

蚁群算法研究与实现

数学与计算机科学学院 计算机科学与技术专业
105032004002 郑加明 指导教师: 黄添强

【摘要】本文论述了蚁群算法的研究现状与存在的问题,并在此基础上实现了一个简单的蚁群算法原形系统。将该系统应用于聚类分析,然后通过实验验证它的性能。实验表明,蚁群算法在求解复杂优化问题方面具有优越性。

【关键词】蚁群算法;聚类分析;优化;数据挖掘

1. 引言

在我们的日常生活中,常常可以看到成群结队的蚂蚁群落。数以百万计的蚂蚁是如何组成一个群落?一些学者在观察蚂蚁的觅食过程中发现,蚂蚁总能找到巢穴与食物源之间的最短路径。实际上,每只蚂蚁事前并不知道哪一条路径最短,它们只关注很小范围内的局部信息,利用几条简单的规则来进行决策。而蚂蚁的这种群体协作功能是通过遗留在其来往路径上的信息素来完成的。整个蚁群就是通过这种信息素进行相互协作和通信,形成正反馈,使多条路径上的蚂蚁逐渐聚集到最短的那条路径上来的。

蚁群算法(Ant Colony Optimization, ACO)是一种最新发展的模拟蚂蚁群体觅食行为的仿生优化算法。它是由 M.Dorigo 等人受到真实蚁群集体行为的研究成果的启发而首先提出来的^[1]。他们充分利用蚁群搜索食物的过程与旅行商问题(TSP)之间的相似性^[2],通过人工模拟蚂蚁搜索食物的过程中个体之间的信息交流与相互协作最终找到从蚁群到食物源的最短路径的原理解决了 TSP 问题,取得了很好的效果。

蚁群算法之所以能引起相关领域研究者的注意,就是因为蚁群算法求解复杂优化问题时具有快速性、分布式和全局优化的特性。其中,寻找最优解的快速性是通过信息素的正反馈机制来完成的,而其分布式计算的特征又避免了算法的早熟性收敛。同时,具有贪婪启发式搜索特征的蚁群系统又能在搜索过程的早期就找到可接受的解。

蚁群算法在解决组合优化问题的求解方面表现了突出的适用特征^[3]。通过相关领域研究者的关注和努力,蚁群算法在大量领域获得了应用,比如机器人系统、图像处理、制造系统、车辆路径系统、通信系统等领域^[3]。

蚁群算法是一种用来在图中寻找优化路径的机率型技术。该研究成果可以被应用于数据挖掘的多种应用领域^[4],具有一定的研究价值与应用价值。比如应用蚁群算法解决聚类问题。本文将应用蚁群算法来优化K均值聚类算法^[4],并验证蚁群算法的性能。

2. 蚁群算法的基本思想

蚁群算法在解决一系列组合优化问题中取得了较好效果,因此该算法逐渐引起了许多研究者的注意,并将蚁群算法应用到实际问题中。在蚁群算法中,研究者们提出了人工蚁群的概念。人工蚁群与真实蚁群有许多相同之处,也有一些人工蚁群特有的本领。一方面,人工蚁群是真实蚁群行为特征的一种抽象,将真实蚁群觅食行为中核心的部分抽象给人工蚁群。另一方面,由于人工蚁群是需要解决实际问题中的复杂优化问题,因此为了能够使蚁群算法更加有效,人工蚁群还具备了一些真实蚁群所不具备的特有本领。

2.1 真实蚁群和人工蚁群的异同

2.1.1 真实蚁群和人工蚁群共同点^[5]

人工蚁群大部分的行为特征都是来源于真实蚁群，它们具有的共同特征主要表现为：

(1) 人工蚁群和真实蚁群一样，是一群相互合作的群体。蚁群中的每只蚂蚁都能够建立一个解，蚂蚁个体通过相互的协作在全局范围内找出问题较优的解。

(2) 人工蚁群和真实蚁群共同的任务是寻找起点（蚁穴）和终点（食物源）的最短路径（最优目标）。

(3) 人工蚁群和真实蚁群都是通过信息素进行间接通讯。在人工蚁群算法中信息素轨迹是通过状态变量来表示的。状态变量用一个二维信息素矩阵 t 来表示。矩阵中的元素 t_{ij} 表示在节点 i 选择节点 j 作为移动方向的期望值。初始状态矩阵中的各元素设置初值，随着蚂蚁在所经过的路径上释放信息素的增多，矩阵中的相应项的值也随之改变。人工蚁群算法就是通过修改矩阵中元素的值，来模拟真实蚁群中信息素更新的过程。

(4) 人工蚁群还应用了真实蚁群觅食过程中的正反馈机制。蚁群的正反馈机制，使得问题的解向着全局最优的方向不断进化，最终能够有效地获得相对较优的解。

(5) 人工蚁群和真实蚁群都存在着一种信息素挥发机制。这种机制可以使蚂蚁逐渐忘记过去，不会受过去的经验的过分约束，这有利于指引蚂蚁向着新的方向进行搜索，避免过早收敛。

(6) 人工蚁群和真实蚁群都是基于概率转移的局部搜索策略。蚂蚁在转移时，所应用的策略在时间和空间上是完全局部的。

2.1.2 人工蚁群的特有功能^[3, 5]

从真实蚁群的行为中获得的启发来设计人工蚁群的功能还具备了真实蚁群不具有的一些特性：

(1) 人工蚁群算法存在于一个离散的空间中，它们的移动是一个状态到另一个状态的转换。

(2) 人工蚁群算法具有一个记忆它本身过去行为的内在状态。

(3) 人工蚂蚁释放信息素的量，是由蚁群所建立的问题解决方案优劣程度函数决定的。

(4) 蚁群算法更新信息素的时机是随不同问题而变化不反映真实蚁群的行为。如：有的问题中蚁群算法在产生一个解后改变信息量，有的问题中则做出一步选择就更改变信息量，但无论哪一种方法，信息量的更新并不是随时可以进行的。

(5) 为了改善系统的性能，蚁群算法中可以增加一些性能，如：前瞻性、局部优化、原路返回等。在很多应用中蚁群算法可以在局部优化过程中相互交换信息。

2.2 蚁群算法的基本思想

有了人工蚁群的定义，我们就可以将蚁群算法应用于复杂优化问题的求解。现在，我们开始分析蚁群的最短路径搜索过程。我们已经知道在这个过程中，信息素起到了最重要的作用。信息素多的路径，经过的蚂蚁就越多，经过一段时间的正反馈过程，就会有更多的蚂蚁聚集到这条路径上来。

假设有两条路径从蚁穴通向食物源，见图2-1中的B-C-E与B-D-E两条路径，B-C-E路径长度是B-D-E长度的两倍，分别为2和1个单位。现在假设每单位时间间隔 Δt 有30只蚂蚁从B到E，并有30只蚂蚁从E到B。这些蚂蚁都以 Δt 时间走完1个单位的速度行走。蚂蚁在行走时，一只蚂蚁可以在1个单位长度上留下浓度为1个单位的信息素。

设在 $t = 0$ 时刻，有30只蚂蚁要在B点经过，并且此时B-C-E和B-D-E两条路径上的信息素都为0。由于此时两条路径上的信息素相同，所以30只蚂蚁选择两条路径的概率相等。我们不妨假设分别有15只蚂蚁选择了这两条路径。同样，30只蚂蚁在D通往B时也分别有15只蚂蚁选择了这两条路径。

经过 Δt 时间后，从蚁穴出发选择路径B-D-E的15只蚂蚁提前到达了食物源。每只蚂蚁就在路径B-D-E上留下1个单位的信息素，15只蚂蚁在路径B-D-E上留下了15个单位的信息素。同时从食物源出发选择路径E-D-B的15只蚂蚁，在路径上留下15个单位的信息素。所以在路径B-D-E上共留下了30个单位的信息素。而从蚁穴出发选择路径B-C-E的15只蚂蚁已经到达C，并在B-C路径上留下15个单位的信息素。同时从食物源出发的选择路径E-C-B的15只蚂蚁也已经到达C，并在E-C路径上留下了15个单位的信息

素，所以B-C-E整条路径上的信息素为15。（见图2-2）

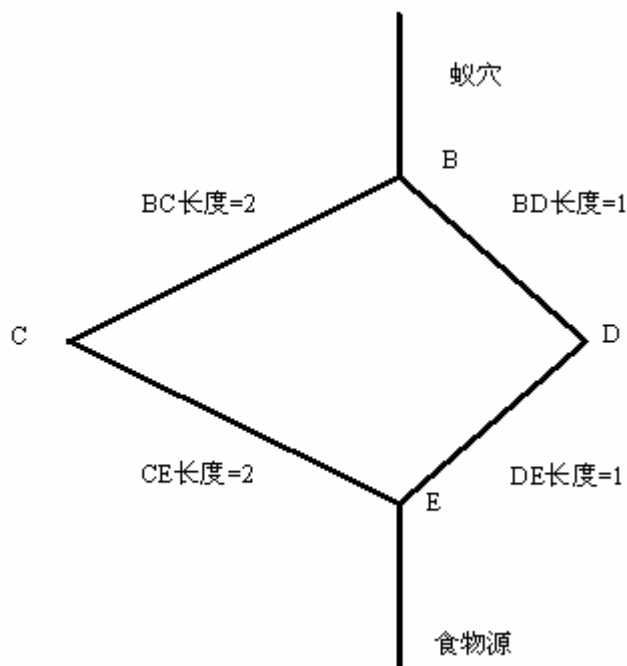


图2-1 简易蚁群算法路径搜索图

在 $t=1$ 时刻，同样又有30只蚂蚁要从蚁穴出发到达食物源。此时蚂蚁发现路径B-D-E上的信息素浓度为30个单位，而路径B-C-E上的信息素浓度为15个单位。这时蚂蚁选择这两条路径的概率就不一样了，设选择路径B-C-E概率是选择路径B-D-E概率的两倍。不妨设有20只蚂蚁选择了路径B-D-E，有10只蚂蚁选择了路径B-C-E。也就是路径B-D-E上的蚂蚁数是路径B-C-E上的两倍。（见图2-3）

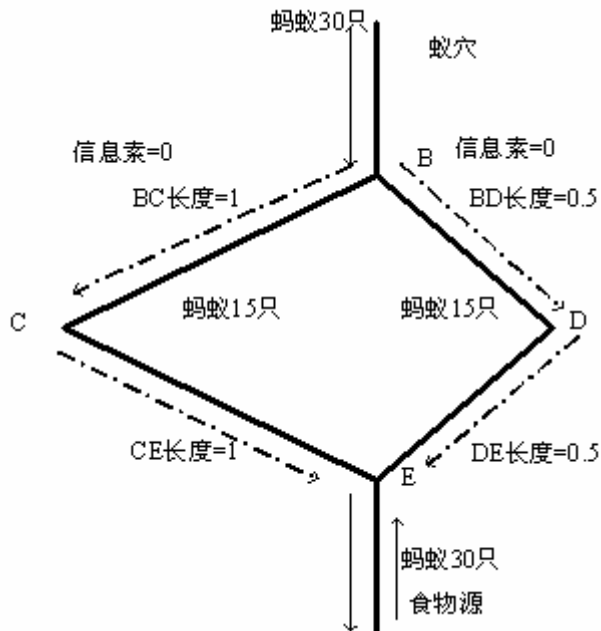


图2-2 $t=0$ 时刻路径搜索图

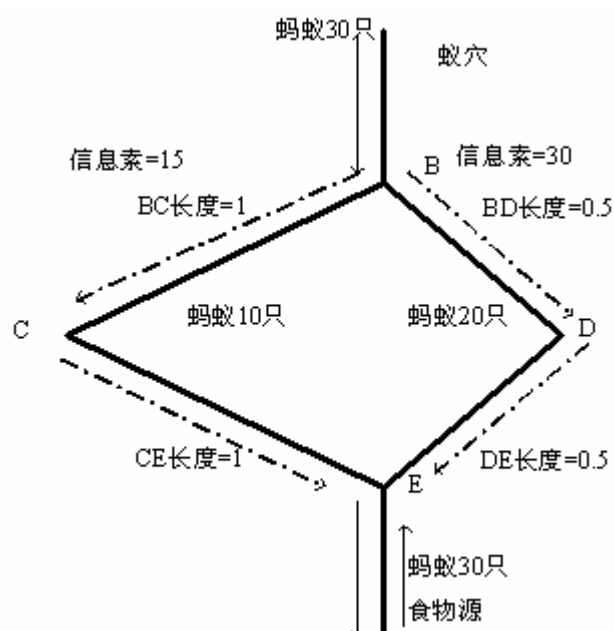


图2-3 $t=1$ 时刻路径搜索图

如此下去，蚂蚁在路径B-D-E上的信息素浓度将越来越浓，选择路径B-D-E的蚂蚁将越来越多。这个过程将一直持续到所有的蚂蚁最终都选择了路径B-D-E。这样蚂蚁就找到了从蚁穴通往食物源的最短路径B-D-E。

现在我们可以总结蚁群算法的基本思想^[3]：如果在给定点，一只蚂蚁要在不同的路径中选择，那么，那些被先行蚂蚁大量选择的路径（也就是信息素越浓的路径）被选中的概率就越大，信息素越浓意味着路径越短，也就意味着找到了一个更好的解。

2.3 蚁群算法的实现过程

在蚁群算法中,蚂蚁之间相互协作可以较好地解决复杂优化问题。蚂蚁既能共同地行动,又能独立地工作。每只蚂蚁都能够找到一个解,虽然有可能不是较好的解。蚂蚁之间不能直接通讯,但是可以通过信息素指引着蚂蚁之间的信息交换。在实现蚁群算法时,实现蚂蚁个体的功能是非常重要的。只要蚂蚁个体的功能完善了,并且拥有了信息素的间接通讯方式,蚁群算法就基本上实现了。

