



# 蚁群最优化——模型、算法及应用综述<sup>①</sup>

刘士新<sup>1</sup>, 宋健海<sup>2</sup>, 唐加福<sup>1</sup>

(1. 东北大学信息科学与工程学院, 辽宁 沈阳 110004;

2. 上海宝信软件股份有限公司钢铁 ERP 事业部, 上海 201900)

**摘要:** 蚁群系统所具有的分布式组织模型对解决复杂组合优化问题、分布控制问题提供了很好的思路, 因此对蚁群系统行为及其自组织能力的研究已经引起了许多研究者的兴趣. 蚁群最优化 (ant colony optimization, ACO) 是蚁群系统的一个重要研究领域, 新模型、新方法、新应用不断出现. 论文对 ACO 的仿生学机理进行了描述, 综合 ACO 在不同种类的组合优化问题上的应用, 建立了 ACO 的一般化模型. 对 ACO 的典型实现过程和关键实现要素进行了分析, 指出了不同 ACO 算法的本质区别. 结合旅行商问题、二次指派问题以及网络路由问题等典型组合优化问题, 概述了 ACO 在静态组合最优化和动态组合优化问题中的应用. 最后讨论了 ACO 在建模、实现以及理论研究等方面的未来方向.

**关键词:** 人工生命; 蚁群系统; 群体智能; 元启发式算法; 组合优化; 综述

**中图分类号:** C934

**文献标识码:** A

**文章编号:** 1000-5781(2004)05-0496-07

## Ant colony optimization review: Modelling, algorithms and applications

LIU Shi-xin<sup>1</sup>, SONG Jian-hai<sup>2</sup>, TANG Jia-fu<sup>1</sup>

(1. School of Information Science & Engineering, Northeastern University, Shenyang 110004, China;

2. Metallurgical ERP Business Dept., Shanghai Baosight Software Company, Ltd. Shanghai 201900, China)

**Abstract:** The study on ant colonies behavior and their self-organizing capacities has been interesting for many researchers because it provides models of distributed organization which are useful to solve difficult combinatorial optimization problems (COPs) and distributed control problems. Ant colony optimization (ACO) is one important research area of ant colonies, and new models, methods and applications are presented increasingly. The bionics mechanism of ACO is described in this paper. Synthesizing various applications of ACO to different COPs, a general ACO model is presented. The typical implementation process and key elements of ACO are analyzed, and essential differences among different ACO algorithms are observed. The applications of ACO to static and dynamic COPs, such as traveling salesman problem, quadratic assignment problem, network routing problem are reviewed. The developing tendency on modeling, implementing and theory research is discussed.

**Key words:** artificial life; ant colony system; swarm intelligence; meta-heuristics; combinatorial optimization; review

① 收稿日期: 2002-12-19; 修订日期: 2004-07-10.

基金项目: 国家自然科学基金资助项目(70301007); 国家 863 计划 CIMS 主题资助项目(2002AA412010); 辽宁省博士启动基金资助项目(20021011); 沈阳市自然科学基金资助项目(1022036-1-04).

## 0 引言

蚁群系统(ant colony system)的行为方式和自组织能力,以及系统所具有的分布式组织模型对解决复杂组合优化问题、分布控制问题以及聚类分析问题提供了很好的解决思路<sup>[1~4]</sup>.目前,对蚁群系统的研究已经成为计算机科学领域的一个重要的研究方向,国内也有一些学者开始了这方面的研究<sup>[5,6]</sup>.组合优化问题是蚁群系统应用最广泛的一个领域,一般称之为蚁群最优化(ant colony optimization, ACO).ACO已经成为相关领域的热门研究方向,新模型、新方法、新应用层出不穷,本文根据作者掌握的文献,针对ACO模型、算法和应用研究进展进行了综述.

## 1 ACO 模型

ACO思想来源于对自然界中蚁群寻找食物过程的观察.在觅食过程中,蚂蚁在它所经过的路径上留下浓度与食物源质量成比例的信息素(pheromone),并能够感知信息素的存在及其浓度,以此指导自己的运动方向,倾向于朝着信息素浓度高的方向移动.于是,蚁群的集体行为便表现出一种信息正反馈现象:某一路径上走过的蚂蚁越多,则后来者选择该路径的概率就越大,因此质量好、距离近的食物源会吸引越来越多的蚂蚁,信息素浓度的增长速度会更快.蚂蚁个体之间就是通过这种信息的交流达到寻找食物和蚁穴之间最短路径的目的<sup>[7]</sup>.意大利学者Dorigo, Maniezzo和Colomi等人正是基于蚁群这种觅食活动,首先提出了ACO的最早形式——蚂蚁系统(ant system, AS)<sup>[7~9]</sup>,并应用在TSP旅行商问题中.AS算法被提出之后,其应用范围逐渐广泛,算法本身也不断被完善和改进,形成了一系列的ACO算法.

