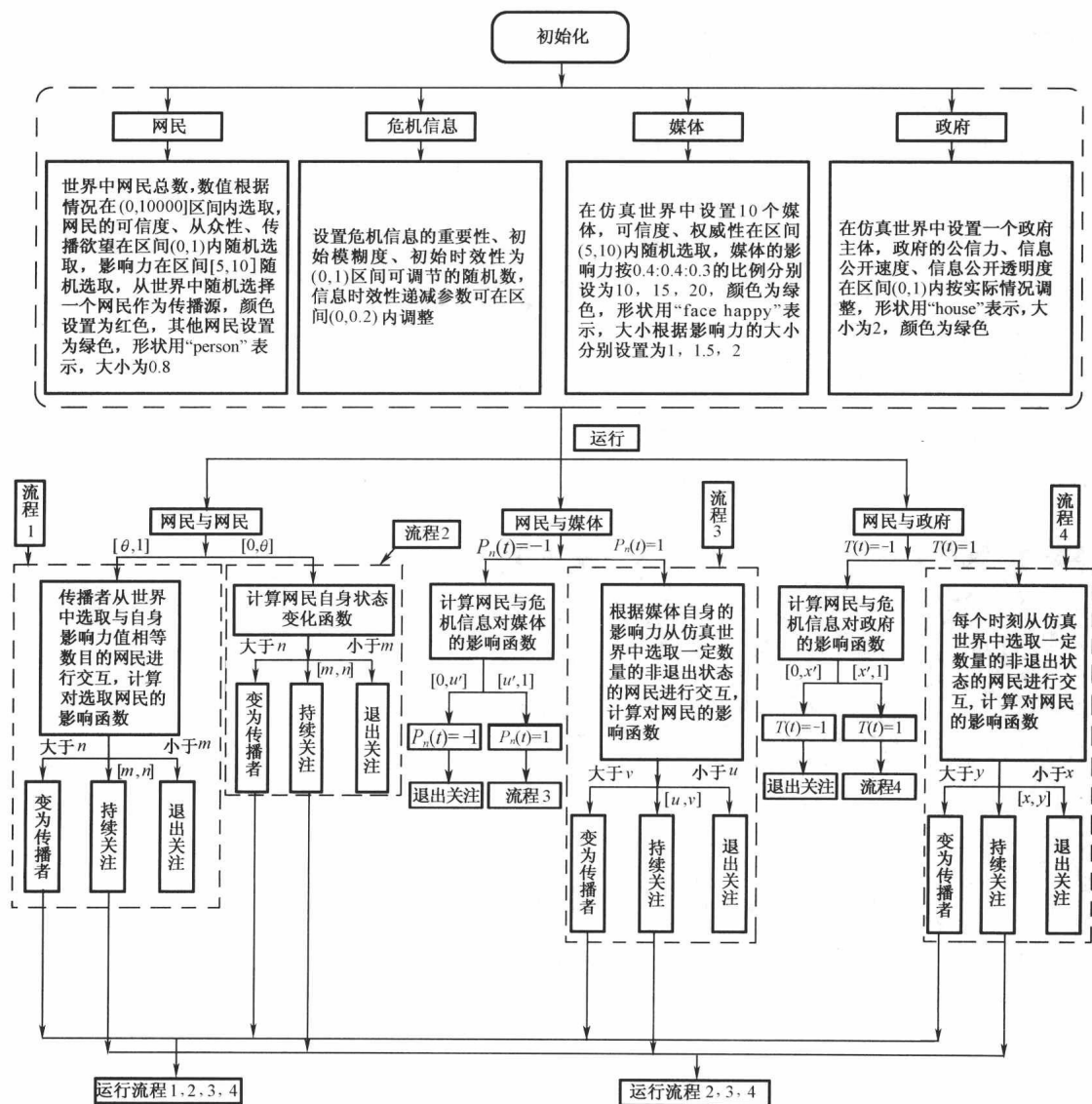


图9-38 网民、媒体、政府之间的交互仿真流程图

网民与媒体、政府之间的 NetLogo 仿真界面如图 9-39 所示。

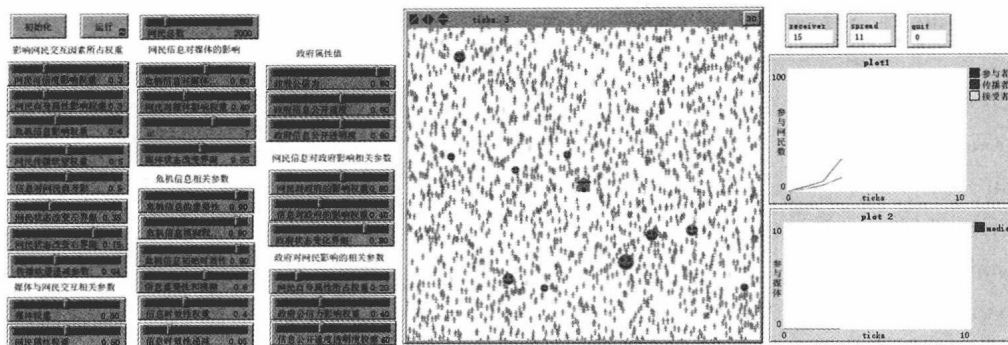


图 9-39 网民与媒体、政府之间的 NetLogo 仿真界面

(2) 参数设置

在网民与网民交互过程中,设置公式(9-5)中的网民可信度影响权重为 0.3、网民自身属性影响函数为 0.3、危机信息影响权重为 0.4,设置公式(9-6)中网民传播欲望影响权重为 0.5、信息对网民自身影响权重为 0.5。

在网民与危机信息的交互过程中,设置公式(9-7)中的网民对媒体影响权重为 0.4、危机信息对媒体影响权重为 0.6,在公式(9-8)中设置媒体影响权重为 0.5,网民自身属性权重为 0.5。

在网民与政府的交互过程中,设置公式(9-9)中的网民对政府的影响权重为 0.6、危机信息对政府的影响权重为 0.4,设置公式(9-10)中网民自身属性权重为 0.2、政府公信力影响权重为 0.4,信息公开速度和透明度权重为 0.6。

设置危机信息的重要性、初始时效性、初始模糊度都为 0.9,政府的公信力、信息公开透明度、信息公开速度分别为 0.5,0.85,0.85,设置图 9-39 中的主体态度变化界限 m, n, u, v, x, y 分别为 0.35,0.75,0.35,0.75,0.35,0.75,媒体状态改变界限 u' 为 0.55,政府状态改变界限 x' 为 0.8。网民传播欲望递减参数、信息时效性递减参数分别为 0.04,0.05,信息模糊度递减参数与政府的信息公开速度和信息公开透明度成正比。

(3) 仿真代码

网民与媒体、政府之间的交互仿真程序如下:

```
breed [ netizens netizen ]
breed [ media medium ]
breed [ governments government ]
globals
[ a b Ia d a1 a2 a3 a4 a5 It a0 v0 Iv β1 β2 m n b1 b2 b3 b4 u1 mc mfi c1 c2 c3 c4 c5 x1 Gv Gc Gt ]
[ x y u v ]

netizens-own
[ spread?
quit?
n-influence      ;影响力
credibility      ;可信度
```

```

conformity      ;从众性
spread-desire   ;传播欲望

```

```
]
```

```
media-own
```

```
[ spread?
```

```
quit?
```

```
m-credibility  ;可信度
```

```
m-influence    ;影响力
```

```
finality       ;权威性
```

```
]
```

```
governments-own
```

```
[ spread?
```

```
quit?]
```

```
to setup
```

```
clear-all
```

```
ask patches [ set pcolor 9.9 ]
```

```
