

基于 TSP 问题的蚁群算法综述

郭平 鄢文晋

(重庆大学计算机学院 重庆 400044)

摘要 蚁群算法是受现实蚂蚁群体行为启发而得出的一类仿生算法。本文以解决 TSP 问题为基础,系统地介绍了蚁群算法从诞生到成熟过程中几个代表性的算法。在阐述算法基本思想的前提下,着重论述算法的创新之处。

关键词 蚁群算法, 仿生, TSP 问题, 创新

The Review of Ant Colony Algorithm Based on TSP

GUO Ping YAN Wen-Jin

(School of Computer Science, Chongqing University, Chongqing 400044)

Abstract Ant colony algorithm is a kind of Bionic Algorithm inspired from the behaviors of the ant colony. Based on the TSP problem, this paper introduces some typically algorithms of the ant colony algorithm systematically in the period from the ant colony algorithm naissance to the mature of the ant colony algorithm. With declaring of the basic principle of the ant colony algorithm, the paper focuses on the innovation of the ant colony algorithm.

Keywords Ant colony algorithm, Bionic, TSP problem, Innovation

1 引言

蚁群算法^[1]最初是通过对蚂蚁群落的观察,受蚁群行为特征启发而得出的。蚂蚁是一种群居昆虫,在觅食、清理巢穴等活动中,彼此依赖、相互协作共同完成特定的任务。就个体来讲,单个蚂蚁的智力和体力是极其有限的,服务于整个群落的生存与发展;就群体来讲,蚁群在行为上的分工协作、在完成过程中所体现的自组织特征等反应出蚁群具有较高的智能和自我管理能力,具有很高层次组织性,这使得蚁群能够完成一些复杂的任务,例如求解旅行商问题^[2],动态网络路由问题^[3]等。

蚁群算法特点是并发性、鲁棒性、正反馈性等。在蚁群算法求解问题的过程中,利用蚁群在问题空间中同时构造问题的多个解体现了算法的并发性。蚁群不会因为单个蚂蚁寻找到较差的解或者因为问题空间发生改变而使得算法丧失作用,这体现了算法的鲁棒性。在蚂蚁构造问题解的过程中,以蚁群觅食行为为例,会在经过的解路径上释放信息素,而解空间中获得信息素越多的路径,对蚂蚁的吸引力就越大,使更多的蚂蚁经过该路径并进一步在上面释放信息素,这体现了算法的正反馈性。

TSP 问题是典型的 NP 完全问题,许多算验证法及算法效率测试都以 TSP 问题为基础。在蚁群算法研究中,第一个蚁群算法,蚂蚁系统,就是在 TSP 问题的基础上提出来的。而后,依据 TSP 问题,又提出了蚁群算法系列中具有代表性的蚁群系统,最大-最小蚂蚁系统。

本文以 TSP 问题为基础,对蚁群算法的基本问题及典型的蚁群算法进行了综述。涉及到算法的基本问题、算法描述、算法改进及意义。通过研究总结了蚁群算法的发展历程和实现思想。本文第 2 节简要地介绍 TSP 问题和具有启发意义的三类蚁群行为;第 3 节依照时间的先后顺序,介绍蚁群算法系列中具有代表性的几个算法,着重阐述相关算法的实现思

想及主要的创新点;最后,对全文进行了总结,并提出了蚁群算法研究中应注意的一些问题。

2 TSP 问题和蚁群行为

在对蚁群算法的研究中,用于算法验证与效率测试最多的例子是 TSP 问题。例如,第一个蚁群算法—蚂蚁系统^[4],以及对蚂蚁系统进行改进而产生的蚁群系统,最大-最小蚂蚁系统等蚁群算法,都是基于 TSP 问题对算法进行验证。

2.1 TSP 问题

TSP 问题是给定一个城市的集合以及城市之间的旅行代价,寻找经过每个城市一次且仅一次并最终回到起始城市旅行代价最小的路径。如果构造一个图如下:图中的顶点为城市,顶点间的边表示城市间的交通线,边上的权为沿该交通线旅行的费用。那么,TSP 问题就抽象为在这个图中寻找最短哈密尔顿回路。