考虑一般的极小化问题 $(S, f, \Omega)$ ,其中 $S$ 为问题的候选解集,对应 $S$ 中的每个候选解 $s, s \in S$ ,有目标函数 $f(s)$ 与之相对应, $\Omega$ 为问题的约束集合.问题是寻找一个全局最优解 $s_{opt} \in S, s_{opt}$ 具有最小目标函数值且满足约束集合 $\Omega$ .在ACO中,问题 $(S, f, \Omega)$ 具有以下特性<sup>[10]</sup>:

1) 一个解可以划分为若干个组成部分,记所

有组成部分集合为 $C = \{c_1, c_2, \dots, c_N\}$ ;

2) 给定 $C$ 中的一个元素序列 $x = \{c_i, c_j, \dots, c_k, \dots\}$ ( $x$ 可以包含 $C$ 中全部元素或仅包含部分元素),则 $x$ 对应问题的一个求解状态,记所有可能的序列集合为 $X$ ,满足约束集 $\Omega$ 的序列集合为 $\bar{X}$ ,  $\bar{X} \subseteq X$ ;

3) 记问题可行解集为 $S^*, S^* \subseteq S$ ,则对 $\forall s \in S^*, \exists x \in \bar{X}$ 与之相对应;

4) 记 $x_i \in X$ 为问题求解过程所处的某一状态,对应部分解的目标函数值为 $J(x_i)$ ,在 $x_i$ 序列末尾加入新元素 $c_j \in C$ 后对应求解状态 $x_j$ ,对应部分解的目标函数值为 $J(x_j)$ ,则 $J(x_i) \leq J(x_j)$ .

基于以上特性,ACO中人工蚂蚁构造问题解的过程可以描述为在一个完全图 $G = (C, L)$ 上的移动过程, $G$ 中 $C = \{c_1, c_2, \dots, c_N\}$ 为节点集合,连接节点 $c_i$ 和 $c_j$ 的边记为 $(i, j)$ ,根据问题特性被赋予不同意义的权系数和浓度为 $\tau_{ij}$ 的信息素, $L = \{(i, j) | i, j = 1, 2, \dots, N, i \neq j\}$ 为边的集合.人工蚂蚁从某一节点 $c_i$ 开始,根据约束集 $\Omega$ 约束、信息素浓度 $\tau_{ij}$ 和问题的启发信息,按一定策略选择下一节点 $c_j$ ,直至达到一个完整解状态.蚂蚁在转移过程中同步或延迟地对其所经过边的信息素浓度进行更新,而每个边上信息素的浓度也会随着时间的推移根据一定的比例挥发.

在ACO中,人工蚂蚁除了自然具有的能力外,还被赋予了记忆功能和启发计算能力,记忆能力使得蚂蚁能够记载自己的行走路线,启发计算能力使得蚂蚁能够评估转移到的下一个节点对最终解带来的好处,除此之外,在有些算法中为了提高求解效果,还对人工蚂蚁赋予了局部搜索能力,即在构造一个完整解之后,再应用局部搜索算法对其构造的解进行改进.

## 2 ACO 实现

引入以下符号:

$\tau_{ij}$ ——某一求解状态下边 $(i, j)$ 上的信息素浓度;

$\eta_{ij}$ ——某一求解状态下蚂蚁由节点 $c_i$ 转移到 $c_j$ 的启发优先系数,针对具体问题根据启发式规则确定;

$k$ ——算法中蚂蚁的编号;

$M_k(i)$ ——当前处于节点  $c_i$  的蚂蚁  $k$  下一步可以访问的节点集合, 根据约束集  $\Omega$  确定;

$\alpha$ ——信息素浓度  $\tau_{ij}$  的相对重要性度量;

$\beta$ ——启发优先系数  $\eta_{ij}$  的相对重要性度量;

$q$ ——在  $[0, 1]$  区间随机产生的服从均匀分布的随机数;

$q_0$ ——蚂蚁选择下一节点的概率度量阈值,

$$0 \leq q_0 \leq 1;$$

$\delta$ ——学习步长;