在蚁群算法中,蚂蚁个体需要实现以下四个重要功能^[3, 5]:

(1) 基于概率的局部搜索策略:实际问题中,蚂蚁经常要从一个状态点移动到另一个状态点。经过这样有限步的移动,每只蚂蚁都建立了一个问题的解。在每一步的移动过程,蚂蚁应用基于概率的局部搜索策略选择移动方向。这个策略主要基于蚂蚁的记忆以及信息素浓度,有时还有具体问题的局部信息等。

(2) 蚂蚁的记忆:蚂蚁的记忆存储了关于蚂蚁过去的信息。这些记忆的信息可以用于计算所解决方案的价值或每一步移动的贡献。在一些组合优化问题中,利用蚂蚁的记忆可以正确引导蚂蚁构建方案的解。例如在TSP问题中,利用蚂蚁的记忆可以记录蚂蚁已经走过的城市,并将其置于一个禁忌表中。这样就避免蚂蚁重复访问这些城市,使得蚂蚁构造出来的解可以满足TSP问题的约束条件。所以,蚂蚁可以只使用关于局部的信息,就能构造出问题的解。

(3) 释放信息素:我们已经知道蚂蚁之间的协作是通过信息素来完成,信息素给蚁群算法产生了正反馈的重要作用。那么,蚂蚁何时释放信息素以及释放信息素的量,由具体问题的特征来决定的。蚂蚁可以在构造解的同时逐步释放信息素,也可以构造完一个解后再释放信息素,也可以两种方法相结合。而蚂蚁释放的信息素的量与蚂蚁建立解决方案的优劣程度成正比。

(4) 蚂蚁决策表:蚂蚁决策表是由信息素函数与启发信息函数共同决定的,它是一种概率表。蚂蚁使用这个表来指导其搜索最有吸引力的移动方向。在蚂蚁移动时,不仅基于概率的局部搜索策略,还增加了信息素挥发机制。这样就避免了蚂蚁过早收敛,出现早熟。

蚁群算法就是通过蚂蚁个体功能来构造解,并释放信息素来协调,最终找到一个较优的解决方案。

3. 蚁群算法的研究现状与存在问题

近年来,关于蚁群算法的研究主要集中在如何改善蚁群算法的性能方面。经过多个领域的研究工作者的改进,已经出现了一些效果较好的改进型蚁群算法。通过这些改进的算法,我们可以看到蚁群算法的优缺点。一些研究者还提出了对未来蚁群算法的展望,对蚁群算法的研究前景充满信心。

3.1 蚁群算法的研究现状

自从1996年,M.Dorigo等人提出基本蚁群算法,并应用其求解TSP问题,取得较好效果之后,许多研究者开始研究如何改进蚁群算法的性能。这里,我们就对一些效果较好的改进型蚁群算法进行论述。

3.1.1 最早蚁群系统的改进

1996年,M.Dorigo等人提出的蚁群系统^[3](Ant Colony System, ACS),改善了最早提出的蚁群算法模型(蚂蚁系统)。蚂蚁系统能够在合适的时间内找到较好的解,但是不适合用于求解大规模的复杂优化问题。M.Dorigo等人在蚂蚁系统的基础上做了以下三个方面的改进。

(1) 提供了一个更好的状态转移选择策略。在蚂蚁系统中,蚂蚁的状态转移策略完全依靠概率进行选择。而在蚁群系统中,蚂蚁使用了不同的选择策略,称为伪随机比例规则^[5]。这个规则使用了一个参数 Q_0 ,来调节蚂蚁探索新路径的程度。

(2) 全局更新只应用于当前最优的蚂蚁路径上。如果在蚁群系统中对所有蚂蚁构造出来的解都进行更新,那么就会大大降低搜索最优路径的效率,使得蚁群不能很快集中到最优的路径上来。因此,现在我们只对最优的蚂蚁路径上信息素进行更新。其他路径由于挥发机制信息素会逐渐减少,从而进一步

增大了最优路径和最差路径之间信息素的差异。这样,就会使得蚂蚁选择更倾向于最优路径的选择,从而提高蚁群的搜索效率。

(3) 在构造解的过程中,应用局部信息更新信息素。在蚁群系统中,蚂蚁在构造路径的同时进行局部更新,而在每次循环后再对路径进行一次全局的更新。

将改进后的蚁群系统应用于TSP问题的求解,可以取得很好的效果。

3.1.2 最大-最小蚂蚁系统^[3]

针对蚁群算法容易出现早熟收敛行为,研究者提出最大-最小蚂蚁系统(Max-Min Ant System, MMAS)。MMAS与ACS有许多相似之处,比如全局更新只用于当前最优的蚂蚁路径上。但在MMAS中,为了避免早熟收敛,在每个解的元素上的信息素的浓度被限制在 $[t_{\min}, t_{\max}]$ 区间内。这样就避免了路径上的信息素被过度增加或过度挥发,使得蚂蚁能更好地搜索更优的解。另外在MMAS中,将信息素浓度初始化为 t_{\max} ,使得蚂蚁在初始阶段更多地搜索新的解决方案。

在MMAS中,应用了一种信息素平滑化(Pheromone Trail Smoothing)机制^[5]。这种机制的基本思想是,通过增加选择信息素浓度较低路径的概率,以提高探索新解的能力。这种平滑机制有助于对解进行更有效的探索。同时,这种机制可以降低MMAS对信息素下限的敏感度。

将改进后的最大-最小蚂蚁系统应用于TSP问题的求解,可以取得比其他蚁群更好的效果。

3.1.3 多重蚁群算法^[3]

基于基本的蚁群算法来求解TSP问题,研究者提出将基本的蚁群算法扩展到多重蚁群算法。这种蚁群算法由几个蚁群来协同解决TSP问题。这里我们将一重蚁群定义作一个蚁群群体层。在多重蚁群算法中,蚁群群体层的交互作用使用了正负两种信息素效应。这样,蚁群就能更好地交换问题解决过程的规划信息,并保持它们在搜索过程中的多样性。

引入了多重蚁群之后,几个蚁群群体层的相互作用,能找到更好的解。在某些时间点上,各蚁群群体层可以相互交换好的解的相关信息。如果交换的信息量适当,则多重蚁群算法就能在不同的处理器上分配不同的蚁群,从而轻易地实现并行处理。有研究者详细研究了在蚁群间具有不同信息交换方式的多重蚁群算法的性能表现,而且比较了多起点单蚁群算法中不同数量的蚁群行为,并以TSP等问题作为测试问题,取得了较好的效果。

3.1.4 具有变异特征的蚁群算法^[3]

蚁群算法是一种新型的模拟进化算法,初步的研究已经表明该算法具有许多优良的性质。但该算法也存在一些不足之处,比如计算时间较长、执行复杂度较高等。吴庆洪等人提出了具有变异特征的蚁群算法^[6]。也就是在基本蚁群算法中可引入变异机制,使得该蚁群算法具有较快的收敛速度,提高了算法的运行效率。该算法采用了逆转变异方式,随机地进行变异,以增大进化时所需的信息素。这种变异机制充分利用了2-opt方法简洁高效的特点^[6],因而具有较快的收敛速度。经过这种变异算子作用后,这一代解的性能会有明显改善,从而也能改善整个群体的性能,减少计算时间。该算法已经通过了计算机仿真的验证,结果表明该方法可以较好地解决以上的不足。

3.1.5 自适应蚁群算法^[3]

蚁群算法加速收敛和早熟停滞现象之间存在矛盾,针对这个问题,有研究者提出了一种的自适应蚁群算法。这种算法可以在加快收敛和避免早熟、停滞现象之间取得很好的平衡。该算法可以根据优化过程中解的分布平衡度,自适应地调整路径选择概率的确定策略和信息量更新策略。

自适应蚁群算法,提出了全局更新信息素的规则,并引入蚁群中蚂蚁分工与协同学习、协同工作的思想。这些改进,可以大大提高蚁群算法的自适应能力。

自适应蚁群算法主要有:基于调节信息素挥发度的自适应蚁群算法、具有分工的自适应蚁群算法以及基于协同学习机制的蚁群算法等^[5]。

目前,自适应蚁群算法在求解TSP问题中,已经得到了很好的效果。

3.2 蚁群算法的优点

(1) 蚁群算法能较好地解决复杂优化问题。蚁群算法是一种结合了分布式计算、正反馈机制和贪婪式搜索的算法。在求解复杂优化问题中, 蚁群算法具有很强的搜索较优解的能力。正反馈机制, 使得蚁群可以快速地发现较优的解。分布式计算避免了蚁群算法出现早熟收敛。而贪婪式搜索有助于在搜索过程中早期就找出可接受的解, 提高系统的运行效率。

(2) 蚁群算法具有很强的并行性。蚁群算法适用于并行操作, 在求解复杂优化问题时, 不仅可以从算法自身的改进出发来提高求解效率, 也可以从算法的执行模式出发来进行优化。

(3) 蚁群算法具有很好的可扩充性。因为蚁群算法可以不通过蚂蚁个体之间直接通信, 而是通过信息素进行间接通信, 所以蚁群算法就具有很好的可扩充性。

(4) 蚁群算法具有较强的鲁棒性^[3]。对于蚁群算法模型稍加改进, 就可以应用于其他问题中。在研究蚁群算法的过程中, 许多研究者都通过了求解 TSP 问题来研究蚁群算法。当然蚁群还可以应用到其他多种领域的问题, 比如本文将蚁群算法应用在聚类分析上。

(5) 蚁群算法还具有易于与其他方法融合的特性。如可以将蚁群算法与遗传算法^[7]和模拟退火算法^[8]相结合。而且这种融合可以改善算法的性能, 是蚁群算法优化方法中的一种。

3.3 蚁群算法的缺点

(1) 对于大规模复杂优化问题, 蚁群算法需要较长的搜索时间^[3]。蚁群中蚂蚁个体的运动是随机的, 虽然通过信息素的间接协调, 能够让蚁群向较优的路径, 但是当问题规模很大时, 蚁群很难在较短时间内从大量杂乱无章的路径中找出一条较好的路径。这主要是因为, 在蚁群搜索早期, 各个路径上信息量相差不明显。虽然有正反馈机制, 但是需要经过很长一段时间, 才能使得较好路径上的信息素浓度明显高于其他路径上的信息素浓度。所以对于大规模复杂优化问题, 算法的收敛需要较长时间。

(2) 蚁群容易出现停滞现象, 出现早熟收敛^[5]。也就是当搜索进行到一定时间后, 所有蚂蚁个体所发现的解完全一致, 不能进一步对解进行搜索, 所以不利于发现更好的解。因此很多研究者对蚁群算法进行改进, 以期望避免算法早熟收敛, 以搜索更优的解。

(3) 蚁群算法还没有严格的数学解释^[5]。目前还未能提出一个完善的理论分析, 对蚁群算法的有效性进行论证。蚁群算法作为一种模拟进化算法, 其研究也才刚刚开始, 有许多问题有待进一步的研究, 如算法的收敛性和理论依据等。

3.4 蚁群算法的展望

针对蚁群算法的优缺点, 研究者提出了许多解决的方案, 但对于蚁群算法的优化研究也才刚刚开始。而这些初步的研究已经在求解复杂优化问题上体现了算法的优越性。种种的实验结果表明, 蚁群算法是一种很有发展前景的算法。

目前, 蚁群算法主要有以下几个值得研究的课题:

(1) 对于蚁群算法, 重要的就是开发出求解问题的算法模型^[3, 5], 使求解问题更加切实有效。而在工程实际的应用中, 算法模型的收敛性和算法的复杂度是值得研究的问题。

(2) 利用蚁群算法的全局性避开了局部最优^[3], 利用局部搜索算法加快了求解的过程, 寻求两者的完善结合, 将成为一个值得研究的课题。

(3) 蚁群算法和其他模拟进化算法的结合^[5], 形成混合模拟进化算法模型, 也会使求解问题变得更加有效。

(4) 从复杂系统和复杂性科学的角度看^[3, 5], 蚁群系统是一个复杂系统。蚁群系统寻找从蚁穴到食物源最短路径的过程, 是整个蚁群之间相互作用、相互影响、相互联系、相互协调的结果。蚁群算法本质上具有概率搜索的特征。因此有研究者提出, 可以将蚂蚁视为微观粒子, 利用量子力学波函数的方法研究某时刻粒子坐标的概率振幅。也可以将蚂蚁为图论中的节点, 建立蚁群算法的图论模型, 进而分析其在不同问题中约束条件及其之间的关系。对于蚁群算法理论问题的深入研究, 可以为推广蚁群算法的应用领域奠定重要的理论基础。

4. 简单蚁群算法原形系统的设计

理解了蚁群算法的基本原理，并全面了解了蚁群算法的研究现状和存在问题后，我们就可以开始设计一个简单的蚁群算法原形系统。

4.1 蚁群算法原形系统的设计思想

通过对蚁群算法原理的理解，我们可以将蚁群算法原形系统的伪代码设计如表4-1所示。

表 4-1 蚁群算法原形系统伪代码

<pre> Procedure: ACO Algorithm Begin While (ACO has not been stopped) do Schedule activities Ant's allocation and moving (蚂蚁的分配和移动); Local pheromone update (局部信息素的更新); Global pheromone update(全局信息素的更新); End schedule activities End While End Procedure </pre>