任意两个城市 A, B, 如果 A 到 B 的旅行代价和 B 到 A 旅行代价相等,称这样的 TSP 问题是对称的 TSP 问题 (symmetric traveling salesman problem, STSP), 否则称为不对称的 TSP 问题 (asymmetric traveling salesman problem, ATSP)。通常,在没有特别申明的情况下所提及的 TSP 问题指对称的 TSP 问题。

n 个顶点的 TSP 问题中的路径指顶点的序列: $L = x_1, x_2, \dots, x_n$, 其中 x_i 与 x_{i+1} ($1 \leq i \leq n-1$) 之间有边。一条路径被称为合法路径, 如果 $x_i \neq x_j$ ($i \neq j, 1 \leq i \leq n, 1 \leq j \leq n$)。

TSP 问题本质上是数学优化问题,可以形式化地描述为:

$$\text{Min}(\sum_{i=1}^{n-1} d(x_i, x_{i+1}) + d(x_n, x_1))$$

其中, $d(x_i, x_{i+1})$ 表示顶点 x_i 与顶点 x_{i+1} 之间的距离。算法研究表明, TSP 问题是 NP 完全问题, 其计算复杂度为 $O(n!)$ 。

自 TSP 问题提出以来,其求解方法得到了不断的改进。

目前已经可以对上万个城市的 TSP 问题进行求解^[5]。近年来,以蚁群行为为基础的蚁群算法已成为一种较为有效的 TSP 问题求解方法。

2.2 蚁群行为

蚁群的行为是整体协作,相互分工,以一个整体去解决一些对单个蚂蚁看上去是不可能完成的任务。就目前来讲,蚁群至少有三个方面的行为特征对算法研究有很好的启发意义,分别是觅食行为^[6]、任务分配^[7]、死蚁堆积^[8]。

蚁群的觅食行为指蚂蚁从巢穴出发寻找食物并且将食物搬回巢穴的行为。当蚂蚁出外寻找食物时,会在自己走过的路径上释放一种称为信息素的物质,后续的蚂蚁一般更愿意走那些信息素强度更高的路径。这样,较短路径上单位时间内通过的蚂蚁数目较多,留下的信息素也较多(浓度更高),对蚂蚁产生了更强的吸引,使得更多的蚂蚁走较短的路径。这就形成了一个正反馈机制,使得最终所有的蚂蚁都走蚁穴到食物源的最短路径。

蚁群的任务分配指蚁群内部能根据所需要完成的任务合理分配蚂蚁数量。研究表明^[9],从事不同任务的蚂蚁之间的比例是会变动的,也就是说,从事任务 A 的蚂蚁可能会根据需要而从事任务 B。在基于这种蚁群行为的模型中,规定了每只蚂蚁的反应阈值,同时每一个任务都有一个相关的激励(task-related stimuli)。反应阈值与激励间的关系是:当某个任务的激励数值比蚂蚁的反映阈值大,那么这只蚂蚁就会加入到该任务执行者的行列中。蚁群中,如果一群蚂蚁在执行某任务时出现执行滞后情况,那么该任务的激励数值就会变大,直到其数值大于某些未执行该任务的蚂蚁的反映阈值,从而吸引更多的蚂蚁来完成该任务,由此实现了任务的自动分配。

死蚁堆积与蚁穴清理行为:在蚁群的巢穴中,工蚁会对死蚂蚁进行搬运,使得那些死蚂蚁堆积到一起。在这一堆积行为中,死蚁堆越大对工蚁的吸引力越大,使得更多的死蚂蚁被堆积到该蚁堆中,这样同样也形成了一个正反馈现象。但是,并不是说所有的死蚂蚁都会堆积到一起,而是最终会形成几个规模较大的蚁堆。

研究者对蚁群的这 3 种行为特征进行了深入的研究,并应用于最短路径优化、任务分配调度以及数据聚类等问题领域,得到了效率较高的许多启发式算法^[10~12]。

3 几个典型的蚁群算法

这里,我们仅讨论与 TSP 问题相关的蚁群算法。在蚁群算法研究及实现中,并不是直接模拟现实蚁群,而是采用人工蚂蚁(artificial ant)^[13]。人工蚁群与现实蚁群的区别主要包括:

(1)人工蚂蚁是有一定的记忆能力的,它可以记住已经走过的路径,以保证不会重复走相同的城市。现实的蚁群是没有记忆的,蚂蚁间的信息交换主要依靠留在所经过路径上的信息素。

(2)人工蚂蚁不仅仅是依据信息素来确定要走的路径的,还依据一定的启发信息,比如相邻边的长度,这意味着人工蚂蚁具有一定的视觉能力,而真实蚂蚁几乎没有视觉。

(3)人工蚂蚁是生活在一个离散的时间环境下的。我们仅考虑人工蚂蚁位于某个城市,而不考虑蚂蚁在城市间的移动过程,即只考虑在某些离散时间点上的蚂蚁。而现实世界中的蚂蚁处于一个连续的时间维中。