$\rho$ ——信息素挥发系数;

$k_{gb}$ ——从算法开始执行到目前为止产生了最好解的蚂蚁编号;

$k_{ib}$ ——在当前循环中产生了最好解的蚂蚁编号;

则在 ACO 中, 处于节点  $c_i$  的蚂蚁  $k$  根据式(1)选择下一节点  $c_j$

$$c_j = \begin{cases} \arg \max_{c_i \in M_k(i)} \{ [\tau_{iz}]^\alpha \cdot [\eta_{iz}]^\beta \} & \text{如果 } q \leq q_0 \\ 0 & \text{其他} \end{cases} \quad (1)$$

式(1)中,  $0$  是依概率  $p_k(i, o)$  从  $M_k(i)$  中随机选择的

$$p_k(i, o) = \begin{cases} \frac{[\tau_{io}]^\alpha \cdot [\eta_{io}]^\beta}{\sum_{c_i \in M_k(i)} [\tau_{iz}]^\alpha \cdot [\eta_{iz}]^\beta} & \text{如果 } c_o \in M_k(i) \\ 0 & \text{其他} \end{cases} \quad (2)$$

在算法迭代过程中,  $\tau_{ij}$  的值依据式(3)更新

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \delta (\Delta\tau_{ij}^D + \Delta\tau_{ij}^S) \quad (3)$$

式(3)中,  $\Delta\tau_{ij}^D$  为信息素延迟更新量(delayed pheromone update), 即每个循环(每个蚂蚁完成一次解的构造过程)之后计算出来的更新量.  $\Delta\tau_{ij}^S$  为信息素同步更新量(step-by-step pheromone update), 在每个蚂蚁转移到一个新节点后计算出来的更新量.

ACO 的不同形式主要区别在于: (1) 求解过程中下一节点的选择策略; (2) 信息素浓度的更新策略. ACO 的一般流程如过程 1 所示, 其中 {} 中的内容为注释, 未说明的符号意义同前. 步骤 6°

的终止条件可以是算法达到预定的循环次数或经过指定次数的循环后最优解没有得到改进.

### [过程 1]

1° 初始化:  $\tau_{ij} = \tau_0, i, j = 1, 2, \dots, N, i \neq j, T_{gb} = \emptyset$ ;  $T_{gb}$  为蚂蚁  $k_{gb}$  经过的节点集合

2° For  $k := 1$  To  $m$  Do  $\{m$  为系统中蚂蚁的数量

设置蚂蚁  $k$  的出发节点  $c_{k1}$ ;

更新  $M_k(k1)$ ;

$T_k := \{c_{k1}\}$ ;

$T_k$  为第  $k$  个蚂蚁经过的节点集合

End-for

3° For  $n := 2$  To  $N$  Do

For  $k := 1$  To  $m$  Do

根据式(1)、(2)选择下一节点  $c_{kn}$

更新  $M_k(kn)$ ;

$T_k := T_k \cup \{c_{kn}\}$ ;  $T_k$  元素按加入顺序排列

End-for

For  $k := 1$  To  $m$  Do

$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \delta \cdot \Delta\tau_{ij}^S$ ;

{同步更新信息素浓度}

End-for

End-for

4° 蚂蚁采用局部搜索方法改进步骤 3° 中构造的解;

5° For  $k := 1$  To  $m$  Do

计算每个蚂蚁构造并经过局部搜索方法改进后的解的目标函数值;

End-for

如果当前最优解得到改进, 则更新  $T_{gb}$ ;

针对每一边  $(i, j)$ , 依据  $\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \delta \cdot \Delta\tau_{ij}^D$  延迟更新信息素浓度;

6° If 满足终止条件 Then 输出  $T_{gb}$ ;

Else 转步 2

## 3 ACO 应用

ACO 已经在很多的组合优化问题中获得了成功的应用<sup>[3]</sup>, 这些问题可以大体分为两类: 一类是静态组合优化问题, 这类问题的参数特性不会在求解过程中发生变化, 如 TSP(traveling salesman problem)、

QAP(quadratic assignment problem)、SOP(sequential ordering problem)、VRP(vehicle routing problem)、JSP(job-shop scheduling problem)等;另一类是动态组合优化问题,这类问题的参数特性会在求解过程中发生变化,如网络路由问题等。

