浙江大学

博士学位论文

# 一种新型的智能优化方法—人工鱼群算法

姓名：李晓磊

申请学位级别：博士

专业：控制科学与工程

指导教师：钱积新

2003.1.1

# 摘 要

优化命题的解决存在于许多领域，对于国民经济的发展也有着巨大的应用前景。随着优化对象在复杂化和规模化等方面的提高，基于严格机理模型的传统优化方法在实施方面变得越来越困难。

本文将基于行为的人工智能思想通过动物自治体的模式引入优化命题的解决中，构造了一种解决问题的架构 — 鱼群模式，并由此产生了一种高效的智能优化算法 — 人工鱼群算法。

文中给出了人工鱼群算法的原理和详细描述，并对算法的收敛性能和算法中各参数对收敛性的影响等因素进行了分析；针对组合优化问题，给出了人工鱼群算法在其中的距离、邻域和中心等概念，并给出了算法在组合优化问题中的描述；针对大规模系统的优化问题，给出了基于分解协调思想的人工鱼群算法；给出了人工鱼群算法中常用的一些改进方法；给出了人工鱼群算法在时变系统的在线辨识和鲁棒PID的参数整定中两个应用实例；最后指出了鱼群模式和算法的发展方向。

在应用中发现，人工鱼群算法具有以下主要特点：

- 算法只需要比较目标函数值，对目标函数的性质要求不高；
- 算法对初值的要求不高，初值随机产生或设定为固定值均可以；
- 算法对参数设定的要求不高，有较大的容许范围；
- 算法具备并行处理的能力，寻优速度较快；
- 算法具备全局寻优的能力；

鱼群模式和鱼群算法从具体的实施算法到总体的设计理念，都不同于传统的设计和解决方法，同时它又具有与传统方法相融合的基础，相信鱼群模式和鱼群算法有着良好的应用前景。

**关键词** 人工智能，集群智能，动物自治体，人工鱼群算法，优化

# ABSTRACT

Optimization problems need to be solved in many fields, and good solutions to the problems may lead to great economic benefit. With the increasing of the complexity and scale of the optimization problems, classical optimization methods that based on the strict modeling become very difficult to be carried out.

The principle of artificial intelligent that based on animal's behaviors can be used for the solving of optimization problems via animats' method, then an architecture of problem solving, which we called fish school method, was defined, and a efficient intelligent optimization algorithm based on it was introduced which we called artificial fish school algorithm (AFSA).

The principle and description of AFSA was described in detail. The convergence property and the influence of parameters to it was discussed; The definitions of distance, neighborhood and center were introduced to the solving of combinatorial optimization problems; For the solving of complex large scale systems, an extended artificial fish swarm algorithm based on decomposition and coordination method was described; then, some methods of improving the property of AFSA was proposed; and two applications of AFSA was discussed, one is for the parameter estimation of time varying systems, the other is for the tuning of robust PID parameters; at last, the ways where AFSA should go on was proposed.

Some characters of AFSA is as follows:

- The algorithm just compares the values of object functions, so it has no special requirement for the problems.
- The algorithm is insensitive to the initial values, that is , either randomized or fixed initial values are all OK.
- The algorithm can tolerate wide range values of parameters.
- The algorithm has fast optimizing speed for the abilities of parallel processing.
- The algorithm has global search abilities of optimization.

The fish schools method and AFSA have different ways to the traditional method, and they have good foundations of merging with them, then we hope that they can be widely used in future.

**Key Words** artificial intelligent, swarm intelligent, animats, optimization

# 致　谢

在此，谨向我的导师钱积新教授表示忠心的感谢。导师严谨的治学态度和长者风范在学术和生活中为我们树立了学习的楷模。钱老师渊博的知识、丰富的经验、严谨的治学、忘我工作的作风、诲人不倦的学者风范对我影响至深，使我终身难忘。再次向钱老师致以崇高的敬意和衷心的感谢！

衷心感谢邓赤女士、邵之江老师、王慧老师、赵均老师等系统所的老师们，感谢他们所给予的热忱帮助！感谢河海大学常州分校的薛云灿副教授的合作和帮助！难忘与浙江大学城市学院的刘健翔博士一起生活的日子！

感谢柯晶博士、冯少辉博士、陈尚兵博士、于江涛博士、李威武博士、李卫东博士、徐祖华博士、郑永凌硕士、陈国金博士、刘育明博士、刘富春博士、江爱朋博士、黄锦涛博士等同学的帮助。

最后，深深感谢辛勤抚育我长大的父母和给予我鼓励和支持的岳父母及其他家人，特别感谢我的爱人綦焱女士，在作者的学习和工作过程中，给予了极大的支持、理解、关心和照顾，并为作者生下了乖巧可爱的女儿李昕瞳小姐，在她一周岁之际，祝她健康成长、一生幸福！

谨以此文献给所有关心和帮助过我的人们！

李晓磊

二〇〇三年一月

于求是园

# 绪 论

## 1 课题目的和意义

优化是在多种（有限种或无限种）决策中挑选最好决策的问题。它广泛应用于工业、农业、国防、工程、交通、金融、化工、能源、通信等许多领域，如在资源利用、结构设计、调度管理、后勤供应等许多领域中产生了巨大的经济效益和社会效益。优化在结构力学、生命科学、材料科学、环境科学、控制论等其他科学研究领域也有广泛应用。国内外的应用实践表明，在同样条件下，经过优化技术的处理，对系统效率的提高、能耗的降低、资源的合理利用及经济效益的提高等均有显著的效果，而且随着处理对象规模的增大，这种效果也更加显著。这对国民经济的各个领域来说，其应用前景是巨大的。

随着生产的迅速发展，各类工程问题的优化计算越来越成为人们急需解决的问题。在工业界，激烈的市场竞争和多变的市场需求使得现代工业正逐步向大型化和自动化方向发展，安全、平稳、优质和高效正越来越成为企业生产所追求的目标。工业自动化是实现这一目标的基本条件和重要保证。工厂装置规模的不断扩大使得生产操作和装置运行的好坏与工厂经济效益的关系更加密切，从而对工业过程自动化提出了更高的要求。其着眼点从基础控制发展到先进控制，进而发展到在整个企业范围内集成市场供应、能源与原材料价格、人事、财务、设备、库存及过程操作现状等信息来优化管理与控制生产，以实现企业的生产过程综合自动化。当前，人们已从基于离散工业的计算机集成制造系统CIMS与基于连续工业的计算机集成生产系统CIPS认识到，仅靠单纯提高个别生产装置的控制与优化以及只寻求局部最优的生产模式已远远不能适应现代工业的生产要求，只有对整个工段、车间乃至全厂进行先进控制及在线优化才是当今企业技术进步和持续发展的最佳选择([梁昔明 2000])。

目前，基于严格机理模型的开放式方程建模与优化已成为国际上公认的主流技术方向。许多工程公司和各大科研机构纷纷投入大量的人力物力对系统的建模与优化进行深入细致的研究，企图取得突破性的进展。然而，基于严格机理模型所得到的优化命题往往具有方程数多、变量维数高、非线性强等特点，这使得相关变量的存储、计算及命题的求解都相当困难。比如，一个典型的流程工业系统全联立方程的维数通常在上万甚至数十万之多，这其中包括了所有的单元模型方程、物性计算方程、流程联接方程，求解如此庞大的优化命题本身就已相当困难，更难以到工业现场实施在线应用([沈静珠 1994])。不仅工业界存在优化问题，在国民经济的各个领域中也存在着相当多的涉及因素多、规模大、难度高和影响广的优化命题，如运输中的最优调度、生产流程的最优排产、资源的最优分配、农作物的合理布局、工程的最优设计以及国土的最优开发等等，所有这些问题的解决也必须有一个强有力的优化工具来进行求解。常规的优化算法面对这样的大型问题已无能为力，无论是在计算速度、收敛性、初值敏感性等方面都远不能满足要求。因此，针对常规优化算法遇到的难点研究智能的优化方法来解决这些问题是很有意义的。

## 2 国际国内研究状况和进展

传统的优化方法根据问题的性质不同，通常将问题划分为线性规划问题、非线性规划问题、整数规划问题和多目标规划问题等，相应的有一些较成熟的常规算法，如应用于线性规划问题的单纯形法，应用于非线性规划的Newton法、共轭梯度法、序列二次规划法（SQP）等，应用于整数规划问题的分支定界法、割平面法和动态规划法等([马振华 1997], [归行茂 & 李重华 1993])。这些方法都是基于严格的数学模型的，当模型复杂的时候，如变量的维数多、约束方程数多、非线性强等，或模型不能用显式的方程来表达时，这些方法往往不能进行有效求解，或者求解的时间过长，如组合优化问题中的组合爆炸；或者求解的效果差，如陷入局部极值、初始值

直接影响寻优的结果等。

人们从生物进化的机理和一些物理现象中受到启发，提出了许多用以解决复杂优化问题的新方法，因其高效的优化性能、无需问题特殊信息等优点，受到各领域广泛的关注和应用([Osman 1996])。研究较多的有模拟退火(SA)([Kirkpartick & Gelatt 1983])、遗传算法(GA)([Holland 1992])、禁忌搜索(TS)([Glover 1989])、进化规划(EP)([Fogel 1993])、进化策略(ES)([Fogel 1994])等。

SA、GA和TS作为具有全局优化性能的该类算法代表，与人工神经网络统称为四大现代启发式算法，具有区别于常规算法的搜索机制和特点。SA是基于Mente Carlo迭代求解的一种全局概率型搜索算法，首先由Kirkpatrick等人用于组合优化([Kirkpartick & Gelatt 1983])；GA是Holland研究自然遗传现象与人工系统的自适应行为时，借鉴“优胜劣汰”的生物进化与遗传思想而提出的一种全局性并行搜索算法([Holland 1992], [Zames 1981])；GA注重父代与子代遗传细节上的联系（强调染色体操作）；EP和ES则侧重父代与子代表现行为上的联系（强调物种层的行为变化）；TS是Glove模拟智能过程而提出的一种具有记忆功能的全局逐步优化算法([王凌 & 郑大钟 2000])。

各种启发式算法在优化机制方面存在一定的差异，但优化流程却具有较大的相似性，均是一种“邻域搜索”框架。算法从一个(一组)初始解出发,在算法参数的控制下通过邻域函数产生若干邻域解,按接受准则(确定性、概率性或混沌方式)更新当前状态，然后按参数修改准则调节控制参数，如此重复以上搜索过程直至算法终止准则满足，最终得到问题的优化解。搜索方式（并行或串行）、邻域函数、状态更新方式、算法参数及其修改方式、算法收敛准则是算法的核心。SA是一种串行优化算法，每步仅随机尝试当前状态邻域中的一个状态，同时通过控制“温度”来控制状态更新概率，从而在搜索过程中具有避免局部极小的能力并最终趋于全局最优。GA是一种并行优化算法，尤其具有很大的隐含并行性，在码空间以一定的概率通过对种群的不断选择、交叉和变异操作达到群体并行进化目的。TS是一种串行优化算法，通过记忆近期操作的存储结构来避免状态的

重复出现，并结合藐视准则实现全局快速搜索。初始温度、退温温度、状态产生方式、抽样稳定准则是影响SA性能的关键因素；种群数目、复制、交叉及变异操作和操作概率是影响GA性能的关键因素；禁忌表大小、邻域函数结构与数量是影响TS性能的关键因素([王凌 & 郑大钟 2000])。

这类算法在解决问题时同样会受到变量维数的局限性，更会因此带来计算时间的增加，同时，通常不能全面直接的将优化问题的约束进行表达，并且在实施时，需要一定的设计技巧，如GA中的编码问题([Goldberg 1989]，[De Jong 1975]，[邢文训 & 谢金星 1999])。

近年来，随着人工智能和人工生命的兴起，出现了一些新型的仿生算法。其中较具代表性的有蚁群算法和粒子群算法。蚁群算法由意大利学者M. Dorigo 等人提出([Dorigo 1996]，[张纪会 & 高齐圣 2000])，他们称为蚁群系统（Ant Colony System），并用它来求解旅行商问题(TSP)([Dorigo 1997a]，[Dorigo 1997b]，[Dorigo & Bonabeau 2000])、指派问题（assignment problem）([Maniezzo & Colorni 1994])、job-shop 调度问题([Colomi & Dorigo 1994])等，取得了一系列较好的结果。在蚁群算法中，利用了如下自然现象：一只蚂蚁仅仅追随其他蚂蚁留下的尾迹，最终会在无数通往食物源的路径中找到最短的一条（如图 1 所示）([H lldobler 1990])。

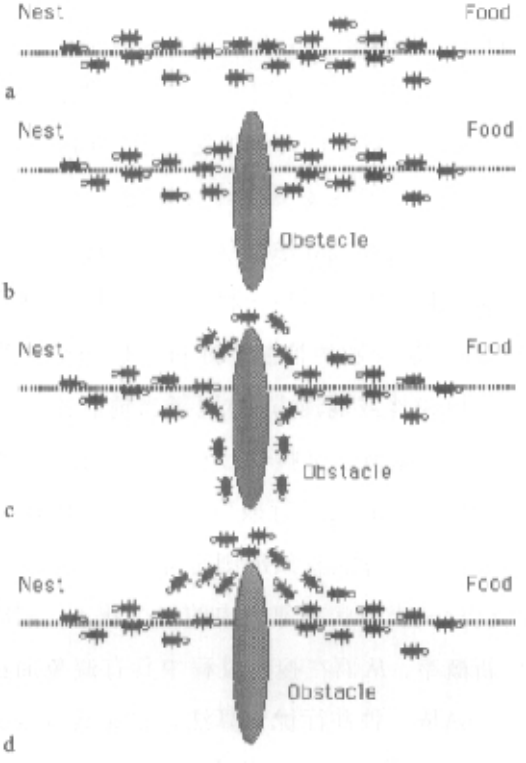



图 1 真实蚁群行为示意图

该算法也存在一些缺陷，如：与其他算法相比，该算法需要较长的搜索时间；容易出现停滞现象，即搜索进行到一定程度

后，所有个体所发现的解完全一致，不能对解空间进一步进行搜索，不利于发现更好的解。对于这些问题，许多学者已经提出了若干改善方法，如M. Dorigo提出的Ant-Q system([Luca & Gambardella 1995])，Thomas等人提出的Max-Min ant system([Thomas & Holger 1997])等。

J. Kennedy,和R.C. Eberhart从模拟动物（包括人类）的社会行为中发展了粒子群优化算法([Kennedy & Eberhart 1995], [Eberhart & Kennedy 1995], [Shi & Eberhart 1999])。算法中每个粒子在解空间中“飞行”的过程中参照自身的经历和同伴粒子的飞行经历来动态的调整自身的速度，从而达到在全空间中搜索最优解的目的。算法中如果动量项的系数较大，粒子群有可能错过最优解，造成在远离最优解的空间中发散；其次，在算法后期所有的粒子趋于同一，失去了粒子的多样性，使得算法的收敛速度减慢，以至于无法达到规定的精度。当然，有很多工作针对其进行了相应的改进([Angeline 1998] ,[Shi & Eberhart 1998])。

## 3 论文各部分的主要内容

将人工智能的思想应用于问题的寻优是一个自然而然的思路，随着计算设备和软件技术的发展，我们有能力实施一些高级的计算方法，使得这一思路逐渐变得现实和实用。而应用动物自治体的模式来定义实体，让他们在问题空间中自主的活动，从而达到解决问题的目的，这就是我们工作的核心。

文章的结构如下：

- 第一章介绍了集群智能和鱼群模式的概念；
- 第二章提出了一种新型的智能优化方法：人工鱼群算法，并对它的收敛性和各参数对收敛性的影响进行了分析；
- 第三章给出了人工鱼群算法在组合优化问题中的应用；
- 第四章给出了基于分解协调方法的人工鱼群算法应用；
- 第五章给出了人工鱼群算法的一些改进方法；

- 第六章给出了基于人工鱼群算法的系统辨识和鲁棒 PID 参数整定方法两个应用实例。
- 最后对全文进行了总结，并指出了今后的发展方向。

# 第一章 鱼群模式概论

我们有幸生活在这样一个多姿多彩的星球上，多样的物种、多彩的生活，无不使我们惊叹这大自然的杰作。集群是生物中常见的一种生存现象，如昆虫、鸟类、鱼类、动物、微生物乃至我们人类等。如图 1.1 是在博茨瓦纳的奥卡万戈河三角洲观察到的红嘴奎利亚雀在黄昏时分返回公共栖息地时形成一个巨大的群的壮观景象，这在自然界中并非少见。生物的这些特性是在漫长的进化过程中逐渐形成的，对他们的生存和进化有着十分重要的影响，同时这些方式为人类解决问题的思路带来了不少启发和鼓舞（[Gelenbe & Schmajuk 1997]）。因此，近年来，有不少科学家对生物的行为进行了广泛的研究，并逐渐形成了一种基于生物行为的人工智能模式。

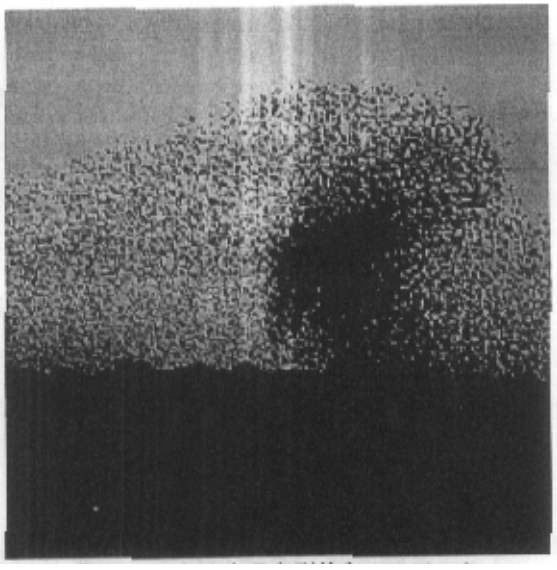


图 1.1 大自然中观察到的鸟类聚群现象

## 1.1 人工智能

人工智能广义的讲，是关于人造事物（理论的或工程的）的智能行为，而智能行为包括知觉、推理、学习、交流和在复杂环境中的行为（[Nilsson 1999]）。长期以来，围绕人工智能有许多争议，就“机器能否思考？”、“什么才算真正的智能？”等问题也没有很确切的共识，某些概念对于特定的研究方法有特定的定义和解释。尽管人工智能发展了四十多年了，人

们利用许多方法创造出了一些实用的系统和理论，但人们不得不承认这些远未达到人类所期望的智能水平。总结这些方法，通常能够分为两大类：

第一类是符号处理的方法，他们基于 Newell 和 Simon 的物理符号系统（[Newell & Simon 1976]）的假说，这一类方法通常被称为经典人工智能。这类方法中突出的方法是将逻辑操作应用于说明性知识库。

第二类方法是基于生物进化和行为的方法。他们通常采用自下而上的方式。这类方法中突出的方法是“Animat Approach”（[Blumberg & Meyer 1998]，[Brooks 1990]，[Maes 1990]，[Wilson 1991]）和“Genetic Algorithms”。

经典的人工智能方法主要是面向能力的模式，通常是把人的高级智能能力加以描述并赋予被设计的对象。它往往采用自上而下的构造方式：由设计者对将要解决的问题建立一个抽象的符号表达的系统模型，然后设计者将自己对这一问题的认识思想和解决思路尽可能完美的用程序的方式表达出来，从而解决这个问题（[蔡自兴 & 徐光祐 1996]），如图 1.2 所示。经典的人工智能的特点可以总结如下：

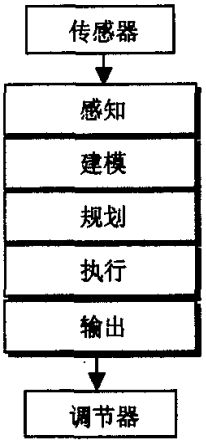


图 1.2 经典人工智能结构

- 通常具备一个存储中心，用来存储诸如感知、分类、目的、意图等系统所需要的信息；
- 通常需要一个控制中心，用来生成感知和分类、意图和行为等之间的逻辑关系；
- 需要一个完备、精确的环境模型；
- 通常是序贯执行的方式，如图 1.2 所示。
- 设计的策略与设计的目标之间存在精确的一致关系；
- 通常忽视主体的自学习和自适应行为。

基于生物进化和行为的人工智能方法强调智能主体对环境的自适应行为（[Meyer 1997]），通常采用自下而上的设计方法（[戴汝为 & 周登勇 2000]）：首先设计单个实体的感知、行为等机制，然后将一个或一群实体

放置于环境之中，让他们在与环境的交互作用中达到问题的解决。以下是它的几个主要特点([Brooks 1991a], [Brooks 1991b])：

- 内嵌的（Situatedness）

  实体置身于环境之中，他不只依靠于对环境的抽象的符号描述，而是直接的与环境接触，并进行感知([Brooks 1986])；

- 物化的（Embodiment）

  与生物类似，实体具备感官和形体的结构，他们的行为成为动态环境的一部分，它们的感官能够通过与环境的接触及时反馈，从而认识环境([Lenat & Feigenbaum 1991],[ [Moravec 1984]])；

- 自治的（Autonomous）

  它在环境中表现出自治性的活动，也就是我们观察到的智能行为，这些行为的实现，除去自身的运算、逻辑机制外，也来源于它所置身于的环境；

- 突现的（Emergence）

  它们的智能是突现出来的([Minsky 1961], [Nicolis & Prigogine 1977])，有时是通过与环境的直接交互作用，有时是间接的通过各组件的交互作用，所以，有时难以说明是哪一种内在的因素造成了外部所突现出来的某一能力([Horswill & Brooks 1988])。

## 1.2 集群与集群智能

一个集群通常定义为一群自治体的集合，他们通过相互间直接的或者间接的通讯，从而通过全体的活动来解决一些分布式的难题([Hoffmeyer 1994] ,[Hoffmeyer 1997], [Bonabeau & Théraulaz 2000])。在这里，自治体是指在一个环境中具备自身活动能力的一个实体，其自身力求简单，通常不必具有高级智能，但是，他们的集群的活动所表现出来的则是一种高级智能才能达到的活动，我们称这种活动为集群智能。

长期以来，以群居生活的昆虫 - 如蚂蚁、蜜蜂、黄蜂、白蚁等，引

起了包括自然学家和艺术家在内的人们的兴趣。比利时诗人 Maurice Maeterlinck 曾经感慨道:“到底是谁在主宰着它们？是谁在维持它们的秩序、预测它们的未来、精心为它们制作计划、维持它们的平衡？”。这些确实是一些待解的迷。

群居昆虫中的每一个个体看上去都有自身的行动方式，并且整个群在整体上呈现出高度的有组织性。显然，在所有个体活动的完美集成过程中不需要任何的指导。事实上，研究社会性昆虫的科学家发现在群体中的协作是高度自组织的，它们的协调行为是通过个体之间的交互行为直接实现，或者个体与环境的交互行为间接实现的。虽然这些交互行为非常简单，但是他们聚在一起却能解决一些难题。这种潜在方式的集群智能已经逐渐为人们所认识，并得到应用。

Craig W. Reynolds 对鸟群等的行为进行了模拟([Reynolds 1987])。每只虚拟的鸟都作为一个独立的因素，它们通过感知周围局部的动态环境来确定自身飞行的路线，在仿真过程中每只虚拟的鸟主要采用了三个行为：1、防止碰撞：避免与周围的同伴发生碰撞；2、速度匹配：试图使自身的速度与周围伙伴的速度一致；3、中心聚拢：试图向附近的同伴聚拢。研究表明，一定数量的这种虚拟鸟能够在复杂的环境中聚集成群并自由避开障碍物。