3.1 蚂蚁系统(Ant System)

蚂蚁系统是第一个蚁群算法,它是 M. Dorige 等人于 1991 年首先提出来的^[1]。蚂蚁系统有三种基本模型,分别是蚁周模型、蚁密模型、蚁量模型。

三种模型的实现大致相同,主要区别是在信息素的更新方式上。在用蚂蚁系统解决 TSP 问题时,蚁量模型和蚁密模型是蚂蚁在构建一条合法路径的过程中进行信息素的更新的,当蚂蚁走过一条边之后,就对该边进行信息素的更新,后文将这种更新称为局部更新。而蚁周模型是在所有蚂蚁都构建了一条合法路径之后才对各边进行信息素更新的,后文将这种更新称为全局更新,并且三者中蚂蚁释放信息素的量上面也不同。蚁密模型中,蚂蚁在自己所走过的边上所释放的信息素是一个常量 Q ,而蚁量模型中,蚂蚁在自己所走过的边上释放的信息素是 Q/d_{ij} ,其中 Q 是一个常量,而 d_{ij} 是蚂蚁走过边的长度。蚁周模型中蚂蚁释放信息素的量在后文说明。

由于蚁周模型是三种模型中性能最好的,下面主要从蚁周模型的角度来讨论蚂蚁系统。

蚂蚁系统的基本思想是:(1)预先初始化各边信息素强度以及各蚂蚁的禁忌表。各蚂蚁按照一定的概率规则,在禁忌表的制约下选择下一个要到达的结点,直到最终形成一条合法路径。(2)计算各蚂蚁所产生的路径长度,路径长度是路径中各边长度之和。(3)更新各边的信息素。各边先进行信息素挥发操作,然后根据各蚂蚁产生的路径长度获取蚂蚁所释放的信息素。(4)当所有蚂蚁均完成了信息素的更新操作之后,记录当前的最短路径,并且对禁忌表以及信息素的增加值 $\Delta\tau_{ij}(t, t+1)$ 进行初始化,并转到步骤 2。依此循环下去,直到满足算法的终了条件为止,比如解无法得到进一步的改进或者达到了事先规定的循环次数。

在蚂蚁系统具体包括了三个方面的内容。

第一、初始化。对于每条边上的信息素初始化为一个较小的数值 τ_0 ;对每只蚂蚁,需要一个禁忌表记录自己已经走过的结点,初始化其禁忌表为该蚂蚁所在的结点,禁忌表长度为 1。蚂蚁在各边上释放信息素的量被初始化为 0。

第二、蚂蚁构造路径。蚂蚁按照一定的概率确定下一步要到达的城市。概率的计算如(1)式。

$$p_{ij}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{u \in allowed} [\tau_{iu}(t)]^\alpha [\eta_{iu}]^\beta} & \text{if } j \in allowed \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

(1)式表示蚂蚁在 t 时刻由城市 i 选择城市 j 的概率。 α 是信息素在概率计算中的权重,它的值越大,信息素在蚂蚁选择下一个要到的城市中起到的作用越大。 β 是启发因子(在 TSP 问题中常以 d_{ij} 的倒数来表示)在概率计算中所占的权重,它的值越大,启发因子在蚂蚁选择城市的过程中所起的作用越大。Allowed 是不在蚂蚁禁忌表中的城市集合。(1)式说明,蚂蚁不会选择禁忌表中的城市,这样就保证了了解的合法性。

第三、对信息素的操作。在蚁周模型中,当所有的蚂蚁都找到了一条合法路径之后,就进行信息素的更新,如(2)式。

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \quad (2)$$

其中, $\tau_{ij}(x)$ 表示在 x 时刻边 ij 上的信息素。 ρ 是信息素维持因子, $1-\rho$ 是信息素的挥发因子。 $\Delta\tau_{ij}(t, t+1)$ 是所有蚂蚁在边 ij 上所释放的信息素的总和,如(3)式。

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^m \Delta\tau_{ij}^k(t, t+1) \quad (3)$$

其中 m 是蚂蚁的数量, $\Delta^k \tau_{ij}(t, t+1)$ 是蚂蚁 k 在 t 到 $t+1$ 时间内在边 ij 上所释放的信息素,

$$\Delta^k \tau_{ij}(t, t+1) = \begin{cases} \frac{Q}{L_k} & \text{如果蚂蚁 } k \text{ 所形成的路径包括边 } ij \\ 0 & \text{否则} \end{cases}$$

