

# 一类新型改进的广义蚁群优化算法

张代远

(南京邮电大学 计算机学院, 江苏 南京 210003;  
江苏省无线传感网高技术研究重点实验室, 江苏 南京 210003;  
南京邮电大学 计算机技术研究所, 江苏 南京 210003)

**摘要:**提出了一类新型蚁群优化算法。该算法改进了概率选择函数,将概率选择函数由严格单调增函数推广为有界函数,给出了蚂蚁在某一源节点选择下一个节点的更一般的表达式。证明了算法收敛的重要定理:即对足够大的迭代次数,改进的广义蚁群优化算法至少找到最优解一次的概率趋近于1。提出了信息素渐近平衡原理。在信息素更新规则中,引入了信息素残留率函数、信息素增量函数。证明了渐近信息素在最优路径上将会趋于一个正数,而在非最优路径上将会趋于0。最后,计算机仿真实验结果表明,无论是获得的最优解的质量还是算法的收敛速度,文中提出的改进的广义蚁群优化算法都优于传统的蚁群优化算法。

**关键词:**人工智能;蚁群优化算法;收敛性;信息素更新规则

**中图分类号:**TP31

**文献标识码:**A

**文章编号:**1673-629X(2012)06-0039-06

## A New Improved Generalized Ant Colony Optimization Algorithm

ZHANG Dai-yuan

(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;  
Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China;  
Institute of Computer Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

**Abstract:** A new improved generalized ant colony optimization algorithm (IGACO) is proposed in this paper. The selected probability functions are generalized from strictly increasing continuous functions to bounded functions, which gives a more general form of expression for the probability of selecting the next node. An important theorem is proved for describing the convergence of IGACO algorithm, i. e. for a sufficiently large number of algorithm iterations, the probability of finding the globally optimal solution at least once tends to 1. A principle of pheromone asymptotic balance is proposed. In the pheromone update rule, the residual rate function of pheromone and the global increasing function of pheromone are presented. Prove that the residual pheromone tends to a positive number on the edges that are globally optimal solution, and tends to 0 on the edges that are not globally optimal solution. Finally, the computational simulation shows that, compared with traditional ant colony optimization algorithm, the IGACO algorithm has good performance both on globally optimal solution and convergent speed.

**Key words:** artificial intelligence; ant colony optimization algorithm; convergence; pheromone update rule

## 0 引言

Marco Dorigo 等学者在真实蚂蚁觅食行为的启发下提出了蚁群优化算法(Ant Colony Optimization algorithm, ACO 算法),它是组合优化问题的一种后启发式近似求解方法。

在蚁群优化算法中,一些人工蚂蚁模拟真实蚂蚁通过环境的间接交流来求解问题<sup>[1]</sup>。优良解的构造就是这些人工蚂蚁之间交互合作的结果。为了讨论方

便,文中也将人工蚂蚁简称为蚂蚁。

虽然蚁群算法已经获得了广泛的应用,但是很多学者只是从应用的角度表明了该算法的有效性。例如,实验表明,蚁群优化算法在许多组合优化的困难问题<sup>[2,3]</sup>的应用中是有效的,但是科学家们对于蚁群优化算法能够成功的机理几乎给不出深刻的理论解释。但是对于算法的性能有一些研究。例如,Meuleau 和 Dorigo<sup>[4]</sup>证明了 ACO 算法与一类统计梯度下降算法有密切联系,对一类特殊的 ACO 算法,它将以概率 1 收敛到局部极优解。文献[5,6]研究了一类蚁群算法的收敛性。Gutjahr<sup>[5]</sup>对一类特殊的 ACO 算法,证明了能够以概率  $1 - \varepsilon$  收敛到最优解,并称为基于图的蚁群

收稿日期:2011-11-06;修回日期:2012-02-10

基金项目:江苏高校优势学科建设工程资助项目(yx002001)

作者简介:张代远(1957-),男,教授,博士,研究生导师,研究方向为人工智能、计算机体系结构、计算机应用等。

系统(GBAS)。Stützle 和 Dorigo<sup>[7]</sup>给出了一个简单的最优解的收敛性证明,他们将这类算法取名为ACO<sub>gb,τ<sub>min</sub></sub>算法。与传统的ACO算法一样,ACO<sub>gb,τ<sub>min</sub></sub>算法中信息素蒸发率是常数<sup>[7,8]</sup>。ACO<sub>gb,τ<sub>min</sub></sub>算法的概率选择函数的分子和分母的各项是正的指数函数。Zhang<sup>[9]</sup>提出了广义蚁群优化算法,该算法推广了ACO<sub>gb,τ<sub>min</sub></sub>算法,其概率选择函数的分子和分母的各项是严格单调增连续函数。因此,ACO<sub>gb,τ<sub>min</sub></sub>算法中的概率选择函数仅仅是广义蚁群优化算法的概率选择函数的一个特例。Zhang<sup>[9]</sup>证明了广义蚁群优化算法能收敛到最优解并且给出了一些应用<sup>[10]</sup>。文中进一步改进了Zhang<sup>[9]</sup>提出的广义蚁群优化算法,称为改进的广义蚁群优化算法(Improved Generalized Ant Colony Optimization algorithm, IGACO 算法)。