可见，在集群的过程中没有所谓的领导者，单个个体的行为也相对比较简单，个体看上去是盲目与随机的，但整体的行为却是有序与有目的的。这样就使得该类问题的设计不同于传统设计方法，从而引入了自下而上的设计方法 — 动物自治体模式（Animats）。

## 1.3 动物自治体

动物自治体通常指自主机器人或动物模拟实体，它主要是用来展示动物在复杂多变的环境里面能够自主的产生自适应的智能行为的一种方式([Wilson 1991], [Dean 1998])，其结构如图 1.3，其主要的特性有以下

几点：

- 它具有物化机制，具备感官和形体的结构等；
- 它是置身于环境的，直接的与环境进行交互的作用，既能感知环境，也能改变环境；
- 它的行为是自适应的，通过与环境的交互作用，能够自主的作出反应；
- 能在复杂的环境中执行多任务；
- 具备多种行为，并且能够并行分布执行；
- 当它们被组合在一起的时候，高级智能行为往往能在它们的个体的简单行为中突现出来。

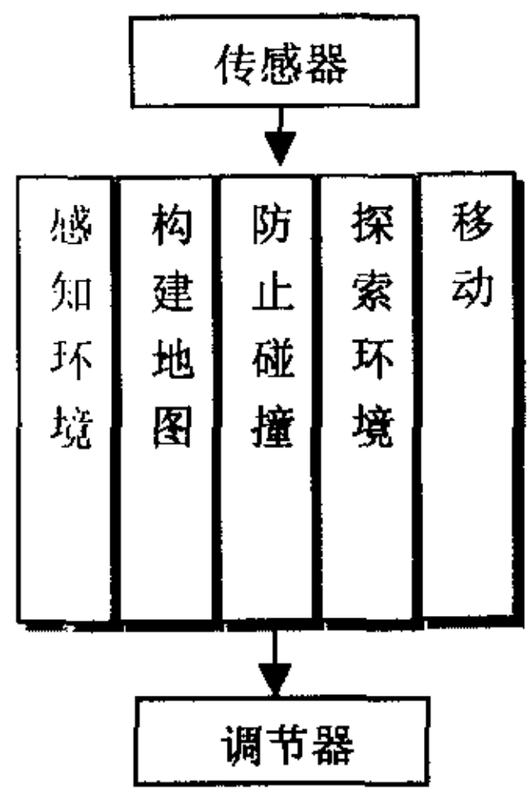


图 1.3 动物自治体结构

Animats 方法存在两个方面，一个是问题方面，一个是解决问题方面，主要是环境的建立和自治体的构造两个主要因素，这两者形成一个刺激和应激相互作用的过程。

环境通常可用以下 FSM（*finite-state machine*）（[Riolo 1987], [Rivest & Schapire 1987]）模型表示：

$$Q(t+1)=F(Q(t),A(t))$$
$$E(t+1)=G(Q(t),A(t))$$

其中，$A$ 为环境的输入，主要是自治体的活动，$E$ 为环境的输出，作为对自治体的感官的刺激，$Q$ 反映了当前环境的状态。

方程中，第一个等式说明环境的下一个状态是当前状态和自治体活动的函数，第二个等式说明对自治体的下一个刺激是环境的当前状态和其自身活动的函数。

在 FSM 模型中，对于环境的复杂性的反映不是很透明，以至于自治体对环境的某些状态难以感受到，通常引入以下 SSM(*sensory-state machine*) 模型([Wilson 1991])，

$$\{E(t+1)\}=f(E(t),A(t))$$

表示，下一次的刺激是当前刺激和活动的函数。$E(t+1)$不是唯一的，而是一个有限的集合。虽然这一模型不能完备的描述环境，但它是比较实用的，它在一定程度上的不确定性反应了环境的相对复杂性。

如果考虑一定的历史信息，则可以构造以下二阶 SSM 模型：

$$\{E(t+1)\} = f_2(E(t), A(t), E(t-1), A(t-1))$$

自治体的合理架构就是能在环境的刺激下做出最好的应激活动。一种理想的架构模式就是将主要的刺激－应激关系做成一个映射，使得对环境的刺激能够立即做出正确的反应或活动，更理想的情况就是能够像动物或人类那样能形成条件反射或具备自学习能力，从而能够做出更好的应激活动。

## 1.4 鱼群模式

在本论文中，将动物自治体的概念引入优化算法中，采用了自下而上的思路，应用了基于行为的人工智能方法，形成了一种新的解决问题的模式，因为是从分析鱼类的活动出发的，所以称为鱼群模式。该模式应用于寻优中，形成了人工鱼群算法。下面就介绍一下鱼群模式中的一些主要观点。

### 1.4.1 视觉

最原始、最直观的寻优方法我想应该是最优方向法，或者叫爬山法，它在实施过程中仅仅需要比较一下目前状态和目的状态的优劣，然后选择较优的状态，这样一步步向最优状态迈进。当然，单纯的爬山法存在一些缺陷，如步长的选择过大将造成算法的振荡而无法收敛，步长选择过小会造成算法收敛速度过慢，且容易陷入局部极值中。但它对目标问题的性质要求很低，甚至可以不是显式的数学表达式，并且实施简单，比较符合生物的行为方式。

生物的视觉是极其复杂的，它能快速感知大量的空间事物，这对于任何仪器和程序都是难以与之相比的，为了实施的简便和有效，在鱼群模式中我们应用了如下的方法来实现虚拟人工鱼的视觉：

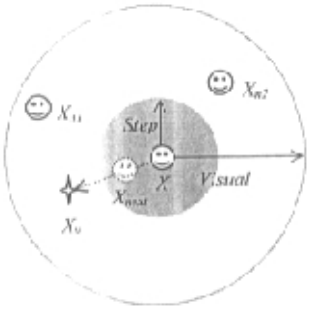



图 1.4 人工鱼的视野和移动步长

如图 1.4 所示，一虚拟人工鱼实体的当前状态为 $X$，$Visual$ 为其视野范围，状态 $X_v$ 为其在某时刻视点所在的位置，如果该位置的状态优于当前状态，则考虑向该位置方向前进一步，即到达状态 $X_{next}$；如果状态 $X_v$ 不比当前状态更优，则继续巡视视野内的其他位置。巡视的次数越多，则对视野内的状态了解更全面，从而对周围的环境有一个全方面立体的认知，这有助于做出相应的判断和决策，当然，对于状态多或无限状态的环境也不必全部遍历，允许一定的不确定性对于摆脱局部最优，从而寻找全局最优是有帮助的。

其中，状态 $X=(x_1,x_2,\cdots,x_n)$，状态 $X_v=(x_1^v,x_2^v,\cdots,x_n^v)$，则该过程可以表示如下：

$$x_i^v = x_i + Visual \cdot Rand(), i = n \cdot Rand()$$

$$X_{next} = \frac{X_v - X}{\|X_v - X\|} \cdot Step \cdot Rand()$$

式中，$Rand$ 函数为产生 0 到 1 之间的随机数，$Step$ 为移动步长。

由于环境中同伴的数目是有限的，因此在视野中感知同伴（如图 1.4 中 $X_{n1}$，$X_{n2}$）的状态，并相应的调整自身状态的方法与上式类似。

### 1.4.2 鱼群行为分析

鱼类与我们的关系相当密切，也较为我们所熟知，通常我们可以观察到如下的鱼类行为：

觅食行为：这是生物的一种最基本的行为，也就是趋向食物的一种活

动；一般可以认为它是通过视觉或味觉来感知水中的食物量或浓度来选择趋向的，因此，以上所述的视觉概念可以应用于该行为。

聚群行为：这是鱼类较常见的一种现象，大量或少量的鱼都能聚集成群，这是它们在进化过程中形成的一种生存方式，可以进行集体觅食和躲避敌害。

追尾行为：当某一条鱼或几条鱼发现食物时，它们附近的鱼会尾随其后快速游过来，进而导致更远处的鱼也尾随过来。

随机行为：鱼在水中悠闲的自由游动，基本上是随机的，其实它们也是为了更大范围的寻觅食物或同伴。

以上是鱼的几个典型行为，这些行为在不同时刻会相互转换，而这种转换通常是鱼通过对环境的感知来自主实现的，这些行为与鱼的觅食和生存都有着密切的关系，并且与我们优化问题的解决也有着密切的关系。

行为评价是我们用来模拟鱼能够自主行为的一种方式。在解决优化问题中，可以选用两种简单的评价方式：一种是选择最优行为执行，也就是在当前状态下，哪一种行为向优的方向前进最大，就选择这一行为；另一种是选择较优行为前进，也就是任选一种行为，只要能向优的方向前进即可。

### 1.4.3 人工鱼

人工鱼是真实鱼个体的一个虚拟实体，用来进行问题的分析和说明，它可以用 animat 的概念来实现。如图 1.5 所示，借助于面向对象的分析方法，可以认为人工鱼就是一个封装了自身数据信息和一系列行为的一个实体，可以通过感官来接收环境的刺激信息，并通过控制尾鳍来作出相应的应激活动。

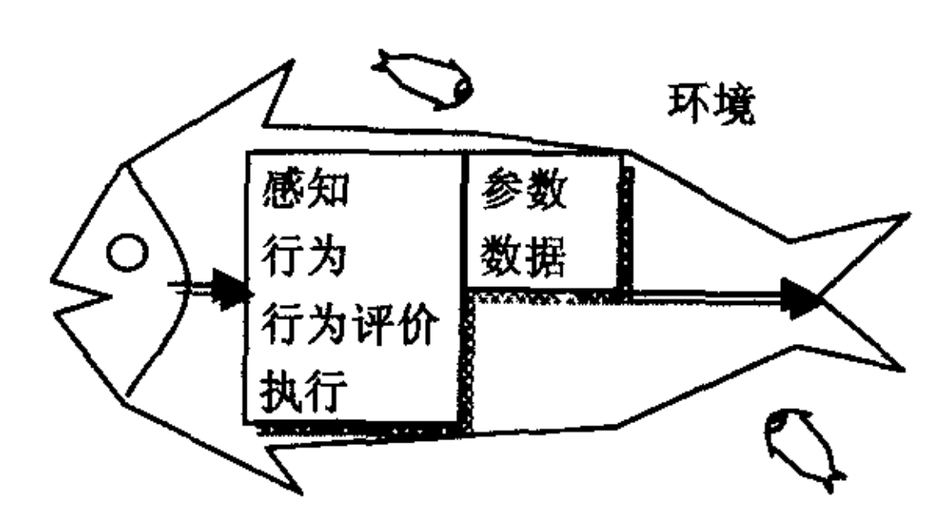


图 1.5 人工鱼实体

人工鱼所在的环境主要是问

题的解空间和其它人工鱼的状态，它在下一时刻的行为取决于目前自身的状态和目前环境的状态（包括问题当前解的优劣和其他同伴的状态），并且通过它还通过自身活动的同时来影响环境，进而影响其他同伴的活动。

### 1.4.4 问题的解决

问题的解决是通过自治体在自主的活动过程中以某种形式表现出来的。在寻优过程中，通常会有两种方式表现出来：一种形式是通过人工鱼最终的分布情况来确定最优解的分布，通常随着寻优过程的进展，人工鱼往往会聚集在极值点的周围，而且，全局最优的极值点周围通常能聚集较多的人工鱼；另一种形式是在人工鱼的个体状态之中表现出来的，即在寻优的过程中，跟踪记录最优个体的状态，就类似于遗传算法等的方式。

## 1.5 结论

鱼群模式不同于传统的问题解决方法，提出了一种新的模式，从而在下文中引出了一种新的优化模式一人工鱼群算法，使得这一模式具备并行分布处理、参数和初值的鲁棒性强等能力。

# 第二章 基本人工鱼群算法

在动物的进化过程中，经过漫长的自然界的优胜劣汰，形成了形形色色的觅食和生存方式，这些方式为人类解决问题的思路带来了不少启发和鼓舞([Gelenbe & Schmajuk 1997])。动物一般不具备人类所具有的复杂逻辑推理能力和综合判断能力的高级智能，它们的目的是在个体的简单行为或通过群体的简单行为而达到或突现出来的。

动物行为具有以下几个特点：1）适应性：动物通过感觉器官来感知外界环境，并应激性的做出各种反应，从而影响环境，表现出与环境交互的能力；2）自治性：动物有其特有的某些行为，在不同的时刻和不同的环境中能够自主的选取某种行为，而无需外界的控制或指导；3）盲目性：不像传统的基于知识的智能系统，有着明确的目标；单个个体的行为是独立的，与总目标之间往往没有直接的关系；4）突现性：总目标的完成是在个体行为的运动过程中突现出来的；5）并行性：各个体的行为是实时的、并行进行的。

根据前面介绍的鱼群模式的思想，利用鱼群的特点提出了一种新型的优化方法一人工鱼群算法([李晓磊 & 钱积新 2001]，[李晓磊 & 邵之江 2002])。

## 2.1 前言

在一片水域中，鱼生存的数目最多的地方一般就是本水域中富含营养物质最多的地方，依据这一特点来模仿鱼群的觅食等行为，从而实现全局寻优，这就是鱼群算法的基本思想。

正如鱼群模式中提到的，鱼类的活动中，觅食行为、聚群行为、追尾行为和随机行为与我们的寻优命题的解决有着较密切的关系，如何利用简便有效的方式来构造并实现这些行为将是算法实施的主要问题。

觅食行为主要认为就是循着食物多的方向游动的一种行为，在寻优算法中则是向较优方向前进的迭代方式，如鱼群模式中的视觉概念；

在聚群行为中，我们可以借鉴 Reynolds 的思想([Reynolds 1987])，在此我们对每条人工鱼规定了这样两个规则：1）尽量向临近伙伴的中心移动；2）避免过分拥挤，这样就能基本实现人工鱼的聚群能力；

追尾行为就是一种向临近的最活跃者追逐的行为，在寻优算法中可以理解为是向附近的最优伙伴前进的过程。

下面，就对这几种行为进行详细的描述和实现。

## 2.2 人工鱼模型(AF-artificial fish)

算法基于 animats 的模式，采用了自下而上的设计方法，所以，首先着重构造人工鱼的模型。这里主要采用了面向对象的技术，并用 C++语言的伪代码形式来说明。通常，人工鱼的模型可以用如下的类来描述：

```
class Artificial_fish
{
Various:
    float AF_X[n];                  //AF's position
    float AF_step;                  //the distance that AF can move for each step
    float AF_visual;                //the visual distance of AF
    float try_number;               //attempt time in the behavior of prey
    float delta;                    //the condition of jamming
Functions:
    float AF_foodconsistence();     //the food consistence of AF's current position
    void AF_move();                 //AF move to the next position
    float AF_follow();              //the behavior of follow
    float AF_prey();                //the behavior of prey
    float AF_swarm();               //the behavior of swarm
    int AF_evaluate();              //evaluate and select the behavior
    void AF_init();                 //to initialize the AF

    Artificial_fish();
```

```
    virtual ~Artificial_fish();
};
```

这样，人工鱼的自身信息和一些行为封装在一个类里面，并且它的状态可以为其它同伴所感知到。

### 2.2.1 一些定义

人工鱼个体的状态可表示为向量 $X=(x_1,x_2,\cdots,x_n)$，其中 $x_i(i=1,\cdots,n)$ 为欲寻优的变量；人工鱼当前所在位置的食物浓度表示为 $Y=f(X)$，其中 $Y$ 为目标函数值；人工鱼个体之间的距离表示为 $d_{i,j}=\|X_i-X_j\|$；*Visual* 表示人工鱼的感知距离；*Step* 表示人工鱼移动的最大步长；$\delta$ 为拥挤度因子。

### 2.2.2 行为描述

2.2.2.1 觅食行为

设人工鱼当前状态为 $X_i$，在其感知范围内随机选择一个状态 $X_j$，如果在求极大问题中，$Y_i<Y_j$（或在求极小问题中，$Y_i>Y_j$，因极大和极小问题可以互相转换，所以以下均以求极大问题讨论），则向该方向前进一步；反之，再重新随机选择状态 $X_j$，判断是否满足前进条件；这样反复尝试 *try_number* 次后，如果仍不满足前进条件，则随机移动一步。

其伪代码描述如下：

```
float Artificial_fish::AF_prey()
{
    for (i=0;i<try_number;i++)
    {
        X_j = X_i + Rand()·Visual;
        if  (Y_i < Y_j)
```

$$X_{i|next} = X_i + Rand() \cdot Step \cdot \frac{X_j - X_i}{\|X_j - X_i\|};$$

*else*

$$X_{i|next} = X_i + Rand() \cdot Step;$$

*}*

*return AF_foodconsistence(* $X_{i|next}$ *);*

*}*

2.2.2.2 聚群行为

设人工鱼当前状态为 $X_i$，探索当前邻域内（即 $d_{i,j} < Visable$）的伙伴数目 $n_f$ 及中心位置 $X_c$，如果 $\frac{Y_c}{n_f} > \delta Y_i$，表明伙伴中心有较多的食物并且不太拥挤，则朝伙伴的中心位置方向前进一步；否则执行觅食行为。

伪代码描述如下：

*float Artificial_fish::AF_swarm()*

*{*

$n_f = 0;\ X_c = 0;$

*for (j=0;j<friend_number;j++)*

*if(* $d_{i,j} < Visual$ *)* { $n_f$ ++; $X_c += X_j$; }

$X_c = \frac{X_c}{n_f};$

*if* $(\frac{Y_c}{n_f} > \delta Y_i)$

$$X_{i|next} = X_i + Rand() \cdot Step \cdot \frac{X_c - X_i}{\|X_c - X_i\|};$$

*else*

*AF_prey();*

*return AF_foodconsistence(* $X_{i|next}$ *);*

*}*

2.2.2.3 追尾行为

设人工鱼当前状态为 $X_i$，探索当前邻域内（即 $d_{i,j} < Visable$）的伙伴中 $Y_j$ 为最大的伙伴 $X_j$，如果 $\frac{Y_j}{n_f} > \delta Y_i$，表明伙伴 $X_j$ 的状态具有较高的食物浓度并且其周围不太拥挤，则朝伙伴 $X_j$ 的方向前进一步；否则执行觅食行为。

伪代码描述如下：

```
float Artificial_fish::AF_follow()
{
    Y_max = -∞ ;
    for (j=0;j<friend_number;j++)
        if( d_{i,j} < Visual && Y_j > Y_max)
            { Y_max = Y_j ; X_max = X_j ; }
    n_f = 0 ;
    for (j=0;j<friend_number;j++)
        if( d_{max,j} < Visual )  { n_f ++ ; }
    if  ( Y_max / n_f > δY_i )
            X_{i|next} = X_i + Rand() · Step · (X_max - X_i) / ||X_max - X_i|| ;
        else
            AF_prey();
    return AF_foodconsistence( X_{i|next} );
}
```

2.2.2.4 随机行为

随机行为的实现较简单，就是在视野中随机选择一个状态，然后向该方向移动，其实，它是觅食行为的一个缺省行为。

### 2.2.3 行为选择

根据所要解决的问题性质，对人工鱼当前所处的环境进行评价，从而选择一种行为。如较常用的评价方法就是选择各行为中使得向最优方向前进最大的行为，也就是各行为中使得人工鱼的下一个状态最优的行为，如果没有能使下一状态优于当前状态的行为，则采取随机行为。

## 2.3 算法描述

鉴于以上描述的人工鱼模型及其行为，每个人工鱼探索它当前所处的环境状况（包括目标函数的变化情况和伙伴的变化情况），从而选择一种行为，最终，人工鱼集结在几个局部极值的周围，一般情况下，在讨论求极大问题时，拥有较大的 *AF_foodconsistence* 值的人工鱼一般处于值较大的极值域周围，这有助于获取全局极值域，而值较大的极值区域周围一般能集结较多的人工鱼，这有助于判断并获取全局极值。如图 2.1 所示，满意解域 S 就是所获取的全局极值域，再根据该域的特性来获取较精确的极值，如通常情况下，可以选择域内各人工鱼的重心作为极值。

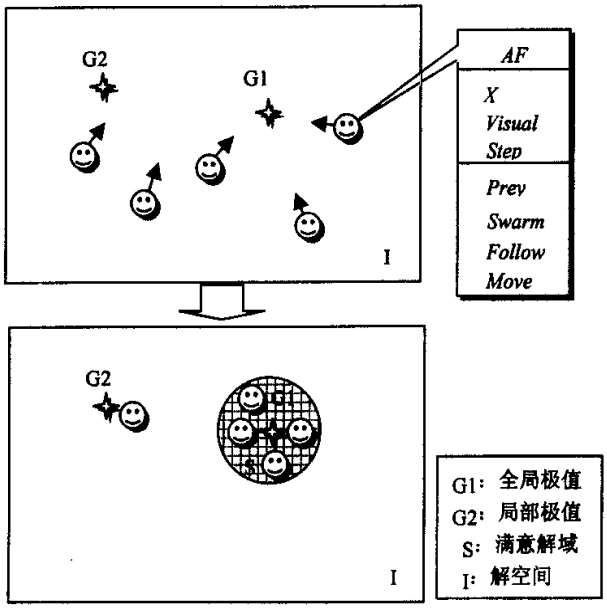



图 2.1 算法示意图

算法的描述如图 2.2 所示。算法对变量的初值要求不高，通常由*::AF_init()*来初始化 *AF* 为随机分布在变量域内的值；算法的终止条件可以根据实际情况设定，如通常的方法是判断连续多次所得值的均方差是否小于预期的误差，或者直接规定迭代的次数。

```
procedure Artificial_Fishschool_Algorithm
    ::AF_init();
    while the result is satisfied do
        switch (::AF_evaluate())
            case value1:
                ::AF_follow();
            case value2:
                ::AF_swarm();
            default:
                ::AF_prey();
        end switch
        ::AF_move();
        get_result();
    end while
end Artificial_Fishschool_Algorithm
```

图 2.2 AFA 的算法描述

## 2.4 仿真实验研究

寻优对象选取了下面的非线性目标函数：

$$\max f(x,y)=\frac{\sin(x)}{x}\frac{\sin(y)}{y}$$
$$s.t.\quad x\in[-10,10] \qquad (2.1)$$
$$y\in[-10,10]$$

如图 2.3 所示，该函数的极点位于（0,0）处，极值为 1，可以看出，该非线性函数在全局极大值的周围密布着许多局部极值，通常的寻优算法极易陷入局部极值或在各局部极值间振荡，比较适用于验证算法的性能。