大部分的蚁群算法都是在蚂蚁系统的基础上发展而来的。它们都是将蚂蚁系统与具体问题相结合,并且在蚂蚁系统的基础之上引入一些新的控制机制。故,通常都是把蚂蚁系统认为是蚁群算法的先驱与研究的基础。

3.2 蚁群系统

蚂蚁系统 (Ant Colony System, ACS) 是第一个蚁群算法,在解决规模较小的 TSP 问题的时候效果很好,但是随着问题的规模扩大,蚂蚁系统就会出现收敛速度过慢的问题。在 1996 年, M. Dorigo 在蚂蚁系统的基础之上又提出了蚁群系统^[13]。

蚁群系统的基本思想是:将 m 只蚂蚁按照一定的规则(随机)放于 n 个结点。每一只蚂蚁通过伪随机规则,也称状态转移规则创建一条合法路径。在创建路径的过程中,每一只蚂蚁通过局部更新规则对自己所走过的边进行信息素的更新操作。当所有的蚂蚁都完成了路径的构造之后,再对最佳路径上的边进行信息素的全局更新。

蚁群系统较蚂蚁系统改进的地方主要体现在三个方面。

第一、蚁群系统全局更新时仅针对当前最好路径上的边进行,更新规则如(4)式所示。

$$\tau(i, j) = (1 - \lambda) \cdot \tau(i, j) + \lambda \cdot \Delta\tau(i, j) \quad (4)$$

其中, λ 是信息素衰退因子, $\Delta\tau(i, j)$ 是当前最好路径长度的倒数。

第二、蚁群系统在蚂蚁创建路径的过程中所使用的状态转移规则不同于蚂蚁系统,如(5)式所示。

$$s = \begin{cases} \arg \max_{u \in allowed} \{ [\tau(i, u)]^a \cdot [\eta(i, u)]^b \} \\ \text{if } q \leq q_0 \text{ (exploitation)} \\ (1) \text{ otherwise (biased exploration)} \end{cases} \quad (5)$$

其中, $q \in (0, 1)$ 是一个随机数, $q_0 \in (0, 1)$ 是一个常量,它在算法的求解效率与算法的运行效率之间起平衡作用。一般地,想使算法收敛于全局最优解,获得较高的求解效率就务必要求搜索范围尽可能地大,不能局限在已有路径这个空间周围,而这不可避免就使得算法的运行效率降低。反之,假如要想算法的运行速度加快,获得满意的运行效率,那么算法的搜索空间就不能太大,这样使得算法有收敛于局部最优解的风险。蚁群系统的做法是:当 $q \leq q_0$ 时,则增强已有的较好的路径上的信息素,即选择当前转移概率最大的那个城市。而当 $q > q_0$ 时,就按照蚂蚁系统中的选路方式来选择下一个要到的城市,以扩大搜索空间。这样做的优点是在保障算法求解效率的基础上同时又提高了算法的运行效率。

第三、蚁群系统在对信息素更新时,除了进行全局更新,还要进行信息素的局部更新。局部更新如(6)式所示。

$$\tau(i, j) = (1 - \rho) \cdot \tau(i, j) + \rho \cdot \Delta(i, j) \quad (6)$$

其中, $\rho \in (0, 1)$ 。M. Dorigo 在文[13]中给出了 $\Delta\tau(i, j)$ 三种可能的取值:一是依据 Q 学习(Q-Learning)^[14]方法取值为,

$$\gamma \cdot \max_{z \in allowed(j)} \tau(j, z)$$

其中, $\gamma \in [0, 1)$, 依据这种方法而得出的蚁群系统称之为 Ant-Q 算法^[15]。另外两种可能的取值是让 $\Delta\tau(i, j)$ 取值为 0 或者为 τ_0 。实验表明^[13], $\Delta\tau(i, j)$ 按 Q 学习方法取值和让 $\Delta\tau(i, j)$ 为 τ_0 在性能上差异不大,并且较 $\Delta\tau(i, j)$ 为 0 时优越。

又由于 $\Delta\tau(i, j)$ 取值为 τ_0 时计算量小,所以 $\Delta\tau(i, j)$ 常被设置为 τ_0 。

在关于蚁群系统的后续研究中, ACS 引入了局部搜索(Local Search)策略,它使用 restrict 3-opt^[16], 这个搜索策略可以同时满足 ATSP 和 STSP 两类问题,并且可以明显提高算法的性能。