蚁群算法的整个过程包含三个部分：蚂蚁的分配和移动、局部信息素的更新和挥发以及全局信息素的更新。

(1) 蚂蚁可以随机分配也可以根据实际问题分配。分配完蚂蚁之后，就是蚂蚁的移动。在蚂蚁的移动过程中，蚂蚁 k 根据公式 (4-1) 的规则在时刻 t 由节点 i 移动到下一节点 j 。

$$s = \begin{cases} \arg\{\max_u x_u = allowed_k(t)[t_{iu}(t)^a h_{iu}^b]\} & q \leq Q_0 \\ S & \text{其他} \end{cases} \quad (4-1)$$

其中 $t_{iu}(t)$ 表示在 t 时刻的信息素浓度， h_{iu} 是特定问题的启发信息， a 表示信息素浓度重要性的参数， b 表示启发信息重要性的参数。 q 是一个均匀分布在 $[0, 1]$ 区间上的随机数， Q_0 是预先确定的参数 ($0 \leq Q_0 \leq 1$)， $allowed_k(t)$ 表示了 t 时刻蚂蚁可以选择的节点。 S 也是从 $allowed_k(t)$ 中选择出来的节点的索引，它是通过下面的概率公式 (4-2) 从 $allowed_k(t)$ 中选择出来的节点。

$$p_{ij} = \begin{cases} \frac{[t_{ij}]^a [h_{ij}]^b}{\sum_{u \in allowed_k(t)} [t_{iu}]^a [h_{iu}]^b}, j \in allowed_k(t) \\ 0 & \text{其他} \end{cases} \quad (4-2)$$

(2) 局部信息素的更新和挥发是通过公式 (4-3) 来完成的。其中， $0 < \Psi \leq 1$ 是一个常数， t_0 是信息素浓度的初始值。应用公式 (4-3) 的信息素更新规则，可以使得蚂蚁对那次选择的路径的吸引力越来越小。这样就是真正起到了信息素的挥发作用。

$$t_{ij}(t+1) = (1 - \Psi)t_{ij}(t) + \Psi \times t_0 \quad (4-3)$$

(3) 全局信息素的更新按照公式 (4-4) 规则进行更新。其中， $0 < r \leq 1$ 是控制信息素衰减过程

的一个参数， $\Delta t_{ij}(t) = Q/F$ ， F 是当前问题目标函数的最优值。

$$t_{ij}(t+1) = (1-r)t_{ij}(t) + r\Delta t_{ij}(t) \quad (4-4)$$

4.2 蚁群算法原形系统的设计流程

蚁群系统是一个简单的蚁群算法原形系统，本文已经论述了它在三个方面的改进。现在，我们就这个系统进行设计。

蚁群系统的工作流程可以表述如下，根据实际问题设置好系统的参数，并根据一些初始化规则，分配蚂蚁的位置。接着蚂蚁通过重复地应用状态转移规则，构造解的路径。在构造路径的同时，蚂蚁也通过应用局部更新规则来修改已访问路径上的信息素浓度。当所有蚂蚁都完成了它们的路径，找出这些蚂蚁中最优解的路径，并应用全局更新规则再次对该路径上的信息素浓度进行修改。这些规则可以让蚂蚁受启发信息和信息素信息的指导，朝着信息素浓度高的路径移动。简单说，这些规则可以让更好的路径上有更多的蚂蚁选择。

本系统的设计流程如图4-1所示。

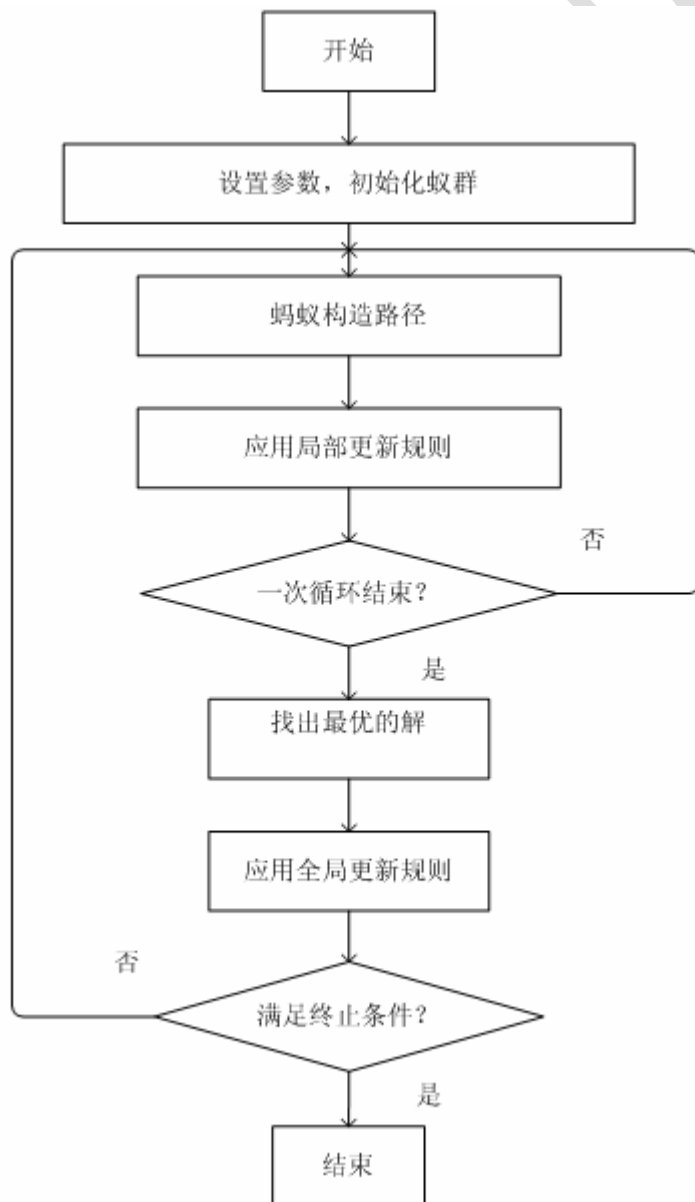


图4-1 简单蚁群原形系统流程图

4.3 蚁群算法原形系统的性能分析

由于蚁群算法的性能要根据实际问题才能分析,所以这里我们就根据最常见的旅行商问题进行讨论。我们所设计的蚁群算法原形系统是一种结构上的启发算法模型。在旅行商问题中,蚂蚁每一步选择转移城市之前都要考虑所有可能的城市集合,其时间复杂性为 $O(n^2)$ 。所以,系统的大部分运行时间都用于计算可选城市的效用。因此,如果构造的城市子集合不合适,将使得蚂蚁花费很长时间从子集合中选择要访问的城市。为了提高系统的搜索效率,尤其对于大规模复杂优化问题,可以将选择城市的数量限制在一个合适的子集或候选表内,该方法也被称为候选集合策略。实验表明,该方法能够在1500个城市以上的旅行商问题中找出好的解决方案^[5]。生成一个解决方案所需要的时间随着城市数量增加稍微大于线性的增长,比起没有候选表时产生的二次增长所需要的时间要少得多。

我们还知道这样的系统容易出现早熟收敛或者运行时间过长等问题。这些问题主要是信息素的更新规则选取不当所造成。

4.4 蚁群算法原形系统的优化改进

许多的研究表明,这样的蚁群算法原形系统具有很强的发现较好解的能力,并且一般不容易陷入局部最优。但是,这种系统模型也存在着一些缺陷,如:需要较长的搜索时间,当问题规模较大进也容易陷入局部最优,产生早熟收敛。蚁群中个体的运动是随机的,在系统运行的初期,各个路径上的信息素浓度相差不明显,通过正反馈机制,使得较好路径上的信息素浓度能够逐渐增大。但是当群体规模较大时,就很难在较短的时间内从大量杂乱无章的路径中找出较好解的路径。

蚁群算法原形系统的优化改进,主要可以从以下三方面出发:

(1) 从系统的总体结构出发:在这里,我们需要考虑的是蚁群系统的总体结构和组织模式。还需要考虑到系统中各个子部分的信息联系模式。比如要将该系统扩展到多群体蚁群系统和混合蚁群系统时,就需要几个蚁群协同解决问题。我们在文中已经论述过了关于蚁群群体层之间的信息交换。而这些改进就是从系统的总体结构出发,进行的优化。

(2) 从系统的具体参数设定和调整策略出发:在这里,我们需要对系统参数的选择和变化模式进行设定。我们已经知道的最大最小蚁群算法就是对信息素浓度加以限制的改进方法。我们还知道在基本蚁群算法中引入变异机制,对参数进行变异式改进^[5, 6],可以使算法既有较快的求解速度又有较高的求解精度。自适应蚁群算法、动态蚁群算法就是一种分布平衡的蚁群算法,可以在加速收敛和防止早熟、停滞现象之间取得很好的平衡。也可以将随机特征引入到算法参数的设定上,这样就可以形成随机扰动蚁群算法。还可以采用基于目标函数值的信息素分配策略,根据目标函数数值来自适应调整蚁群的搜索方向。通过许多研究者的验证表明,这些改进方案可以较好地优化系统的性能。

(3) 从与其他智能算法相结合出发:我们知道蚁群算法具有容易与其他智能算法相结合的优点,所以可以通过结合,达到算法之间优势互补的功效。

5. 基于蚁群算法的聚类算法

蚁群算法可以应用到多种问题领域,现在我们就将以上设计的原形系统应用到聚类分析中^[9],并验证其性能。

5.1 K均值聚类算法的基本思想^[4]

我们知道“物以类聚,人以群分”这一现象,聚类分析就是从这里出发的。它是根据不同对象之间的差异和特定的准则作模式样本的分类。聚类分析是一种数学分析方法,是统计学的一个分支,也是无导师监督的机器学习方法,属于NP难问题。

聚类分析是知识发现的重要方法,在图像识别、信息检索、数据挖掘、统计学、机器学习、空间数

据库、生物学以及市场营销等领域有着广泛的应用。

目前常用的简单聚类算法是K均值聚类算法^[4]。K均值算法是一种基于划分方法的聚类算法，它是最早提出的较为经典的聚类算法之一。K均值聚类算法的主要思想是：对N个对象给出K个划分，其中每个划分代表一个簇。首先，随机地选择K个对象，每个对象初始地代表一个簇的中心（平均值）。对剩余的每个对象，根据其与各个簇中心的距离，将它赋给最近的簇。然后重新计算每个簇的平均值，对所有模式样本与每个簇的平均值相比较，把对象赋给最相似的某个簇。不断重复这个过程，直到簇中的对象都是“相似的”，而不同簇中的对象都是“相异的”，即目标函数收敛使平方误差函数值最小。通过K均值算法的聚类，可以将模式样本（见图5-1）划分成几个簇，聚类结果见图5-2。

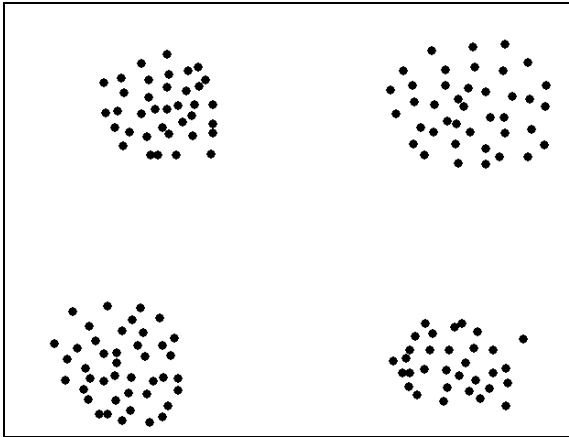


图5-1 待聚类的模式样本

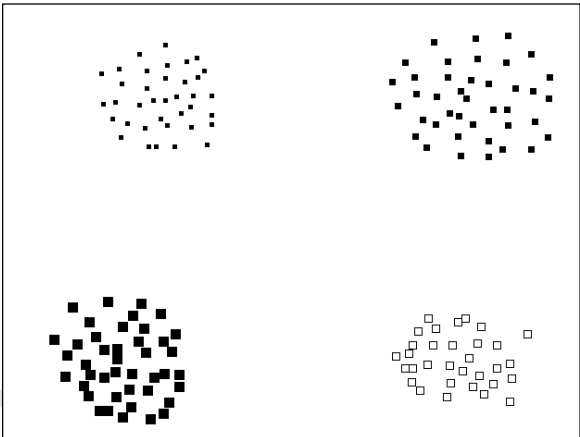


图5-2 应用K均值聚类后的各个簇

K均值算法的主要流程如表5-1所示。

表5-1 K均值聚类算法

算法：K均值聚类
输入：簇的数目 K 以及模式样本总数 N
输出：类内距离最短的 K 个簇
1. 任意选择 K 个样本作为初始的簇中心；
2. repeat
4. 按照样本与簇中心距离最短选择样本所对应的簇；
5. 计算簇的平均值，得到新的 K 个簇的中心；
3. until K 个簇的中心稳定；