概率选择函数形式的一般化有利于计算,有助于拓广算法的应用领域,在理论研究上也具有重要意义。在文中,概率选择函数的形式更加一般化,IGACO 算法只要求组成概率选择函数的分子和分母的各项为有定义、有界的正函数就可以保证算法收敛到最优解。文中提出了信息素渐近平衡原理。引入了信息素残留率函数、信息素增量函数。对信息素残留率函数进行了建模。文中提出的信息素增量函数的形式与文献[9]中的形式完全不同。提出了渐近信息素等一些新概念,并证明了渐近信息素在最优路径上将会趋于一个正数,而在非最优路径上将会趋于0。最后,通过TSP问题验证了IGACO算法是非常有效的,要优于传统的蚁群优化算法。

## 1 改进的广义蚁群优化算法的基本原理

### 1.1 改进的广义蚁群优化算法的基本概念

假设 $\Omega$ 是某一类空间(如度量空间), $D \subseteq \Omega$ , $\Phi$ 是定义在 $D$ 上的泛函。

1) 假设有限非空集合 $D = \{c_1, c_2, \dots, c_{N_i}\} \in \Omega$ ,  $D$ 中至少有2个结点;

2) 在 $D$ 上定义泛函 $\Phi(D) = F(c_1, c_2, \dots, c_{N_i})$ ;

3) 找一个最优代价函数 $f^* \in F$ ,使得 $f^*(D) = \min \Phi(D) = \min \Phi(c_1, c_2, \dots, c_{N_i})$ ,最优代价函数的值记为 $s^* = f^*(D) = f^*(c_1, c_2, \dots, c_{N_i})$ 。

$$f^*(D) = \min \Phi(D) \quad (1)$$

$$s^* = f^*(D) \quad (2)$$

概率选择函数的表达式如下(函数的参数是当前所处状态的信息):

$$P(r(t), d(t)) = \frac{F(\tau(r(t), d(t), t))}{Q(\tau(r(t), d(t), t))} \quad (3)$$

其中 $P(r(t), d(t))$ 表示在 $t$ 时刻,由当前源结点 $r(t)$ 出发,到达目标结点 $d(t)$ 之间的选择概率,且有

$0 < F(\tau(t)) < \infty, 0 < F(\tau(t))/Q(\tau(t)) \leq 1, \tau_i(t) \in [a_i, b_i]$ ,  $t$ 是时间参数, $r(t)$ 表示 $t$ 时刻的源, $d(t)$ 表示 $t$ 时刻的目标, $P(r(t), d(t))$ 表示在 $t$ 时刻,由源向目标的选择概率。

对于选定的某一个源结点 $i \in r(t)$ 和目标结点 $j \in d(t)$ ,也可以将(3)式写成:

$$P(i \in r(t), j \in d(t)) = \frac{F(\tau(i, j, t))}{Q(\tau(i, j, t))} \quad (4)$$

定义1(IGACO 算法):假设 $\Omega$ 是某一类空间, $D \subseteq \Omega$ , $\Phi$ 是定义在 $D$ 上的泛函。在 $t$ 时刻, $r(t) \in D_s$ , $d(t) \in D_d$ , $D_s$ 和 $D_d$ 分别表示源域和目标域, $D_s \subseteq D$ , $D_d \subseteq D$ ,则IGACO算法指的是如下的一类最优化问题:

$$\begin{cases} \min \Phi(D) \\ \text{s. t. } P(r(t), d(t)) = \frac{F(\tau(r(t), d(t), t))}{Q(\tau(r(t), d(t), t))} \end{cases} \quad (5)$$

其中, $F(\tau)$ 、 $Q(\tau)$ 在 $R^n$ 上有定义,而且都是变量 $\tau$ 的有界函数,且满足以下条件 $0 < F(\tau(t))/Q(\tau(t)) \leq 1$ 。为了方便在计算机中实现迭代计算,需要将连续时间 $t$ 离散化并记为 $t_m$ 。令 $t = \{t_m | m = 0, 1, \dots\}$ ,今后假设离散化的时间序列 $t_m$ 满足 $0 \leq t_0 < t_1 < \dots < t_{m-1} < t_m < t_{m+1} < \dots$ 。