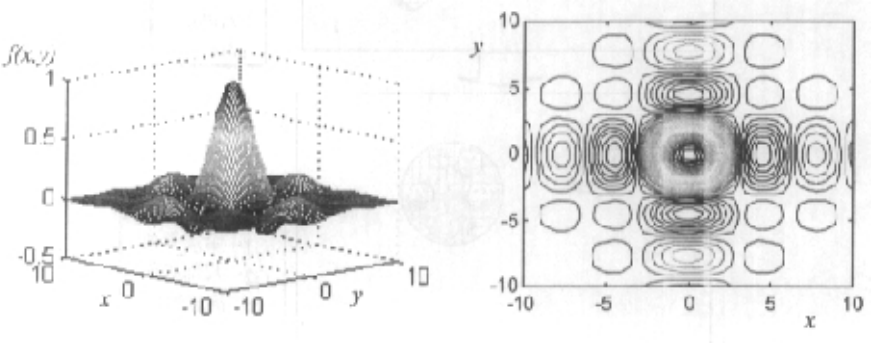
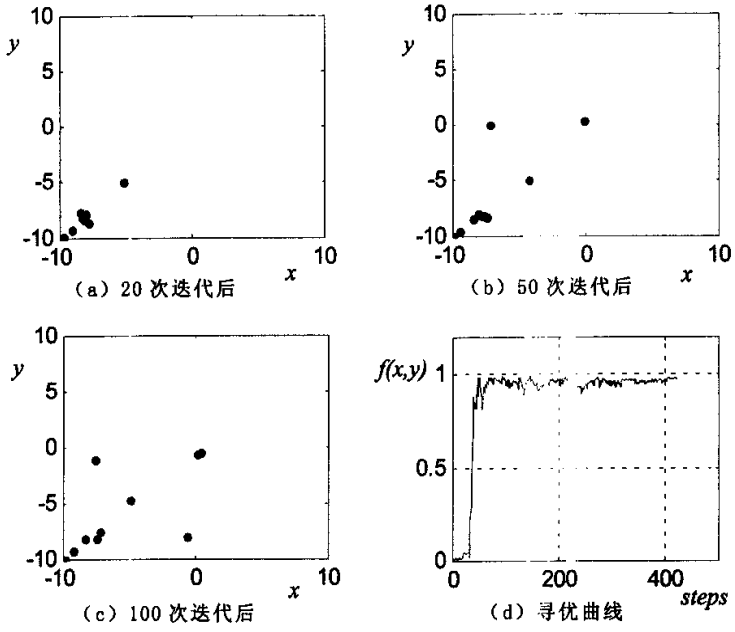



图 2.3 具有多个局部极值的非线性目标函数

下面的仿真试验中，*AF* 的数量为 10 个，为了检验其寻优的效率，令它们的初始位置都处于距全局极值最远的（-10，10）处，其感知距离 *Visual=2.5*，每次移动的步长 *Step=0.3*，拥挤度因子 $\delta=0.618$。

图 2.4 为 *AF* 只执行聚群行为的仿真结果，如图 2.4（a）中，*AF* 迅速聚集在最近的局部极值点附近，图 2.4（b）中已经有 *AF* 到达了全局极值的附近，图 2.4（c）中有更多的 *AF* 到达了全局极值点的附近，且有少数 *AF* 聚集到了其他的局部极值点附近，从寻优曲线可以看出，*AF* 能较快的跳出局部极值点，从而较快的找到全局极值。




（a）20 次迭代后　（b）50 次迭代后　（c）100 次迭代后　（d）寻优曲线

图 2.4　聚群行为仿真结果

图 2.5 为 *AF* 只执行追尾行为的仿真结果，如图 2.5（a）中，*AF* 迅速聚集在最近的局部极值点附近，图 2.5（b）中 *AF* 又快速向附近的极值点聚集，图 2.5（c）中有更多的 *AF* 聚集到了附近的局部极值点的附近，并有 *AF* 开始到达了全局极值点附近，从寻优曲线可以看出，*AF* 能较快的聚集到某个极值点，但可能会陷于其中。

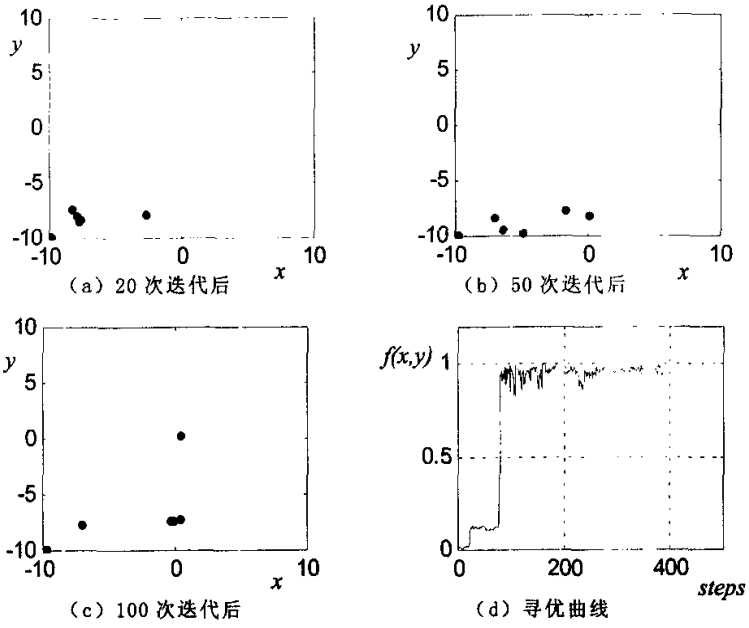



图 2.5 追尾行为仿真结果

图 2.6 为完整的人工鱼群算法的仿真结果，从图中可以看出，*AF* 能迅速向全局极值点附近聚集，从寻优曲线可以看出，*AF* 能很快的搜索到全局极值点并较快的稳定在满意解域内。

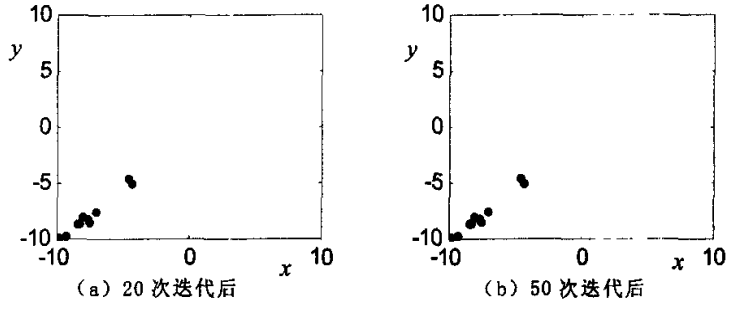
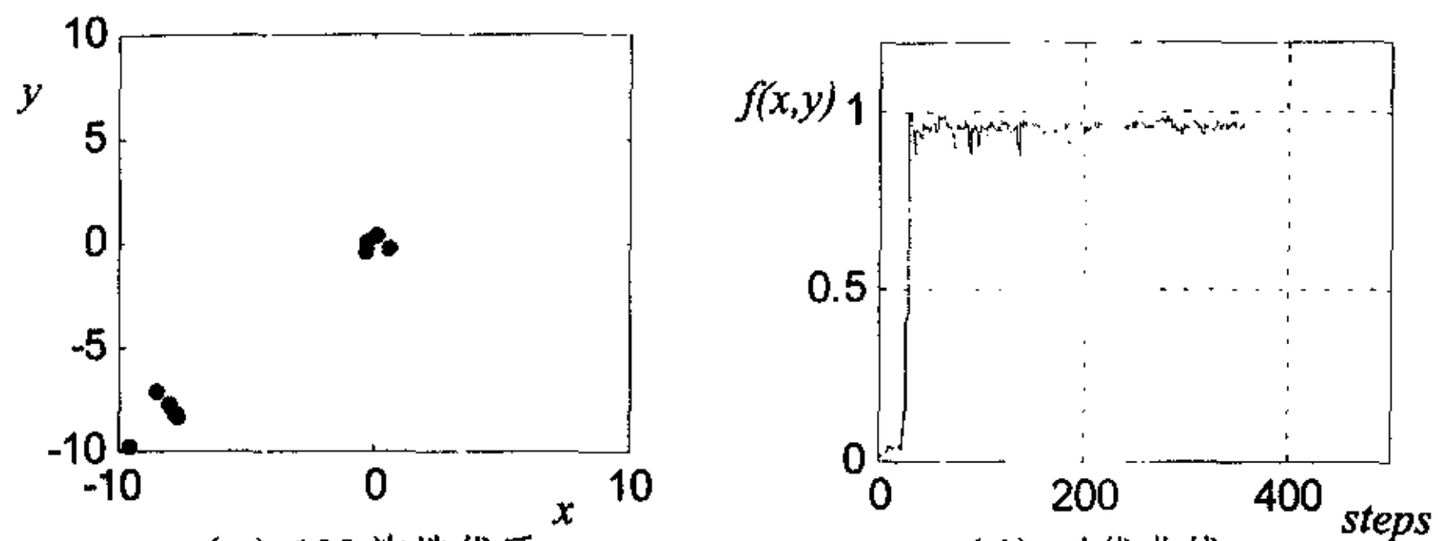



图 2.6 鱼群算法仿真结果




（c）100 次迭代后　　（d）寻优曲线

图 2.6(续)　鱼群算法仿真结果

由上可得出如下结论：

- 聚群行为能够很好的跳出局部极值，并尽可能的搜索到其他的极值，最终搜索到全局极值。
- 追尾行为有助于快速的向某个极值方向前进，加快寻优的速度，并防止 *AF* 在局部振荡而停滞不前。
- 鱼群算法在对以上两种行为进行评价后，自动选择合适的行为，从而形成了一种高效快速的寻优策略。

## 2.6 算法对照

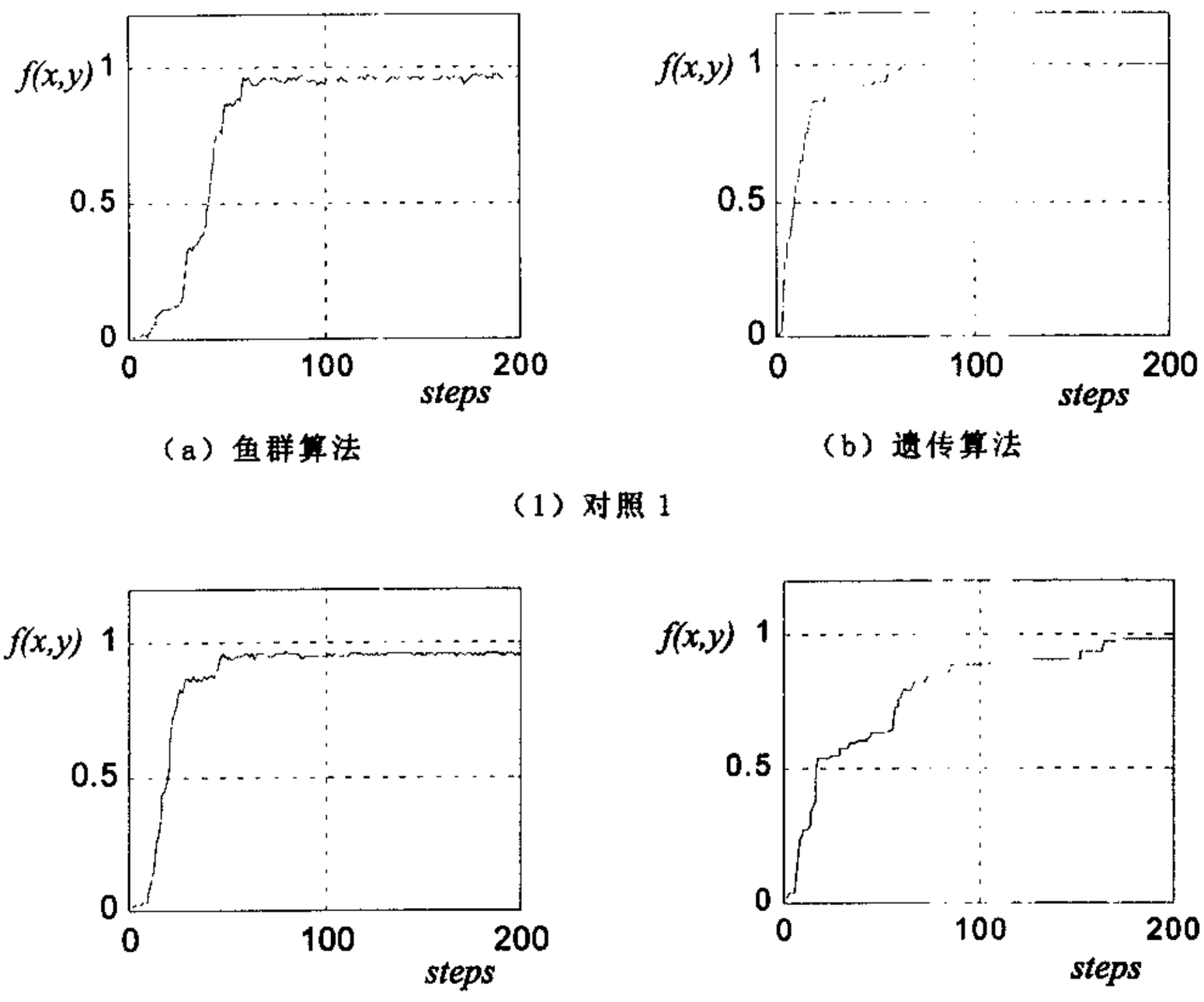
遗传算法作为具有全局优化能力的代表算法之一，受到了长期的、广泛的关注，作者将鱼群算法与之进行了比较。

由于这两种算法都存在一定的随机性，因此，在对照中分别进行了 10 组实验，最后将结果求平均，横坐标为各算法中迭代的次数。

在本实验中，鱼群算法的参数同上；遗传算法中，编码长度为 10，交叉位数为 4，变异位数为 2。图 2.7 展示了实验的结果；为了增强可比性，在对照 1 中，鱼群算法的人工鱼个体数为 10，遗传算法的种群数为 10；在对照 2 中，鱼群算法的人工鱼个体数为 50，遗传算法的种群数为 50。

可见，鱼群算法中当人工鱼个体的数目较少时，还不能体现出它的优

势，当然，对于遗传算法来说，种群数较少时也具有容易陷入局部极值和早熟的可能；当人工鱼个体的数目增加时，鱼群算法的收敛速度得以提高，而遗传算法则由于种群数的增多减缓了进化的速度，可见鱼群算法中蕴含着集群智能的优势。




（a）鱼群算法　　（b）遗传算法

（1）对照 1

（c）鱼群算法　　（d）遗传算法

（2）对照 2

图 2.7 鱼群算法与遗传算法的比较

## 2.7 人工鱼群算法全局收敛的基础

对于一种算法，其收敛性往往是人们所首要关心的问题。在人工鱼群算法中，人工鱼的觅食行为奠定了算法收敛的基础，聚群行为增强了算法收敛的稳定性和全局性，追尾行为则增强了算法收敛的快速性和全局性，其行为评价也对算法收敛的速度和稳定性提供了保障。总的来说，算法中对各参数的取值范围还是很宽容的，并且对算法的初值也基本无要求。

算法中，使人工鱼逃逸局部极值实现全局寻优的因素主要有以下几点：

- 觅食行为中 *try_number* 的次数较少时，为人工鱼提供了随机游动的机会，从而能跳出局部极值的邻域；
- 随机步长的采用，使得在前往局部极值的途中，有可能转而游向全局极值，当然，其相反的一面也会发生的，就是在去往全局极值的途中，可能转而游向局部极值，这对一个个体当然不好判定他的祸福，但对于一个群体来说，好的一面往往会具有更大的机率；
- 拥挤度因子的引入限制了聚群的规模，只有较优的地方才能聚集更多的人工鱼，使得人工鱼能够更广泛的寻优。
- 聚群行为能够促使少数陷于局部极值的人工鱼向多数趋向全局极值的人工鱼方向聚集，从而逃离局部极值；
- 追尾行为加快了人工鱼向更优状态的游动，同时也能促使陷于局部极值的人工鱼向趋向全局极值的更优人工鱼方向的追随而逃离局部极值域。

## 2.8 各参数对收敛性能的影响分析

由于算法存在一定的随机性，在相同参数下，收敛过程和结果也存在一定的差异，所以在以下的讨论中，将针对每一种参数连续多次进行全局寻优收敛实验作为一组数据，然后对多组数据进行统计分析([浙江大学 1980])，从而确定各参数的性质。通常数据的均值(Mean)表征本组数据的优劣，其标准误差(SE)则可以反映该组数据的稳定性。分析软件使用了 SPSS for windows 统计分析软件。

不能求取高精度的数值解是人工鱼群算法的一大缺陷，它只能快速找到全局极值的邻域，在以下的仿真过程中，以找到全局最优值 1.0 的±1%的邻域为收敛准则。

在以下分析算法收敛性能的过程中，我们将基于如下所示的二维 *sinc* 函数来进行：

$$Max f(x,y)=\sin\sqrt{(x-10)^2+(y-10)^2}\Big/\sqrt{(x-10)^2+(y-10)^2} \qquad (2.2)$$

$$s.t.\quad 0\le x\le 20,\ 0\le y\le 20$$

由图 2.8 可以看出，该函数的全局极大值的周围被无穷个极小值（波谷）所包围，再外围是无穷个局部极大值（波峰），在最远端的四个角上又是处于局部极大值。使用该函数作为实验函数应该具有一定的代表性。现有的一些寻优算法在本问题的寻优过程中由于初值等原因往往不能 100%的找到全局最优解，并且存在收敛速度等方面的问题。

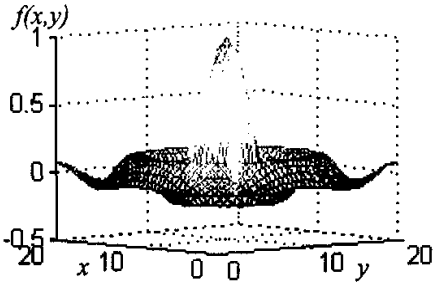


图 2.8 二维 Sinc 函数

对于该函数，应用人工鱼群算法寻优时，从任何初始值出发，都能收敛到全局最优值，并且收敛时间最短为几十毫秒，仿真的硬件环境为 Intel Celeron 400 CPU，软件环境为 Microsoft Windows 98，编程语言为 Microsoft Visual C++。

### 2.8.1 视野和步长

在觅食行为中，人工鱼的个体总是尝试向更优的方向前进，这就奠定了算法收敛的基础。如图 2.9 所示，人工鱼随机的巡视在其视野范围中某点的状态 $X_i$，如果发现比当前状态 $X$ 更好，那么，它就向状态 $X_i$ 的方向前进一步到达状态 $X_{next}$；如果状态 $X_i$ 并不比状态 $X$ 好，那么它继续随机巡视视野范围内的状态，如果巡视次数达

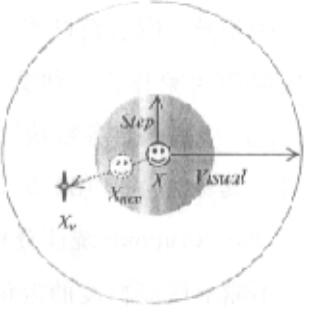
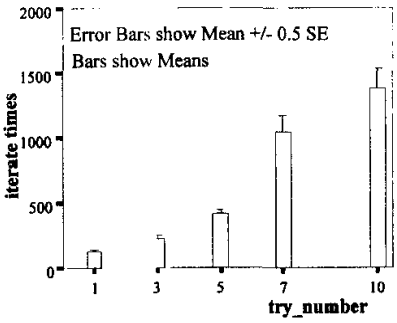


图 2.9 人工鱼的视野和移动步长

到一定的次数（*try_number*）仍旧没有找到更优的状态，那么就做随机的游动。

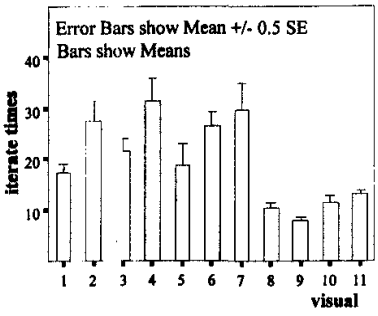
由于每次巡视的视点都是随机的，所以不能保证每一次觅食行为都是向着更优的方向前进的，这在一定程度上减缓了收敛的速度，但是从另一方面看，这又有助于人工鱼摆脱局部极值的诱惑，从而去寻找全局极值。为了测试其摆脱陷入局部极值的能力，在实验中令所有的人工鱼的初始状态都位于（0，0）处，这对算法能否全局收敛无疑是一个严峻的考验。如图2.10是*try_number*在不同值时各进行 10 次仿真的统计结果。可见，*try_number* 的次数越多，人工鱼摆脱局部极值的能力就会越弱，当然，对于局部极值不是很突出的优化问题，增加 *try_number* 的次数可以减少人工鱼的随机游动而提高收敛的效率。在使用中发现，对于普通的非线性目标函数，*try_number* 为 1 时的收敛效果也是比较满意的。




*AF number=30 visual=10 step=3 delta=0.618*

图 2.10 *try_number* 参数仿真结果统计

图 2.11 是参数 *Visual* 在不同取值下各 10 次仿真结果的统计图，人工鱼的初始状态为随机的。由于视野对算法中各行为都有较大的影响，因此，它的变化对收敛性能的影响也是比较复杂的。当视野范围较小时，人工鱼的觅食行为和随机游动比较突出；视野范围较大时，人工鱼的追尾行为和聚群行为将变得比较突出。总体来看，视野越大，越容易使人工鱼发现全局极值并收敛。

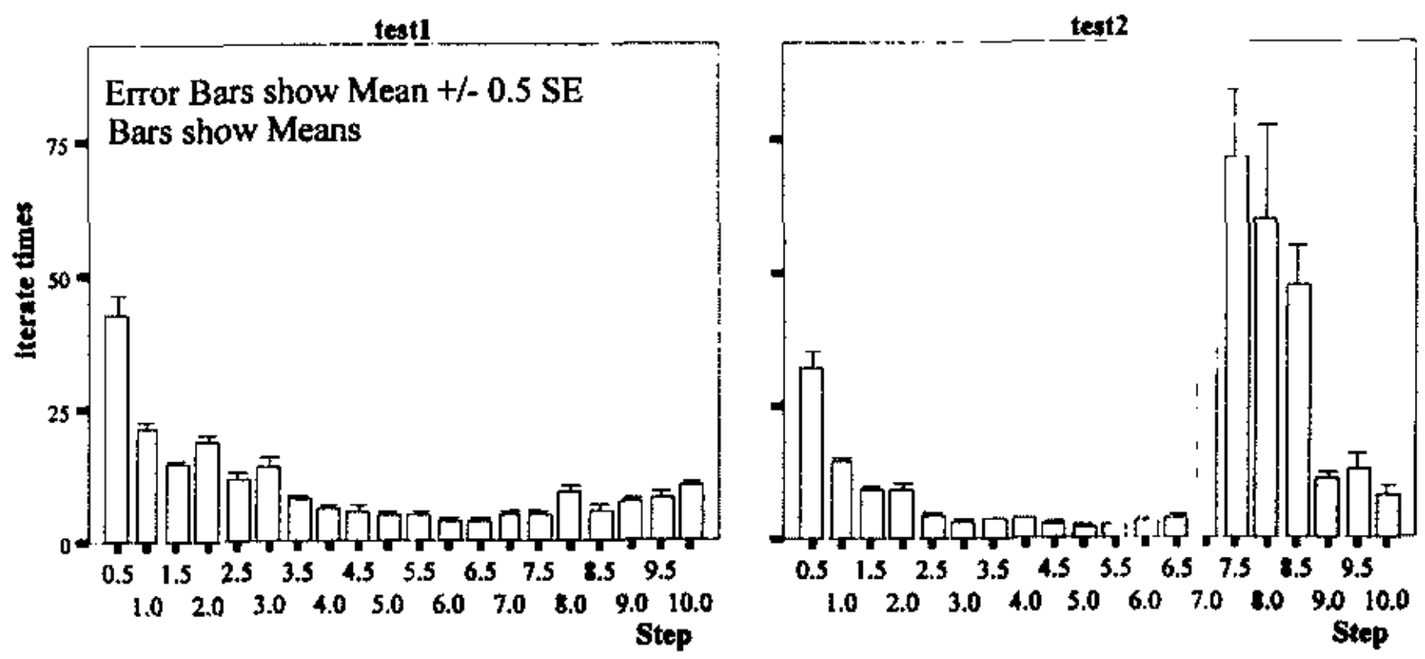



*AF number=30 step=3 delta=0.618*

图 2.11 *AF visual* 参数仿真结果统计

图 2.12 是参数 *Step* 在不同取值下各 10 次仿真结果的统计图，人工鱼的初始状态为随机的。在算法的描述中，我们使用了随机步长的方式，即移动步长为 *Random(Step)*（在 *0~Step* 之间随机取值）。在图中，test1 是