### 3.1 在 TSP 中的应用

TSP 是 ACO 应用得最早、最多的一类组合优化问题。针对 TSP,ACO 模型中节点集合  $C$  由所有城市构成,边的集合  $L$  由各城市间的连接组成,边  $(i, j)$  的权重  $d_{ij}$  为城市  $i$  和  $j$  间的距离,对于对称 TSP,  $d_{ij} = d_{ji}$ ,对于非对称 TSP,  $d_{ij} \neq d_{ji}$ ,启发优先系数一般设置为  $\eta_{ij} = 1/d_{ij}$ 。

文献[8, 9] 首先将 AS 算法应用在 TSP 中,设计了 3 种算法:Ant-cycle, Ant-density 和 Ant-quantity,其中 Ant-cycle 求解效果最好,求解过程中依式(2)计算出的概率选择下一节点  $c_o$  (相当于  $q_0$  取值为 0)。对信息素浓度只进行延迟更新。

文献[11, 12] 基于 AS 算法和 Q 学习算法<sup>[13]</sup> 设计了 Ant-Q 算法,算法中  $\Delta\tau_{ij}^D$  采用两种计算方法:一种是全程最好(global best)更新算法;另一种是每个循环内最好(iteration best)更新算法。取  $\Delta\tau_{ij}^S = \gamma \cdot \max_{c_j \in M_k(i)} (\tau_{jx})$ ,  $\gamma$  为对下一转移节点评估的折扣系数。文献[14, 15] 对 Ant-Q 中  $\Delta\tau_{ij}^S$  值的计算进行了简化,取  $\Delta\tau_{ij}^S = \tau_0$ ,称简化后的算法为 ACS(ant colony system),并采用 2-Opt 和 3-Opt 局部搜索算法对蚂蚁构造的解进行改进,得到了很好的求解效果。

文献[16] 提出了一种改进的 AS 算法——AS<sub>rank</sub>,求解过程中依式(2)计算出的概率选择下一节点  $c_o$ 。对  $\tau_{ij}$  采用延迟更新方法,首先对全部蚂蚁构造的解依目标函数值由小到大排序,并选择排在前边的  $\sigma$  个蚂蚁对  $\tau_{ij}$  进行更新,排序越小,更新贡献量越大。

文献[17] 基于 AS 算法设计了 MMAS 算法,MMAS 在求解过程中依式(2)计算出的概率选择下一节点  $c_o$ 。在  $\tau_{ij}$  的更新策略上只采用延迟更新方法,  $\Delta\tau_{ij}^D$  计算方法和 Ant-Q 相同,为了防止算法的停滞,MMAS 对  $\tau_{ij}$  的值设置了上下限,并采用 2-Opt 等局部搜索算法改进由蚂蚁构造的解。

文献[18] 在 ACS 中引入了杂交算子。文献[5] 在 AS 算法中引入变异机制,充分利用了 2-Opt 简

洁高效的特点,使得该方法具有较快的收敛速度。文献[19] 首先在 AS 算法的基础上提出了相遇算法,提高了 AS 算法中蚂蚁一次周游的质量,然后将相遇算法与采用并行策略的分段算法相结合,提出一种基于 AS 算法的 TSP 分段求解算法。文献[20] 通过自适应地改变 AS 算法的挥发度等系数,实现了保证收敛速度的条件下提高解的全局最优性。文献[21] 将 ACS 应用推广到多目标 TSP 中。

### 3.2 在 QAP 中的应用

QAP 是继 TSP 后,ACO 在组合优化问题领域的又一个重要应用方面<sup>[3, 22]</sup>。目前针对 QAP 的算法多是利用 QAP 和 TSP 的相似性而对求解 TSP 的 ACO 算法的推广,并针对 QAP 问题的自身特点对算法做了相应的调整。

QAP 是将  $n$  个设备指派到  $n$  个地点,地点  $i$  和  $j$  ( $i, j = 1, 2, \dots, n; i \neq j$ ) 之间的距离为  $d_{ij}$ ,设备  $r$  和  $s$  ( $r, s = 1, 2, \dots, n; r \neq s$ ) 之间的流量为  $a_{ij}$ 。记  $\Psi$  为  $n$  个非重复的自然数  $\{1, 2, \dots, n\}$  的一个排列,排列中的第  $i$  个数值  $\Psi(i)$  表示设备  $i$  被指派到位置  $\Psi(i)$ ,记全部排列的集合为  $S(n)$ ,则 QAP 问题是寻找  $\Psi^* \in S(n)$ ,使得  $f(\Psi^*)$  极小

$$f(\Psi^*) = \arg \min_{\Psi \in S(n)} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \cdot d_{\Psi(i)\Psi(j)} \quad (4)$$