定义2:从源结点到目标结点所经历的时间称为一个迭代周期(或称为一次迭代)。

定义3: $m$ 个时间上相邻的一次迭代称为 $m$ 个迭代周期(或称为 $m$ 次迭代)。

定理1(IGACO 算法的收敛性定理):假设在一次迭代中经历了有限个结点。若函数 $F(\tau)$ 、 $Q(\tau)$ 在 $R^n$ 上有定义,而且都是变量 $\tau$ 的有界函数,且满足以下条件 $0 < F(\tau(t))/Q(\tau(t)) \leq 1$ ,则对于一个任意小的正数 $\varepsilon > 0$ 和一个充分大的迭代次数 $m$ ,IGACO算法在 $m$ 次迭代内至少找到一次最优解的概率 $P$ 满足以下不等式

$$P^* \geq 1 - \varepsilon$$

或者

$$\lim_{m \rightarrow \infty} P^* = 1$$

证明:假设总共进行了 $m$ 个迭代周期,在一个迭代周期中有 $n+1$ 步,中间经历了 $n$ ( $1 < n < \infty$ )个结点。若假设在第 $k$ 个迭代周期中获得了最优解,最优解的结点序列是: $r(t), c_{k1}(t), c_{k2}(t), \dots, c_{kn}(t), d(t)$ ,则在第 $k$ 个迭代周期中求得最优解的概率是

$$P(r(t), c_{k1}(t), c_{k2}(t), \dots, c_{kn}(t), d(t)) = \prod_{j=1}^{n+1} P_{kj} \quad (6)$$

其中 $P_{k1} = P(r(t), c_{k1}(t))$ ,  $P_{k2} = P(c_{k1}(t), c_{k2}(t))$ ,  $\dots$ ,  $P_{k,n+1} = P(c_{kn}(t), d(t))$ ,每个 $P_{kj}$ 由概率

选择函数计算。显然,在一个迭代周期中,只要有任何一个中间结点选择不正确,就不能求得最优解。于是,至少有一步没有正确选择中间结点的概率是

$$\bar{P}_k = 1 - \prod_{j=1}^{n+1} P_{kj} \quad (7)$$

显然,在  $m$  个迭代周期完成之后仍然没有求得最优解的概率是  $\bar{P} = \prod_{k=1}^m \bar{P}_k$ , 于是,在  $m$  个迭代周期完成之后至少求得一次最优解的概率是

$$P^* = 1 - \prod_{k=1}^m \bar{P}_k = 1 - \prod_{k=1}^m (1 - \prod_{j=1}^{n+1} P_{kj}) \quad (8)$$

根据定理的条件显然有  $0 < P_{kj} \leq 1$ , 这样就能保证

$$\lim_{m \rightarrow \infty} P^* = \lim_{m \rightarrow \infty} (1 - \prod_{k=1}^m (1 - \prod_{j=1}^{n+1} P_{kj})) = 1 \quad (9)$$

也就是说,对于一个任意小的数  $\varepsilon > 0$  和一个充分大的迭代次数  $m$ , IGACO 算法在  $m$  次迭代内至少找到一次最优解的概率  $P$  满足不等式  $P^* \geq 1 - \varepsilon$ , 或者  $\lim_{m \rightarrow \infty} P^* = 1$ 。

证毕

## 1.2 信息素更新过程

### 1.2.1 信息素渐近平衡原理与信息素更新规则

信息素的更新规则对于求得最优解至关重要。从物理概念上看,在开始的时候,路径上没有信息素,所以一旦蚂蚁留下信息素就会很快蒸发掉,或者说这时候蒸发率高,而信息素残留率很低。经过一段时间之后,随着路径上蚂蚁数量的增多,信息素的蒸发量也越来越大,散发到空气中的信息素又会反过来沉积到路径上来。到一定的时间之后,这种蒸发和沉积就会达到一种动态平衡。达到动态平衡之后,残留在路径上的信息不再变化,而维持着一个恒定的数值,或者说,信息素残留率达到了一个恒定的数值。基于以上的考虑,文中作者提出了一个基本原理—信息素渐近平衡原理。

信息素渐近平衡原理:假设给定一个图,图中有确定数量的结点,结点之间有连通的边,并有  $n$  只蚂蚁在结点之间不停地随机走动并在边上留下信息素,则随着时间的推移,任何路径上残留的信息素都将达到稳定的确定值。