随机步长的统计结果，test2 是采用固定步长的统计结果。可以看出，随着步长的增加，收敛的速度得到了一定的加速，但在超过一定的范围后，又使得收敛速度减缓，尤其在 test2 中，步长过大时，有时会出现振荡现象而大大影响收敛的速度。还可以看出，采用随机步长的方式在一定程度上防止了振荡现象的发生，并使得该参数的敏感度大大降低了，但最快的收敛速度还是最优固定步长的收敛速度，所以，对于特定的优化问题，我们可以考虑采用合适的固定步长或变尺度方法来提高收敛的速度。




*AF number=30 visual=10 delta=0.618*

图 2.12 *step* 参数仿真结果统计

### 2.8.2 拥挤度因子 $\delta$（*delta*）

拥挤度因子用来限制人工鱼群聚集的规模，在较优状态的邻域内希望聚集较多的人工鱼，而次优状态的邻域内希望聚集较少的人工鱼或不聚集人工鱼。其选取规则通常如下：

在求极大的问题中：

$$\delta = 1/(\alpha n_{\max}) \quad (0 < \alpha < 1)$$

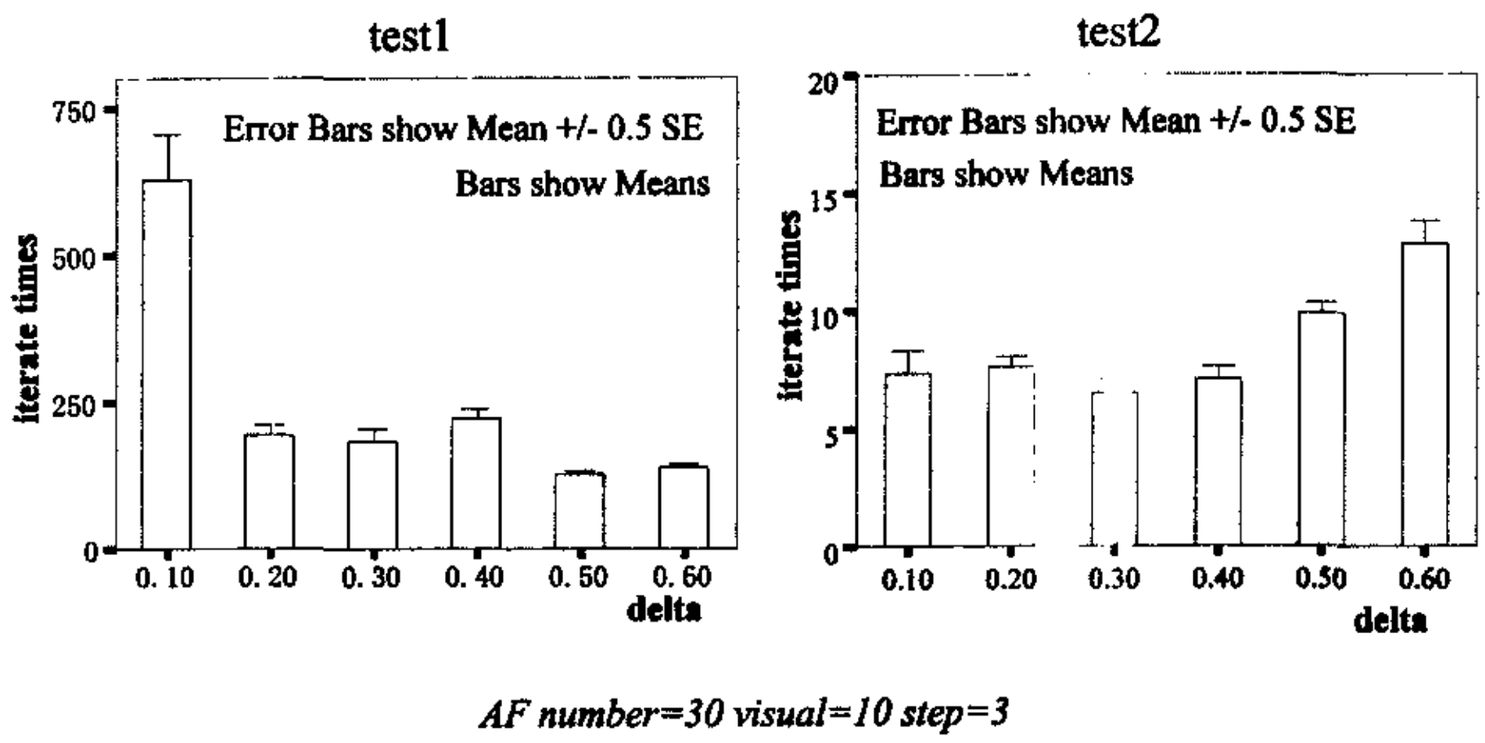
式中，$\alpha$ 为极值接近水平，$n_{\max}$ 为期望在该邻域内聚集的最大人工鱼数目。例如，如果希望在接近极值 90%水平的邻域内不会有超过 10 个人工鱼聚集，那么取 $\delta = 1/(0.9\times10) \approx 0.11$。这样，如果 $Y_c/(Y_i n_f) < \delta$，人工鱼就认为 $Y_c$ 状态过于拥挤，其中 $Y_i$ 为人工鱼自身状态的值，$Y_c$ 为人工鱼所感知的某状态的值，$n_f$ 为周围伙伴的数目。

在求极小的问题中：

$\delta = \alpha n_{\max} \quad (0 < \alpha < 1)$

式中，$\alpha$为极值接近水平，$n_{\max}$为期望在该邻域内聚集的最大人工鱼数目。例如，如果希望在接近极值 90%水平的邻域内不会有超过 10 个人工鱼聚集，那么取$\delta = 0.9 \times 10 = 9$。这样，如果$Y_c n_f / Y_i > \delta$，人工鱼就认为$Y_c$状态过于拥挤，其中$Y_i$为人工鱼自身状态的值，$Y_c$为人工鱼所感知的某状态的值，$n_f$为周围伙伴的数目。

由于本试验函数是极大值问题，所以，$\delta$越大，表明允许的拥挤程度越小，如图 2.13 所示，在 test1 中，各人工鱼的初始状态为（0，0），可以看出，*delta* 越大，人工鱼摆脱局部极值的能力越强；test2 中的初始状态为随机，可以看出，*delta* 越大，收敛的速度会有些减缓，这主要因为人工鱼在逼近极值的同时，会因避免过分拥挤而随机走开，或者受其它人工鱼的排斥作用，不能精确逼近极值点。可见，虽然$\delta$的引入避免了人工鱼过度拥挤而陷入局部极值，另一方面，该参数会使得位于极值点附近的人工鱼之间存在相互排斥的影响，而难以向极值点精确逼近，所以，对于某些局部极值不是很严重的具体问题，可以忽略拥挤的因素，从而在简化算法的同时也加快了算法的收敛速度和提高结果的精确程度。

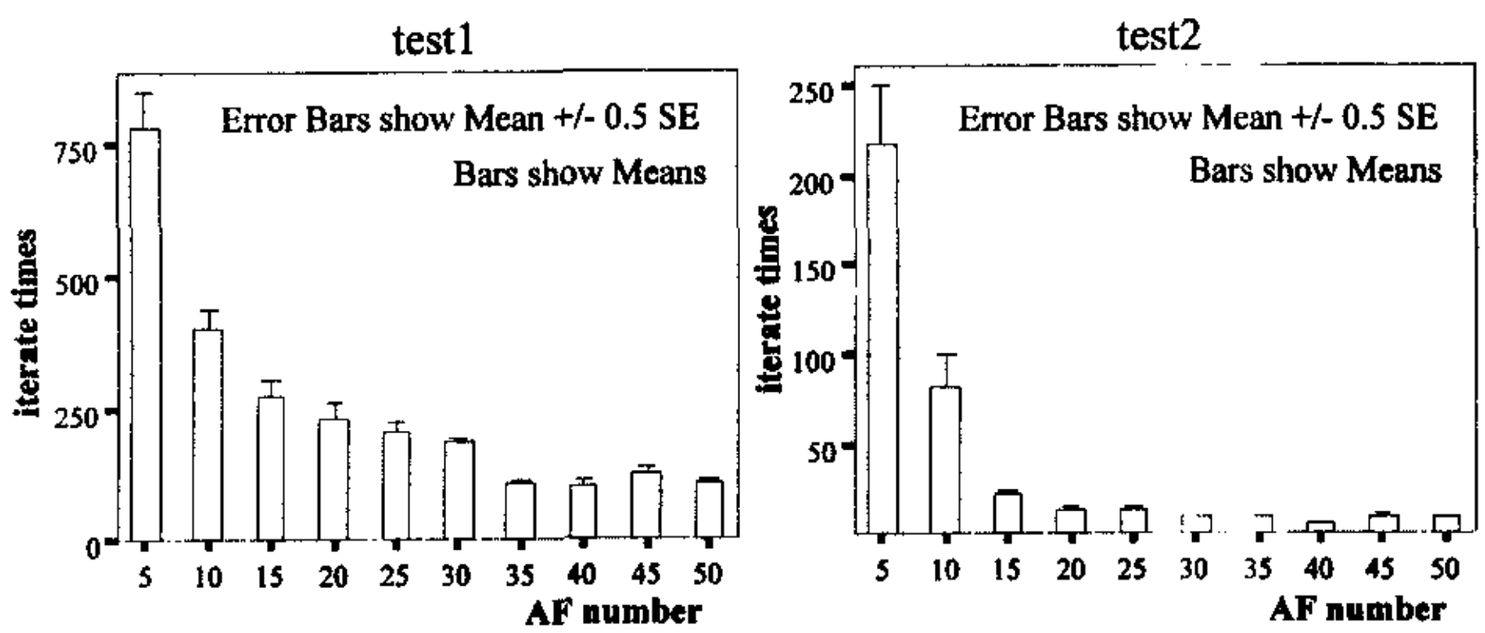



*AF number=30 visual=10 step=3*

图 2.13 *delta* 参数仿真结果统计

## 2.8.3 人工鱼的个体数目

人工鱼群算法是集群智能的一个应用，这也是算法具备自己的一些特性的根本原因，其中最主要的应该是群体的概念，所以，在算法中，由一个人工鱼个体单独迭代 100 次，和 10 条鱼一起迭代 10 次的效果是迥然不同的。如图 2.14 所示，test1 中人工鱼的初始状态为（0，0），test2 中的初始状态为随机状态，每个参数值做 10 次仿真作为统计样本。可以看出，人工鱼的数目越多，跳出局部极值的能力越强，同时，收敛的速度也越快（从迭代次数来看），当然，付出的代价就是算法每次迭代的计算量也越大了，因此，在使用过程中，满足稳定收敛的前提下，应尽可能的减少个体的数目。

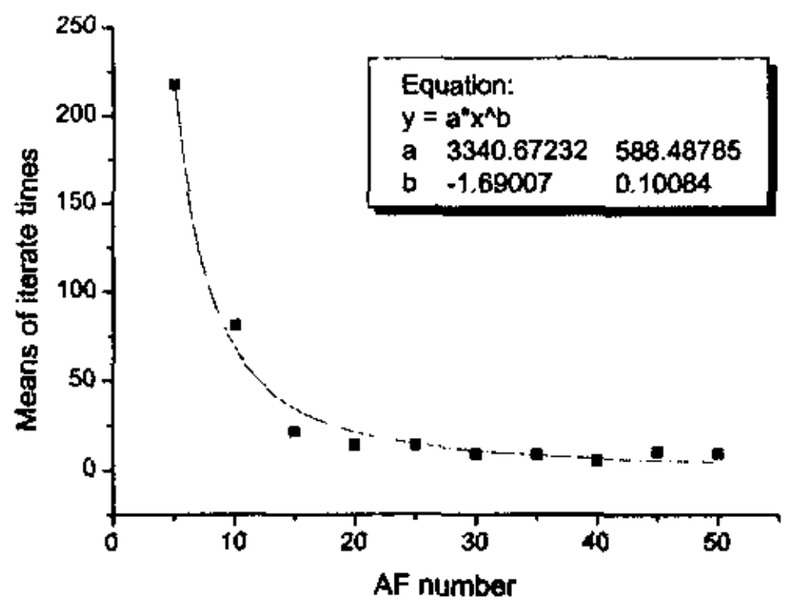


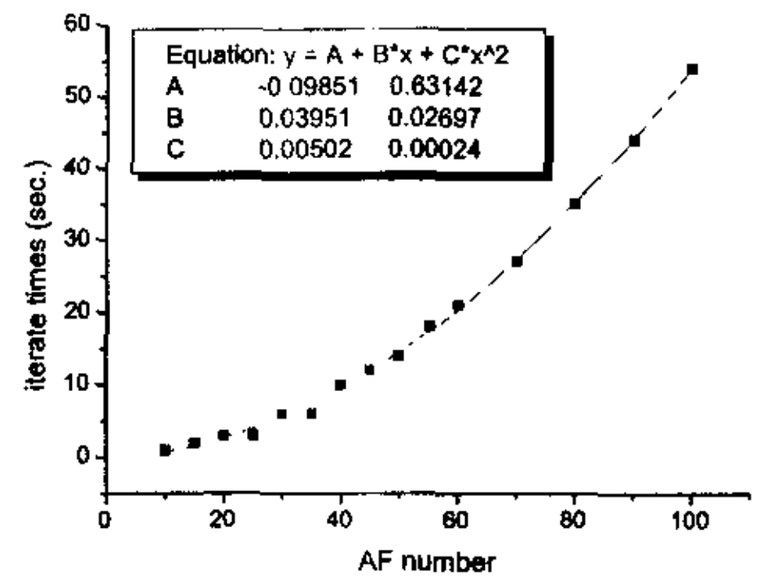

*AF visual=10 step=3 delta=0.618*

图 2.14 不同人工鱼数目的仿真结果统计

### 2.8.4 参数对计算时间的影响

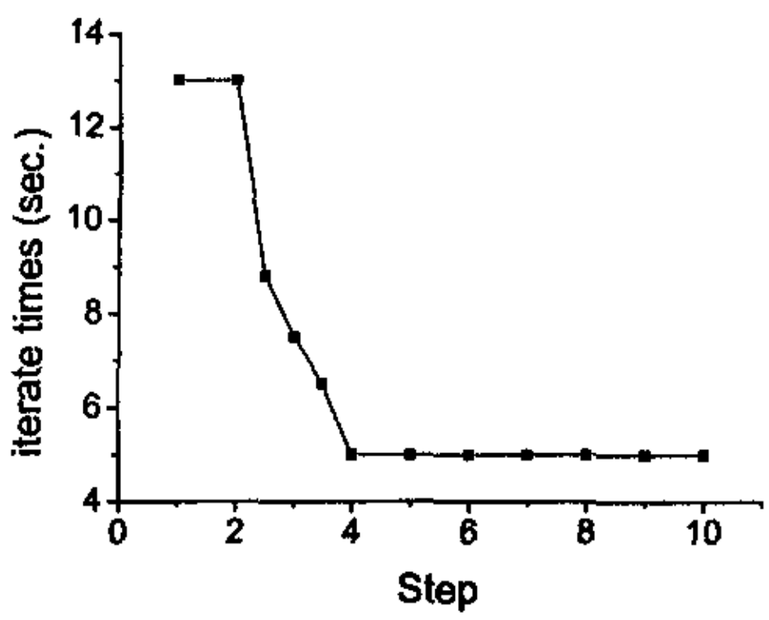
图 2.15 中，(b)～(e)是各参数在不同取值下的每 1000 次迭代的计算时间。可以看出，参数 *Step*、*Visual*、*delta* 的影响都是有限的，而人工鱼数目的影响通过回归分析可以看出是呈二次函数上升的；图 2.15(a)中，对人工鱼的数目与收敛迭代次数关系的回归分析可以看出，*AF number* 的增加对迭代次数的减少是呈幂指数下降的，这或许能补偿一下由于人工鱼数目增加而造成的计算量增加的问题，因此，合理选择人工鱼的数目是提高算法效率的关键。这一规律是否具有一般性还有待进一步研究，如与目标函数的性质有无关系、与算法实现的代码优化有无关系等。




*Visual=10, Step=3, delta=0.618*

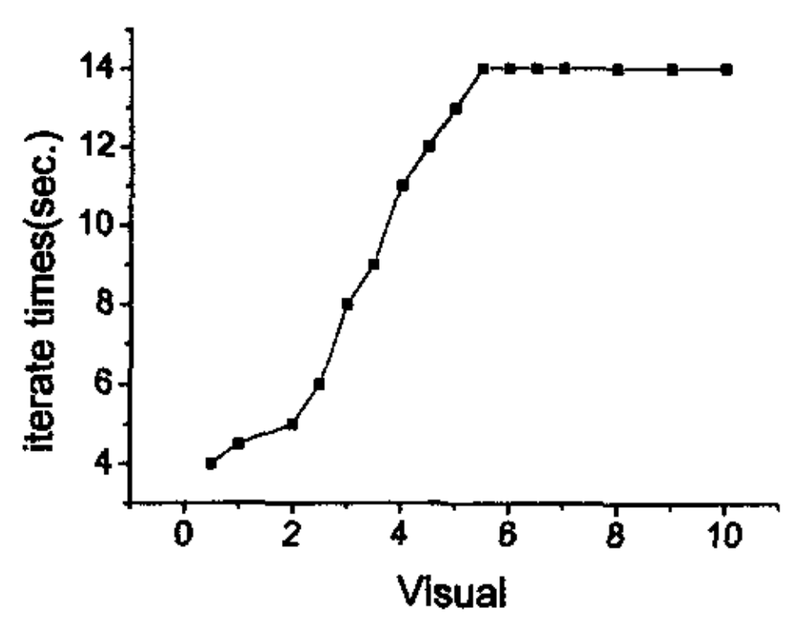
(a)




*Visual=3, AF number=50, delta=0.618*

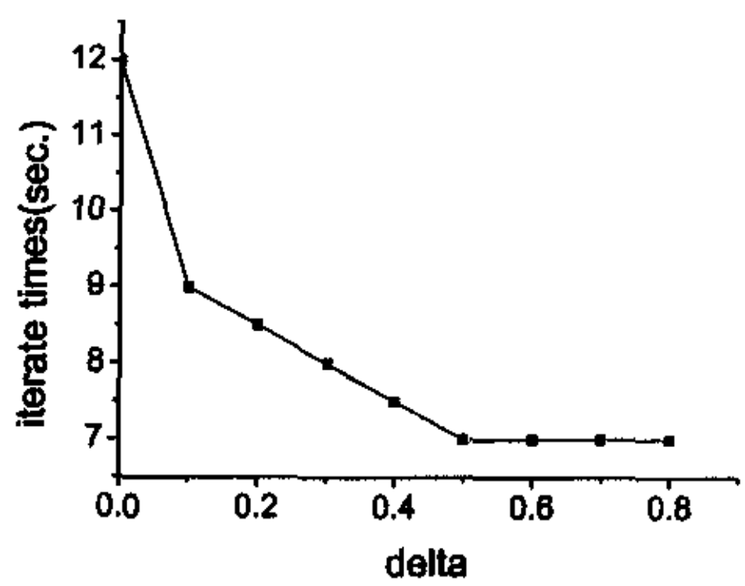
(c)

(b)




*AF number=50, Step=3, delta=0.618*

(d)




*Visual=3, Step=3, AF number=50*

(e)

图 2.15 各参数对计算时间的影响

## 2.9 结束语

人工鱼群算法采用了自下而上的设计思路，从 *AF* 的个体行为出发，达到了最终结果的突现，为优化问题的解决提供了一条新的思路。总结以上的研究，可以得出鱼群算法的以下特点：

- 并行性：多个 *AF* 并行的进行搜索；
- 简单性：算法中仅使用了目标问题的函数值；
- 全局性：算法具有很强的跳出局部极值的能力；
- 快速性：算法中虽然有一定的随机因素，但总体是在步步向最优搜索；
- 跟踪性：随着工作状况或其他因素的变更造成的极值点的漂移，本算法具有快速跟踪变化的能力。

由以上的研究可以看出，本算法仅仅获取的是系统的满意解域，对于精确解的获取还需借助于一些策略，如分阶段寻优和变参数寻优等。

由以上讨论可以看出，虽然各个参数有各自的取值规则，但总体来说，整个算法对各参数的取值范围的容许度还是相当大的，如对实验函数(2.2)来说，人工鱼个体数选 10~50，*Step* 取 1~10，*Visual* 取 1~11，取 $\delta>0.1$，均能快速稳定的收敛，且在上千次的实验中全局收敛的次数为 100%；还可以看出，算法对初值基本无要求，如实验函数中，即使位于全局最优点最远端的某个局部极值中也能较快速的全局收敛。所以，算法采用自下而上的设计模式，各个体行为之间具有相对的独立性和互补性，使得整个算法具有较稳定的收敛性能。

# 第三章 组合优化问题的人工鱼群算法应用

## 3.1 引言

组合优化问题涉及经济管理、交通运输、通信网络等领域，主要是寻找离散事件的最优编排、分组、次序或筛选等，是运筹学中的一个经典而重要的分支。一些组合优化问题随着规模的扩大，往往造成计算规模的迅速增加，被称为 NP、NP-C 和 NP-hard 复杂性问题。如装箱判定问题，旅行商问题（TSP）等（[邢文训 & 谢金星 1999]）。对于此类问题，大都采用寻找近似算法或启发式算法来寻求一个满意解或目前最好解([Reeves 1993], [Polya 1948])。下面，我们用人工鱼群算法对此类问题进行了求解，并以 TSP 问题为例进行了仿真实验，发现它依然保持了快速收敛的特性。

## 3.2 基本概念

### 3.2.1 组合优化问题

组合优化问题的数学模型（[邢文训 & 谢金星 1999]）描述为：

$$\min f(X)$$
$$s.t.\ g(X) \geq 0$$
$$X \in D$$

其中，$f(X)$为目标函数，$g(X)$为约束函数，$X$为决策变量，$D$表示有限个元素组成的集合。通常，一个组合优化问题可用三参数$(D,F,f)$表示，其中 $D$ 表示决策变量的定义域，$F$ 表示可行解域$F=\{X \mid X \in D, g(X) \geq 0\}$，$f$ 表示目标函数，满足$f(X^*)=\min\{f(X) \mid X \in F\}$的可行解$X^*$称为问题的最优解。

### 3.2.2 人工鱼群算法中的距离和邻域

在组合优化问题的解决中，距离、邻域等概念依然是很重要的，但它们的形式与距离空间中的定义有着很大的不同([Ahuja & Ergun 2002], [Ahuja & Orlin 2000], [秦裕瑗 1994])，以下是在人工鱼群算法中应用的一些定义的分析和描述：

**定义 1** 对于组合优化问题$(D,F,f)$，决策变量$X_1$和$X_2$之间的距离表示如下：

$$\text{Distance}(X_1,X_2)=|X_1-X_2|+|X_2-X_1|$$

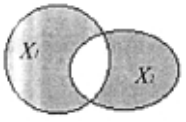



图 3.1 距离示意图

表示不同时属于$X_1$和$X_2$的元素的个数，如图 3.1 中阴影部分的元素个数。

**定义 2** 对于组合优化问题$(D,F,f)$，如下

$N(X,k)=\{X'|\text{Distance}(X,X')<k,X'\in D\}$

称为$X$的$k$-距离邻域，$X'\in N(X,k)$称为$X$的一个邻居。

**定义 3** 对于组合优化问题$(D,F,f)$，

$$\text{Center}(X_1,X_2,\cdots,X_m)=\bigcup_{i=1}^{m}\bigcup_{\substack{j=1\\ j\neq i}}^{m}(X_i\cap X_j)$$