在应用 ACO 求解 QAP 时,首先将  $n$  个设备进行排列,然后根据排列顺序依次为每个设备选择一个地点。在 ACO 模型中,图  $G$  中节点集合  $C$  由  $n$  个地点组成,边的集合  $L$  为所有节点的全连接,信息素浓度  $\tau_{ij}$  对应设备  $i$  被指派到位置  $j$ ,人工蚂蚁随机地选择图  $G$  中的某个节点出发,走访每个节点一次后完成一个解的构造过程。

在文献[23] 中,对于设备  $i$ ,根据式(5)计算的概率值为其选择位置  $j$

$$p_{ij}^k = \begin{cases} \frac{\alpha \cdot \tau_{ij} + (1 - \alpha) \cdot \eta_{ij}}{\sum_{r \in M_k(i)} (\alpha \cdot \tau_{ir} + (1 - \alpha) \cdot \eta_{ir})}, & \text{如果 } j \in M_k(i) \\ 0 & \text{其它} \end{cases} \quad (5)$$

式中:  $\eta_{ij}$  的值为目标函数值下界<sup>[24]</sup> 的倒数。信息素浓度  $\tau_{ij}$  的更新方法同 Ant-cycle 类似,局部搜索算法采用文献[25] 介绍的禁忌搜索算法。

文献[26]针对 QAP 设计了 ANTS-QAP 算法,对于位置  $i$ ,根据式(5)计算的概率值为其选择设备  $j$ ,  $\eta_{ij}$  的值与文献[24]中的方法类似,但计算复杂性要小.  $\tau_{ij}$  的更新算法同 Ant-cycle 类似. 局部搜索算法同文献[23].

文献[17]介绍了 MMAS 在 QAP 中的应用,针对 QAP,式(2)中取  $\eta_j = 1$ ,即不采用启发式规则. 局部搜索算法采用文献[25]介绍的禁忌搜索算法和 2-Opt 算法,结果表明采用禁忌搜索算法求解效果更好.

文献[27]针对 QAP 设计了 FANT 算法,算法中仅使用一个蚂蚁,在算法执行过程中对  $\tau_{ij}$  更新的学习步长作自适应调整. 文献[28]针对 QAP 设计了 HAS-QAP 算法,与一般 ACO 算法不同, HAS-QAP 并没有将信息素浓度  $\tau_{ij}$  用在解的构造过程中,而是用在对解的改进过程中. 文献[29]设计的 ANTabu 算法与 HAS-QAP 具有相似的结构,但采用并行机制实现.

### 3.3 在网络路由问题中的应用

蚁群最优化在动态组合优化问题中的应用主要集中在网络通信方面,特别是网络路由选择问题上. 网络路由选择问题可以简单描述为在一个代表通信网络的全连接图中求解任何两个节点的最短路径问题. 虽然在静态环境下该问题可以通过多项式算法求解<sup>[30]</sup>,但在节点间流量(边的权重)随时间变化情况下,该问题的求解变得非常困难.

文献[31]针对网络路由问题设计了算法 AntNet. 与一般 ACO 算法不同,在 AntNet 中,对于任一有向弧  $(i, j)$ ,有  $N - 1$  个信息素浓度值  $\tau_{ijd} \in [0, 1]$ ,  $d = 1, \dots, N$  且  $d \neq i$  与之对应,  $\tau_{ijd}$  代表从节点  $i$  到达目的节点  $d$  时有向弧  $(i, j)$  上的信息素浓度,与弧  $(i, j)$  相对应的还有启发优先系数  $\eta_{ij}$ ,其值可以根据当前时刻由节点  $i$  向节点  $j$  待发送的数据队列长度计算,等待队列越短,  $\eta_{ij}$  值越大. 算法中每个蚂蚁有一个源节点  $s$  和目的节点  $d$ ,在转移过程中,处于节点  $i$  的蚂蚁  $k$  根据由  $\tau_{ijd}$  和  $\eta_{ij}$  计算的概率值选择下一节点  $j$ . 在构造由  $s$  到  $d$  的路径过程中,蚂蚁经历与数据转移同样的延迟时间,因此,蚂蚁由  $s$  到达  $d$  所经历的时间  $T_{sd}$  可以用作对该路径质量的评价. 当蚂蚁  $k$  完成一次路径构造过程后,根据该路径的质量对其所经过弧上的信息素浓度进行更新,而对

于在路径构造过程中未经过弧的信息素浓度进行挥发计算.