实验研究表明^[13], ACS 在解决规模较大的 TSP 问题时,可以取得比较令人满意的结果。

3.3 最大-最小蚂蚁系统

在 1997 年, T. Stutzle 等人在对蚂蚁系统的实验分析及应用研究中,提出了最大-最小蚂蚁系统(Max-Min Ant System, MMAS)^[17]。

最大-最小蚂蚁系统的基本思路是:在算法中,将各边上的信息素的浓度控制在 $(\tau_{\min}, \tau_{\max})$ 范围之内, τ_{\min} 和 τ_{\max} 是算法中信息素所能够取值的下限值和上限值。在算法的初始化的时候,将各边的信息素的数值初始化为信息素的上限数值。最大-最小蚂蚁系统会增强当前循环中最优路径上的信息素强度。而另外边上的信息素由于挥发作用而进一步地减少,这样就加剧了各边信息素的差异,提高了算法的运行效率。但是这样可能会导致解过快收敛于一个局部最优解。为了避免出现这种情况,算法中引入了平滑机制。

MMAS 在蚂蚁系统的基础之上最主要的改进包括以下四个方面。

第一、算法只对当前循环中所产生的最短路径进行信息素的更新操作,这样做的目的是使得算法在最短路径附近搜索,从而逐步找出全局最优解。这与 ACS 中选择更新路径有区别, MMAS 是选择当前循环的最佳路径进行更新,它仅仅考虑当前循环中所产生的所有路径。而 ACS 是考虑算法运行至今所产生的所有的路径。前者称为 S_g (solution of iteration best), 后者常用 S_{gb} (solution of global best) 来表示。

第二、为了避免算法过快地收敛于局部最优解, MMAS 将各边上信息素的量控制在一个比较固定的范围之内,分别用 τ_{\max} , τ_{\min} 来表示,这也是算法的主要特点,算法的名称也由此而来。具体的控制方法是:当目前边上的信息素的值已经超过 τ_{\max} 时,就把该边上信息素的数值定为 τ_{\max} , 如果某边上的信息素的数值比 τ_{\min} 都还小时,就把该边上的信息素的数值定为 τ_{\min} 。文[17]中给出了 τ_{\max} 的数值是变动的,在算法第 t 次循环时, $\tau_{\max}(t) = n/L_{\min}$, L_{\min} 是算法运行过程中当前所找出的最短路径。而 $\tau_{\min} = \text{const}/\Phi * n^2$, 这里 Φ 是 TSP 问题中各边的平均长度, n 是 TSP 问题中的结点数。在文[18]中, τ_{\max} 按(7)式计算:

$$\tau_{\max} = \frac{1}{1 - \rho} * \frac{1}{f(s^{opt})} \quad (7)$$

其中, ρ 是信息素的挥发因子, $f(s^{opt})$ 是问题的最优解。 τ_{\min} 按(8)式计算,

$$\tau_{\min} = \frac{\tau_{\max} \cdot (1 - \sqrt[n]{\rho_{best}})}{(avg - 1) \cdot \sqrt[n]{\rho_{best}}} \quad (8)$$

第三、MMAS 并没有像 AS 那样将各边上的信息素初始化为一个比较小的数值,而是设为信息素的上限数值 τ_{\max} 。之所以这样设定,主要是由实验分析而得出的,经实验^[17]表明,将各边的信息素的数值初始化为 τ_{\max} 可以提高算法的效率。

第四、为了扩展蚁群的搜索空间,从而使得最终找出全局最优解,在 MMAS 中还引入了平滑机制(Smoothing)。平滑

机制就是尽可能缩小各边上的信息素的差异。平滑机制如(9)式所示:

$$\tau_{ij}^*(t) = \tau_{ij}(t) + \delta(\tau_{\min}(t) - \tau_{ij}(t)) \quad (9)$$

这里 $\delta \in [0, 1]$, 代表平滑的强度。如果 δ 取值为 0, 平滑机制将不发生作用。当 δ 取值为 1, 平滑机制使得各边上的信息素的数值设定成了 $\tau_{\min}(t)$, 这就意味着平滑机制使得各边上的信息素均返回到了初始数值, 消去了当前搜索所导致的各边上的信息素的差异。 $\tau_{ij}^*(t)$ 和 $\tau_{ij}(t)$ 分别代表了平滑之后与平滑之前边 ij 上的信息素的值。平滑机制的引入可以显著地提升算法的求解效率。