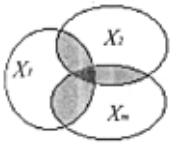



图 3.2 中心示意图

称为决策变量$X_1,X_2,\cdots,X_m$的中心，如图 3.2 中深灰色阴影中的元素，灰色阴影中的元素作为次中心考虑。

## 3.3 人工鱼群算法描述

算法的原理与基本鱼群算法一致，在描述和实现中有些地方进行了适当的调整。如为了算法的简便易行，最优值的获取方式采用了跟踪记录最优个体状态的方法，因此，引入了公告板（*AF_bulletin*）用来记录最优状

态；为了解决带约束的优化问题，引入了约束行为（*AF_subject*），扩展了算法的适用范围；为了更有效的增强算法的效能，用移动策略（*AF_movestrategy*）替代了行为评价。

### 3.3.1 人工鱼(AF, Artificial Fish)模型

人工鱼的模型依然采用面向对象的方法构造如下：

```
class Artificial_fish
{
Variables:
    float AF_X[n];              //AF's position
    float AF_visual;            //the visual distance of AF
    float AF_step;              //the moving distance of each step
    float AF_delta;             //the factor of congestion
    int AF_trynumber;           //the times that AF tries each step
Functions:
    float AF_foodconsistence(); //the food consistence of AF's current position
    float AF_subject();         //the subject to AF
    float AF_bulletin();        //to check the bulletin board
    float AF_follow();          //the behavior of following
    float AF_prey();            //the behavior of preying
    float AF_swarm();           //the behavior of swarming
    void AF_randmove();         /the behavior of random moving
    int AF_movestrategy();      //the strategy of moving
    void AF_init();             //to initialize the AF
};
```

### 3.3.2 符号定义

一些符号的定义说明如下：

决策变量 $X$ 表示人工鱼的状态 *AF_X*；目标函数值 $Y=f(X)$ 表示人工鱼当前状态的食物浓度 *AF_foodconsistence*； $d_{i,j}=\mathrm{Distance}(X_i,X_j)$ 表示人工

鱼 $x_i$ 和人工鱼 $x_j$ 之间的距离；$\delta$ 表示拥挤度因子 *AF_delta*；*trynumber* 表示人工鱼每次移动最大的试探次数；*Visual* 表示人工鱼的视野范围 *AF_visual*；*Step* 表示人工鱼的移动步长 *AF_step*；*AF_Number* 表示参与寻优的人工鱼的数目。以下的介绍都以求极大问题为例进行说明。

### 3.3.3 人工鱼的行为描述

#### 3.3.3.1 觅食行为

设人工鱼当前状态为 $x_i$，在其视野范围内随机选择一个状态 $x_j$，如果 $Y_i < Y_j$，则向该方向前进一步；反之，再重新随机选择状态 $X_j$，判断是否满足前进条件；试探 *trynumber* 次后，如果仍不满足前进条件，则执行其他行为（如随机移动行为）。

伪代码描述如下：

```
float Artificial_fish::AF_prey()
{
    for (i=0;i<try_number;i++)
    {
        X_j = Random(N(X_i,Visual));
        if  (Y_i > Y_j)
            { X_i|next = X_j; return AF_foodconsistence(X_i|next); }
    }
X_i|next = X_j;
return AF_foodconsistence(X_i|next);
}
```

$Random(N(X_i,Visual))$ 表示在 $x_i$ 的 *Visual*-距离的邻域内随机取一个邻居。

#### 3.3.3.2 聚群行为

设人工鱼当前状态为 $X_i$，探索其邻域的伙伴数目 $n_f$，如果

$\frac{n_f}{N} < \delta,(0<\delta<1)$，表明伙伴中心有较多的食物并且不太拥挤，并且 $Y_i < Y_c$，则向中心位置 $X_c$ 前进一步；否则执行其他行为（如觅食行为）。

伪代码描述如下：

```
float Artificial_fish::AF_swarm()
{
        n_f =| N(X_i,Visual) | ;
        X_c = Center(N(X_i,Visual))
        if  (n_f/N < δ && Y_i < Y_c)
            X_i|next = X_c ;
        else
            AF_prey();
return AF_foodconsistence( X_i|next );
}
```

3.3.3.3 追尾行为

设人工鱼当前状态为 $X_i$，探索其邻域内状态最优的邻居 $X_{\max}$，如果 $Y_i < Y_{\max}$，并且 $X_{\max}$ 的邻域内伙伴的数目 $n_f$ 满足 $\frac{n_f}{N} < \delta,(0<\delta<1)$，表明 $X_{\max}$ 的附近有较多的食物并且不太拥挤，则向 $X_{\max}$ 的位置前进一步；否则执行觅食行为。

伪代码描述如下：

```
float AF_type_A::AF_follow()
{
        Y_max = MAX(f(X_max), X_max ∈ N(X_i,Visual) ;
        n_f =| N(X_max,Visual) | ;
        if( Y_i < Y_max && n_f/N < δ )
            X_i|next = X_max ;
        else
```

*AF_prey();*

*return AF_foodconsistence(* $X_{i|next}$ *);*

*}*

#### 3.3.3.4 约束行为

在寻优过程中，由于聚群行为、随机行为等操作的作用，容易使得人工鱼的状态变得不可行，这时就需要加入相应的约束来对其进行规整化，使它们由无效状态或不可行状态转变成可行的。

#### 3.3.3.5 公告板

公告板用来记录最优人工鱼个体的状态。各人工鱼个体在寻优过程中，每次行动完毕就检验自身状态与公告板的状态，如果自身状态优于公告板状态，就将公告板的状态改写为自身状态，这样就使公告板记录下历史最优的状态。

#### 3.3.3.6 移动策略

移动策略是原行为评价的一种延伸，可以依旧采用原行为评价的模式，也可以采取一定的行动策略，如先进行追尾行为，如果没有进步再进行觅食行为，如果还没有进步则进行聚群行为，如果依然没有进步就进行随机移动行为。

### 3.3.4 小结

有了底层的人工鱼的模型，算法的展开就在一群人工鱼的自主活动中开始了。整个算法没有高层的指挥者，也不需要关于命题的先验知识的启发，每条人工鱼就按照自己的规则游动着，命题的满意解就这样在公告板上显示出来。

算法整体表现出快速向极值域收敛的特性，随机移动行为的存在使得寻优活动更加全面的展开，从而可能到达全局极值域内。

在应用中，本算法可以针对问题的性质进行灵活的简化，也可以与其他的有效算法进行融合，以进一步提高运算速度和提高寻优结果的精度。

## 3.4 计算实例研究

旅行商问题（TSP, *traveling salesman problem*）是一个典型的组合优化问题，并且是 NP-hard 问题([马良 2000])，下面就以对称距离的 TSP 问题为例来验证算法的性能。

旅行商问题就是一个商人欲到 $n$ 个城市推销商品，每两个城市之间的距离为 $d_{ij}$，如何选择一条道路使得商人每个城市走一遍后回到起点所走过的路径最短。它的数学模型描述为([邢文训 & 谢金星 1999])：

$$\min \sum_{i \neq j} d_{ij} x_{ij}$$

$$s.t. \sum_{j=1}^{n} x_{ij} = 1, i = 1,2,\cdots,n,$$

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1,2,\cdots,n,$$

$$\sum_{i,j \in s} x_{ij} \leq |s| - 1, 2 \leq |s| \leq n-2, s \subset \{1,2,\cdots,n\},$$

$$x_{ij} \in \{0,1\}, i, j = 1,2,\cdots,n, i \neq j$$

其中，$d_{ij}=d_{ji}$ 时，称为对称距离 TSP，否则称非对称距离 TSP；

在 TSP 问题中，两个决策变量 $A=\{a_1,a_2,\cdots,a_n\}$ 和 $B=\{b_1,b_2,\cdots,b_n\}$ 之间的距离表示如下：

$$\text{distance}(A,B) = \sum_{i=1}^{n} sign(|a_i - b_i|) \qquad \text{其中：} \ sign(x) = \begin{cases} 0, & x=0 \\ 1, & x>0 \\ -1, & x<0 \end{cases}$$

那么他们的 $k$-距离邻域可以表示为:

$$N(X,k) = \{X' \mid \text{distance}(X',X) < k, X' \in D\}$$

$X_1, X_2, \cdots, X_m$ 的中心为 $\text{Center}(X_1, X_2, \cdots, X_m) = \underset{i=1,\cdots,m}{Most}(x_i^1, x_i^2, \cdots, x_i^m)$，其中 *Most* 操作符表示取其中多数共有或最相近的值。

以下的 TSP 问题的数据来自 http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/；硬件环境是 Intel Celeron 400，软件环境是 Microsoft windows98 + Visual C++ 6；图 3.3 是对 Ulysses22 城市（22 个城市）TSP 问题的寻优曲线，算法的参数为 *AF_Number* = 9，*Visual*=16，*trynumber*=500，其它参数被简化掉了，其最好解是 7013，实际寻优的最好解是 7013，200 步迭代的计算时间约为 11 秒；图 3.4 是对 Att48 城市（48 个城市）TSP 问题的寻优曲线，算法的参数为 *AF_Number*=9，*Visual*=45，*trynumber*=300，其它参数被简化掉了，其中最好解是 10628，实际寻优的最好解是 10886，500 次迭代的计算时间约为 70 秒。

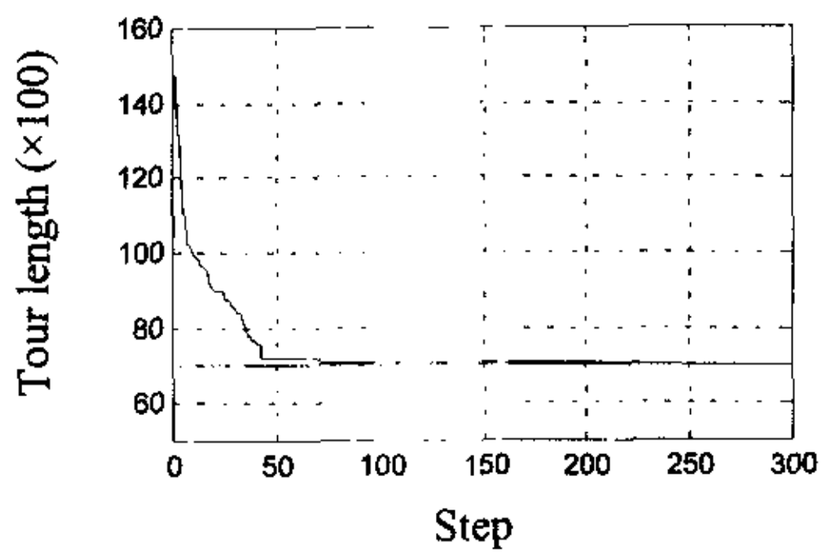


图 3.3 Ulysses 22 城市 TSP 问题的寻优曲线

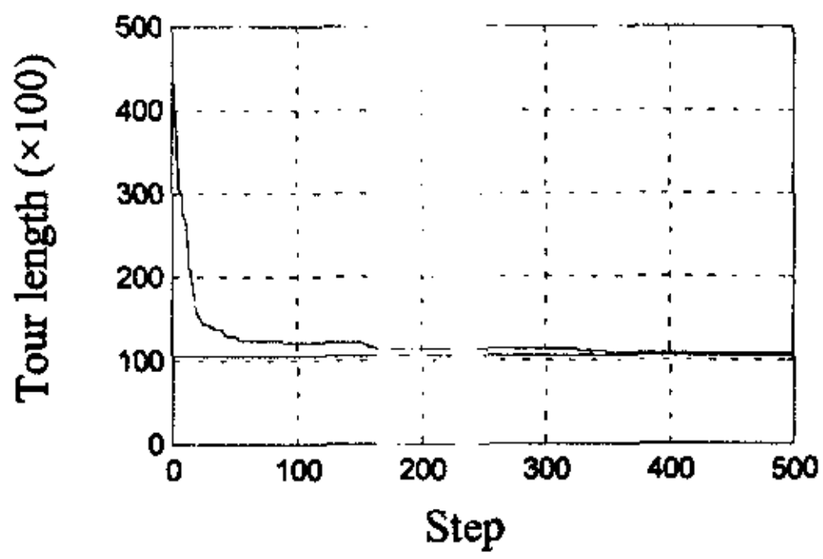


图 3.4 Att48 城市 TSP 问题的寻优曲线

在仿真实验中，为了反映算法对组合优化问题的解决能力，没有针对 TSP 问题的特点添加任何的启发规则，完全依靠人工鱼群算法对解空间的寻优能力，可见，人工鱼群算法具有较快的收敛速度，但是随着问题规模的扩大，其寻优的精度会有所降低，这是有待改进的地方。

## 3.5 结 论

通过对 TSP 的仿真可以看出，人工鱼群算法在离散问题的解决方面也具备较快的收敛能力。

作为一种新型的智能优化方法，它还有许多需要进一步改进的地方；其较快的收敛速度，可以应用于一些有实时性要求的地方；对于一些精度

要求不高的场合，也可以用它快速的得到一个可行解；另外，对问题的了解不必很深，甚至不需要问题的精确描述，也使得它的应用范围得以延伸。

# 第四章 基于分解协调的人工鱼群算法

## 4.1 引言

随着被处理对象规模的增大，优化处理对系统效率的提高、能耗的降低、资源的合理利用及经济效益的提高等的效果越显著，随着现代生产规模的迅速发展，大规模系统工程问题的优化计算越来越成为人们急需解决的问题，然而，随着对象规模的增大，优化命题的方程数增多、变量维数变高、非线性更强等特点，使得命题的求解相当困难。

大系统的优化问题一般有整体优化和分散优化两种方法，其中分解与协调优化方法是一种比较好的思路，即将系统分解为多个子系统，对各子系统进行优化和协调运算，以求得大系统的最优解。但目前的分解协调算法如目标协调法、模型协调法等都存在计算速度、收敛性、初值敏感性等方面的问题，往往不能达到满意的效果([沈静珠 1994])。

下面针对大系统中常见的变量维数高、约束方程多等特点，基于基本的人工鱼群算法，采用分解协调的思想，引入了鱼的协调行为，构造了一种扩展的人工鱼群算法([李晓磊 & 钱积新 2003])。

## 4.2 系统的分解与鱼群的分类

通常，大系统可以通过变换分解为一些子系统，一般子系统之间会存在一定的耦合关系，如果没有耦合关系，那么问题会变得更加简单了，对各子系统分别寻优的解将是大系统的最优解。在对大系统进行分解后，根据各子系统来设计各类人工鱼相应的寻优目标和行为，从而将鱼群分为多个不同的种类。

对于大多数工业过程，可以抽象为目标可分解的优化命题来描述，即具有如下和的形式：

$$\min J = \sum_{k=1}^{N} f_k(x)$$
$$s.t.\quad s_i(x) \geq 0, \quad i = 1,2,\cdots,m$$
$$h_j(x) = 0, \quad j = 1,2,\cdots,l$$
$$x = [x_1, x_2, \cdots, x_n]^T$$

于是我们可以将原系统分解为$N$个子系统，分解的原则主要是降低子系统所需要寻优的变量维数和约束方程的数目：

$$\min j_k = f_k(x_{local}) \qquad (k = 1,\cdots,N)$$
$$s.t.\quad s_i(x) \geq 0, \quad i \in [1,2,\cdots,m]$$
$$h_j(x) = 0, \quad j \in [1,2,\cdots,l]$$
$$x_{local} = [x_p, \cdots, x_q]^T \subseteq [x_1, x_2, \cdots, x_n]^T$$

这样我们可以设想用$N$类鱼来分别表示这$N$个子系统，然后用$N$类鱼群通过基本鱼群算法分别求各子系统的最优解。通常情况下，由于子系统间和约束条件的耦合性，所得的解都不是所期望的全局最优解。而随着鱼群协调行为的引入，这个问题得到了很好的解决。

## 4.3 基于分解和协调思想的鱼群算法描述

算法采用自下而上的设计方法，设计的关键就是底层的人工鱼个体行为的实现，算法的进行也就是人工鱼个体的自适应的行为活

```
procedure Artificial_Fishschool_Algorithm
    ::AF_init();
while the result is satisfied do
    switch (::AF_evaluate())
        case value1:
            ::AF_follow();
        case value2:
            ::AF_swarm();
        default:
            ::AF_prey();
    end switch
    ::AF_move();
    ::AF_coordinate();
    ::AF_subject();
    ::AF_bulletin();
    get_result();
end while
end Artificial_Fishschool_Algorithm
```

图 4.1 基于分解协调思想的鱼群算法迭代原理

动，最优解将在该过程中突现出来。算法中每条人工鱼作为一个实体，该实体中封装着自身的状态和行为，实体每活动一次就是算法的一次迭代，图4.1描述了带有约束和协调行为的人工鱼群算法迭代的基本原理。

在描述鱼群的各个行为之前，先说明一些符号的定义。

### 4.3.1 符号说明

人工鱼个体的全局状态可表示为向量 $X=(x_1,x_2,\cdots,x_n)$，其中 $x_i(i=1,\cdots,n)$ 为欲寻优变量的总体；向量 $X_{local}=(x_p,\cdots,x_q)$ 为某类人工鱼所寻优的局部变量；人工鱼当前所在位置的总体的目标函数值表示为 $Y$，它所代表的子系统的目标函数值表示为 $G$；人工鱼个体之间的距离表示为 $d_{i,j}=\|X_i-X_j\|$；人工鱼的感知距离为 *Visual*；人工鱼移动的步长为 Step；$\delta$ 为拥挤度因子。

以下的描述均以求极小值来说明。虽然不同种类的人工鱼所封装的数据不同，但各种行为的原理大致相同，所以，以下主要以假设类名为 *AF_type_A* 的人工鱼的 C++语言伪代码来说明。

### 4.3.2 人工鱼行为描述

#### 4.3.2.1 公告板

算法采用公告板的方式来获取最优解。各人工鱼个体在寻优过程中，每次行动完毕就检验自身状态与公告板的状态，如果自身状态优于公告板状态，就将公告板的状态改写为自身状态，这样就使公告板记录下历史最优的状态。

#### 4.3.2.2 约束行为

优化命题中约束条件的处理是在约束行为中进行的。由于人工鱼的个体在寻优过程中寻到的解存在有效解和无效解之分，所以在鱼的实体对象

中，加入约束条件，主要用来判断解的有效性和转化一部分无效解为有效解。

在判断和转化过程中，如果解满足总系统的约束，则该解为全局有效的，返回 Valid 值；如果仅满足子系统的约束，则该解为本地有效的，返回 localValid 值；其他情况则认为是无效解，返回 inValid 值。

#### 4.3.2.3 觅食行为

设人工鱼当前局部状态为 $X_{local,i}$，在其视野范围内随机选择下一个局部状态 $X_{local,j}$，如果 $G_i > G_j$，则向该方向前进一步；反之，则随机移动一步。

其 C++语言伪代码描述如下：

```
float AF_type_A::AF_prey()
{
    save_current_state();
    X_{local,j} = X_{local,i} + Rand() · Visual;
    if  (G_i > G_j)
    {
        X_{local,i|next} = X_{local,i} + Rand() · Step · (X_{local,j} − X_{local,i}) / ||X_{local,j} − X_{local,i}||;
        return AF_localfood();
    }
    restore_original_state();
    X_{local,i|next} = X_{local,i} + Rand() · Step;
return AF_localfood();
}
```

#### 4.3.2.4 聚群行为

设人工鱼当前状态为 $X_i$，探索当前邻域内（即 $d_{i,j} < Visable$）的伙伴数目 $n_f$ 及中心位置 $X_c$，如果 $Y_c n_f < \delta Y_i \quad (\delta > 1)$，（求极大问题时为

$\frac{Y_c}{n_f} > \delta Y_i \quad (\delta < 1)$)，表明伙伴中心并不太拥挤，则朝伙伴的中心位置方向前进一步；否则执行觅食行为。

伪代码描述如下：

```
float AF_type_A::AF_swarm()
{
    n_f = 0; X_c = 0 ;
    for (j=0;j<friend_number;j++)
        if( d_{i,j} < Visual )  {  n_f ++; X_c += X_j ;  }
    X_c = X_c / n_f;
    if  (Y_c n_f < δY_i)
        X_{i|next} = X_i + Rand() · Step · (X_c - X_i) / ||X_c - X_i||;
    else
        AF_prey();
return AF_globalfood( X_{i|next} );
}
```

4.3.2.5 追尾行为

设人工鱼当前状态为 $X_i$，探索当前邻域内（即 $d_{i,j} < Visable$）的伙伴中 $Y_j$ 为最小的伙伴 $X_j$，如果 $Y_j n_f < \delta Y_i \quad (\delta > 1)$，(求极大问题时为 $\frac{Y_c}{n_f} > \delta Y_i$ $(\delta < 1)$)，表明伙伴 $X_j$ 的状态较优并且其周围不太拥挤，则朝伙伴 $X_j$ 的方向前进一步；否则执行觅食行为。

伪代码描述如下：

```
float AF_type_A::AF_follow()
{
    Y_min = +∞ ;
    for (j=0;j<friend_number;j++)
        if( d_{i,j} < Visual && Y_j < Y_min )  {  Y_min = Y_j ;  X_min = X_j ;  }
```

$n_f = 0$;

for (j=0;j<friend_number;j++)

if( $d_{\min,j} < Visual$ ) { $n_f$ ++ ; }

if ( $Y_{\min} n_f < \delta Y_i$ )

$$X_{i|next} = X_i + Rand() \cdot Step \cdot \frac{X_{\min} - X_i}{\|X_{\min} - X_i\|};$$

else

AF_prey();

return AF_globalfood( $X_{i|next}$ );

}

4.3.2.6 协调行为

鱼群的协调行为主要是指在不同鱼类的个体之间进行的一种信息交换活动。可以理解为，在人工鱼与本类鱼伙伴进行聚群和追尾的同时，参照其他类鱼的状态来调整自身状态的一种行为。

如图4.2所示的示意图，寻优空间S中有三类人工鱼（*AF_type_A, AF_type_B, AF_type_C*），每类鱼有两条，在寻优过程的某一时刻，A类鱼的个体I发现了距其最近的两个异类鱼：B类鱼II和C类鱼III，于是，他就根据它们所局部寻优的部分变量来调整自己相应的变量。

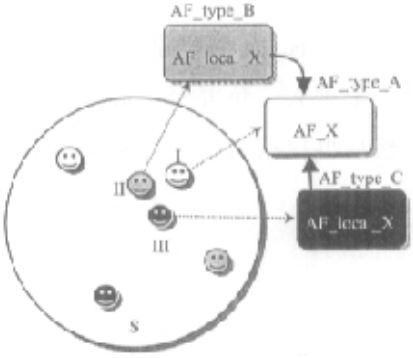



图 4.2 鱼群协调行为示意图

整个过程伪代码描述如下：

void AF_type_A::AF_coordinate()

{

*do*

*{*

$$X_{\min} = \underset{j}{Minimum}(\|X_i - AF_type_T(j) :: X_i\|);$$

$$(j = 1, \cdots, AF_number)$$

$$x_{i|next} = x_i + Random(Step)(x_{\min,i} - x_i) / \|x_{\min,i} - x_i\|;$$

$$(i \in \{[p, \cdots, q] | AF_type_T :: X_{local}\})$$

*}while(AF_type_T enumerate all AF types except itself)*

*return void;*

*}*

其中， $AF_type_T :: X = (x_1, x_2, \cdots, x_n)$（$T \in all\ AF\ types\ except\ A$）为其他类别的人工鱼的全局状态。