### 3.4 在其他组合优化问题中的应用

文献[32]针对 SOP<sup>[33]</sup>设计的 HAS-SOP 中,对代表 SOP 的完全图  $G = (V, A)$  重新定义:边  $(i, j)$  的权重为  $l_{ij} = t_{ij} + p_j$ ,  $t_{ij}$  代表完成工作  $i$  后开始工作  $j$  需要的准备时间,  $p_j$  代表工作  $j$  的加工时间. 如果工作  $i$  是工作  $j$  的前序工作,则令  $l_{ij} = \infty$ . 则可将求解 TSP 的 ACO 算法推广到 SOP 中,不同点在于每个蚂蚁的出发点均为节点 0,并且在构造解的过程中遵循 SOP 的特定约束.

文献[34]将 AS 算法应用于 JSP,模型中图  $G$  的节点集合  $C = O \cup \{o_0\}$ ,其中  $O = \{\dots, o_{jm}, \dots\}$  为所有工件要经过的工序集合,  $o_{jm} \in O$  为工件  $j$  需要在机器  $m$  上完成的工序,加工时间为  $d_{jm}$ ,  $o_0$  为虚工序. 边的集合  $L$  由两部分组成:一部分为  $O$  中所有节点的全连接,其中连接同一工件各工序的边由于工序间的先后关系设定为有向边,另一部分为节点  $o_0$  到各工件的第 1 个工序节点的有向连接. 在求解过程中,每个蚂蚁从节点  $o_0$  出发,根据边上信息素的浓度  $\tau_{ij}$ 、启发优先系数  $\eta_{ij}$  以及约束集  $\Omega$  的约束随机选择下一节点,直到构造一个完整的解.

文献[35]将 AS<sub>rank</sub> 算法应用于 VRP 中,模型中图  $G$  的节点集合  $C = \{n_0, n_1, \dots, n_n\}$ ,  $n_0$  代表车库,共有  $M$  辆车在  $n_0$  停放. 在求解过程中每个蚂蚁按照如下方法构造行使路径:从  $n_0$  出发,根据边上信息素的浓度  $\tau_{ij}$ 、启发优先系数  $\eta_{ij}$  以及约束集  $\Omega$  的约束随机选择下一节点. 如果由于车辆能力  $D$  或最大行使里程  $L$  的限制而不能继续走访其它顾客时,返回节点  $n_0$ ,然后开始新的行程,直到构造问题的一个解为止才完成一次构造过程. 算法中  $\eta_{ij}$  基于文献[36]的启发式算法设计,局部搜索算法采用基于 2-Opt 的启发式算法.

## 4 讨论及结论

ACO 虽然在许多类型组合优化问题求解中得到了很好的应用,但仍有很多方面值得深入研究.

(1) ACO 把每个蚂蚁作为一个 agent,构成一个多 agent 系统. 目前出现的算法中,每个蚂蚁具

有相同的特性和能力,并通过信息素实现彼此通信.在以后的研究中可以考虑给每个蚂蚁赋予更多的智能,并具有学习能力;在一个系统中,可以考虑对蚂蚁进行分工,不同的蚂蚁具有不同的能力并完成不同的任务,彼此协调,共同实现系统目标.另外,尽管 ACO 应用非常广泛,但其基本实现是非常相似的,如何针对 ACO 的 agent 进行程序框架设计,使得该 agent 框架具有广泛的适用性也是一个需要深入研究的问题.

(2) 本文给出了针对多种组合优化问题的 ACO 建模方法,但绝非该类问题的唯一建模方法,比如,文献[10]中针对 QAP 设计的模型中节点集合  $C$  就由全部设备和全部节点构成,而且事先并未对设备(位置)排定顺序,蚂蚁在构造解的过程中交替选择设备节点和位置节点.因此,同样一类组合优化问题,可以建立不同的模型,不同的模型中信息素浓度  $\tau_{ij}$  被赋予不同的意义,问题的解空间也因模型而异,如何针对具体问题设计最合适的模型需要深入研究.