5.2 当前K均值聚类算法存在的主要问题

应用K均值算法来聚类时，当最终簇是密集的，而簇与簇之间区别明显时（见图5-3），它的效果较好（见图5-4）。

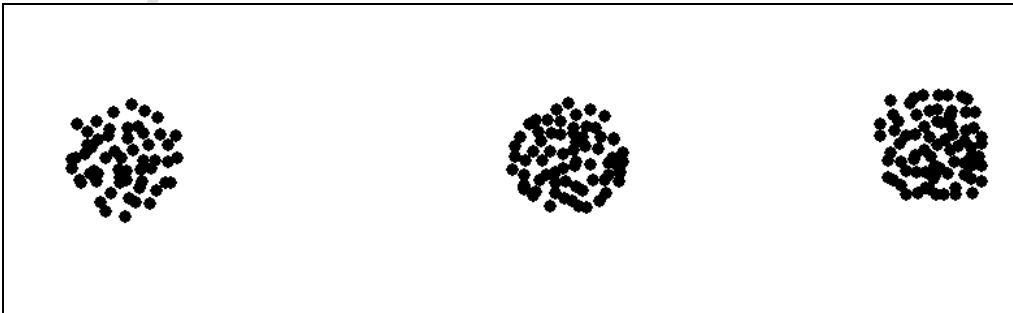


图5-3 簇密集且簇与簇之间区别明显的模式样本

应用K均值出现这样正确的划分几乎是肯定的。对于处理大数据集，K均值算法具有相对可伸缩的

和高效率性。

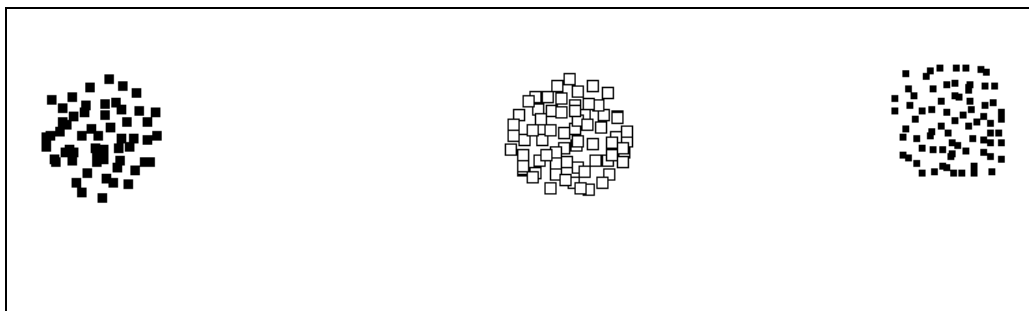


图5-4 应用K均值聚类后的各个簇

但是，对于K均值算法还存在一些不足的地方。

(1) 必须事先给出要生成簇的个数K。由于在实际问题中，大量的模式样本不可能明确知道其具体可能生成簇的个数。所以K均值不能很好地处理未知簇的个数的聚类问题。

(2) K均值聚类算法的性能与初始簇中心的选取有很大关系。当初始时，簇的中心选择不恰当可能使得聚类出来结果不符合实际（见图5-5所示）。由于初始簇中心选择不恰当，使得程序容易陷入局部最优，而造成停滞。

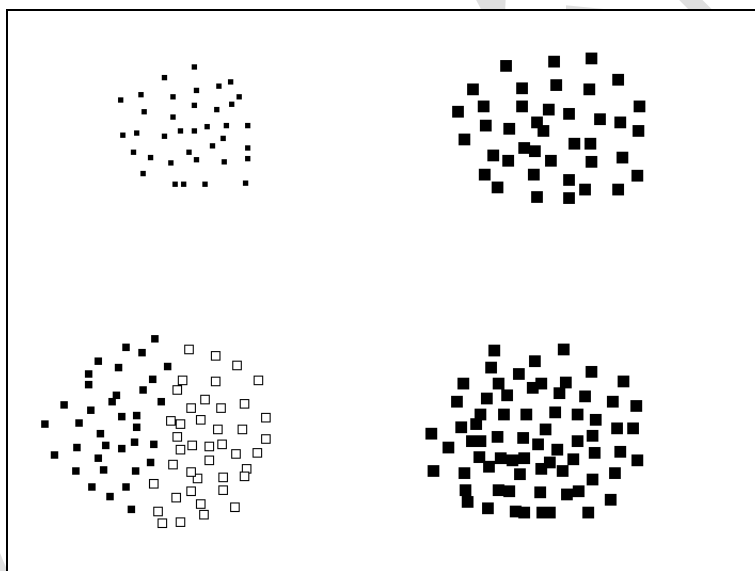


图5-5 初始簇的中心选择不恰当所造成的聚类错误

(3) K均值聚类算法对于“噪声”和孤立点数据比较敏感，少量的该类数据能够对平均值产生极大的影响。在图5-3中加入一些孤立点，并用K均值进行聚类，可以看到聚类结果出现错误（见图5-6所示）。

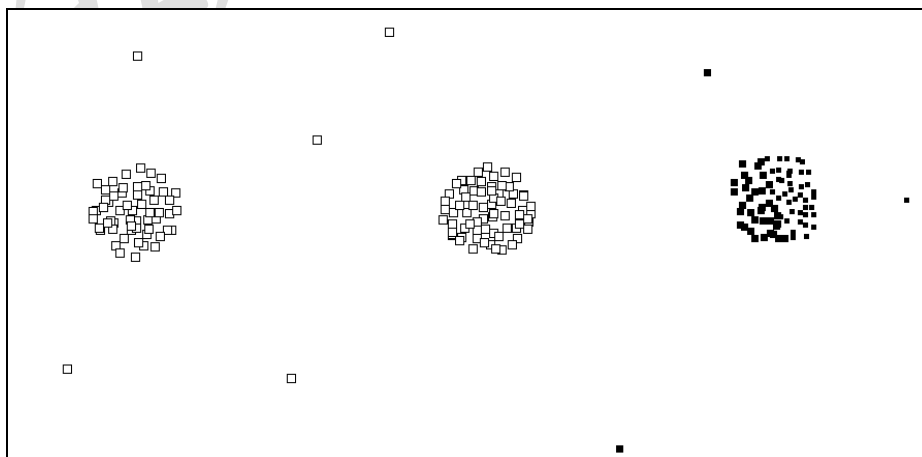


图5-6 少量孤立点所造成的聚类错误

5.3 应用蚁群算法原形系统优化K均值聚类算法

针对K均值聚类算法的不足，可以利用蚁群算法相结合，来改进算法的聚类效果。本文将应用上述所设计的简单蚁群算法原形系统优化K均值聚类算法^[12]，并通过实验验证其性能。

5.3.1 蚁群聚类算法的设计方案

由于K均值聚类算法容易受到初始信息以及少量孤立点的影响^[12]，而造成聚类结果出现偏差。对此我们引入蚁群算法原形系统对K均值算法进行改进。

首先根据聚类问题，定义变量（如表5-2所示）。

表 5-2 变量定义

N ：	模式样本总数；
M ：	蚂蚁总数量；
V ：	模式样本的属性个数；
K ：	聚类中心个数；
S ：	存放蚂蚁构造的解（长度为 N ）；
NC ：	迭代次数；
t ：	存放信息素的 $N \times K$ 矩阵；
w ：	$N \times K$ 的矩阵，其元素 $w_{ij} = \begin{cases} 1, & \text{如果 } x_i \in \text{Cluster}_j \\ 0, & \text{如果 } x_i \notin \text{Cluster}_j \end{cases}$
x ：	存放模式样本的数据信息， x_{iv} 是样本 i 的第 v 个属性值；
m ：	$K \times V$ 的聚类中心的矩阵， m_{jv} 表示类 j 中所有样本的属性 v 的均值；
p ：	$N \times K$ 的矩阵，用于存放蚂蚁选择聚类中心的概率；
d ：	$N \times K$ 的矩阵，用于存放各样本到各聚类中心的距离；
h_{ij} ：	表示样本到聚类中心的能见度， $h_{ij} = \frac{1}{d_{ij}}$ ；

其中， m_{jv} 按照公式（5-1）计算。这样就可以得到各个簇的中心位置。接着我们可以定义聚类目

标函数如公式（5-2）所示。该公式满足：

$$\begin{cases} \sum_{j=1}^K w_{ij} = 1, i = 1, \dots, N \\ \sum_{j=1}^K w_{ij} \geq 1, j = 1, \dots, K \end{cases} \quad \text{条件。}$$

$$m_{jv} = \frac{\sum_{i=1}^N w_{ij} x_{iv}}{\sum_{i=1}^N w_{ij}}, j = 1, \dots, K, v = 1, \dots, V \quad (5-1)$$

$$F(w, m) = \sum_{j=1}^K \sum_{i=1}^N \sum_{v=1}^V w_{ij} \|x_{iv} - m_{jv}\|^2 \quad (5-2)$$

有了以上定义之后，我们就可以开始研究蚂蚁构造解的策略。根据蚁群原形系统，蚂蚁移动的选择策略按照公式（4-1）进行。对此我们结合聚类问题，将此规则修改成公式（5-3）。

根据公式（4-2），将下面的概率公式（5-3）作为 S 选择所属的簇的概率。

$$s = \begin{cases} \text{选择 } \max_{j \in R} \{[t_{ij}]^a [h_{ij}]^b\}, & \text{如果 } q \leq q_0 \\ S, & \text{否则} \end{cases} \quad (5-3)$$

$$p_{ij} = \frac{[t_{ij}]^a [h_{ij}]^b}{\sum_{k=1}^K [t_{ik}]^a [h_{ik}]^b}, j = 1, \dots, K \quad (5-4)$$

接着，我们开始设计基于蚁群算法的K均值聚类算法，其设计流程如图5-7。

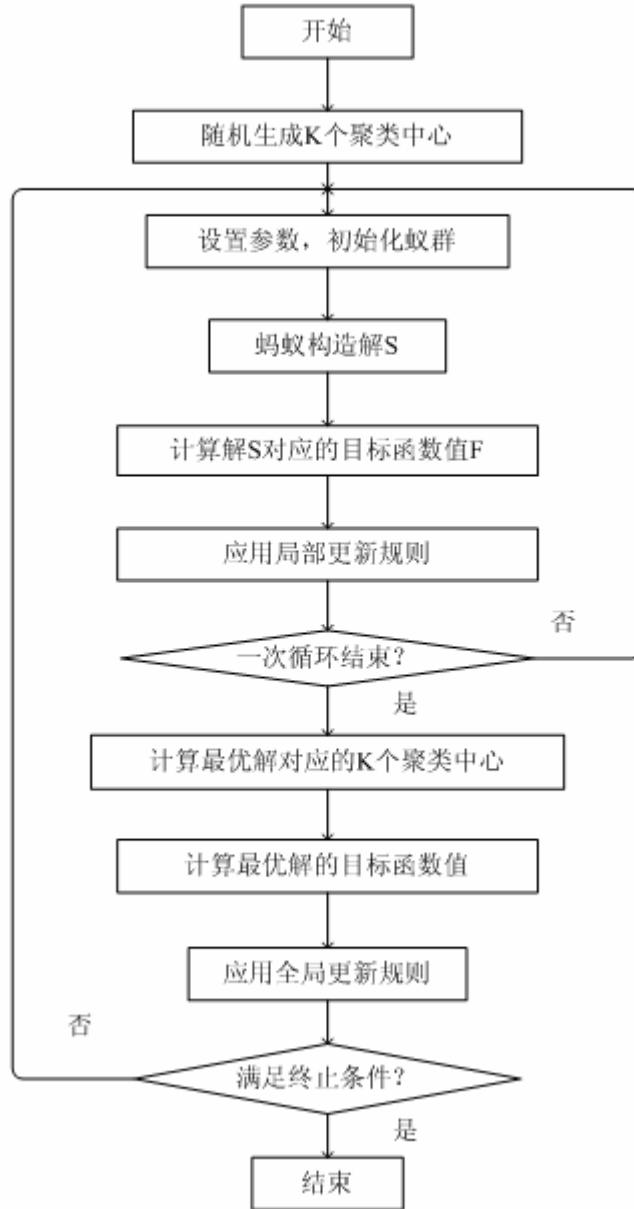


图5-7 基于蚁群算法的K均值聚类算法流程

K均值聚类算法需要事前输入簇的个数K，这里我们就不改进这个。对于算法的初始化，可以用随机产生K个簇中心的方法，也可以应用随机构造解来初始化簇的中心。这里我们应用随机构造解的方法来初始化。随机构造解的方法是通过将所有模式样本随机地附给所属的簇。

每个蚂蚁都通过随机构造解，来产生初始簇，应用公式（5-1）计算各个蚂蚁的簇中心，并根据目标函数（5-2）计算其目标函数值。找出目标函数值最小的蚂蚁所构造出的解，并将该蚂蚁的簇中心作为初始的簇中心。

接着，需要设置参数，在该算法中主要的参数如表5-3所示。设置好参数，蚂蚁就可以按照我们设计的规则构造解。当所有蚂蚁完成一次循环后，找出最优的解。并将最优解所对应的聚类中心作为当前

最优的聚类中心，并计算当前最优的目标函数值。根据公式（4-4）我们进行全局信息素更新。以上过程一直重复，直到满足结束条件为止。

表 5-3 参数定义

a ：	表示信息素浓度重要性的参数；
b ：	表示启发信息重要性的参数；
Ψ ：	局部信息素更新参数（ $0 < \Psi \leq 1$ ，本算法暂时不用局部更新，以简化算法）；
r ：	用来控制信息素衰减过程的一个参数（ $0 < r \leq 1$ ）；
Q_0 ：	用来控制按概率转移的阈值（ $0 \leq Q_0 \leq 1$ ）；
M ：	蚂蚁的数目。

5.3.2 蚁群聚类算法的实现过程

根据蚁群聚类算法，我们先设计一个蚂蚁类，如表5-4所示。该类定义了蚂蚁在K均值聚类问题中的一些基本功能。

表 5-4 蚂蚁类 CAnt 的定义

class CAnt	
{	
public:	
double F;	//各样本点到各自聚类中心的距离和
int *S;	//存放蚂蚁构造的解
CDoublePoint *Center;	//聚类中心
CPoint **Cluster;	//存放K个簇的栈
int *Top;	//各个簇的样本个数
void Init_Ant(int i,int j,CPoint*);	//初始化蚂蚁
int N;	//模式样本总数
int K;	//聚类中心个数
CPoint *AllData;	//存放N个样本数据信息
void Random_solution();	//随机分配解
void Update_center();	//更新聚类中心
void Add_all_toCluster();	//将所有的样本添加到相应的簇
void Fun();	//计算目标函数的值
void Out_cluster();	//输出各个簇
virtual ~CAnt();	//释放蚂蚁
};	