reset-ticks
```

```
setup-constant
```

```
setup-netizens
```

```
setup-midia
```

```
setup-governments
```

```
end
```

```
to setup-constant
```

```
set a1 网民可信度影响权重
```

```
set a2 网民自身属性影响权重
```

```
set a3 危机信息影响权重
```

```
set It 危机信息的重要性
```

```
set a0 危机信息模糊程度初始值
```

```
set v0 危机信息初始时效性
```

```
set  $\beta$ 1 信息重要性和模糊程度权重
```

```
set  $\beta$ 2 信息时效性权重
```

```
set Iv 信息时效性递减参数
```

```
set m 网民状态改变左界限
```

```
set n 网民状态改变右界限
```

```
set a4 网民传播欲望权重
```

```
set a5 信息对网民自身影响权重
```

```
set d 传播欲望递减参数
```

```
set b1 网民对媒体影响权重
```



```

set b2 危机信息对媒体影响权重
set u1 媒体状态改变界限
set u 媒体影响网民左界限
set v 媒体影响网民右界限
set b3 媒体权重
set b4 网民属性权重
set c1 网民对政府的影响权重
set c2 信息对政府的影响权重
set x1 政府状态变化界限
set c3 网民自身属性所占权重
set c4 政府公信力影响权重
set c5 信息公开速度透明度权重
set x 政府影响网民左界限
set y 政府影响网民右界限
set Gv 政府信息公开速度
set Gt 政府信息公开透明度
set Gc 政府公信力
end

to setup-netizens
  set-default-shape netizens "person"
  create-netizens 网民总数 [
    set color green
    set size 0.8
    set n-influence random 7 + 1
    set credibility (random 10)/10 + 0.1
    set conformity (random 10)/10 + 0.1
    set spread-desire (random 10)/10 + 0.1
    setxy random-xcor random-ycor
  ]
  ask one-of netizens [ get-spread ] ;从网民中任选一个作为传播源
end

to setup-media
  set-default-shape media "face happy"
  create-media 10 [
    set color green
    set m-credibility (random 6 + 5)/10
    set finality (random 6 + 5)/10
    setxy (random-xcor * 0.90) (random-ycor * 0.90)
  ]