(3) ACO 中蚂蚁根据信息素浓度  $\tau_{ij}$  和启发优先系数  $\eta_{ij}$  来计算选择下一节点的概率,并在算法执行过程中延迟和(或)同步更新  $\tau_{ij}$ . 采用这样的策略,蚂蚁在构造一个新解的过程中需要做探索新解(exploration)和利用求解经验(exploitation)的平衡.根据算法特点,一般可以采用4种平衡策

略:①在计算选择下一节点的概率时考虑参数  $\alpha$  和  $\beta$  的平衡;②对  $\tau_{ij}$  的更新方法采用更加合理的策略,比如使其变化曲线呈饱和曲线形式,或对其设定上下限;③在对  $\tau_{ij}$  更新前,对目标函数值进行标度处理(scaling),使每个解对  $\tau_{ij}$  的更新贡献量不至于因目标函数值差别而过于悬殊;④可以考虑在算法的初始阶段采用一定的策略,使得蚂蚁经过不同的节点构造解,充分扩展蚂蚁的搜索空间,然后再利用已知解的信息对求解空间进行深度探索.

(4) 虽然 ACO 在很多组合最优化问题的应用中都获得了良好的求解效果,但在理论上目前还未见到一般性的最优收敛性证明.文献[37]针对一种特殊的 ACO — GBAS 证明出了以下结论:给定任意小的正数  $\epsilon > 0$  和一些固定的算法参数,经过  $t \geq t_0$  次循环后,GBAS 将以概率  $P_t \geq 1 - \epsilon$  收敛到最优解,其中  $t_0$  为循环代数,由  $t_0 = f(\epsilon)$  决定. GBAS 与 AS 算法非常相似,主要不同点是 GBAS 执行过程中只有当解得到改进时才更新  $\tau_{ij}$ . 文献[37]的结论为其它 ACO 的最优收敛性证明打下了基础.

总之,ACO 作为一种新的且很有前途的研究领域,处于人工生命与运筹学的交叉位置.随着对其研究的不断深入,ACO 的模型将更加丰富,应用将更加广泛,相应的理论基础也会更加完善.

## 参考文献:

- [1] Dorigo M, Bonabeau E, Theraulaz G. Ant algorithms and stigmergy[J]. Future Generation Computer Systems, 2000, 16: 851—871.
- [2] Bonabeau E, Dorigo M, Theraulaz G. Inspiration for optimization from social insect behaviour[J]. Nature, 2000, 406: 39—42.
- [3] Dorigo M, Caro G D, Gambardella L M. Ant Algorithms for Discrete Optimization[J]. Artificial Life, 1999, 5(3): 137—172.
- [4] Martin M, Chopard B, Albuquerque P. Formation of an ant cemetery: Swarm intelligence of statistical accident[J]. Future Generation Computer Systems, 2002, 18: 893—901.
- [5] 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.
- [6] 张纪会, 高齐圣, 徐心和. 自适应蚁群算法[J]. 控制理论与应用, 2000, 17(1): 1—5.
- [7] Dorigo M, Maniezzo V, Colomi A. The ant system: Optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics—Part B, 1996, 26(1): 29—41.
- [8] Colomi A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies[A]. Proceedings of ECAL91—European Conference on Artificial Life[C]. Paris, France: Elsevier Publishing, 1991. 134—142.
- [9] Colomi A, Dorigo M, Maniezzo V. An Investigation of some properties of an “Ant algorithm”[A]. Proceedings of the Parallel Problem Solving From Nature Conference (PPSN92)[C]. Brussels, Belgium: Elsevier Publishing, 1992. 509—520.
- [10] Dorigo M, Stützle T. The ant colony optimization metaheuristic: Algorithms, applications, and advances[A]. Metaheuristics Handbook[M]. Glover F, Kochenberger G eds., International Series in Operations Research and Management Science, Kluwer, 2001.
- [11] Gambardella L M, Dorigo M. Ant-Q: A reinforcement learning approach to the traveling salesman problem[A]. Proceedings of ML