#### 4.3.2.7 行为选择

根据所要解决的问题性质，对人工鱼当前所处的环境进行评价，从而选择一种合适的行为。一般按照进步最快的原则来选择。

### 4.3.3 结果的获取

最简单的方案就是看公告板的状态是否稳定在了满意的误差界内，最终公告板的值就是系统的最优值，公告板的状态就是系统的最优解；另一种获取结果的方式就是取得人工鱼群聚集最多的区域，可以求得系统的一个或几个满意解域，然后再根据需要进行相应的处理。

### 4.3.4 补充说明

针对不同性质的问题，算法可以相应的进行简化。如对于简单的单峰函数，只使用觅食行为就可以了，那样就是一定意义上的爬山寻优法了。

算法中每种行为都存在一定的随机性，一般会觉得这将造成算法收敛效率的降低，而事实并非如此，由于群的效应，使得整体结果是在快速向

极值域收敛的，随机性的存在正使得寻优活动更加全面的展开，从而实现全局的最优。

## 4.4 计算实例

如图4.3所示的换热系统([沈静珠 1994])，四股热流的热容流量均为15000W/℃，流线III的冷流体（进口温度150℃）的热容流量为13500W/℃，其余两股冷流体的热容流量为10000W/℃。位于流线III上的换热器的传热系数均为135W/(m²·℃)，其余的换热器的传热系数均为100W/(m²·℃)。要求三股冷流体的出口温度分别达到400℃，450℃，500℃。求总换热面积最小时的换热面积最优分配。

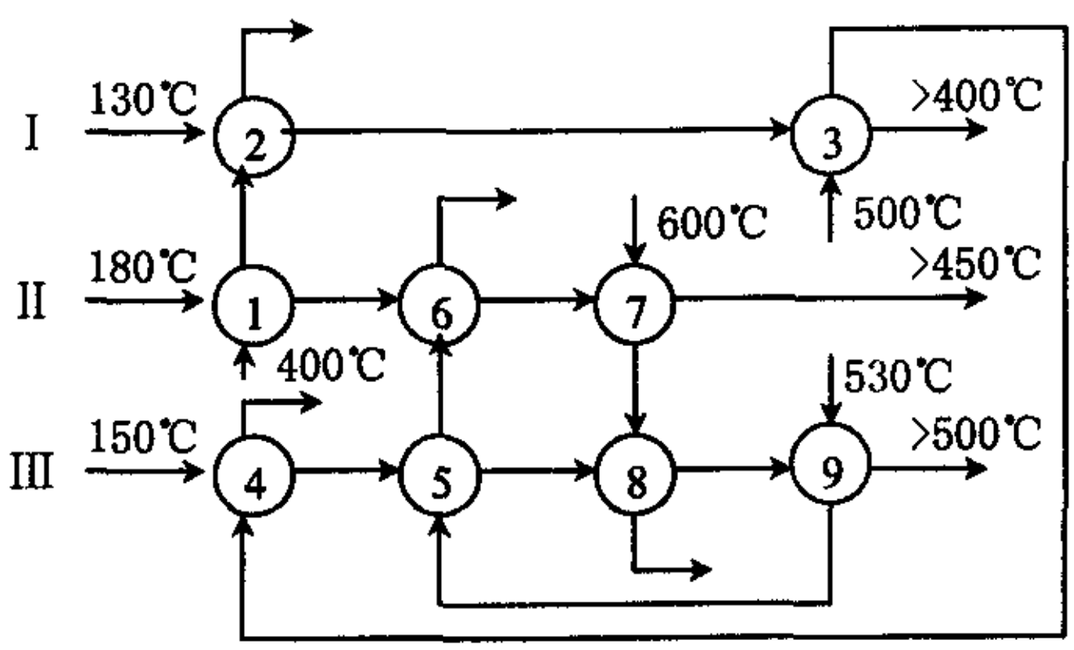



图 4.3 换热器系统

由传热方程及热量衡算方程，得各换热器的模拟方程为

$$q_i = A_i K_i \Delta t_i$$

$$q_i = (WC_p)_i (T_{i,I} - T_{i,O}) = (wc_p)_i (t_{i,O} - t_{i,I}) \quad (i = 1, \cdots, 9)$$

式中 $$\Delta t_i = \frac{(T_{i,I} - t_{i,O}) - (T_{i,O} - t_{i,I})}{\ln \frac{T_{i,I} - t_{i,O}}{T_{i,O} - t_{i,I}}}$$

$A_i, K_i$分别为第i个换热器的换热面积和传热系数；$T_{i,I}, T_{i,O}, (WC_p)_i$分别为第i个换热器热流体的进、出口温度及热容流量；$t_{i,I}, t_{i,O}, (wc_p)_i$分别为第

i个换热器冷流体的进、出口温度及热容流量(如图4.4所示)。

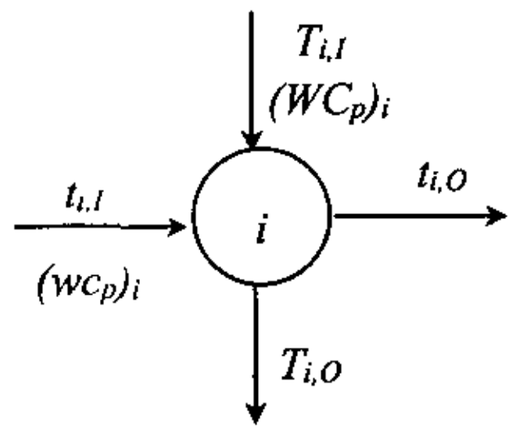


图 4.4 第 $i$ 个换热器

要求总换热面积最小，则系统的目标函数可以表示如下：

$$\min J = \sum_{i=1}^{9} A_i$$

$$s.t.\ A_i K_i \Delta t_i - (WC_p)_i (T_{i,I} - T_{i,O}) = 0$$

$$A_i K_i \Delta t_i - (wc_p)_i (t_{i,O} - t_{i,I}) = 0$$

$$t_{1,O} - t_{6,I} = 0 \quad t_{2,O} - t_{3,I} = 0 \quad t_{4,O} - t_{5,I} = 0$$
$$t_{5,O} - t_{8,I} = 0 \quad t_{6,O} - t_{7,I} = 0 \quad t_{8,O} - t_{9,I} = 0$$
$$T_{1,O} - T_{2,I} = 0 \quad T_{3,O} - T_{4,I} = 0 \quad T_{5,O} - T_{6,I} = 0$$
$$T_{7,O} - T_{8,I} = 0 \quad T_{9,O} - T_{5,I} = 0$$

按4股热流体的路线可以将系统分解为如图4.5所示4个子系统。

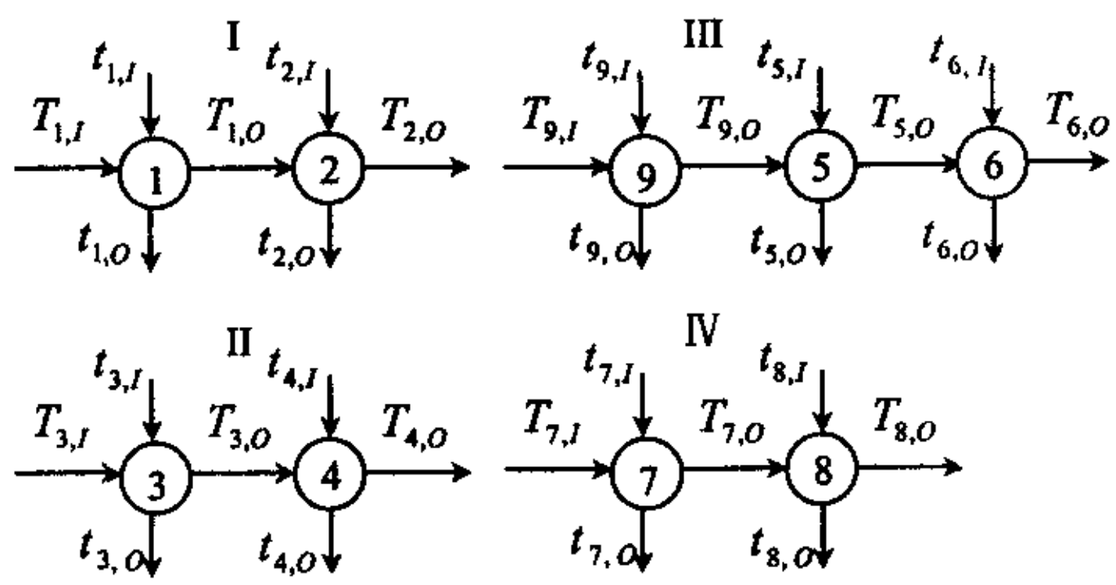


图 4.5 4个子系统

### 4.4.1 计算结果

在实验中每类人工鱼的数目在10个左右时就能稳定的收敛到满意的解域，移动步长*Step*的取值从10~200，视野范围*Visual*的取值从20~400都能保证比较稳定的收敛，并且迭代100次的时间约5秒钟，仿真用的计算机为Intel Celeron 400，软件环境为Microsoft Visual C++ 6.0。由于本算法中存在一定的随机性，所以为了验证本算法的收敛性能，连续进行了

多次计算。以下仿真结果的参数取值为：每类人工鱼的个数均为15个，移动步长*Step*=50，视野范围*Visual*=300，在仿真中没有使用拥挤度因子$\delta$。如表4.1和表4.2是连续10次计算所得的最优值和解，图4.6是连续10次的计算结果进行平均后所得的迭代收敛曲线，图4.7是所得结果最优的一次（第9次）计算过程的迭代收敛曲线。

**表4.1 连续10次计算所得的最优值**

| 次数 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 最优值 $\sum_{i=1}^{9} A_i / m^2$ | 713.5 | 710.1 | 705.7 | 705.3 | 713.6 | 710.9 | 706.0 | 710.7 | 705.2 | 711.4 |

**表4.2 系统的最优解**

| 换热器 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $T_{i,1}$/℃ | 400.00 | 394.40 | 500.00 | 382.36 | 529.99 | 529.01 | 600.00 | 592.45 | 530.00 |
| $T_{i,o}$/℃ | 394.40 | 332.04 | 382.36 | 301.69 | 529.01 | 362.16 | 592.45 | 359.11 | 529.99 |
| $t_{i,1}$/℃ | 180.00 | 130.00 | 223.54 | 150.00 | 239.63 | 188.40 | 438.68 | 240.72 | 499.99 |
| $t_{i,o}$/℃ | 188.40 | 223.54 | 400.00 | 239.63 | 240.72 | 438.68 | 450.00 | 499.99 | 500.00 |
| $A_i$/$m^2$ | 3.94 | 50.28 | 138.78 | 60.91 | 0.38 | 196.25 | 7.46 | 247.18 | 0.04 |

如图4.6和图4.7所示，由于人工鱼的初始状态是随机产生的，所以在计算初期一般没有有效解，随着人工鱼个体间的协调行为和自身的约束行为，人工鱼的状态逐渐出现有效状态；同时，随着人工鱼的觅食行为和追尾行为，使所得的解快速向全局和局部最优方向收敛；还是在同时，随着人工鱼的聚群行为，使得一部分鱼离开局部最优区域，再去寻取其他的区域，从而能最终到达全局最优解域。

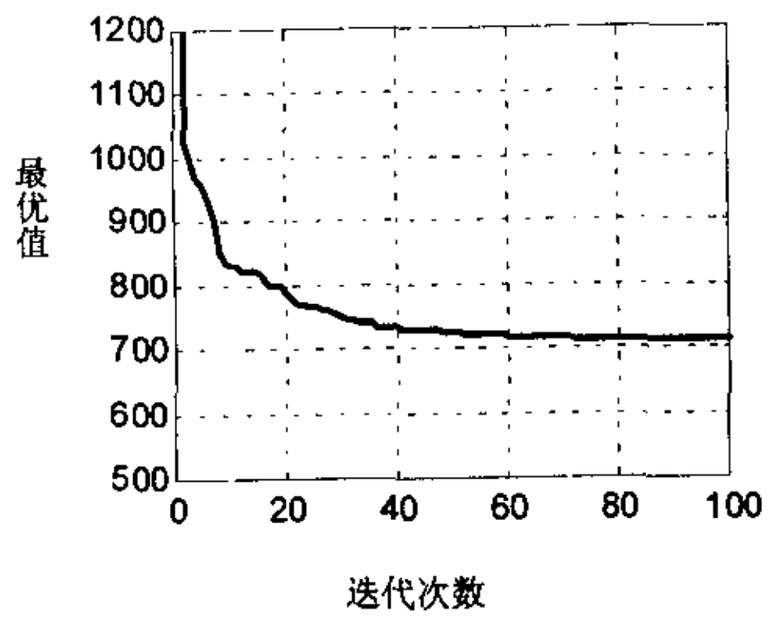



图4.6 10次计算的平均收敛曲线

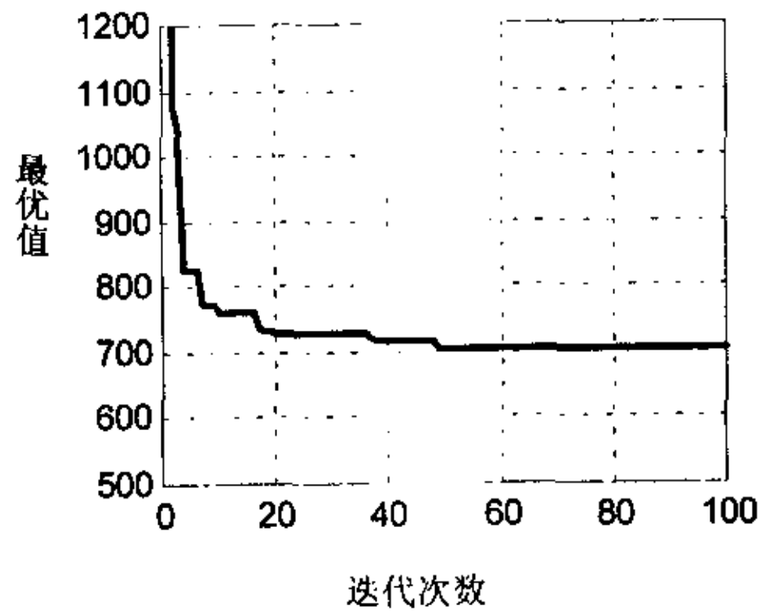



图4.7 结果最优计算的收敛曲线

### 4.4.2 结果比较

以下是多种优化方法针对该实例的优化结果和特点分析([沈静珠 1994])：

4.4.2.1 模型协调法

模型协调法是将大系统分解为几个子系统，算法分为两个等级，在低等级各子系统分别用Lagrange乘子法计算，计算各子系统的输出$Y_i$，在高等级调整各子系统的输出参数。本算法对于方程数大于未知数维数的情况不能求解。该法对本系统所得的最优结果为739.777/$m^2$

4.4.2.2 复合形法

复合形法是将单纯形法推广用于带约束条件的优化方法。他在不同初值的计算结果为：

| 初值/$m^2$ | | 迭代次数 | 最优值 $\sum_{i=1}^{9} A_i / m^2$ |
|---|---|---|---|
| $i=1,2,4,5,6,8$ | $A_i=10$ | 279 | 706.6 |
| | $A_i=100$ | 274 | 701.7 |
| | $A_i=300$ | 467 | 703.8 |
| $A_i^+$ | | 668 | 705.0 |

表中：$A_1^+=80$，$A_2^+=12$，$A_4^+=44$，$A_5^+=13$，$A_6^+=90$，$A_8^+=66$

它的收敛速度较慢，特别是决策变量的维数多时更为突出。

4.4.2.3 模拟退火法

模拟退火法是将优化计算与物理上物体逐渐退火过程进行类比而提出的一种优化方法。

模拟退火法中取 $A_i^L = 0, A_i^U = 300, \alpha = 0.96, S = 0.2, T_I / T_E = 100$，在不同初始“温度”时计算结果如下：

| $T_I$ | 10000 | 100.00 | 10.00 | 1.00 |
|---|---|---|---|---|
| $\sum_{i=1}^{9} A_i$ | 865.2 | 725.1 | 720.3 | 714.8 |

在计算过程中，各个参数的选取都比较重要，否则将不利于寻求全局最优或造成运算时间过长。

## 4.5 结论

从以上的计算实例可以看出，采用基于分解协调思想的人工鱼群算法具有以下几个特点：

- 算法对变量的初始值无要求，可以是随机产生的值，包括无效解也可以；
- 算法中仅仅使用了目标函数的函数值，对函数的性质没有特殊要求；
- 算法的收敛性好，大都能收敛到比较满意的解；
- 参数的敏感性低，算法中参数的选择要求不高，参数的变动对结果的影响也不大；
- 分解与协调思想的引入为复杂的大系统的优化问题的解决提供了一条可行的途径。

# 第五章 人工鱼群算法的改进方法

## 5.1 引言

在前面部分的论述中可以看出，鱼群算法在解决问题时存在一些优点：

算法中只需要比较目标函数值，因此对目标函数要求不高；

算法对初值的要求不高，初值随机产生或设定为固定值均可以；

算法对参数设定的要求不高，有比较大的容许范围；

同时，它也存在一些有待改进的地方：

随着人工鱼数目的增多，将会需求更多的存储空间，也会造成计算量的增长；

由于视野和步长的随机性和随机行为的存在，使得寻优的精度难以很高；

由于鱼群模式可以采用面向对象的方式来实现，所以，对功能的扩展和改造有着良好的基础。

## 5.2 改进的鱼群算法

在以上所述的鱼群算法中，当寻优的域较大或处于变化平坦的区域时，一部分 AF 将处于无目的的随机移动中，这影响了寻优的效率，下面引入生存机制和竞争机制加以改善。

### 5.2.1 生存机制

在此，我们引入生存周期的概念，即随着人工鱼所处环境的变化，赋

予人工鱼一定的生存能力，这样，就使得人工鱼在全局极值附近拥有最强的生命力，从而具有最长的生命周期；位于局部极值的人工鱼将会随着生命的消亡而重生，如随机产生该 $AF$ 的下一个位置，从而展开更广的搜索，这样不仅节省了存储空间，同时也提高了寻优能力和效率。

生存机制的描述如下：

$$h=\frac{E}{\lambda T};\quad \begin{cases} h\geq 1 & ::\text{AF_move} \\ h<1 & ::\text{AF_init} \end{cases}$$

式中：$h$ 为生存指数；$E$ 为 $AF$ 当前所处位置的食物能量值；$T$ 为 $AF$ 的生存周期；$\lambda$ 为消耗因子，即单位时间内消耗的能量值。

即当 $AF$ 所处位置的食物能量足以维持其生命时，$AF$ 将按照正常的行为寻优，否则，当此处的能量低于其生命的维持所需时，通常此时处于非全局极值点附近，寻优通常会没有结果，所以，强制该 $AF$ 初始化，例如，可以随机产生该 $AF$ 的位置，使其跳出该区域，这样就相当于利用相同的存储空间增加了寻优 $AF$ 的个数，从而提高了算法的效率。

### 5.2.2 竞争机制

竞争机制就是实时的调整 $AF$ 的生存周期，其描述如下：

$$T=\varepsilon\frac{E_{\max}}{\lambda}$$

式中：$E_{\max}$ 为当前所有的 $AF$ 中所处位置的食物能量的最大值；$\varepsilon$ 为比例系数。

即随着寻优的逐步进展，$AF$ 的生存周期将被其中最强的竞争者所提升，从而使得那些处于非全局极值点附近的 $AF$ 能有机会展开更广范围的搜索。

### 5.2.3 仿真结果

仿真的对象及参数同前面第一章仿真对象（2.1）中所讨论的一致，引入的新参数为：$\lambda = 0.01$，$\varepsilon = 0.618$。

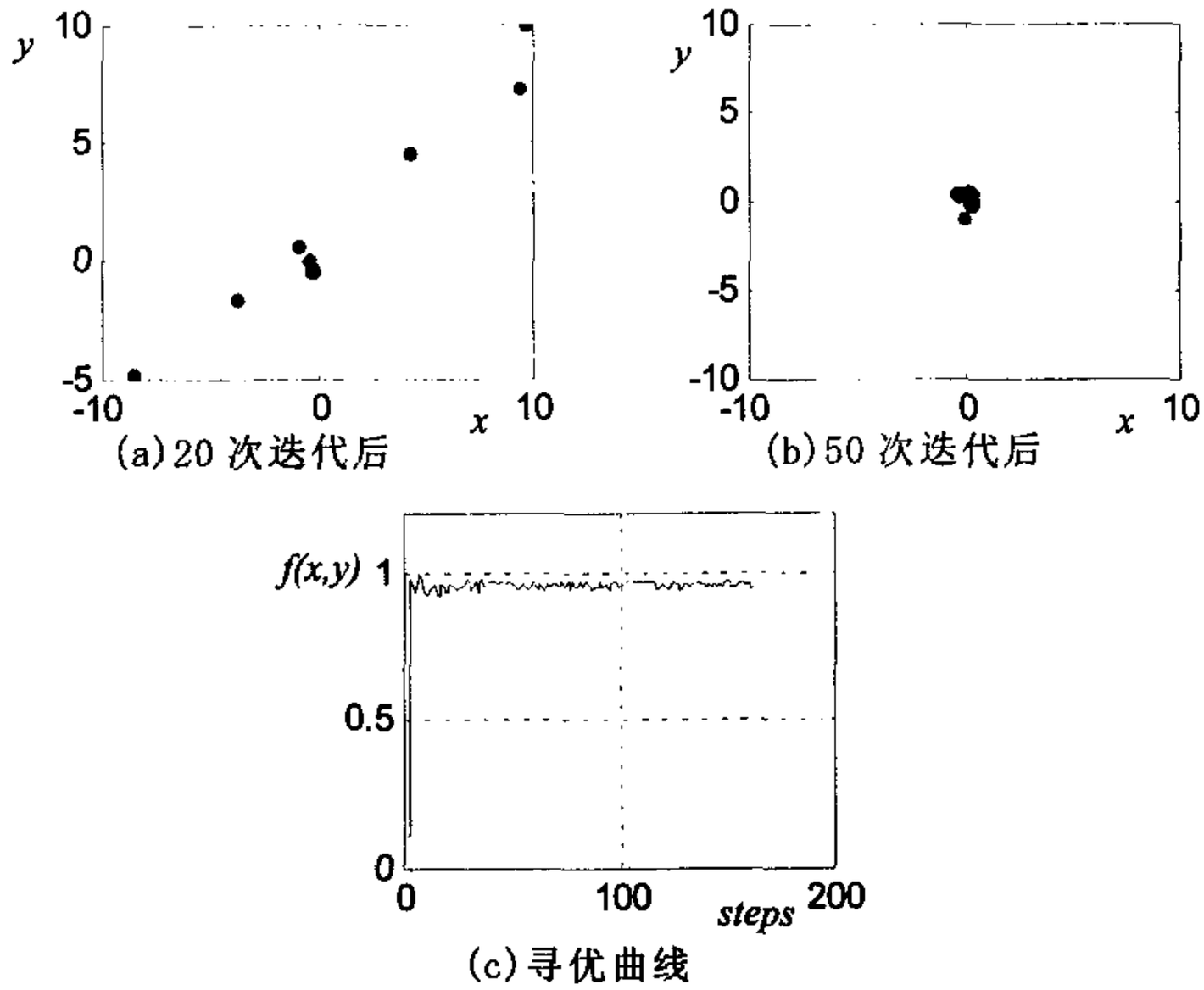



图 1 改进鱼群算法的仿真结果

由图 1 可以看出，算法的收敛速度和搜索的效率得到了明显提高。

## 5.3 视野的改进

在鱼群模式所讨论的视野概念中，由于视点的选择是随机的，移动的步长也是随机的，这样，虽然能在一定程度上扩大寻优的范围，尽可能保证寻优的全局性，但是，会使得算法的收敛速度减慢，有大量的计算时间浪费在随机的移动之中。