```

```

    ifelse random 100 < 40          ;根据影响力设置媒体的形状大小
    [ set m-influence 10 set size 1 ]
    [ ifelse random 100 > 70
      [ set m-influence 20 set size 2 ]
      [ set m-influence 15 set size 1.5 ]
    ]
  ]
end

to setup-governments
  set-default-shape governments "house"
  create-governments 1
  [ set color green
    set size 2.0
  ]
end

to get-spread          ;定义传播者
  set spread? true
  set quit? false
  set color red
end

to get-receiver        ;定义接收者
  set spread? false
  set quit? false
  set color yellow
end

to become-quit         ;定义退出者
  set spread? false
  ;set spread-count 0
  set quit? true
  set color gray
end

to go
  if ticks > 20 [ stop ]
  go-netizens           ;网民交互
  netizens-influence-media ;网民对媒体的影响

```



```

media-influence-netizens           ;媒体对网民的影响
netizens-influence-governments     ;网民对政府的影响
governments-influence-netizens     ;政府对网民的影响
update-global-Ia                   ;更新危机信息的模糊程度
move
tick
end
to go-netizens
ask netizens with [ color = red or color = yellow ] [
  if spread-desire > d [set spread-desire spread-desire - d] ;参与网民的传播欲望不断下降
]
ask netizens with [ spread? = true ] [
  if n-influence > 0 [set n-influence n-influence - 1] ;传播者的影响力随时间逐渐减少
  set a credibility
]
ask netizens with [ spread? = true ]
  [ if count netizens with [ color ! = gray ] > 40
    [ ask n-of n-influence netizens with [ color ! = gray ]
      [ ifelse ( a1 * a + a2 * conformity * spread-desire + a3 * (  $\beta 1 * I_t * ( a_0 - 0.3 * I_a * ticks ) + \beta 2 * ( v_0 - I_v * ticks ) ) ) < m$ 
        [ become-quit ]
        [ ifelse ( a1 * a + a2 * conformity * spread-desire + a3 * (  $\beta 1 * I_t * ( a_0 - 0.3 * I_a * ticks ) + \beta 2 * ( v_0 - I_v * ticks ) ) ) > n$ 
          [ get-spread ]
          [ get-receiver ]
        ]
      ]
    ]
  ]
  if n-influence = 0 [ become-quit ]
]
ask netizens with [ color = red or color = yellow ]
  [ ifelse ( a4 * spread-desire + a5 * (  $\beta 1 * I_t * ( a_0 - 0.3 * I_a * ticks ) + \beta 2 * ( v_0 - I_v * ticks ) ) ) < m$ 
    [ become-quit ]
    [ ifelse ( a4 * spread-desire + a5 * (  $\beta 1 * I_t * ( a_0 - 0.3 * I_a * ticks ) + \beta 2 * ( v_0 - I_v * ticks ) ) ) > n$ 
      [ get-spread ]
      [ get-receiver ]
    ]
  ]
]