- 95, Twelfth International Conference on Machine Learning[C]. Prieditis A, Russell S, eds. Tahoe City, CA, Morgan Kaufmann, 1995. 252—260.
- [12] Dorigo M, Gambardella L M. A study of some properties of ant-Q[A]. Proceedings of the Fourth International Conference on Parallel Problem Solving From Nature[C]. Voigt H M, Ebeling W, Rechenberg I, Schwefel H S, Eds. Berlin: Springer-Verlag, 1996. 656—665.
- [13] Watkins C J C H. Learning with Delayed Rewards[D]. Psychology Department, Cambridge: University of Cambridge, 1989.
- [14] Dorigo M, Gambardella L M. Ant colonies for the traveling salesman problem[J]. BioSystems, 1997, 43: 73 — 81.
- [15] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53 — 66.
- [16] Bullnheimer B, Hartl R F, Strauss C. A New Rank Based Version of the Ant System: A Computational Study[R]. Vienna: SFB Adaptive Information Systems and Modelling in Economics and Management Science, 1997.
- [17] Stützle T, Hoos H H. MAX — MIN ant system[J]. Future Generation Computer Systems, 2000, 16: 889—914.
- [18] 陈 焯. 带杂交算子的蚁群算法[J]. 计算机工程, 2001, 27(12): 74—76.
- [19] 吴 斌, 史忠植. 一种基于蚁群算法的 TSP 问题分段求解算法[J]. 计算机学报, 2001, 24(12): 1328—1333.
- [20] 王 颖, 谢剑英. 一种自适应蚁群算法及其仿真研究[J]. 系统仿真学报, 2002, 14(1): 31—33.
- [21] 马 良, 蒋 馥. 多目标旅行售货员问题的蚂蚁算法求解[J]. 系统工程理论方法应用, 1999, 8(4): 23—27.
- [22] Stützle T, Dorigo M. ACO algorithms for the quadratic assignment problem[A]. In Corne D, Dorigo M, Glover F, eds., New Methods in Optimization[M]. New York: McGraw-Hill, 1999.
- [23] Maniezzo V, Colomi A. The ant system applied to the quadratic assignment problem[J]. IEEE Transactions on Knowledge and Data Engineering, 1999, 11(5): 769—778.
- [24] Gilmore P. Optimal and suboptimal algorithms for the quadratic assignment problem[J]. Journal of SIAM, 1962, 10: 305—313.
- [25] Taillard D. Robust taboo search for the quadratic assignment problem[J]. Parallel Computing, 1991, 17: 443 — 455.
- [26] Maniezzo V. Exact and Approximate Nondeterministic Tree-search Procedures for the Quadratic Assignment Problem[R]. Università di Bologna, Sede di Cesena, Italy, 1998.
- [27] Taillard D, Gambardella L M. Adaptive Memories for the Quadratic Assignment Problem[R]. IDSIA, Lugano, Switzerland: IDSIA-A-87-97, 1997.
- [28] Gambardella L M, Taillard D, Dorigo M. Ant Colonies for the QAP[R]. IDSIA-4-97, IDSIA, Lugano, Switzerland, 1997.
- [29] Talbi E-G, Roux O, Fonlupt C, Robillard D. Parallel Ant Colonies for the quadratic assignment problem[J]. Future Generation Computer Systems, 2001, 17: 441—449.
- [30] Bertsekas D. Network Optimization: Continuous and Discrete Models[M]. Belmont, MA: Athena Scientific, 1998.
- [31] Di Caro G, Dorigo M. AntNet: Distributed stigmergetic control for communications networks[J]. Journal of Artificial Intelligence Research, 1998, 9: 317—365.
- [32] Gambardella L M, Dorigo M. An ant colony system hybridized with a new local search for the sequential ordering problem[J]. Journal on Computing, 2000, 12: 1—29.
- [33] Escudero L F. An inexact algorithm for the sequential ordering problem[J]. European Journal of Operational Research, 1988, 37: 232—253.
- [34] Colomi A, Dorigo M, Maniezzo V, et al. Ant system for job-shop scheduling[J]. JORBEL—Belgian Journal of Operations Research, Statistics and Computer Science, 1994, 34(1): 39—53.
- [35] Bullnheimer B, Hartl R F, Strauss C. An improved ant system algorithm for the vehicle routing problem[J]. Annals of Operations Research, 1999, 89: 319—328.
- [36] Paessens H. The savings algorithm for the vehicle routing problem[J]. European Journal of Operational Research, 1988, 34: 336—346.
- [37] Gutjahr W J. A graph-based ant system and its convergence[J]. Future Generation Computer Systems, 2000, 16: 873—888.

#### 作者简介:

刘士新(1968—), 男, 辽宁调兵山人, 博士, 副教授, 研究方向: 制造执行系统、项目管理、最优化方法等;

宋健海(1965—), 男, 浙江绍兴人, 博士, 上海宝信软件股份有限公司高级工程师, 研究方向: 钢铁企业 ERP、MES 及企业信息系统整体解决方案等;

唐加福(1965—), 男, 湖南东安人, 博士, 教授, 研究方向: 生产计划与调度、进化算法、Fuzzy 优化理论与方法等。