在 MMAS 中, 同样引入了局部搜索策略。引入局部搜索策略的好处是一方面可以加快算法收敛, 提高算法性能; 另一种使得蚁群在尽可能大的搜索空间中搜索, 产生全局最优解。MMAS 中, 对于 STSP 问题采用的启发式搜索策略是 2-opt^[19], 对于 ATSP 问题则采用的是 reduced 3-opt^[19]。实验表明^[32], 最大-最小蚂蚁系统是解决 TSP 问题最好的启发式算法之一。

3.4 蚁群优化算法

随着对蚁群算法研究的深入, 蚁群算法的应用领域也随之不断扩充。继 TSP 问题之后, 蚁群算法又相继应用于 QAP^[20], VRP^[21] 等组合优化问题, 网络路由问题^[22], 图的着色问题^[23] 以及机器人路径规划^[24] 等问题。蚁群算法与不同具体问题相结合而产生了不同的蚁群算法。这些不同的蚁群算法在外在形式和内在运行机制方面都有相似之处。M. Dorigo 等人于 1999 年提出了蚁群优化算法 (Ant Colony optimization, ACO)^[25], 该算法给出了蚁群算法的一个一般化的框架。蚁群优化算法的提出在蚁群算法发展历程中具有重要的意义。

ACO 的主要思想是: 如果求解的问题能够转换为在一个图中寻找最优路径, 那么 ACO 就能够用于寻找满足给定限定条件的最优路径, 进而完成对问题的求解。ACO 对蚁群算法的实现进行了高度的概括, 将蚁群算法主要概括为蚂蚁的行为, 信息素的挥发操作和一些守护操作 (daemon_actions)。其中守护操作是可选的。ACO 的伪代码如下:

```

procedure ACO ()
    while (termination_criterion_not_satisfied)
        schedule_activities
        ants_generation_and_activity ();
        pheromone_evaporation ();
        daemon_actions ();
    end schedule_activities
end while
end procedure

```

蚂蚁的行为可以描述为一群蚂蚁同时异步地在问题空间中相邻的状态之间移动, 可以形象化为在一个图中的相邻结点之间移动。移动依靠问题的启发信息和信息素所决定的选择策略以及问题本身的限制条件。通过移动, 逐步地构建问题的解。一旦蚂蚁构建出了问题的解, 或者蚂蚁正在构建问题解的过程时, 蚂蚁会对解进行评估, 然后依据解的好坏来释放一定的信息素到路径上。而这些信息素将进一步指导后续蚂蚁构造路径。在 ACO 中, 蚂蚁的行为用 ant_generation_and_activity() 来表示。

信息素的挥发操作可以描述为图的边上的信息素随着时间而逐渐消失。这样做是为了避免算法过快收敛于局部解, 信息素的挥发操作同样有利于蚂蚁对新的解空间进行搜索, 进而找出更好的解。在 ACO 中, 信息素的挥发操作作用 pheromone_evaporation() 来表示。

守护操作可以描述为实现一些蚂蚁所不能完成的集中控制任务。比如一些局部搜索策略; 比较各蚂蚁所产生的路径长度, 仅仅对其中的最短路径再次进行信息素的全局更新等等。在 ACO 中, 守护操作可以用 daemon_actions() 表示。

需要指出的是, ACO 对以上三个行为之间的进度和具体实现方式并没有严格的限制, 比如它们三者之间是否需要同步或者是否要彼此并行且独立。这就意味着在利用 ACO 解决实际问题的時候, 可以自由安排以上三个行为之间的实现进度以及具体的实现方式。

ACO 适用于离散优化问题的求解, 相应问题的特征在文 [26] 中有详细的说明。如果说, 前期的蚁群算法基本都是围绕蚂蚁系统而展开的, 那么在蚁群优化算法提出之后, 有关蚁群算法的研究和相关应用就基本上是围绕蚁群优化算法而进行的。