```

```

]
end

to netizens-influence-media
  ask media with [ color ! = gray ]
    [ if (  $b3 * \mu * (\text{count netizens with [ color = red or color = yellow ]} / \text{网民总数} + b4 * (\beta1 * It * (a0 - 0.3 * Ia * \text{ticks}) + \beta2 * (v0 - Iv * \text{ticks}))) > u$ 
      [ if random 100 < 50 [ get-spread ] ]
    ]
end

to media-influence-netizens
  ask media with [ spread? = true ] [
    if m-influence > 0 [ set m-influence m-influence - 1 ]
    if m-influence = 0 [ become-quit ]
    set mc m-credibility
    set mfi finality
  ]
  ask media with [ spread? = true ]
    [ ifelse count netizens with [ color ! = gray ] > 30
      [ ask n-of m-influence netizens with [ color ! = gray ]
          [ ifelse (  $b1 * mc * mfi + b2 * \text{conformity} * \text{spread-desire}$  ) < u
              [ become-quit ]
              [ ifelse (  $b1 * mc * mfi + b2 * \text{conformity} * \text{spread-desire}$  ) > v
                  [ get-spread ]
                  [ get-receiver ]
                ]
            ]
          ]
      ]
    [ become-quit ] ;当网民参与人数小于 30 时,媒体退出关注
  ]
end

to netizens-influence-governments
  ask governments
    [ if (  $c4 * \mu * (\text{count netizens with [ color = red or color = yellow ]} / \text{网民总数} + c5 * (\beta1 * It * (a0 - 0.3 * Ia * \text{ticks}) + \beta2 * (v0 - Iv * \text{ticks}))) > x$ 
      [ get-spread ] ;网民对政府的影响函数值大于 x 时,政府开始介入危机事件
    ]

```

```

end

to governments-influence-netizens
  ask governments with [ spread? = true ] [
    ifelse count netizens with [ color = red or color = yellow ] > 60
    [ ask n-of 60 netizens with [ color != gray ]
      [ ifelse( c1 * conformity * spread-desire + c2 * (1 - Gc) + c3 * (1 - Gv * Gt) ) < x
        [ become-quit ]
        [ ifelse( c1 * conformity * spread-desire + c2 * (1 - Gc) + c3 * (1 - Gv * Gt) ) > y
          [ get-spread ]
          [ get-receiver ]
        ]
      ]
    ]
  ]
  [ become-quit ]
]

end

to update-global-Ia
  ask governments [ if spread? = true
    [ set Ia 0.5 * Gc + 0.5 * Gv * Gc ] ]
end

to move
  ask netizens
  [ rt random 360
    lt random 360
    fd 1 ]
  ask media
  [ rt random 360
    lt random 360
    fd 1 ]
end

```

(4) 仿真结果

如图 9-40(a) 所示, 设置初始网民数为 2000, 绿色表示潜在传播者或信息接收者, 红色为危机信息传播源, 在仿真世界中设置一个初始传播源。运行四个步长后, 媒体开始参与到网络危机信息传播中来, 而政府还没有介入。如图 9-40(b) 所示, 红色“笑脸”表示参与危机信息传播的媒体, 黄色的小人表示参与危机事件的接收者, 灰色的小人表示退出危机事件的网民。

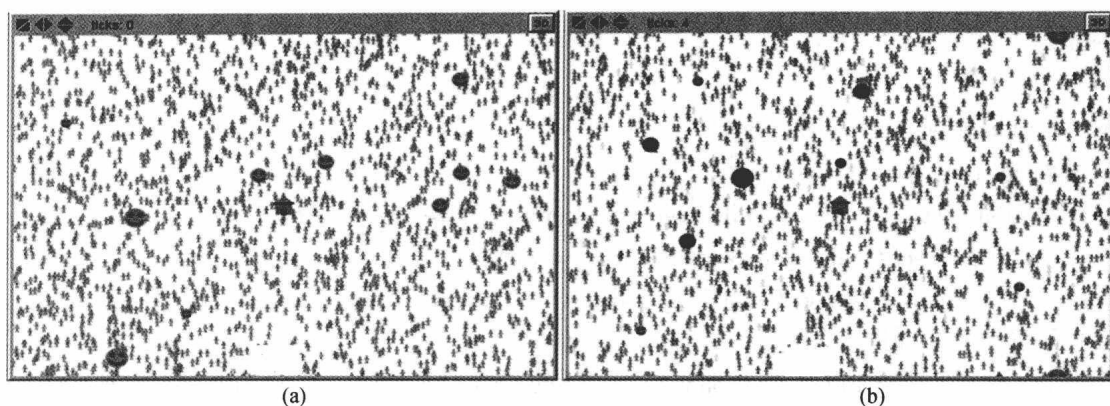


图 9-40 潜伏期各类主体参与数量演示图

运行六个步长后,政府开始介入危机事件,如图 9-41,图上部为 ticks = 6 和 ticks = 7 时仿真世界的图形化显示窗口,下面为对应时刻用绘画功能输出的各种状态下的网民数量变化曲线。