根据K均值聚类算法的特点，结合蚁群算法原形系统的设计模式，设计一个关于K均值的蚁群聚类算法系统类ACOKMeans（如表5-5所示）。该类包含了蚁群聚类所需要的一些功能以及参数。其中主要包括了：

- （1）各个参数的设置
- （2）目标函数的计算；
- （3）聚类中心的计算；
- （4）蚂蚁解的构造；

- (5) 全局信息素的更新;
- (6) 蚂蚁转移策略;
- (7) 最优解的选取;
- (8) 结果的输出。

表5-5 K均值蚁群聚类系统类CACOKMeans的定义

```
class CACOKMeans
{
public:
    double ALPHA;// a 参数

    double BETA;// b 参数

    double RHO; // r 参数

    double Q0; //  $Q_0$  参数

    int M;//蚂蚁数量
    int N;//模式样本总数
    int K;//聚类中心个数
    long double MinF;//存放最优目标函数值
    CPoint *AllData;//存放N个样本数据信息
    double **TAU;//存放信息素浓度
    CDoublePoint *OutCenter;//最优解聚类中心
    CPoint **OutCluster;//存放最优解K个簇的栈
    int *OutTop; //当前最优解簇的样本个数
    CACOKMeans(double rho,double alpha,double beta,double q0,int m);//初始化蚁群聚类系统
    void InitData(); //初始化数据
    void Init_tau(CAnt*);//初始化信息素TAU
    int Init_Center(CAnt*);//初始化最优簇中心
    double ETA(int i,int j); //计算ETA的值
    void BuildSolution(CAnt*,int i);//蚂蚁构造解
    int Min_F(CAnt*); //找出目标函数最小的蚂蚁
    void Global_update_rule(CAnt*,int);//全局更新信息素
    double Transition(int i,int j); //计算公式 (5-4) 分子的值
    double Sum_transition(int i); //计算公式 (5-4) 分母的值
    int Choose(int);//选择某个簇
    int MaxP(int); //选择概率最大的簇
    void Add_all_togoodCluster(CAnt*,int m);//将最优解所有的样本添加到相应的簇
    void Add_all_togoodoutCluster();//按照转移概率最大, 将最优解所有的样本添加到相应的簇
    void Out_good_cluster(); //以转移概率最大的输出
    void Update_good_center();//更新最优解的聚类中心
    virtual ~CACOKMeans(); //解散该系统
};
```

根据图5-7的流程, 我们可以实现算法的主要代码如表5-6所示。这些代码就是根据蚁群系统的主要思想来设计的。从代码的实现, 可以看出一个好的蚁群系统, 可以很容易应用到各种具体的问题中去。而且只要对系统稍加改进就可以, 就可以完善许多功能。在这里我们应用蚁群算法搜索最优解的能力来解决聚类问题, 而且可以较好地避免陷入局部最优。

表5-6 K均值蚁群聚类算法主要代码

```

int main()
{
    int NC=100;//迭代次数
    int a,x,i,j;
    srand((unsigned)time(NULL));
    ACOKMeans *aco=new ACOKMeans(0.4,1.0,2.0,0.6,10);//初始化系统
    aco->InitData();//初始化数据
    Ant *ant;
    ant=new Ant[aco->M];//分配蚂蚁
    for(i=0;i<aco->M;i++)//初始化蚂蚁参数
        ant[i].Init_Ant(aco->N,aco->K,aco->AllData);
    aco->Add_all_togoodCluster(ant,aco->Init_Center(ant));//初始化聚类中心
    aco->Update_good_center();//初始化聚类中心
    aco->Init_tau(ant); //初始化信息素
    for(i=0;i<NC;i++)
    {
        for(j=0;j<aco->M;j++)
        {
            //蚂蚁构造解
            aco->BuildSolution(ant,j);
            ant[j].Add_all_toCluster();
            ant[j].Update_center();
            ant[j].Fun();
        }
        a=aco->Min_F(ant);//找出目标函数最小的蚂蚁
        if(a!=-1)
        {
            x=a;
            cout<<i<<" "<<x<<":"<<MinF<<endl;
            aco->Add_all_togoodCluster(ant,x);
            aco->Update_good_center();//更新当前最优聚类中心
            aco->Global_update_rule(ant,x);//全局更新信息素
        }
    }
    aco->Out_good_cluster();//以最大转移概率输出各个簇
    return 0;
}

```

5.3.3 蚁群聚类算法的性能验证

调试好程序，并开始测试其性能：

(1) 如图5-7，当模式样本可能生成的簇之间的距离明显时：我们分别应用K均值聚类算法和蚁群K均值聚类算法对这种情况进行实验。重复运行100次，实验结果显示未改进的K均值聚类算法产生如图5-8的正确聚类划分的概率为75%，产生如图5-9的错误聚类结果的概率为25%。而应用蚁群K均值聚类算法，能得到更大的正确率。经过测试，重复运行100次，正确率达到98%。通过这次测试，我们可以看到K均值聚类算法解决簇之间的距离明显的情况时，虽然正确率还行，但仍然存在一定程度的错误可能。但是应用基于蚁群的聚类算法，可以使得运行结果的正确率提高到接近100%。通过这次测试，我们可以明显看到蚁群算法的优越性。

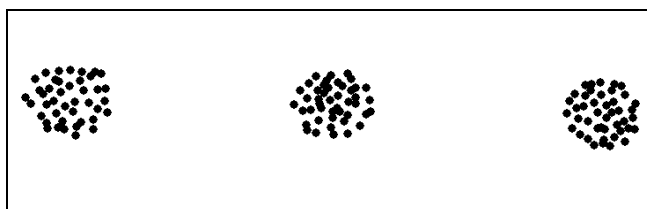


图5-7 簇之间的距离明显的情况

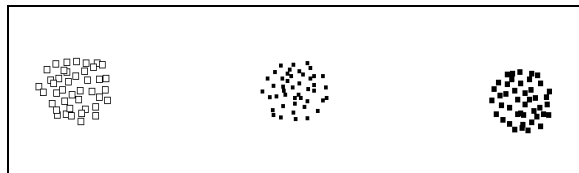


图5-8 正确的聚类效果

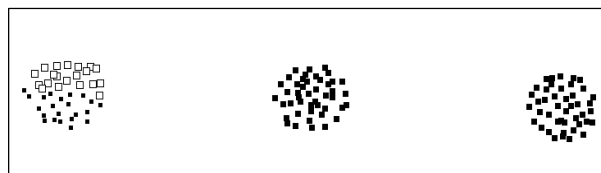


图5-9 产生的错误聚类结果

(2) 当模式样本中可能生成的簇包含的个数差异较大（见图5-10），且簇之间的距离不是很明显时：我们分别用K均值聚类算法和蚁群K均值聚类算法对这种情况进行实验。实验结果表明未改进的K均值聚类算法容易产生如图5-11的错误聚类划分。经过测试，重复运行100次，错误率高达68%。而应用蚁群K均值聚类算法，能较好地改进算法的性能。经过测试，重复运行100次，错误率降到了45%。应用蚁群聚类算法能提高产生正确的聚类效果，如图5-12。通过这次测试，我们可以看到K均值聚类算法解决簇包含的个数差异较大的情况时，性能很差。特别是簇之间的距离不是很明显时，会得到更差的结果。但是应用基于蚁群的聚类算法，可以一定程度上提高运行结果的正确率。当然，蚁群算法对K均值算法在这种情况的改进还没有最有效地解决该问题，但是其优化的结果是值得肯定的。

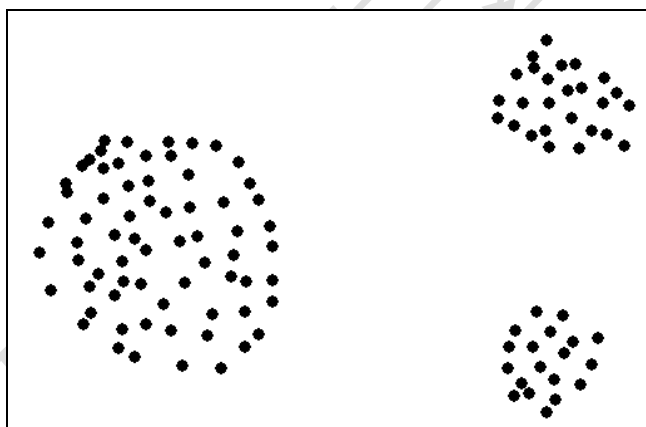


图5-10 簇包含的个数差异较大的情况

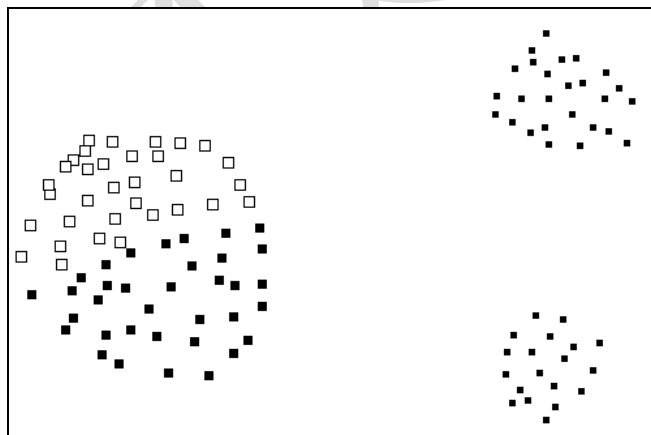


图5-11 产生的错误聚类结果

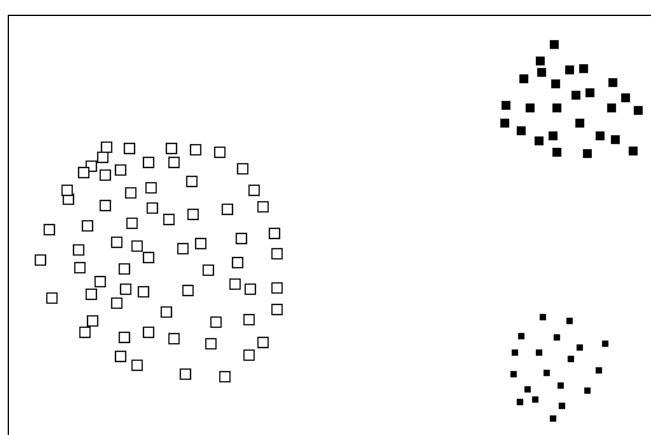


图5-12 正确的聚类效果

(3) 如图5-13，当模式样本中出现了少量的“噪声”和孤立点数据时：我们分别用K均值聚类算法和蚁群K均值聚类算法对这种情况进行实验。实验结果表明未改进的K均值聚类算法容易产生如图

5-14的错误聚类划分。经过测试，重复运行100次，错误率为38%。而应用蚁群K均值聚类算法，能较好地改进算法的性能。经过测试，重复运行100次，错误率降到了2%。

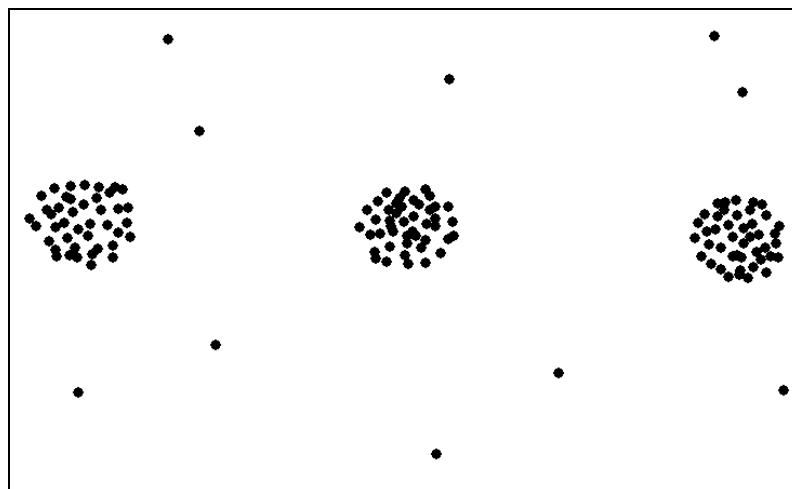


图5-13 出现了少量的“噪声”和孤立点数据的情况

应用蚁群聚类算法能明显提高产生正确的聚类效果，如图5-15。通过这次测试，我们可以看到K均值聚类算法解决模式样本出现少量的“噪声”和孤立点数据的情况时，性能会下降。也就是未改进的K均值聚类算法对孤立点比较敏感，少量的该类数据能够对平均值产生较大的影响。