结论 本文主要从解决 TSP 问题的角度出发, 按照时间先后顺序分别介绍了蚁群算法系列中比较有代表性的几个算法。描绘了蚁群算法从 AS 到 ACO 发展完善的过程。在介绍各算法基本思想的基础上, 着重阐述了算法的主要特点和创新点及其意义, 从上述蚁群算法的发展过程中可以看出, 蚁群算法的核心就是对信息素的合理操作, 整个蚁群利用信息素作为一种间接通信工具而形成解决问题的合力。故, 如果要实现蚁群算法进一步的发展, 关于信息素的操作是一个主要的人手点。目前蚁群算法的应用越来越广, 国际上已经有专门的学术会议来讨论蚁群算法。相信随着研究的深入, 蚁群算法必将可以得到进一步的发展, 进一步拓宽蚁群的应用领域。最后需要补充的是, 在蚁群算法发展的历程中, 除了文中介绍的蚁群算法之外, 还有其他一些同样具有代表性的蚁群算法。比如: 带精英策略的蚂蚁系统 (Ant System with elitist Strategy, AS_{elitist})^[27], 基于优化排序的蚂蚁系统 (Rank-Based Version of Ant System, AS_{rank})^[28] 等等。

参考文献

- Colomni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies [A]. In: Proceedings of ECAL91-European Conference on Artificial Life [C]. Paris, France: Elsevier Publishing, 1991. 134~142
- Dorigo M, Maniezzo V, Colomni A. Ant system: optimization by a colony of cooperating agents [J]. IEEE Trans. Systems, Man, Cybernet. -Part B, 1996, 26(1): 29~41
- Dorigo M, Stutzle T. Ant Colony Optimization [M]. Cambridge, MA: MIT Press, 2004
- Dorigo M. Optimization, learning and natural algorithms [M]. [Ph. D. dissertation]. DEI, Politecnico di Milano, Italy, 1992
- http://www.tsp.gatech.edu/methods/progress/progress.htm
- Deneubourg J L, Aron S, Goss S, et al. The self-organizing exploratory pattern of the Argentine ant [J]. J. Insect Behavior, 1990(3): 159~168
- Bonabeau E, Theraulaz G, Deneubourg J L. Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies [C]. In: Proceedings Royal Society of London B, 1996, 263: 1565~1569
- Deneubourg J L, Goss S, Franks N, et al. The dynamics of collective sorting: robot-like ants and ant-like robots [C]. In: JA Meyer, et al. eds. Conference on Simulation of Adaptive Behavior, Cambridge, 1990. 356~363
- Robinson G E. Regulation of division of labor in insect societies. Ann. Rev. Entomol., 1992, 37: 637~665
- Bonabeau E, Sobkowski A, Theraulaz G, et al. Adaptive task allocation in APIred by a model of division of labor in social insects. In: D Iundh, et al. eds. Biocomputation and Emergent Computing, World Scientific, Singapore, 1997. 36~45

(下转第 194 页)

上,第 i 个可疑的样本点对成分 t_1, t_2, \dots, t_m 的贡献过大,即该样本点严重偏离数据集重心,可将该样本点视为异常点。

事实上也可以不先进行聚类,对样本数据中每个样本都直接利用主成分分析法进行检验,验证其是否是异常点,但是当样本量很大时计算量就会很大,由于绝大多数的样本点都不是异常点,于是很多的计算就会浪费在正常点的验证上,因此一般我们采取先聚类再对不包含在聚类中的数据检验的方式。如果检验不是异常点,就把它归入距离最近的一类中。这样,对于聚类来说,只把真正的异常点作为噪声处理,就不会把本来是正常数据误为噪声而去掉了。

另外,该方法还可以通过调节基于密度聚类算法^[7]的参数 ϵ 和 $MinPts$ 的值来确定异常点的选取方法。如果想最大限度地检测到所有的异常点,我们可以减小参数的值;如果增大参数的值,那么随着参数值增大检测到的异常点为确实为异常点的可能性就更大。

5 实验验证

我们采用来源于 UCI 的 wine 数据集,该数据集包含 185 个数据,有 13 个属性,先利用 DBSCAN 算法进行聚类,可以把数据集分成 3 类,共包含 178 个数据,其它的 7 个数据不包含在任何一个类中,利用 PCA 方法检验,取 $\alpha=0.05$,7 个数据在 95% 的检验水平上都是异常点,这与文[9]中基于距离的异常点挖掘方法所得出的最好结果一致。