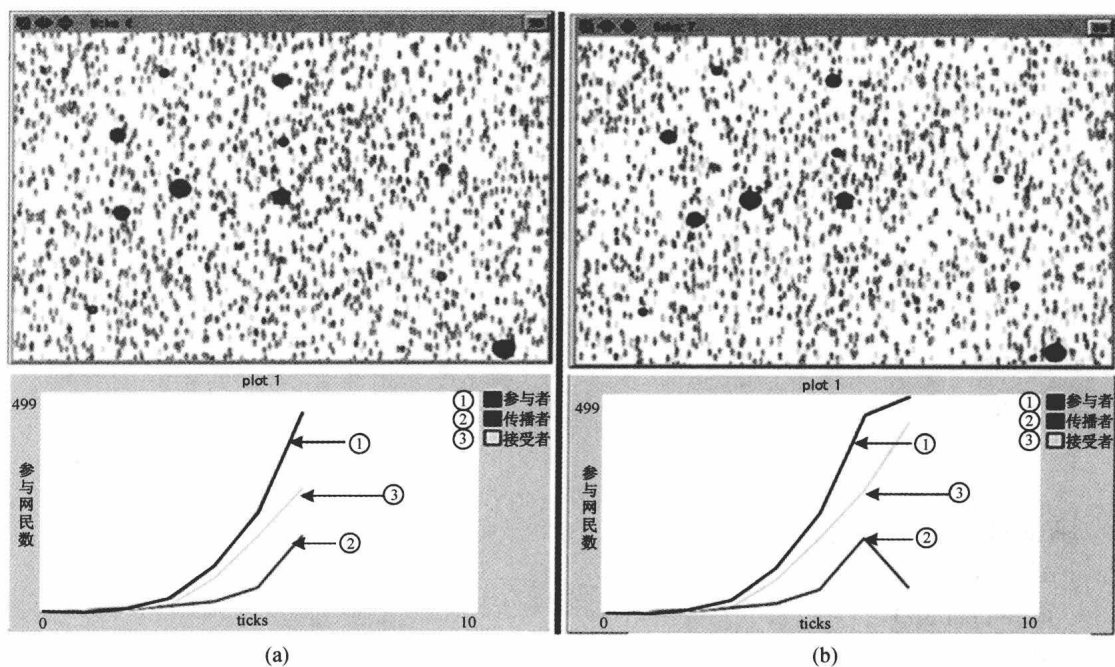


图 9-41 政府介入后各种态度下的网民数量变化仿真结果演示图

(a) 政府开始介入;(b) 政府介入 2 个步长后

运行 20 个步长后,危机事件得到平息,网民、媒体、政府均退出对危机事件的关注,如图 9-42 所示,左面为仿真世界的图形化显示窗口,右面为对应时刻各种状态下的网民数量变化曲线。