下面引入一种自适应步长的方式：

对于人工鱼当前状态 $X = (x_1, x_2, \cdots, x_n)$ 和所探索的下一个状态 $X_v = (x_1^v, x_2^v, \cdots, x_n^v)$，其表示如下：

$$x_i^v = x_i + Visual \cdot Rand(), i = Rand(n)$$

$$X_{next}=\frac{X_v-X}{\|X_v-X\|}\cdot|1-\frac{Y_v}{Y}|\cdot Step(\text{求解极小问题})$$

$$\text{或}\quad X_{next}=\frac{X_v-X}{\|X_v-X\|}\cdot|1-\frac{Y}{Y_v}|\cdot Step(\text{求解极大问题})$$

式中，*Rand* 函数为产生 0 到 1 之间的随机数，*Step* 为移动步长，$Y_v$ 为 $X_v$ 状态的目标函数值，$Y$ 为 $X$ 状态的目标函数值。

即，本次移动步长的大小取决于当前所在的状态和视野中视点感知的状态。

## 5.4 高阶行为模式

将人工鱼以前的行为和经验也封装在对象内部，并对其采取一定的参考和学习，以达到提高算法的效率的目的。

如：

```
class Artificial_fish
{
public:
    float AF_X[n];              //AF's position
    float last_AF_X[n];         //AF's last position
    float evergood_AF_X[n];     //the ever good position of AF
    ... ...
};
```

这样，人工鱼就记忆着曾经的最好状态和上一次的状态，从而为下一次的行为作出更加合理的行为评价。

## 5.5 分段优化方法

算法在优化过程初期虽然具有较快的收敛品质，但在后期却往往收敛较慢，或者无法达到要求的精度，因此，与其他算法相结合，在合适的时

候与它们互相切换，实现各算法之间的优缺点互补，也是一种解决问题的常用方法。

## 5.6 混合优化方法

鱼群模式提供了一种解决问题的架构，其中自然可以应用传统的一些相对成熟的计算方法，而面向对象的方法使得将其他计算方法与鱼群算法的有机融合提供了良好的基础。

例如，如果问题的模型比较熟知，并且目标函数的非线性程度不是非常严重，则可以在觅食行为中使用单纯行法等传统方法来代替在视野中的随机搜索方法，这样，在提高收敛速度的同时，能适当提高收敛的精度，并且还能在一定程度上克服局部极值的问题。

## 5.7 结论

由以上可以看出，人工鱼群算法由于采用了鱼群模式，使得在扩展性、改造性和融合性等方面具备较好的能力。依靠一种理想的算法来解决所有的问题也是不太符合实际的，所以，应用一些措施在一定程度上提高自身算法的品质，另一方面，与其他算法相融合，达到超过单一算法的效能，这也许才是解决一些难题的有效方法([王凌 & 郑大钟 2000a])。

# 第六章 人工鱼群算法的应用研究

## 6.1 引言

在控制领域的很多应用中，都离不开优化命题的解决，并且，这些命题通常都存在严重的非线性和变量维数多等特点，有些应用对实时性还有一定的要求，数学规划的优化方法往往会因为陷入局部极值而影响优化效果，或者因目标函数难以求导等因素而难以实施；一些启发式优化方法，往往会因为计算时间过长，或约束条件不能很好的处理等因素而影响实施。

下面，我们将人工鱼群算法应用于时变系统的在线辨识和鲁棒PID的参数整定两个实例中，取得了较好的效果。

## 6.2 基于人工鱼群算法的参数估计方法

### 6.2.1 前言

时变系统的参数跟踪是目前辨识领域十分活跃的研究课题（[Googwin & Sin 1984]，[Ioannou 1996]，[Ljing 1990]，[Norton & Mo 1990]），虽然目前已有了较多成熟的辨识算法，但这些算法基本上都是在假设模型结构已知的基础上进行的，一些重要的参数都是根据先验知识预先确定的，不易在线辨识，常见辨识算法如最小二乘算法，梯度法等其本质上都是利用在梯度方向上的寻优的局部搜索方法。

人工鱼群算法采用基于动物行为来寻求全局最优的新的思路，具有良好的克服局部极值，取得全局极值的能力。且算法的实现无需目标函数的梯度值等特性，并对搜索空间具有一定的自适应能力，且寻优速度较快，因此可以用来解决此类问题。

### 6.2.2 基于人工鱼群算法的系统辨识

考虑 ARX 模型：

$$A(q^{-1})y(k)=B(q^{-1})u(k-d)+\xi(k)$$

其中 $q^{-1}$ 代表滞后因子，$y(k)$为输出信号，$u(k)$为输入信号，$\xi(k)$为独立于 $u(k)$的白噪声。

$$A(q^{-1})=1+a_1(k)q^{-1}+\ldots+a_n(k)q^{-n}$$

$$B(q^{-1})=b_1(k)q^{-1}+\ldots+b_m(k)q^{-m}$$

为具有时变系数的多项式。

假设 $A(q^{-1})$，$B(q^{-1})$ 未知，则欲辨识参数为：

$$\hat{\theta}=\left[\hat{a}_1,\ldots,\hat{a}_n,\hat{b}_1,\ldots,\hat{b}_m\right]$$

实际参数为：

$$\theta=\left[a_1,\ldots,a_n,b_1,\ldots,b_m\right]$$

在鱼群辨识算法中，人工鱼个体的状态用来表示欲辨识的参数，即 $X=\hat{\theta}$；人工鱼所处位置的食物浓度由目标函数来表示，即：

$$J=\alpha-\frac{1}{2}\sum_{j=k-W}^{k}\left(y(j+1)-\phi^T(j)\theta(k)\right)^2$$

式中：$W$ 为辨识时间窗口宽度，$W$ 越小，则跟踪性能越好，但对噪声很敏感；$W$ 越大，则克服噪声能力较强，跟踪速度较慢。$W$ 一般取 $n+m+2 \sim n+m+6$。

辨识前，根据辨识对象的复杂性选择人工鱼个体的数目，通常情况下，数目越多辨识的精度越高，收敛的速度也会更快，但会相应提高对计算设备内存等的需求，然后，根据先验知识大致确定参数的域，并在域内随机的初始化各人工鱼个体的初始状态。

### 6.2.3 仿真实例

为了检验本算法的性能，本文进行了仿真实验研究。考虑具有多个输

入的搅拌槽混合系统([Murakami 2000])，根据物料和组分平衡并化简可得其方程为：

$$c'(k+1)=a_1c'(k)+\sum_{i=1}^{n}b_i(t)q_i'(k)+d\xi(k)$$

其中：

$$a_1=\exp\left(-\frac{\bar{q}}{V}T\right)$$

$$b_i=\left[1-\exp\left(-\frac{\bar{q}}{V}T\right)\right]\left(\frac{c_i-\bar{c}}{\bar{q}}\right)$$

$$d=\frac{1}{\bar{q}}\left[1-\exp\left(-\frac{\bar{q}}{V}T\right)\right]$$

且 $c'=c-\bar{c}$，$q_i'=q_i-\bar{q}_i$，$\bar{c}$，$\bar{q}$ 和 $\bar{q}_i$ 为工作点值，$c$ 为输出物料组分，$c_i$ 为第 $i$ 个进料组分，$q$ 为输出物料量，$q_i$ 为第 $i$ 个进入物料量。

由前文讨论可概括鱼群辨识算法如下：

1. 人工鱼个体进行初始化；
2. 采样系统的实际输出 $y(k)$；
3. 执行鱼群算法的寻优，找出拥有最大值的人工鱼个体，并纪录其状态；
4. 如果达到了指定的精度或在指定的搜索步数内仍然没有找到更好的个体，则寻优结束，AF 的状态就是当前的辨识参数；
5. 令 $y(k-1)=y(k)$，转（2）。

在仿真过程中，鱼群辨识算法的参数选择如下：人工鱼个体数 100，视野 *Visual*=0.3，移动步长 *Step*=0.1,拥挤度因子 $\delta$=0.618。在仿真过程中，我们考虑了两种参数变化情况：参数 $b$ 发生突变和参数 $b$ 发生缓慢变化。假如进料量为随机

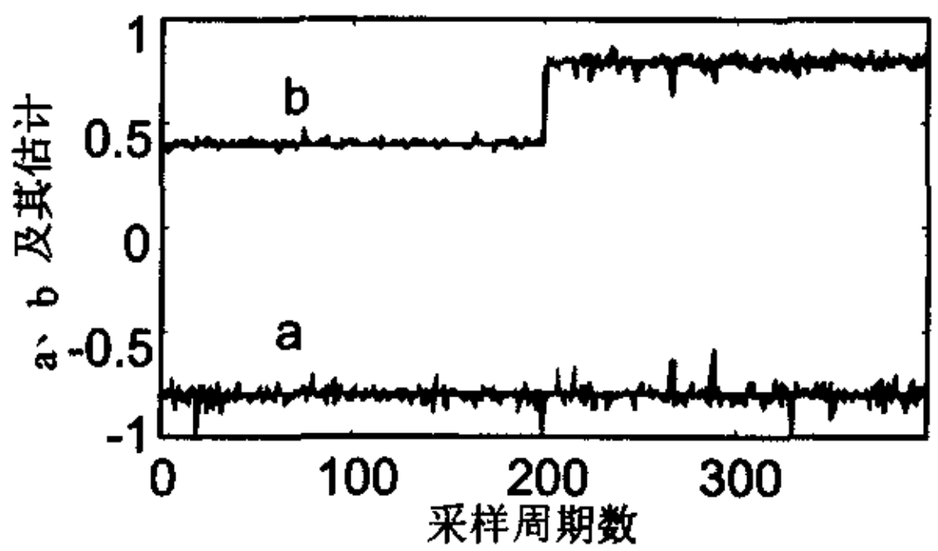



图 6.1 人工鱼群参数估计算法对参数突变的跟踪响应

序列。并假设其中某个输入物料组分在 $t=nT$ 时开始发生变化，则$\hat{b}(k)$也将发生变化。所有的仿真均在有噪声下进行。

图 6.1 为$b(k)$在 $t=nT=200T$ 时发生突变时参数$\hat{a}(k)$，$\hat{b}(k)$跟踪$a(k)$，$b(k)$的响应曲线。其中曲线 a 为$\hat{a}(k)$对$a(k)$的跟踪，曲线 b 为$\hat{b}(k)$ 对$b(k)$的跟踪响应。图 6.2 为$b(k)$自 $t=nT=200T$ 时起发生缓慢变化时参数$\hat{a}(k)$，$\hat{b}(k)$跟踪$a(k)$，$b(k)$的曲线。其中曲线 a 为$\hat{a}(k)$对$a(k)$的跟踪，曲线 b 为$\hat{b}(k)$ 对$b(k)$的跟踪响应。从两图可明显看出，本文所提鱼群辨识算法均具有良好的跟踪性能。另外，本算法的计算速度较快，在 CPU 为 Celeron 400 的计算机上用 Visual C++ 6.0 进行的仿真结果显示，其计算时间均小于 1 秒，故该参数估计算法可适用于实时辨识。

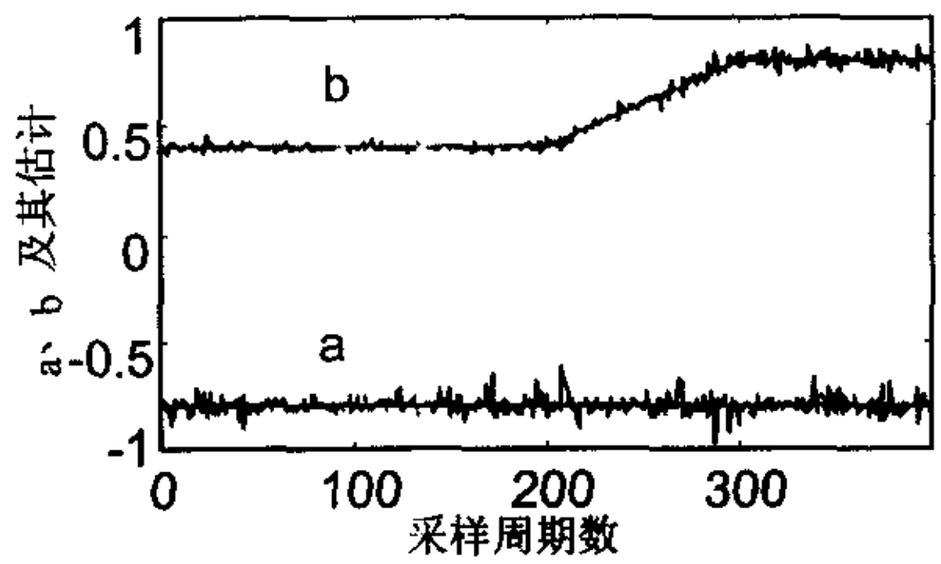



图 6.2 人工鱼群参数估计算法对慢时变参数的跟踪响应

### 6.2.4 结论

以上介绍了一种基于人工鱼群算法的全局搜索参数估计算法，并将其用于一混合系统的在线辨识，并取得了较好的效果，可以预见，基于全局搜索策略的人工鱼群辨识算法非常适用于辨识系统的结构和参数，这为系统辨识和基于辨识的控制系统分析和设计提供了新途径。

## 6.3 基于人工鱼群算法的鲁棒PID控制器参数整定方法研究

### 6.3.1 引言

PID 控制器具有结构简单、实现方便、鲁棒性强、效果满意等特点，并且人们对它的原理、物理意义等比较熟悉，建立起了比较完善的理论体系，所以它在工业过程控制中得到了广泛而长久的应用。对于某一特定的过程对象，常规的 PID 参数整定方法很容易产生一组控制器参数，使得控制器具备满意的控制效果，但随着过程对象的变化，如元器件参数的漂移、设备故障、环境变化等因素的影响，这时常规 PID 参数整定出的参数往往不能再保持满意的控制效果，甚至会出现振荡或发散现象，这在工业现场中是不允许的，通常需要对控制器的参数重新整定([王伟 & 张晶涛 2000])。鲁棒 PID 控制器就是为了克服常规 PID 控制器的这一缺陷而产生的，本文中采用了基于极小一极大原理的鲁棒 PID 控制器([郑泳凌 & 马龙华 2001])，并采用人工鱼群算法对其参数进行整定。

人工鱼群算法对寻优空间的形式和性质没有特殊要求，它原则上是一种基于比较目标函数值的搜索方法，不需要利用导数信息，因而具有较好的全局寻优能力，且寻优速度较快。对典型问题的仿真结果表明，用人工鱼群算法整定的鲁棒 PID 控制器具有良好的鲁棒性和满意的控制效果。

### 6.3.2 鲁棒PID控制器

我们采用了基于极小一极大原理的鲁棒 PID 控制器，也就是寻找一组 PID 参数使得即使过程对象在最恶劣的工况下也能保持可接受的控制品质，即控制器的性能对过程对象的不确定性不敏感。

PID 控制器的形式如下：

$$G_c = K_c(1+\frac{1}{T_i s}+T_d s)\frac{1}{T_f s+1}$$

式中 $K_c$ 为比例增益，$T_i$ 为积分时间，$T_d$ 为微分时间，$T_f$ 为后置滤波器常数。

控制器的性能指标有很多种，如 ISE（*Integral Squared Error*）、ITSE（*Integral Time Squared Error*）、IAE(*Integral Average Error*)、ITAE(*Integral Time Average Error*）等，不同的性能指标对于控制器品质的侧重点不同，如上升时间、超调量等，也就直接影响着整定后 PID 参数的整定值，如本

文与文献[郑泳凌 & 马龙华 2001]采用了不同的性能指标函数，从而使得整定结果存在一定的差异，因此，根据实际情况可以选择或设计合适的性能指标。

通常对于如下的一阶惯性纯滞后模型：

$$G_p = \frac{K}{Ts+1}e^{-\tau s}$$

鲁棒 PID 控制器参数的整定目标为：

$$\min_{K_c,T_i,T_d,T_f} \max_{K,T,\tau} f(K_c,T_i,T_d,T_f,K,T,\tau)$$

例如，如果采用 ISE 性能指标，则参数整定的目标函数为([郑泳凌 & 马龙华 2001])：

$$\min_{K_c,T_i,T_d,T_f} \max_{K,T,\tau} f(K_c,T_i,T_d,T_f,K,T,\tau) = ISE = \int_0^\infty e(t)^2 dt = \sum_{k=1}^{N}(sp(k)-y(k))^2 \cdot h$$

其中：

$$\begin{aligned}
y(k) = \frac{1}{\frac{TT_iT_f}{h^3}+\frac{T_i(T+T_f)}{h^2}+\frac{T_i}{h}} \cdot \{ & KK_c[(1+\frac{T_i}{h}+\frac{T_iT_d}{h^2})sp(k-p) \\
& -(\frac{T_i}{h}+\frac{2T_iT_d}{h^2})sp(k-p-1)+\frac{T_iT_d}{h^2}y_r(k-p-2)] \\
& +(\frac{3TT_iT_f}{h^3}+\frac{2T_i(T+T_f)}{h^2}+\frac{T_i}{h})y(k-1) \\
& -(\frac{3TT_iT_f}{h^3}+\frac{T_i(T+T_f)}{h^2})y(k-2)+\frac{TT_iT_f}{h^3}y(k-3) \\
& -KK_c[(1+\frac{T_i}{h}+\frac{T_iT_d}{h^2})y(k-p)-(\frac{T_i}{h}+\frac{2T_iT_d}{h^2})y(k-p-1) \\
& +\frac{T_iT_d}{h^2}y(k-p-2)]\}
\end{aligned}$$

$p=\frac{\tau}{h}+1$；$h$ 为采样周期；$N$ 为采样次数。

可以看出，优化的目标函数是个很复杂的非线性函数，应用常规的非线性规划优化算法（如 SQP）很容易陷于局部极值，而导致寻优效果较差，应用遗传算法等方法又会带来优化时间过长的问题。同时，鲁棒 PID 控制器参数整定的极小－极大问题不是一个鞍点的优化命题，需要在内层（求

极大）和外层（求极小）之间进行有效的协调，最终才能求得在极小一极大意义下满意的解。SQP 等算法往往会陷入局部极值点或临近初值的边界上，而遗传算法等算法则带来了更长的计算时间。应用人工鱼群算法则能很好的克服以上缺点。

### 6.3.3 基于人工鱼群算法的参数整定

与基本的人工鱼群算法类似，算法中依然采用了觅食、追尾和聚群等基本行为，所不同的是，其行为目标被定义为两种模式，即寻最大和寻最小，在适当的时候进行切换。算法过程描述如下：

```
Procedure Artificial_fish_school_algorithm
Begin
    AF_initialize()
    While not termination-criterion-mini do
        AF_move(MINI) for N steps
         While not termination-criterion-max do
             AF_move(MAX)
            AF_bulletin()
         End
    End
End

Procedure AF_move(type)
  AF_evaluate(AF_prey(type),AF_follow(type),AF_swarm(type))
End
```

其中，*AF_prey* 为觅食行为，*AF_follow* 为追尾行为，*AF_swarm* 为聚群行为，*AF_evaluate* 为行为评价，*AF_bulletin* 为公告板。

由算法过程可以看出，寻优过程为两层，在内层寻最大时，人工鱼认为 $K_c,T_i,T_d,T_f$ 为常数，对 $K,T,\tau$ 进行寻优；在外层寻最小时，认为 $K,T,\tau$ 为常数，对 $K_c,T_i,T_d,T_f$ 进行寻优，最终，当公告板中的最优值连续 $L$ 步依然不变，则近似认为收敛，这样得到的解一般能满足工程中的需要。需要说明的是，由于该命题不存在鞍点，所以在进行外层寻极小时，是在内层寻

极大的基础上逐步向前寻优的，也就是每次迭代人工鱼向极小方向寻优 $N$ 步，$N$ 越小寻优的精度会有所提高，但寻优速度会很慢；$N$ 越大寻优的速度会很快，但寻优到一定程度时会出现振荡现象，影响寻优的精度。

### 6.3.4 实例仿真研究

对于一供气压力系统([Qiang 2000])，可以近似描述为一阶惯性纯滞后系统如下：

$$G'(s)=\frac{0.8}{1.07s+1}e^{-2.1s}$$

仿真过程中，设定值为单位阶跃函数，即 $sp(k)=\begin{cases}1,k\geq 0\\0,k<0\end{cases}$，采样周期为 0.2s，采样次数为 500 次。考虑到本系统的一些特点，在设计系统的优化指标时，我们在 ISE 性能指标的基础上考虑了抑制系统超调和振荡的因素，设计了一种扩展的 ISE 性能指标：

$$\min_{K_c,T_i,T_d,T_f}\ \max_{K,T,\tau} f(K_c,T_i,T_d,T_f,K,T,\tau)=EISE=\lambda\int_0^{\infty}e(t)^2g_1(t)dt+(1-\lambda)g_2(y(t))$$

其中，$g_1(t)$用来抑制振荡，$g_2(y(t))$用来抑制超调，$\lambda$为加权系数。

在对象参数 $K$、$T$、$\tau$ 的不确定度为±40%时的整定结果如表 6.1 所示：

表 6.1 PID 参数整定结果

| 算法 | PID 控制器参数 | | | | 正常工况下指标值 | 最差工况下指标值 |
|---|---|---|---|---|---|---|
| | $K_c$ | $T_i$ | $T_d$ | $T_f$ | | |
| ISE 指标最优参数整定普通 PID | 0.984 | 1.74 | 0.845 | - | 3.3542 | 发散 |
| 人工鱼群算法整定鲁棒 PID | 0.992 | 4.956 | 2.195 | 3.964 | 3.7e-004 | 0.002198 |

图 6.3 所示为按照 ISE 指标最优参数整定规则而非鲁棒意义下整定的一组 PID 参数的输出响应曲线([Zhuang & Atherton 1993])，实线所示为对标称对象特性用这组 PID 参数的响应曲线，点线所示为过程对象参数均偏离标准值的＋40%时的响应曲线，可见当对象特性漂移较大时会出现大的超调和振荡现象，在有的恶劣工况下甚至会出现发散的情况（图中没有给出）；图 6.4 所示为我们在不确定度 40%条件下利用人工鱼群算法对鲁棒 PID 控制器的参数整定后的系统输出响应曲线，实线表示鲁棒 PID 控制器的参数对标称工况的阶跃响应曲线，点线所示为过程对象的参数均偏离标准值的＋40%时的输出响应曲线，虚线所示为过程对象的参数均偏离标准值的－40%时的输出响应曲线，可见，控制器在恶劣的工况下，依然具有可接受的控制品质，原则上讲，在该不确定度的范围内，这组 PID 参数对任意一种工况的控制都是可以接受的。可见，鲁棒 PID 控制器的整定不是单纯针对最好或最差工况的整定就能实现的，需要进行极小－极大意义下的寻优才能实现。