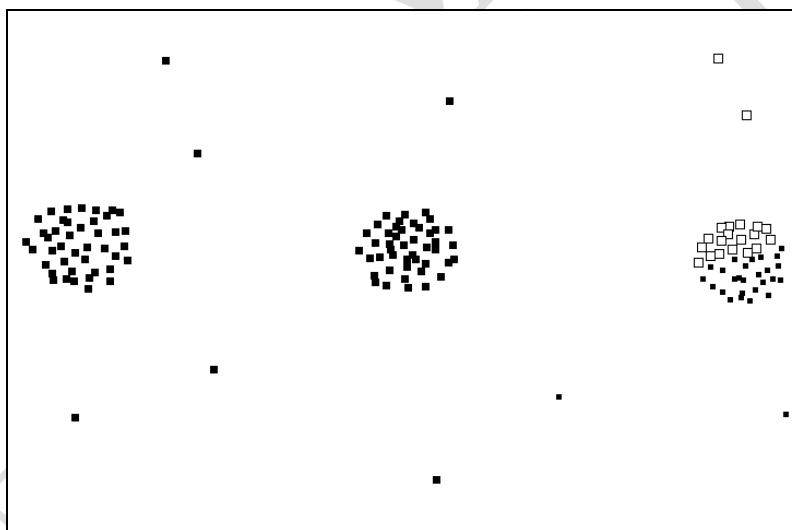


图5-14 产生的错误聚类结果

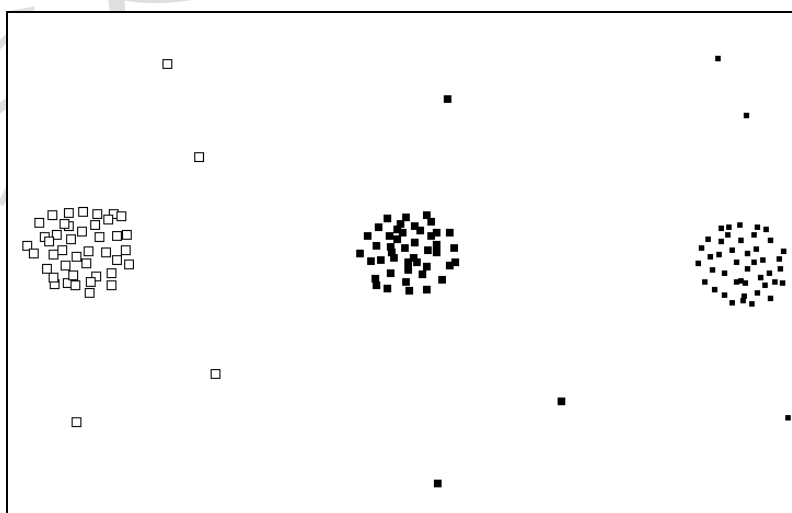


图5-15 正确的聚类效果

但是应用基于蚁群的聚类算法，少量的孤立点数据，不会对平均值产生较大的影响。其正确率依然很高。

(4) 如图5-16，当模式样本中出现“Y”型簇的数据时：我们分别用K均值聚类算法和蚁群K均值聚类算法对这种情况进行实验。实验结果表明未改进的K均值聚类算法，容易产生如图5-17的错误聚类划分。而应用蚁群K均值聚类算法，容易产生如图5-18的错误聚类划分。

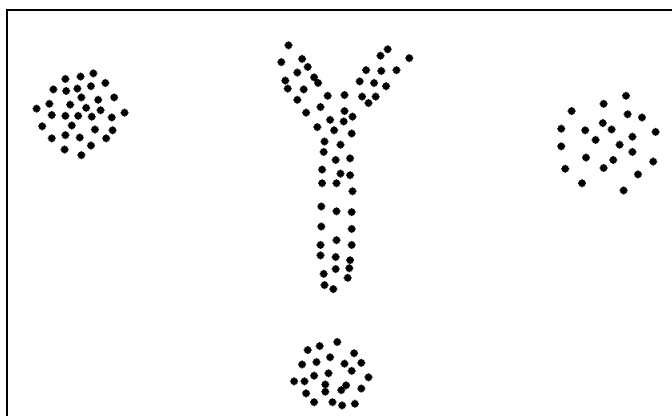


图5-16 含“Y”型簇的情况

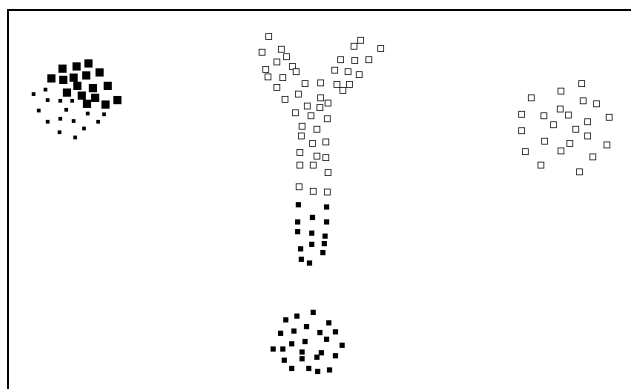


图5-17 K均值聚类算法容易产生的聚类效果

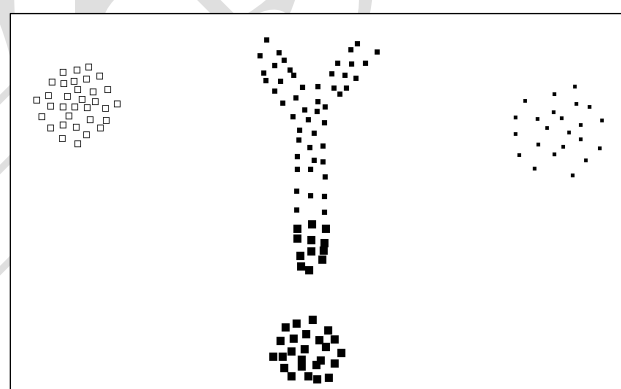


图5-18 蚁群聚类算法容易产生的聚类效果

所以应用蚁群聚类算法未能很好地解决该问题，但是不管怎样我们可以看到应用蚁群聚类算法得到的结果要比K均值得到的结果更符合实际。通过观察我们可以看到，应用蚁群聚类算法时，“Y”型簇的下部容易被归入与它临近的簇。其实这是由K均值聚类算法的特点决定的，由于我们只是在K均值的基础上，应用蚁群算法加以改进，所以未能真正解决这个不足。

如果我们将“Y”型簇的下部去除，如图5-19所示。接着分别应用K均值算法和蚁群聚类算法进行聚类分析。

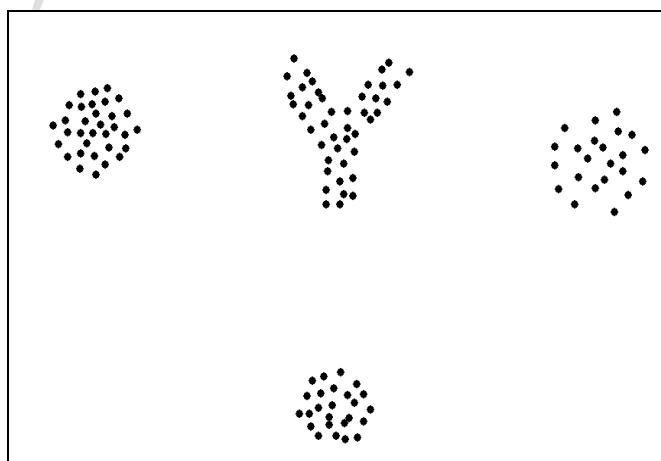


图5-19 K均值聚类算法容易产生的聚类效果

通过这次测试，我们可以看到应用K均值聚类算法解决时，仍然会出现一些错误，这当然是K均值聚类算法的弊病所造成的。但是应用蚁群聚类算法时，就可以很好地解决该问题。所以在处理不规则形状的问题上，蚁群聚类算法比K均值聚类有更好的效果（如图5-20和图5-21）。

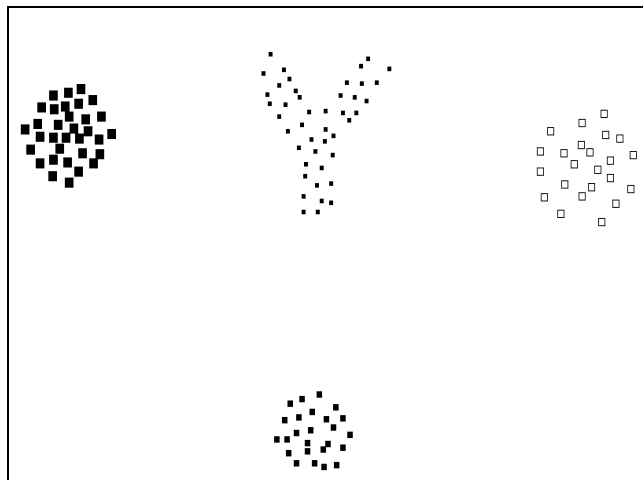


图5-20 蚁群聚类算法产生的聚类效果

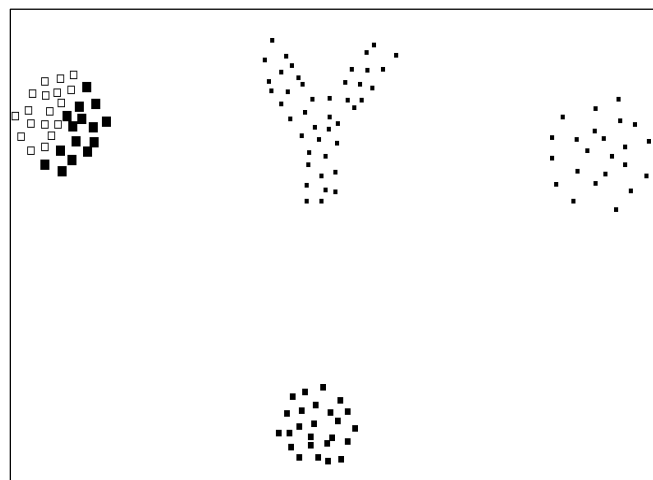


图5-21 K均值聚类算法产生的聚类效果

5.4 蚁群算法原形系统的性能优化与分析

将蚁群算法原形系统应用于K均值聚类方法的优化，并通过实验验证其性能，我们看到了蚁群算法解决复杂优化问题的优越性。当然蚁群算法本身还有一些不足之处，如果可以对这些不足进行优化和改进，那么将会更好地解决实际问题。

对此，我们对蚁群算法原形系统提出以下优化方案：

（1）引入随机扰动机制^[10]，避免大量的无效搜索而使系统出现早熟收敛现象。引入扰动因子，以起到一定的扰动作用。我们可以将信息素最大的那条路径，使用转移概率公式选择，而对于其他路径，则采用随机选择方式。这样就可以将蚂蚁的确定性选择和随机性选择结合在一起。两者的共同作用可以使蚁群算法具有更强的全局搜索能力。

（2）将局部信息素更新与全局信息素更新相结合。使得较优路径上的信息素浓度与较差路径上的信息素浓度相差明显，有利于提高系统的效率。

（3）还可以引入蚂蚁的感觉阈限^[11]，也就是信息素浓度超过这个阈限，蚂蚁才开始选择信息素浓度大的路径。这样，可以让蚂蚁在初始阶段选择较多的不同路径，以获得多样化的解，避免蚂蚁陷入局部最优，让蚂蚁少出现早熟收敛。

6. 总结

蚁群算法是一种最新发展的模拟蚂蚁群体觅食行为的仿生优化算法。该研究成果可以被应用于数据挖掘的多种应用领域，具有一定的研究价值与应用价值。蚁群算法能较好地解决复杂优化问题，具有很强的并行性和可扩充，还具有较强的鲁棒性和易于与其他方法融合的特性。但是，蚁群算法也存在一些不足：不能很好地解决大规模复杂优化问题，出现早熟收敛，蚁群算法没有严格的数学解释等。对此，研究者已经提出了许多蚁群算法的优化和改进的方法。本文通过全面分析蚁群算法的研究现状和存在问题，提出了一个简单蚁群算法的原形系统，并将该系统应用于到聚类分析。通过蚁群算法与K均值聚类算法的结合，来改进K均值聚类存在的一些不足。通过实验测试表明，蚁群算法能在一定程度上改善K均值聚类算法的不足。然而对于蚁群算法的优化研究也才刚刚开始，而这些初步的研究已经在求解复杂优化问题上体现了算法的优越性。种种的实验结果表明，蚁群算法是一种很有发展前景的算法，我们对蚁群算法的研究前景充满信心。

参考文献

- [1] Dorigo, Maniezzo, Colomi. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on SMC, 1996, 26 (1) : 8-41.
- [2] Dorigo, Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem[R]. IEEE Transactions on Evolutionary Computing, 1997, 1(1): 53-56.
- [3] 吴启迪, 汪 镭. 智能蚁群算法及应用[M]. 上海: 上海科技教育出版社, 2004, 3-129
- [4] Margaret H. Dunham. 数据挖掘教程[M]. 北京: 清华大学出版社, 2005, 5: 107-138
- [5] 李士勇. 蚁群算法及其应用[M]. 哈尔滨: 哈尔滨工业大学出版社, 2004, 1-21.
- [6] 吴庆洪, 张纪会, 徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展, 1999, 36(10): 1240-1245.
- [7] Yamazaki T, Komine T, Kawakami T. Geo-technical properties of deep-sea sediments and the in-situ measurement techniques[A]. Proceedings of the Sixth (2005) ISOPE Ocean Mining Symposium [C]. Changsha, Hunan, China, 2005.
- [8] Fan Z J, Koren Y, Wehe D. A Simple traction control of tracked vehicles[A]. Proceeding of the American Conference[C]. Seattle, Washington. 1995.
- [9] Chen MS. Data mining an overview from a database perspective[J]. IEEE Trans on Knowledge and data engineering, 1996, 8 (6) : 866-883
- [10] 郝晋, 石立宝, 周家启. 具有随机扰动的蚁群算法[J]. 系统工程理论与实践, 1999, (3) : 84-87.
- [11] 陈螳, 秦玲, 陈宏建. 具有感觉和知觉特征的蚁群算法[J]. 系统仿真学报, 2003, (11): 1418-1425.
- [12] 李士勇, 赵宝江. 一种蚁群聚类算法[J]. 计算机测量与控制, 2007, 15 (11) 1590-1596

Research and Implementation on Ant Colony Optimization

ZHENG Jia-ming 105032004002 Advisor: HUANG Tian-qiang

Major in Computer Science and Technology College of Mathematics and Computer Science

【Abstract】 In this thesis we discuss the development and problem of Ant Colony Optimization (ACO). According to existed research, we implement a prototype system of ACO. Then we use this system to clustering, and test its performance through our experiments. The result of experiments show that ACO has an advantage in solving complex optimize problem.