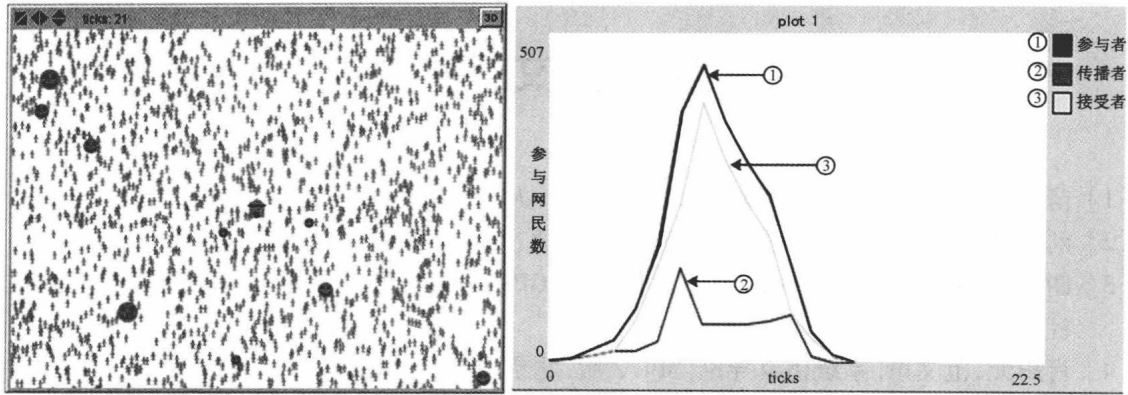


图 9-42 危机平息后仿真结果演示图

参 考 文 献

- [1] 苗东升. 系统科学大学讲稿[M]. 3 版. 北京: 人民大学出版社, 2007.
- [2] 刘思峰, 方志耕, 朱建军, 等. 系统建模与仿真[M]. 北京: 科学出版社, 2012.
- [3] 谢毅, 缪亚萍. 离散事件系统建模仿真及 GPSSWorld 教程[M]. 北京: 清华大学出版社, 2011.
- [4] 肖田元, 范文慧. 系统仿真导论[M]. 2 版. 北京: 清华大学出版社, 2010.
- [5] Jack P C, Kleijnen. Verification and validation of simulation models[J]. European Journal of Operational Research, 1995, 82(6): 145-152.
- [6] 杨群, 邱江, 张庆林. 演绎推理的认知和脑机制研究述评[J]. 心理科学, 2009, 32(3): 646-648.
- [7] 柯匹 M 欧文, 科恩 卡尔. 逻辑学导论[M]. 13 版. 张建军, 潘天群, 顿新国, 译. 北京: 人民大学出版社, 2014.
- [8] 李言俊, 张科, 余瑞星. 系统辨识理论及应用[M]. 北京: 国防工业出版社, 2011.
- [9] 刘兴堂. 复杂系统建模理论、方法与技术[M]. 北京: 科学出版社, 2008.
- [10] 刘思峰, 谢乃明. 灰色系统理论及其应用[M]. 6 版. 北京: 科学出版社, 2013.
- [11] 才华. 基于自组织理论的黑龙省城市系统演化发展研究[D]. 哈尔滨: 哈尔滨工程大学, 2006.
- [12] 苗东升. 系统科学精要[M]. 3 版. 北京: 人民大学出版社, 2010.
- [13] 克拉默 弗里德里希. 混沌与秩序: 生物系统的复杂结构[M]. 柯志阳, 吴彤, 译. 上海: 上海世纪出版集团, 2010.
- [14] 朱华, 姬翠翠. 分形理论及其应用[M]. 北京: 科学出版社, 2011.
- [15] 李学伟, 吴今培, 李雪岩. 实用元胞自动机导论[M]. 北京: 北京交通大学出版社, 2013.
- [16] 祝玉学. 物理系统的元胞自动机模拟[M]. 北京: 清华大学出版社, 2003.
- [17] 谢惠民. 复杂性与动力系统[M]. 上海: 上海科技教育出版社, 1994.
- [18] 周锋. 基于元胞自动机的沿海港口水域船舶交通流演化仿真研究[D]. 武汉: 武汉理工大学, 2012.
- [19] 郭齐胜, 徐享忠. 计算机仿真[M]. 北京: 国防工业出版社, 2011.
- [20] 陈无畏. 系统建模与计算机仿真[M]. 北京: 机械工业出版社, 2013.
- [21] 李伟, 何东之, 李润梅. 基于 Multi-Agent 的多车道交通流的分布式仿真研究[J]. 计算机仿真, 2005, 22(2): 191-194.
- [22] 昊旭光, 杨惠珍, 王新民. 计算机仿真技术[M]. 2 版. 北京: 化学工业出版社, 2008.
- [23] 李柏年, 吴礼斌. MATLAB 数据分析方法[M]. 北京: 机械工业出版社, 2012.
- [24] 刘浩, 韩晶. MATLAB R2012a 完全自学一本通[M]. 升级版. 北京: 电子工业出版社, 2013.
- [25] 薛山. MATLAB 基础教程[M]. 北京: 清华大学出版社, 2011.
- [26] 张志涌, 杨祖樱. MATLAB 教程 R2012a [M]. 北京: 北京航空航天大学出版社, 2010.