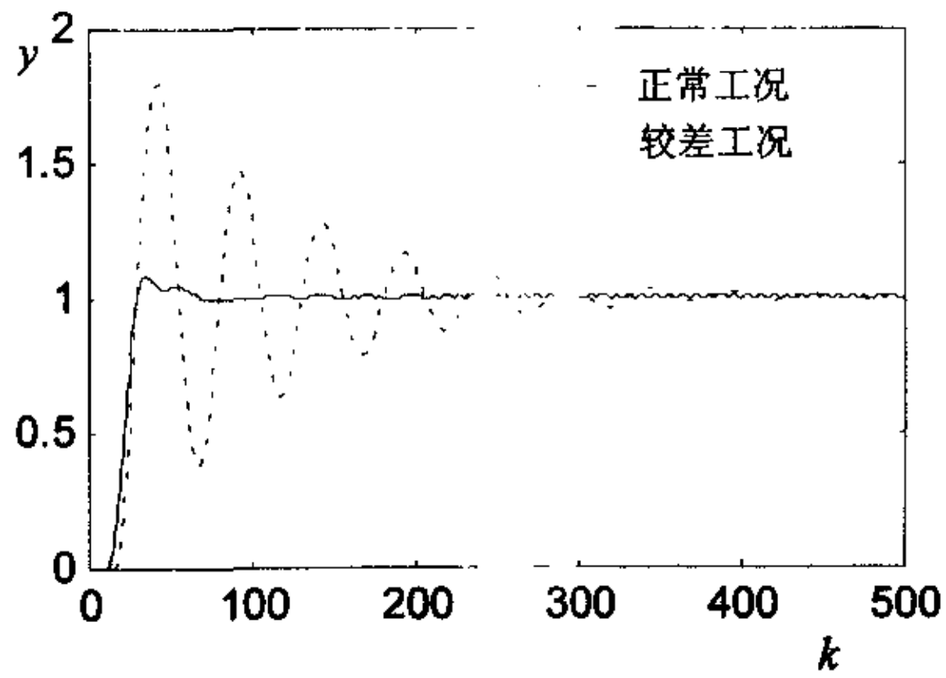


图 6.3 ISE 最优参数（非鲁棒）整定结果

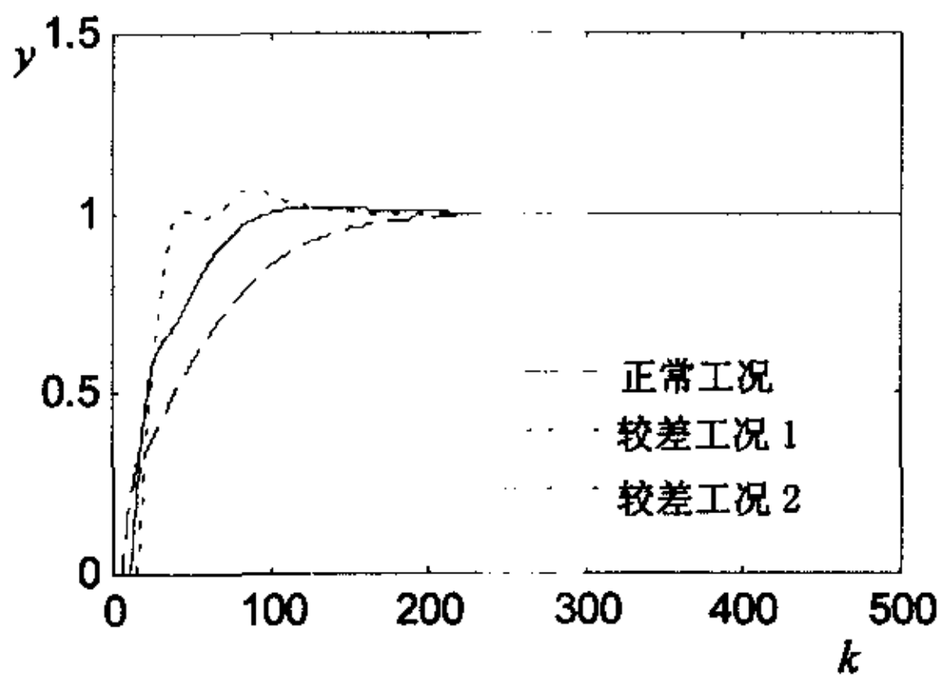


图 6.4 鲁棒 PID 参数整定结果

图6.5所示为人工鱼群算法的寻优曲线，其中人工鱼的数目为20个，视野范围*Visual*为10，移动的最大步长*Step*为1.5，外层每次寻优步数*N*为50，中止指标为最优值在400步依然没有变化，则近似认为收敛。仿真硬件环境为Intel Pentium 4 1.7G，软件环境为Microsoft Windows Millennium + Microsoft visual C++ 6，计算时间约为30秒。算法中参

数的初值是随机产生的，尽管在寻优初期处于极大的发散状态，但是仍然能够以极快的速度收敛下来，从而寻到满意的结果。可见，该算法对初始值没有要求，并且具有较快的收敛速度。

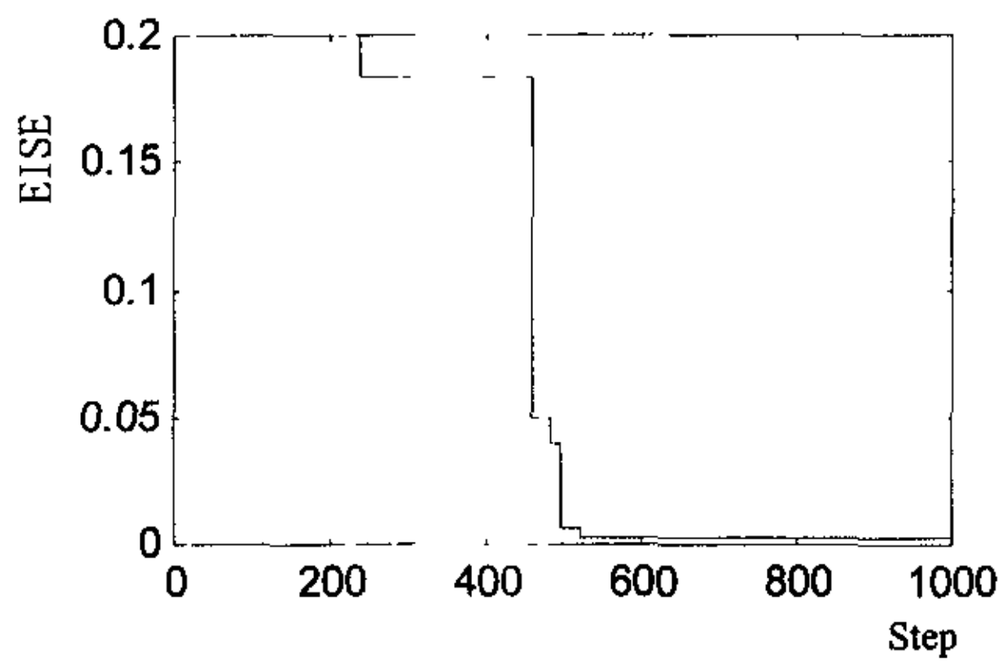



图 6.5 人工鱼群算法寻优曲线

### 6.3.5 结论

可以看出，用人工鱼群算法对鲁棒PID控制器进行参数整定取得了满意的控制效果；同时也可以看出，PID调节的方法本身也具有较强的鲁棒性，这样才使得极小一极大方法与之相结合，实现了鲁棒性更佳的鲁棒PID控制器；鉴于极小一极大问题的特殊性和优化目标函数的复杂性，非线性规划方法（SQP等）与遗传算法等优化方法均有待进一步的改进，因此，应用人工鱼群算法应该是一种合适的选择。

# 第七章 结语

鱼群模式提出了一种新的解决问题的方式，应用该模式所产生的人工鱼群算法在应用于一些问题时取得了一定的效果，并具有以下主要特点：

- 算法只需要比较目标函数值，对目标函数的性质要求不高；
- 算法对初值的要求不高，初值随机产生或设定为固定值均可以；
- 算法对参数设定的要求不高，有较大的容许范围；
- 算法具备并行处理的能力，寻优速度较快；
- 算法具备全局寻优的能力；

鱼群模式提供了一种灵活的架构，我们可以根据问题的性质对人工鱼群算法进行相应的调整，如对于规模小或性质简单的问题，可以通过对人工鱼行为进行简化或干脆忽略某些行为来达到简化算法、提高效率的目的；对于规模大或性质复杂的问题，可以添加相应的有效行为或参数来更好的解决问题，如基于分解协调的人工鱼群算法中引入协调行为，解决了子系统之间的协调问题；对于非连续形式的优化命题或者非数值型的优化命题，可以相应的转化一下视野等概念的表示方法来使得算法可行，并达到解决问题的目的，如组合优化问题的解决。

随着优化命题的复杂程度和规模的不断提高，使用某一单一的优化算法而使问题得到满意的解决，变得越来越困难，单纯依靠改进算法的某些参数或指标已经显得有些力不从心，将成熟的算法进行良好的融合，利用它们的优势互补，共同来解决问题不失为一个好的选择，鱼群模式就为鱼群算法与其他一些优秀算法进行融合和互补提供了一个良好的架构。

鱼群模式和鱼群算法在下一步的工作我认为主要有以下几点：

- 研究更加有效的邻域搜索方法，来增强鱼群模式中视野的概念，从而提高算法的效能；
- 更加深入的研究聚群行为的机理，使得它的设计和实现更加合理，从而使得算法取得事半功倍的效果；
- 研究大规模鱼群算法的存储和执行效率问题。

鱼群模式和鱼群算法形成了一种新型的思路，从具体的实施算法到总体的设计理念，都不同于传统的设计和解决方法，但同时它又能与传统方法相融合，因此，从具体的数学问题到更高层次的管理调度等问题，相信都有良好的应用前景。

# 参考文献

**[Ahuja & Ergun 2002]** Ravinda K. Ahuja, Ozlem Ergun, etc. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*. 123 (2002): 75-102.

**[Ahuja & Orlin 2000]** Ravindra K. Ahuja, James B. Orlin, Dushyant Sharma, Very large-scale neighborhood search, *International Transactions in Operational Research*. 7(2000)301-317.

**[Angeline 1998]** Angeline. Using selection to improve particle swarm optimization. *Proceedings of the 1998 International Conference on Evolutionary Computation*, Anchorage, Alaska, Piscataway NJ/IEEE Press, 1998:84-89.

**[Blumberg & Meyer 1998]** Blumberg, B., Meyer, J. A., Pfeifer, R. and Wilson, S. W.(Eds.). From Animals to Animats. *Proceedings of the Fifth International Conference on Si,ulations of Adaptive Behavior*. The MIT press/Bradford Books. 1998.

**[Bonabeau & Théraulaz 2000]** E. Bonabeau and G. Théraulaz, Swarm smarts, *Scientific American*, March 2000:72-79.

**[Brooks 1986]** Rodney A. Brooks, A Robust Layered control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, RA-2, April, 1986:14-23.

**[Brooks 1990]** Brooks, R.,"Elephants Don't Play Chess," *Robotics and Autonomous Systems*, 6,1990:3-15

**[Brooks 1991a]** Brooks, R.A. 1991. Intelligence without Representation. *Artificial Intelligence*, 47, 1991:139-160.

**[Brooks 1991b]** Brooks, R.A. 1991. Intelligence without Reason. Computers and Thought lecture. In *Proceedings of ICJAI-91, Sidney, Australia.*

**[Colomi & Dorigo 1994]** Colomi A, Dorigo M and Maniezzo V, et al. Ant system for

job scheduling. *Belgian J. of Operations Research Statistics and computer science*, 1994, 34(1):39-53.

**[Dean 1998]** Jeffrey Dean. Animats and what they can tell us. *Trends in cognitive sciences*. 1998,Vol2: 60-67

**[De Jong 1975]** De Jong K. A. An analysis of the behavior of a class of genetic adative systems. Ph. D Dissertation, University of Microfilms, 1975.

**[Dorigo 1996]** Dorigo M., Maniezzo V.,and Colorni A.(1996). The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics* 26:29-41.

**[Dorigo 1997a]** Dorigo M. and Gambardella L. (1997). Ant colonies for the traveling salesman problem. *Biosystems*. 43:73-81.

**[Dorigo 1997b]** Dorigo M. and Gambardella L. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, Vol.1, No.1, April 1997:53-66.

**[Dorigo & Bonabeau 2000]** Marco Dorigo, Eric Bonabeau, Guy Theraulaz, Ant algorithms and stigmergy, *Future Generation Computer Systems*, 16,2000:851-871.

**[Eberhart & Kennedy 1995]** Eberhart, R. C. and Kennedy J. A new optimizer using particles swarm Theory, Proc. *Sixth International Symposium on Micro Machine and Human science* (Nagoya, Japan), IEEE Service Center, Piscataway, NJ, 1995:39-43.

**[Fogel 1993]** Fogel D B. Applying evolutionary programming to selected traveling salesman problems, *Cybernetics and system*, 1993, 24: 27-36.

**[Fogel 1994]** Fogel D B. an introduction to simulated evolutionary optimization, *IEEE Trans on Neural Networks*, 1994, 5(1):3-14.

**[Gelenbe & Schmajuk 1997]** Erol Gelenbe, Nestor Schmajuk, John Staddon, John Reif, Autonomous search by robots and animals: A survey, *Robotics and*

*Autonomous Systems* 22 ,1997:23-34.

**[Glover 1989]** Glover F. Tabu search Part I. *ORSAJ on computing*, 1989, 1:190-206.

**[Goldberg 1989]** Goldberg D. E. Genetic algorithms in search, optimization and machine learning, MA: Addison Wesley, 1989.

**[Googwin & Sin 1984]** Googwin, G., & Sin, K. Adaptive filtering prediction and control. Englewood Cliffs, NJ: Prentice-Hall, 1984.

**[Hoffmeyer 1994]** J. Hoffmeyer, The swarming body, *Proc. 5th Congress of the International Association for Semiotic Studies*, Berkeley, 1994.

**[Hoffmeyer 1997]** Jesper Hoffmeyer, swarm-semiotics, Irmengard Rauch and Gerald F. Carr(eds.): Semiotics Around the World. *Proceedings of the Fifth Congress of the International Association for Semiotic Studies*. Berkeley 1994. Berlin/New York: Mouton de Gruyter 1997, pp937-940.

**[Holland 1992]** Holland J H. Adaptation in natural and artificial systems, Boston: MIT Press, 1992.

**[Hölldobler 1990]** Hölldobler B. and E.O. Wilson. *The ants*. Springer-Verlag, Berlin, 1990.

**[Horswill & Brooks 1988]** Ian D. Horswill and Rodney A. Brooks, Situated Vision in a Dunamic World: Chasing Objects. AAAI-88, St Paul, MN, 1988:796-800.

**[Ioannou 1996]** Ioannou, P.A et al. Robust adaptive control. Englewood Cliffs, NJ: Prentice-Hall, 1996.

**[Kennedy & Eberhart 1995]** J. Kennedy, R.C. Eberhart, Particle swarm optimization, *Proc. IEEE International Conference on Neural Networks*, Perth, Australia, Nov. 27-Dec. 1, 1995, pp. 1942-1948.

**[Kirkpartick & Gelatt 1983]** Kirkpartick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing, *Science*, 1983,220:671-680.

**[Lenat & Feigenbaum 1991]** Douglas B. Lenat and Edward A. Feigenbaum, On the Thresholds of Knowledge, *Artificial Intelligence*, 47,1991:185-250.

**[Ljing 1990]** Ljing, L., Adaption and tracking in system identification-A survey. *Automatica*, 1990, 26, 7-21.

**[Luca & Gambardella 1995]** Luca M, Gambardella and Dorigo M. Ant-Q: an reinforcement learning approach to the traveling salesman problem. *Proc. of 12th Machine Learning Conf.* France: Morgan Kaufmann, 1995, 252-260.

**[Maes 1990]** Maes, P.(ed.), *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, Cambridge, MA: MIT Press, 1990.

**[Maniezzo & Colorni 1994]** Maniezzo V,Colorni A and Dorigo M. The ant systemapplied to the quadratic assignment problem. Technical Report, IRIDIA, 94/28, Univ. Libre De Bruxelles, Belgium, 1994.

**[Meyer 1997]** Jean-Arcady Meyer, From natural to artificial life: Biomimetic mechanisms in animat designs. *Robotics and Autonomous systems* 22, 1997:3-21.

**[Minsky 1961]** Marvin Minsky, Steps Toward Artificial Intelligence, *Proc. IRE* 49, Jan. 1961:8-30.

**[Moravec 1984]** H. P. Moravec, Locomotion, vision and intelligence. In:M. Brady and R. Paul, eds., Robotics Research 1, MIT Press, Cambridge, MA, 1984:215-224.

**[Murakami 2000]** Murakami K. et al. Constrained parameter estimation with applications to blending operations. *Journal of Process Control*,2000, 10, 195-202.

**[Newell & Simon 1976]** Newell, A., and Simon, H. A., "Computer Science as Empirical Inquiry: Symbols and Search," *Communications of the Association for Computing Machinery*, 19(3):113-126,1976.

**[Nicolis & Prigogine 1977]** G. Nicolis and I. Prigogine, Self-Organization in Nonequilibrium Systems, Wiley, New York, NY,1977.

**[Nilsson 1999]** Nilsson N. J. Artificial Intelligence. 北京：机械工业出版社，

1999,9.

**[Norton & Mo 1990]** Norton, J.P., & Mo, S.H. Parameter bounding for time varying systems. *Mathematics and Computers in Simulation*, 1990, 32, 527-534.

**[Osman 1996]** Osman H I. Metaheuristics: a bibliography. *Annals of Operations Research*, 1996,63:513-623.

**[Polya 1948]** Polya G. How to Solve it? Princeton:Princeton University Press,1948.

**[Qiang 2000]** Qiang Bi. Advanced controller auto–tuning and its application in HVAC systems. *Control Engineering Practice* 2000,8:633-644.

**[Reeves 1993]** Reeves C R(Ed). Modern Heuristic Techniques for Combinatorial Problems. Oxford: Blackwell Scientific Publications, 1993.

**[Reynolds 1987]** Reynolds C. W. Flocks, herds, schools: a distributed behavioral model. *Proceeding of SIGGRAPH'87*, Anaheim, California. Computer Graphics 1987:21(4):25-34.

**[Riolo 1987]** Bucket brigade performance. I. Long sequences of classifiers. Genetic Algorithms and Their Applications. *Proceedings of the Second International Conference on Genetic Algorithms*, 1987:184-195.

**[Rivest & Schapire 1987]** A new approach to unsupervised learning in deterministic environments. *Proceeding of the Fourth International Workshop on Machine Learning*. Los Altos, CA: Morgan Kaufmann. 1987:364-375.

**[Shi & Eberhart 1998]** Shi, Y. and Eberhart, R., Parameter selection in particle swarm optimization. In *Evolutionary Programming VII:Proc. EP98*, New York: Springer Verlag 1998:591-600.

**[Shi & Eberhart 1999]** Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, *Proc. 1999 Congress on Evolutionary Computation*, July 6-9 1999:1945-1950.

**[Thomas & Holger 1997]** Thomas stutzle and Holger Hoos. Max-min ant system and

localsearch for combinatorial optimization problems. *Proc. of 2nd Int. Conf. On Metaheuristics*. Wien: Springer-Verlag, 1997.

**[Wilson 1991]** Wilson, S., "The Animat Path to AI," in Meyer, J. A., and Wilson, S.(eds.), *From Animals to Animats; Proceedings of the First International Conference on the Simulation of Adaptive Behavior*, Cambridge, MA: MIT Press/Bradford Books,1991. )

**[Zhuang & Atherton 1993]** Zhuang M. Atherton D P. Automatic tuning of optimum PID controllers. *Proceedings IEE, Pt D*,1993,140:216-224.

**[Zames 1981]** Zames G. Feedback and sensitivity: model reference transfer functions, multiplicative seminorms, and approximate inverses. *IEEE Trans Autom Control*, 1981,AC-26(2):301-320.

**[蔡自兴 & 徐光祐 1996]** 蔡自兴，徐光祐. 人工智能及其应用(第二版). 清华大学出版社，1996.

**[戴汝为 & 周登勇 2000]** 戴汝为，周登勇. 智能控制与适应性. *第三届全球智能控制与自动化大会*, 2000:11-17

**[归行茂 & 李重华 1993]** 归行茂，李重华，柴常智编，数学手册，上海科学普及出版社，1993.

**[李晓磊 & 钱积新 2001]** 李晓磊，钱积新. 人工鱼群算法：自下而上的寻优模式. *过程系统工程年会论文集*，2001:76~82.

**[李晓磊 & 邵之江 2002]** 李晓磊，邵之江，钱积新，一种基于动物自治体的寻优模式：鱼群算法，*系统工程理论与实践*，2002年，第22卷，第11期：32—38.

**[李晓磊 & 钱积新 2003]** 李晓磊，钱积新. 基于分解协调的人工鱼群优化算法研究. *电路与系统学报*, February 2003.1.

**[梁昔明 2000]** 梁昔明，大规模优化理论及算法研究，浙江大学博士后科研工作报告，浙江大学工业控制技研究所，2000.5

**[马振华 1997]** 马振华等，现代应用数学手册：运筹学与最优化理论卷/《现代应用数学手册》编委会编. 一北京：清华大学出版社，1997

**[马良 2000]** 马良，旅行推销员问题的算法综述，*数学的实践与认识*，2000,Vol.30,No.2:156-165.

**[秦裕瑗 1994]** 秦裕瑗，论算法的发现（I）— 组合优化的基本方法，*数学杂志*，1994，Vol.14, No.3:435-444.

**[沈静珠 1994]** 沈静珠编著. 过程系统优化（第二版）. 清华大学出版社, 1994.

**[王凌 & 郑大钟 2000]** 王凌，郑大钟，Meta-heuristic 算法研究进展，*控制与决策*，2000, Vol.15 No.3.

**[王凌 & 郑大钟 2000a]** 王凌，郑大钟，邻域搜索算法的统一结构和混合优化策略，*清华大学学报（自然科学版）*，2000，Vol.40, No.9:125-128.

**[王伟 & 张晶涛 2000]** 王伟，张晶涛，柴天佑. PID 参数先进整定方法综述. *自动化学报*，2000，26（3）：347—355.

**[邢文训 & 谢金星 1999]** 邢文训，谢金星 编著. 现代优化计算方法，北京：清华大学出版社，1999.8

**[张纪会 & 高齐圣 2000]** 张纪会，高齐圣，徐心和. 自适应蚁群算法. *控制理论与应用*. 2000 Vol. 17， No.1:1-3.

**[浙江大学 1980]** 浙江大学数学系高等数学教研组编，概率论与数理统计，人民教育出版社，1980.

**[郑泳凌 & 马龙华 2001]** 郑泳凌,马龙华,钱积新. 鲁棒PID控制器参数整定方法. *化工自动化及仪表*, 2001,28（5）:14—17.

## 作者攻读博士学位期间完成的论文

- 李晓磊，钱积新. 人工鱼群算法：自下而上的寻优模式. *过程系统工程年会论文集*，2001:76~82.
- 李晓磊，邵之江，钱积新，一种基于动物自治体的寻优模式：鱼群算法，*系统工程理论与实践*，2002年，第22卷，第11期：32－38.
- 李晓磊，钱积新. 基于分解协调的人工鱼群优化算法研究. *电路与系统学报*, February 2003.No.1.

## 作者攻读博士学位期间参与的项目

- 国家重点基础研究发展计划/复杂生产制造过程实时、智能控制与优化理论和方法研究项目/多模式、变约束、多目标的生产制造过程实时智能运行操作优化理论与方法研究课题。
- 国家 863 计划/现代集成制造系统技术主题/生产过程操作优化技术及软件课题。

本文工作受到

国家重点基础研究发展计划和国家 863 计划资助