结束语 本文给出了一个基于主成分分析的异常点挖掘

方法,这种方法把基于密度的聚类算法和基于统计的异常点挖掘方法结合了起来,同时又融合了数据维数消减的主成分分析方法。实验表明,该方法在异常点挖掘方面效果明显。

参考文献

- 1 王宏鼎,董云海,等. 异常点挖掘研究进展. 智能系统学报,2006,1(1):67~73
- 2 陈华,李继波. 异常(Outlier)检测算法综述. 大众科技,2005,9:96~97
- 3 李炎,李皓,等. 异常检测算法分析. 计算机工程,2002,28(6):5~6,32
- 4 李之棠,刘颖. 入侵检测中的模糊数据挖掘技术. 计算机工程与科学,2002,24(2):18~21
- 5 钱昌明,李国庆,等. 分类异常点检测算法及在 IDS 模型中的应用. 计算机应用研究,2006,23(4):94~96
- 6 Han Jiawei, Kamber M. Data mining: concepts and techniques. New York: Morgan Kaufmann Publishers, 2001
- 7 Ester M, Kriegel H, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial database with noise. In: Proceedings of the 2nd conference on Knowledge Discovering in Databases and Data Mining, Portland, USA, 1996. 226~231
- 8 王惠文. 偏最小二乘回归分析及其应用. 北京: 国防工业出版社, 1999. 130~184
- 9 Knorr E, Ng R. Algorithms for mining distance-based outliers in large datasets. In: Proceedings of Very Large Data Bases (VLDB'98), New York, USA, 1998. 392~403
- 10 traveling Salesman Problems [J]. Operations Research, 1973, 21:498~516
- 11 Krieger M J B, Billeter J B. The call of duty: Selforganised task allocation in a population of up to twelve mobile robots. Robotics and Autonomous Systems, 2000, 30:65~84
- 12 Maniezzo V. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. INFORMS J. Comput, 1999, 11(4):358~369
- 13 Lumer E, Faieta B. Diversity and adaptation in populations of clustering ants. In: J.-A. Meyer, et al. eds. Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats, Vol. 3, MIT Press/Bradford Books, Cambridge, MA, 1994. 501~508
- 14 Reimann M, Doerner K, Hartl R F. D-ants: savings based ants divide and conquer the vehicle routing problems. Comput. Oper. Res, 2004, 31(4):563~591
- 15 Dorigo M, Gambardella L M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem [J]. IEEE Trans. On Evolutionary Computation, 1997, 1(1):53~66
- 16 Sorges U, Gunes M, Bouazizi I. ara - the ant colony based routing algorithm for manets. In: Proc. of the 2002 ICPP Workshop on Ad Hoc Networks (IWAHN 2002), 2000. 79~85
- 17 Watkins C J C H. Learning with delayed rewards [A]: [PhD Thesis]. University of Cambridge, England, 1989
- 18 Casta D, Hertz A. Ant Can Colour Graphs [J]. Journal of the operational Research Society, 1997, 48:295~305
- 19 Dorigo M, Gambardella L M. A study of some properties of Ant-Q. In: Proc. PPSN IV-4th Int. Conf. Parallel Problem Solving from Nature. Berlin, Germany: Springer-Verlag, 1996. 656~665
- 20 Fan X P, Luo X, Yi S, et al. Path planning for robots based on ant colony optimization algorithm under complex environment [J]. Control and Decision, 2004, 19(2): 166~170
- 21 Johnson D S, McGeoch L A. The travelling salesman problem: a case study in local optimization. In: Local Search in Combinatorial Optimization. E. H. L. Aarts, et al. eds. New York: Wiley and Sons, 1997
- 22 Dorigo M, Di Caro G. The Ant Colony Optimization meta-heuristic. In: New Ideas Optimization. D. Corne, et al., eds. McGraw Hill, London, UK, 1999. 11~32
- 23 Stutzle T, Hoos H. Improvements on the Ant System: Introducing MAX-MIN Ant System [A]. In: Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms [C], Springer Verlag, Wien, 1997. 245~249
- 24 Dorigo M, Caro G D, Gambardella L M. Ant Algorithms for Discrete Optimization [J]. Artificial Life, 1999, 5(3):137~172
- 25 Stutzle T, Hoos H. Max-Min ant system [J]. Future Generation Computer System, 2000, 16:889~914
- 26 Bullnheimer B, Hartl R F, Strauss C. A New Rank-based Version of The Ant System: A Computational Study: [Technical Report POM-03/97]. Institute of Management Science, University of Vienna, Accepted for Publication in the central European Journal for Operations Research and Economics, 1997
- 27 Lin S, Kernighan B W. An effective heuristic algorithm for the Quadratic Assignment Problem [A]: [Technical Report IRIDIA/94~28]. Universite de Bruxelles, Belgium, 1994
- 28 Maniezzo V, Dorigo M, Coloni A. The Ant System Applied to the Quadratic Assignment Problem [A]: [Technical Report IRIDIA/94~28]. Universite de Bruxelles, Belgium, 1994