- [27] 戴金辉. 基于混合神经网络的入侵检测技术的研究[D]. 沈阳:东北大学,2010.
- [28] 卓金武. MATLAB 在数学建模中的应用[M]. 北京:北京航空航天大学出版社,2011.
- [29] 王小川,史峰,郁磊,等. MATLAB 神经网络 43 个案例分析[M]. 北京:北京航空航天大学出版社,2013.
- [30] 陈明. MATLAB 神经网络原理与实例精解[M]. 北京:清华大学出版社,2013.
- [31] 王凡. 上市公司信用风险评估的实证研究[D]. 北京:北京工业大学,2008.
- [32] 郎茂祥,傅选义,朱广宇. 预测理论与方法[M]. 北京:清华大学出版社,北京交通大学出版社,2011.
- [33] 刘思峰,党耀国. 预测方法与技术[M]. 北京:高等教育出版社,2005.
- [34] 邓聚龙. 灰理论基础[M]. 武汉:华中科技大学出版社,2002.
- [35] 邱均平,文庭孝. 评价学:理论·方法·实践[M]. 北京:科学出版社,2010.
- [36] 林森. 复杂系统评价方法研究[D]. 青岛:青岛大学,2007.
- [37] 殷秀萍. 我国农业综合生产能力动态评价与相关问题研究[D]. 哈尔滨:东北农业大学,2013.
- [38] 杜栋,庞庆华,吴炎. 现代综合评价方法与案例精选[M]. 2 版. 北京:清华大学出版社,2008.
- [39] 张炳江. 层次分析法及其应用案例[M]. 北京:电子工业出版社,2014.
- [40] 李春明. 优化方法[M]. 南京:东南大学出版社,2009.
- [41] 梁旭,黄明. 现代智能优化混合算法及其应用[M]. 北京:电子工业出版社,2011.
- [42] 史峰,王辉,郁磊,等. MATLAB 智能算法 30 个案例分析[M]. 北京:北京航空航天大学出版社,2011.
- [43] 司守奎,孙玺菁. 数学建模算法与应用[M]. 北京:国防工业出版社,2011.
- [44] 高飞. 数学建模算法与应用[M]. 北京:人民邮电出版社,2014.
- [45] Metropolis N, Rosenbluth AE, Teller AH, et al. Equation of state calculations by fast computing machines[J]. Journal of Chemical Physics, 1953, 21: 1087-1092.
- [46] Colomi A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies[C]. Proceeding of the First European Conference on Artificial Life. Paris: Elsevier Publishing, 1991: 134-142.
- [47] 段海滨. 蚁群算法原理及其应用[M]. 北京:科学出版社,2005.
- [48] 冯玉蓉. 模拟退火算法的研究及其应用[D]. 昆明:昆明理工大学,2003.
- [49] 窦全胜. 求解优化问题的演化计算方法研究[D]. 长春:吉林大学,2005.
- [50] 霍兰 H 约翰. 隐秩序:适应性造就复杂性[M]. 周晓牧,译. 上海:上海科技教育出版社,2011.
- [51] 傅星. 基于复杂适应系统理论的经济仿真研究[D]. 北京:首都经贸大学,2005.
- [52] 迟妍. 基于复杂适应系统理论的作战模型研究[D]. 长沙:国防科技大学,2004.
- [53] 周哲. 生态工业复杂适应系统研究[D]. 北京:清华大学,2005.
- [54] 米勒 H 约翰,佩奇 E 斯科特. 复杂适应系统:社会生活计算模型导论[M]. 隆云滔,译. 上海:上海科技教育出版社,2012.
- [55] 宋琪. 基于 CAS 理论的地震应急协同物流网络演化与仿真研究[D]. 哈尔滨:哈尔滨工程大学,2012.
- [56] 盛昭瀚,李静,陈国华. 社会科学计算实验基本教程[M]. 上海:上海三联书店,2010.

- [57] NetLogo 用户手册(中文版)[EB/OL]. 张发,译. [2008-10-16]. http://ccl.northwestern.edu/netlogo/4.0.4/docs/NetLogo_manual_chinese.pdf.
- [58] NetLogo User Manual [EB/OL]. [2013-12-19]. <http://ccl.northwestern.edu/netlogo/docs>.
- [59] 王昌伟. 网络危机信息传播机理与仿真研究[D]. 哈尔滨:哈尔滨工程大学,2012.
- [60] 迟准,王昌伟,梁静国,等. 网络危机信息传播仿真研究——基于政府干预的传播[J]. 情报杂志,2012,31(11):23-29.