【Keywords】 Ant Colony Optimization; clustering analysis; Data mining

福建师范大学本科毕业论文(设计)作者承诺保证书

本人郑重承诺： 本篇毕业论文(设计)的内容真实、可靠。如果存在弄虚作假、抄袭的情况，本人愿承担全部责任。

签名： 郑加明

2008 年 5 月 8 日

福建师范大学本科毕业论文(设计)指导教师承诺保证书

本人郑重承诺： 我已按有关规定对本篇毕业论文(设计)的选题与内容进行了指导和审核，坚持一人一题制，确认由作者独立完成。如果存在学风问题，本人愿承担指导教师的相关责任。

签名： 黄添强

2008 年 5 月 8 日

福建师范大学本科毕业论文（设计）文献综述

论文题目：蚁群算法研究与实现

学院：数学与计算机科学学院 专业：计算机科学与技术
学号：105032004002 姓名：郑加明 指导教师：黄添强

1. 引言

在我们的日常生活中，常常可以看到成群结队的蚂蚁群落。数以百万计的蚂蚁是如何组成一个群落？一些学者在观察蚂蚁的觅食过程中发现，蚂蚁总能找到巢穴与食物源之间的最短路径。经研究发现，蚂蚁的这种群体协作功能是通过遗留在其来往路径上的信息素来完成的。整个蚁群就是通过这种信息素进行相互协作和通信，形成正反馈，使多条路径上的蚂蚁逐渐聚集到最短的那条路径上来的。

蚁群算法(Ant Colony Optimization, ACO)是一种最新发展的模拟蚂蚁群体觅食行为的仿生优化算法。它是由 M.Dorigo 等人受到真实蚁群集体行为的研究成果的启发而首先提出来的^[1]。他们充分利用蚁群搜索食物的过程与旅行商问题(TSP)之间的相似性^[2]，通过人工模拟蚂蚁搜索食物的过程中个体之间的信息交流与相互协作最终找到从蚁群到食物源的最短路径的原理解决了 TSP 问题^[2]，取得了很好的效果。蚁群算法是一种用来在图中寻找优化路径的机率型技术。该研究成果可以被应用于数据挖掘的多种应用领域，具有一定的研究价值与应用价值。

2. 蚁群算法的基本原理

1.1 真实蚁群和人工蚁群的共同点

人工蚁群大部分的行为特征都是来源于真实蚁群，它们具有的共同特征主要表现为：

- (1) 人工蚁群和真实蚁群一样，是一群相互合作的群体。蚁群中的每只蚂蚁都能够建立一个解，蚂蚁个体通过相互的协作在全局范围内找出问题较优的解。
- (2) 人工蚁群和真实蚁群共同的任务是寻找起点（蚁穴）和终点（食物源）的最短路径（最优目标）。
- (3) 人工蚁群和真实蚁群都是通过信息素进行间接通讯。在人工蚁群算法中信息素轨迹是通过状态变量来表示的。状态变量用一个二维信息素矩阵来表示。人工蚁群算法就是通过修改矩阵中元素的值，来模拟真实蚁群中信息素更新的过程。
- (4) 人工蚁群利用了真实蚁群觅食过程中的正反馈机制^[3]。蚁群的正反馈机制，使得问题的解向着全局最优的方向不断进化，最终能够有效地获得相对较优的解。
- (5) 人工蚁群和真实蚁群都存在一种信息素挥发机制。这种机制可以使蚂蚁逐渐忘记过去，不受过去经验的过分约束，这有利于指引蚂蚁向着新的方向进行搜索，避免过早收敛。
- (6) 人工蚁群和真实蚁群都是基于概率转移的局部搜索策略。蚂蚁在转移时，所应用的策略在时间和空间上是完全局部的。

1.2 人工蚁群的特有功能

从真实蚁群的行为中获得的启发来设计人工蚁群算法还具备了真实蚁群不具有的一些特性：

- (1) 人工蚁群算法存在于一个离散的空间中，它们的移动是一个状态到另一个状态的转换。
- (2) 人工蚁群算法具有一个记忆它本身过去行为的内在状态。
- (3) 人工蚂蚁释放信息素的量，是由蚁群所建立的问题解决方案优劣程度函数决定的。

(3) 蚁群算法更新信息量的时机是随不同问题而变化不反映真实蚁群的行为。如: 有的问题中蚁群算法在产生一个解后改变信息量, 有的问题中则做出一步选择就更改信息量, 但无论哪一种方法, 信息量的更新并不是随时可以进行的。

(4) 为了改善系统的性能, 蚁群算法中可以增加一些性能, 如: 前瞻性、局部优化、原路返回等。在很多应用中蚁群算法可以在局部优化过程中相互交换信息。

3. 蚁群算法的研究现状

当前对蚁群算法的研究, 除了算法意义上的研究, 还有从仿真模型角度的研究, 并且不断有学者提出对蚁群算法的优化方案。吴庆洪等人提出了具有变异特征的蚁群算法。该算法为了克服蚁群算法收敛较慢的问题, 采用逆转变异方式, 随机地进行变异, 以增大进化时所需的信息素。这种变异机制充分利用了2-opt方法简洁高效的特点, 因而具有较快的收敛速度。经过这种变异算子作用后, 这一代解的性能会有明显改善, 从而也能改善整个群体的性能, 减少计算时间^[4]。魏平等人通过对目标函数的自动适应来调整蚂蚁的路径搜索行为, 同时通过路径选择过程中的多样性来保证得到更多的搜索空间, 以快速找到函数的全局最优解, 从而提出了一种求解函数优化的蚁群算法^[5]。颜晨阳等人通过理论上的证明和实验验证, 提出了一种基于Metropolis接受准则的信息素更新策略, 并通过在不同规模的TSP上的实验, 证明了这种新策略的有效性^[6]。

4. 蚁群算法的应用

由于蚁群算法寻求最短路径的行为类似于旅行商问题(TSP)的解决过程, 因而蚁群算法能在TSP问题实例中得到较优的结果^[2]。蚁群算法在解决组合优化类问题求解方面表现了突出的适用特征。

通过相关领域研究者的关注和努力, 蚁群算法在大量领域获得了应用, 比如机器人系统、图像处理、制造系统、车辆路径系统、通信系统等领域。

蚁群算法的研究成果还可以应用于数据挖掘的多种应用领域, 具有一定的研究价值与应用价值。比如应用蚁群算法解决聚类问题。本文将应用蚁群算法来优化K均值聚类算法。

5. 蚁群算法的评价与展望

5.1 蚁群算法的优点

- (1) 能较好地解决组合优化问题。
- (2) 较强的鲁棒性: 对蚁群算法模型稍加修改, 就可以应用于其他问题。
- (3) 分布式计算: 蚁群算法是一种基于种群的进化算法, 具有并行性, 易于并行实现。
- (4) 易于与其他方法结合, 很容易与多种启发式算法结合, 以改善算法的性能。

5.2 蚁群算法的缺点

- (1) 对于大规模优化问题, 蚁群算法需要较长的搜索时间。
- (2) 容易出现停滞现象, 出现早熟。
- (3) 未能提出一个完善的理论分析, 对它的有效性也没有给出严格的数学解释。

5.3 蚁群算法的展望

对蚁群算法, 重要的就是开发出求解问题的算法模型, 使求解问题更加切实有效。而在工程实际的应用中, 算法模型的收敛性和算法的复杂度是值得研究的问题。

参考文献

- [1] Dorigo, Maniezzo, Colomi. Ant system: optimization by a colony of cooperating agents [J]. IEEE Transactions on SMC, 1996, 26(1):8-41.
- [2] Dorigo, Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem [R]. IEEE Transactions on Evolutionary Computing, 1997, 1(1):53-56.
- [3] 李士勇. 蚁群算法及其应用[C]. 哈尔滨: 哈尔滨工业大学出版社, 2004, 1-21.
- [4] 吴庆洪, 张纪会, 徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展, 1999, 36(10): 1240-1245.
- [5] 魏 平, 熊伟清. 一种求解函数优化的蚁群算法[J]. 计算机科学, 2002, 29(9): 227-229.
- [6] 颜晨阳, 张友鹏, 熊伟清. 一种新的蚁群优化算法信息素更新策略[J]. 计算机应用研究, 2007, 24(7): 86-88.

福建师范大学本科毕业论文(设计)题目审核表

学院： 数学与计算机学院 系或教研室： 软件教研室 时间： 2007 年 11 月 21 日

课 题 情 况	题目名称	蚁群算法研究与实现							
	教师姓名	黄添强	职称	讲师	学位学历	博士	专业方向	数据挖掘	
	课题来源	<input checked="" type="checkbox"/> A. 科研 <input type="checkbox"/> B. 生产 <input type="checkbox"/> C. 教学 <input type="checkbox"/> D. 其它 <input type="checkbox"/> E. 学生自拟					成果类别	A. 论文 <input checked="" type="checkbox"/> B. 设计	
	课题类别	<input type="checkbox"/> A. 工程设计类 <input type="checkbox"/> B. 理论研究类 <input type="checkbox"/> C. 实验研究类 <input type="checkbox"/> D. 软件设计类 <input checked="" type="checkbox"/> E. 综合类 <input type="checkbox"/> F. 其它							
	学生应具备的条件	掌握较强的数学基础，同时具有较熟练的程序设计能力。							
主要研究内容目标特色	蚁群算法(ant colony optimization, ACO)，又称蚂蚁算法，是一种最新发展的模拟昆虫王国中蚂蚁群体觅食行为的仿生优化算法，它是一种用来在图中寻找优化路径的机率型技术。了解蚁群算法研究现状与存在的问题，并在此基础上实现一个简单的蚁群算法原形系统，然后通过实验验证它的性能。								
成果描述	理解蚁群算法的基本原理，掌握一种经典的蚁群算法，设计并实现它，提交一个实用的蚁群算法原形系统。								
成果价值	该研究成果可以被应用于数据挖掘的多种应用领域，具有一定的研究价值与应用价值。								
系或教研室审题意见	同意。 <div style="text-align: right;"> 负责人签字： 严宣辉 2007 年 12 月 1 日 </div>								
学院审批意见	<div style="text-align: right;"> 领导小组组长签字（盖章）： 年 月 日 </div>								

福建师范大学本科毕业论文(设计)开题报告

学院： 数学与计算机科学学院 专业： 计算机科学与技术

课题名称		蚁群算法研究与实现					
姓名	郑加明	学号	105032004002	指导教师	黄添强	职称 学历	讲师 博士
<p>一、开展本课题的意义及工作内容：</p> <p>蚁群算法(Ant Colony Optimization, ACO)是一种最新发展的模拟昆虫王国中蚂蚁群体觅食行为的仿生优化算法。它是一种用来在图中寻找优化路径的机率型技术。该研究成果可以被应用于数据挖掘的多种应用领域，具有一定的研究价值与应用价值。</p> <p>本课题开展的工作内容是，全面了解蚁群算法研究现状与存在的问题，并在此基础上实现一个简单的蚁群算法原形系统，然后通过实验验证它的性能。</p>							
<p>二、课题工作的总体安排及进度：</p> <p>1 月份：借阅蚁群算法相关书籍并上网搜集相关资料。了解蚁群算法的基本思想。</p> <p>2 月份：整理和充实资料，全面了解蚁群算法研究现状与存在的问题，并写好文献综述。</p> <p>3 月份：掌握一种经典的蚁群算法，并实现一个简单的蚁群算法原形系统。</p> <p>4 月份：将蚁群算法原形系统应用于聚类问题，并通过实验验证它的性能，并提出问题。</p> <p>5 月份：修改和完善蚁群算法原形系统，研究论文定稿，准备答辩。</p>							
<p>三、课题预期达到的效果：</p> <p>1. 理解蚁群算法的基本原理。</p> <p>2. 全面了解蚁群算法研究现状与存在问题。</p> <p>3. 实现一个简单的蚁群算法原形系统，并通过解决聚类问题验证其性能。</p>							
<p>四、指导教师意见：</p> <p>蚁群算法是仿生算法的一支新秀，具有广阔的应用前景与良好的应用价值。同意开题。</p> <p style="text-align: right;">签名：黄添强</p> <p style="text-align: right;">2008 年 1 月 15 日</p>							

不够填写可续页

福建师范大学毕业论文(设计)中期检查表

学院： 数学与计算机科学学院 专业： 计算机科学与技术

论文(设计)题目		蚁群算法研究与实现					
姓名	郑加明	学号	105032004002	指导教师	黄添强	职称 学历	讲师 博士
论文(设计)的进度计划	<p>1 月份：借阅蚁群算法相关书籍并上网搜集相关资料。了解蚁群算法的基本思想。</p> <p>2 月份：整理和充实资料，全面了解蚁群算法研究现状与存在的问题，并写好文献综述。</p> <p>3 月份：掌握一种经典的蚁群算法，并实现一个简单的蚁群算法原形系统。</p> <p>4 月份：将蚁群算法原形系统应用于聚类问题，并通过实验验证它的性能。提出问题，并完成研究成果基本框架。</p> <p>5 月份：修改和完善蚁群算法原形系统，研究论文定稿，准备答辩。</p>						
目前已经完成的内容	<p>1、了解了蚁群算法的基本思想。</p> <p>2、通过阅读大量的资料，全面了解了蚁群算法研究现状与存在的问题，并写好了文献综述。</p> <p>3、认真理解了一种蚁群优化算法的设计思想，并了解了其在解决 TSP 问题的效果。</p> <p>4、设计一个简单的蚁群算法原形系统，并研究将其应用于聚类问题的实现方案。</p>						
内容 尚须完成的	<p>1、研究将蚁群算法原形系统应用于聚类问题的实现方案。</p> <p>2、使用 Visual C++ 6.0 实现应用蚁群解决聚类的算法。</p> <p>3、通过实验验证蚁群算法原形系统的性能，并完成研究成果基本框架。</p> <p>4、修改和完善蚁群算法原形系统，论文定稿。</p>						
存在的问题和拟采取的办法	<p>问题 1、如何将蚁群算法应用到数据挖掘领域。</p> <p>问题 2、聚类的几种常用算法，能否应用蚁群算法进行聚类研究。</p> <p>问题 3、对蚁群算法进行优化应注意哪些问题，有哪些主要方法。</p> <p>问题 4、如何设置蚁群算法的各个参数，对于不同的参数产生的结果进行评价。</p> <p>解决方法：继续查阅有关数据挖掘中聚类算法的知识。研究应用蚁群算法进行聚类分析的方法。通过实验实现该方案，并测试蚁群算法的性能。</p>						
指导教师意见	<p style="text-align: center;">毕业设计按计划进度进行，目前进展顺利，望继续努力，尽快完成。</p> <p style="text-align: right;">签名：黄添强</p> <p style="text-align: right;">2008 年 4 月 12 日</p>						

福建师范大学本科毕业论文(设计)指导教师指导记录表

学院： 数学与计算机科学学院 专业： 计算机科学与技术

论文(设计)题目		蚁群算法研究与实现			
姓名	郑加明	学号	105032004002	指导教师姓名 及专业方向	黄添强 数据挖掘
<p>指导内容：</p> <p>讨论确定毕业设计课题，初步确定开展的课题要求。同时，简单介绍课题的意义和背景知识，强调蚁群算法是一种很有趣的算法，且发展前景很好。老师要求写好毕业设计的计划安排，定期阅读蚁群算法相关文献和书籍，并做好反馈工作。</p> <p style="text-align: right;">学生签名：郑加明 教师签名：黄添强 2007 年 12 月 21 日</p>					
<p>指导内容：</p> <p>指导开题报告的书写，并安排下一阶段的课题开展。强调开题报告是做好毕业设计的第一步，需要认真对待。对开题报告中课题研究进度安排作了说明，再次强调了阅读相关文献的重要性。最后，要求在规定时间内完成开题报告。</p> <p style="text-align: right;">学生签名：郑加明 教师签名：黄添强 2008 年 1 月 11 日</p>					
<p>指导内容：</p> <p>批改开题报告，并说明下一阶段的安排。着重就开题报告中课题研究进度进行了修改，强调要严格按照该计划开展课题研究。另外特别指出了蚁群算法在数据挖掘方面的结合，要求在开展后期着重研究这方面的相关知识。</p> <p style="text-align: right;">学生签名：郑加明 教师签名：黄添强 2008 年 1 月 18 日</p>					
<p>指导内容：</p> <p>介绍文献综述包含的相关内容，并指出书写文献综述的注意事项。强调了文献综述对课题研究的重要性，要求认真写好文献综述，并及时反馈。另外，就如何阅读相关文献作了许多指导，介绍了阅读文献的方法。</p> <p style="text-align: right;">学生签名：郑加明 教师签名：黄添强 2008 年 2 月 18 日</p>					
<p>指导内容：</p> <p>指导蚁群算法的基本思想，以及如何更好地理解蚁群算法。介绍了一些常见的智能算法，并强调了蚁群算法与其他智能算法的相似和不同之处。指出了学习仿生优化算法的方法，以及如何更好地应用这些仿生优化算法。</p> <p style="text-align: right;">学生签名：郑加明 教师签名：黄添强 2008 年 2 月 25 日</p>					
<p>指导内容：</p> <p>检查课题开展的前期工作，并总结一些常见问题。对于文献综述的书写提出一些修改意见。要求按时完成文献综述，为继续研究课题奠定基础。另外还提出应该注重蚁群算法研究现状，并思考该算法的优点和存在的不足。</p> <p style="text-align: right;">学生签名：郑加明 教师签名：黄添强 2008 年 3 月 3 日</p>					

<p>指导内容:</p> <p>指导设计一个简单蚁群算法原形系统的方法, 提出应当建立在前期相关文献的阅读上。要求在蚁群算法研究现状和存在问题上, 再提出一个简单的算法原形系统设计方案。另外, 还强调应该开始多阅读数据挖掘方面的文献, 特别是聚类分析方面的知识。</p> <p>学生签名: 郑加明 教师签名: 黄添强 2008 年 3 月 17 日</p>
<p>指导内容:</p> <p>检查蚁群算法原形系统的设计, 并提出改进方案。要求将设计好的系统应用于聚类分析中, 并通过实验验证该系统的性能。强调学习一些常用的聚类分析算法, 并分析它们的不足, 提出应用蚁群算法优化的方案。</p> <p>学生签名: 郑加明 教师签名: 黄添强 2008 年 3 月 24 日</p>
<p>指导内容:</p> <p>针对基于蚁群算法优化 K 均值聚类算法, 提出一些修改的意见。强调系统设计时, 可以从简单开始, 由易到难, 逐步改进和测试。可以针对 K 均值聚类算法存在的某一问题, 提出自己的设计方案。要求设计出来的算法能与未改进前的 K 均值聚类算法形成对比。</p> <p>学生签名: 郑加明 教师签名: 黄添强 2008 年 3 月 31 日</p>
<p>指导内容:</p> <p>指导蚁群聚类算法设计的实现, 并提出设计的操作步骤。要求应用 Visual C++ 6.0 实现该系统, 实验结果要求可视化。强调编写代码时, 应该规范并加以注释。设计好的系统, 首先必须通过一些简单的数据进行测试, 之后实现可视化, 加大测试量, 验证算法的性能。</p> <p>学生签名: 郑加明 教师签名: 黄添强 2008 年 4 月 7 日</p>
<p>指导内容:</p> <p>检查系统的设计效果, 并提出一些改进建议。针对实验结果的分析方法, 进行指导。强调实验数据应当可以保存, 应该设计一些常见的测试数据。另外, 还要求进一步完善该系统并及时完成论文初稿, 反馈存在的问题。</p> <p>学生签名: 郑加明 教师签名: 黄添强 2008 年 4 月 14 日</p>
<p>指导内容:</p> <p>针对论文初稿, 提出许多修改意见。并强调书写论文时, 一定要严格按照学校规定的格式书写。另外, 还检查了应用蚁群算法优化 K 均值聚类算法的系统。针对该系统的设计, 提出一些修改方案, 并强调实验数据要严谨客观。</p> <p>学生签名: 郑加明 教师签名: 黄添强 2008 年 4 月 23 日</p>
<p>指导内容:</p> <p>再次对论文进行修改, 并提出一些改善意见。对论文的摘要进行了修改, 使得论文主题更加鲜明。要求论文中引用部分一定要注明所参考的文献。对论文中图表等格式作了一些修改, 并指出图表应当真正起到简单清晰的表达和表示作用。</p> <p>学生签名: 郑加明 教师签名: 黄添强 2008 年 4 月 30 日</p>
<p>指导内容:</p> <p>提出论文基本上定稿, 但应该继续检查以排除错误。最后就毕业答辩时应该注意的问题, 作了重要说明。特别强调答辩时, 应当表达清楚, 并给出演示。对于系统实现的部分, 应当设置好测试数据, 提高演示效率。</p> <p>学生签名: 郑加明 教师签名: 黄添强 2008 年 5 月 7 日</p>

备注: 1.此表由学生根据老师每次指导论文的内容填写, 指导教师签字后学生保存。表格不够填写可续页。
2.凡学生不主动请教指导教师或弄虚作假, 教师没有完成指导工作量、不负责任者, 均将酌情处理。
3.表格不够填写可续页。

福建师范大学本科毕业论文(设计)答辩（ 小组 ）记录表

学院：数学与计算机科学学院 专业： 计算机科学与技术

论文(设计)题目		蚁群算法研究与实现					
姓名	郑加明	学号	105032004002	指导教师	黄添强	职称 学历	讲师 博士
<p>答辩过程问答情况记录：</p> <p>1、问：你的算法在时间复杂性搜索准确度上与现有的其他蚁群算法相比，性能更优吗？ 答：取决于 K 均值原理。</p> <p>2、问：那个语句说明意味着该句是用来找蚂蚁还是找路径？ 答：找路径最短的蚂蚁。</p> <p>3、与现有算法比，你做了哪些改进？ 答：加入随机扰动等，这些改进我已做过实验。</p> <p style="text-align: right;">记录人：陈星白 2008 年 5 月 10 日</p>							
<p>答辩小组或答辩委员会成员名单及职称：</p> <p>黄榕宁 副教授 陈星白 讲师 唐朝晖 助教</p>							
<p>答辩评审成绩：（ 83 ）</p> <p style="text-align: right;">负责人签名：黄榕宁 2008 年 5 月 10 日</p>							

备注：请在表头（ ）内注明“小组”或“委员会”。

福建师范大学本科毕业论文(设计)答辩（ 委员会 ）记录表

学院：数学与计算机科学学院 专业： 计算机科学与技术

论文(设计)题目		蚁群算法研究与实现											
姓名	郑加明	学号	105032004002	指导教师	黄添强	职称 学历	讲师 博士						
<p>答辩过程问答情况记录：</p> <p>1、问：你的算法的目标函数是什么？ 答：目标函数依实际问题定。</p> <p>2、问：本文最大贡献是什么？ 答：应用蚁群算法与聚类分析相结合，来探索蚁群聚类系统模型的设计方法。</p> <p>3、分析一下你的蚁群算法解决问题的时间复杂度？ 答：时间复杂度与普通 K 均值一样，为 $O(NKT)$。</p> <div style="text-align: right; margin-top: 100px;"> <p>记录人：张仕</p> <p>2008 年 5 月 10 日</p> </div>													
<p>答辩小组或答辩委员会成员名单及职称：</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 50%;">严宣辉</td> <td style="width: 50%;">副教授</td> </tr> <tr> <td>陈荔聪</td> <td>讲师</td> </tr> <tr> <td>张仕</td> <td>讲师</td> </tr> </table>								严宣辉	副教授	陈荔聪	讲师	张仕	讲师
严宣辉	副教授												
陈荔聪	讲师												
张仕	讲师												
<p>答辩评审成绩：（ 90 ）</p> <div style="text-align: right; margin-top: 100px;"> <p>负责人签名：严宣辉</p> <p>2008 年 5 月 21 日</p> </div>													

备注：请在表头（ ）内注明“小组”或“委员会”。

福建师范大学本科毕业论文(设计)成绩表

学院： 数学与计算机科学学院 专业： 计算机科学与技术

论文(设计)题目		蚁群算法研究与实现					
姓名	郑加明	学号	105032004002	指导教师	黄添强	职称 学历	讲师 博士
<p>论文(设计)摘要：</p> <p>蚁群算法(Ant Colony Optimization, ACO)是一种最新发展的模拟昆虫王国中蚂蚁群体觅食行为的仿生优化算法。它是一种用来在图中寻找优化路径的机率型技术。该研究成果可以被应用于数据挖掘的多种应用领域，具有一定的研究价值与应用价值。</p> <p>本文将论述蚁群算法研究现状与存在的问题，并在此基础上实现一个简单的蚁群算法原形系统，然后将系统应用到 K 均值聚类算法的优化。通过实验测试，验证该原形系统的性能。</p>							
指导教师意见 (评语应包括工作态度、能力评价、完成质量和规范性、学术性等)	<p>指导教师评分(百分制)： 96</p> <p>评语：</p> <p>该生对蚁群算法的原理、研究现状与存在问题进行了研究，并用蚁群算法进行聚类，实现了原形系统。用蚁群算法进行聚类具有一定的创新性。该生工作态度认真，勤奋学习，编程能力强。论文结构合理，语言流畅，表达清晰。</p>						
	<p style="text-align: right;">签名：黄添强</p> <p style="text-align: right;">2008 年 5 月 9 日</p>						

<p>评阅人意见</p>	<p>评阅人评分（百分制）：85</p> <p>评语：</p> <p>论文选题角度有新意，使用材料翔实、恰当，理论分析准备，逻辑严密，层次清楚，结构合理，有独创的个人见解，学术性较强。用蚁群算法进行聚类具备一定的创新。</p> <p style="text-align: right;">签名：陈星白</p> <p style="text-align: right;">2008 年 5 月 9 日</p>
<p>答辩评审意见</p>	<p>答辩评分（百分制）：90</p> <p>评语：</p> <p>该生在答辩中能正确回答问题，并演示自己的成果，表现出对自己的工作有深入的了解。该生对蚁群算法和聚类算法有很好的了解，并能结合蚁群算法改进 K 均值聚类算法，聚得较好的结果。</p> <p>课题的理论水平较高，编程难度较高，有一定的创新性。</p> <p style="text-align: right;">负责人签名：严宣辉</p> <p style="text-align: right;">2008 年 5 月 21 日</p>
<p>毕业论文（设计）综合评定成绩：_____（百分制）；_____（五级制）。</p> <p style="text-align: center;">答辩委员会主任（签名）：</p> <p style="text-align: right;">年 月 日</p>	

备注：毕业论文（设计）综合评定成绩＝指导教师成绩*50%＋评阅人成绩*30%＋答辩小组成绩*20%，百分制折合成五级制记分。