可以给出如下的信息素更新规则的伪代码。

Procedure Phomone\_Update

$$\forall (i, j): \tau(i, j, t_m) \leftarrow \delta(t_m) \cdot \tau(i, j, t_{m-1}) \quad (10)$$

$$\text{if } f(s) < f(\hat{s}), \text{ then } \hat{s} \leftarrow s \quad (11)$$

$\forall$  ant  $k$ , if  $M(s) == \text{True}$ ,

$$\text{then } \forall (i, j) \in s, \tau(i, j, t_m) \leftarrow \tau(i, j, t_m) + \mu(t_m) \quad (12)$$

End Procedure

后面将要证明,在很宽松的条件下,上面的信息素更新规则满足信息素渐近平衡原理。

$\tau(t)$  称为信息素,随时间  $t$  变化,  $0 < \tau(t) < \infty$ 。信息素是蚂蚁寻找路径的依据,是一个重要参数。

$\delta(t)$  称为信息素残留率函数(相应地,将  $\nu(t) = 1 - \delta(t)$  称为信息素蒸发率函数),该函数以时间  $t$  作为自变量。在 IGACO 算法中,采用函数式的信息素挥发方式,而在传统方法中,信息素以一个常数比率挥发<sup>[2]</sup>。

$\mu(t)$  是信息素增量函数,该函数以时间  $t$  作为自变量。若寻找的路径是当前的最优路径,则使用该函数作为最优路径的信息素增量,以更新其构造路径上的信息素。

$M(s)$  是信息素更新标志函数,其定义为:

$$M(s) = \begin{cases} \text{TRUE} & \text{if } s \text{ is the global best solution; } s = \hat{s} \\ \text{FALSE} & \text{otherwise} \end{cases} \quad (13)$$

构造可行解之前,蚂蚁被随机放置在某一个结点。然后,根据当前结点的所有邻接边(路径)上的信息素数值,按照(5)式计算概率,蚂蚁依概率选择函数的数值选择下一个结点,并从当前结点移动到下一个结点,直到最后完成一个可行解的构造。当蚂蚁完成解的构造之后,执行信息素更新。

### 1.2.2 信息素残留率函数

开始时,路径上没有蚂蚁走过,因此没有留下信息素,此时若蚂蚁经过此路径并留下少量信息素,则该信息素会被迅速蒸发掉,或者说信息素残留率很低,即  $\delta(t) \rightarrow 0$ 。另一方面,当路径上的蚂蚁逐渐增多时,留下的信息素也逐渐增多,但是由于蒸发因素,最终蚂蚁留下的信息素量和蒸发掉的信息素量会达到平衡,从而使得留在路径上的信息素保持不变。从(10)式可以看出,这时候应该使得  $\delta(t) \rightarrow 1$ 。于是  $\delta(t)$  的取值范围应该在 0 和 1 之间。

由于蚂蚁留下的信息素和蒸发行为最终会达成平衡,根据(10)式,并利用信息素渐近平衡原理可知:时间  $t$  越大,  $\delta(t)$  就越应该趋近于 1,这时可以保证信息素  $\tau(i, j, t)$  基本维持不变而达到平衡。于是  $\delta(t)$  应该是一个值域在 0 与 1 之间的严格单调增函数。

为了对  $\delta(t)$  进行建模,对  $\delta(t)$  进行 Taylor 展开,得到

$$\delta(t + \Delta t) = \delta(t) + \delta'(t)\Delta t + \frac{1}{2!}\delta''(t)\Delta t^2 + \dots \quad (14)$$

假设  $\Delta t$  很小,此时,忽略(14)式的高阶无穷小量,得到

$$\delta(t + \Delta t) \approx \delta(t) + \delta'(t)\Delta t \quad (15)$$

现在考虑  $\delta(t) \rightarrow 0$  的时间  $t$ , 假设  $\Delta t$  很小, 根据 (15) 式有  $\delta(t + \Delta t) \approx \delta'(t)\Delta t$ , 或者近似地有  $\delta(t) \approx \delta'(t)\Delta t$ , 这意味着  $d\delta(t)/dt \approx \delta(t)/\Delta t$ , 或者说,  $d\delta(t)/dt$  与  $\delta(t)$  成正比。

另一方面, 考虑  $\delta(t) \rightarrow 1$  的时间  $t$ , 假设  $\Delta t$  很小, 根据 (15) 式有  $\delta(t + \Delta t) \approx 1 + \delta'(t)\Delta t$ , 或者近似地有  $\delta(t) - 1 \approx \delta'(t)\Delta t$ , 这意味着  $d\delta(t)/dt \approx (\delta(t) - 1)/\Delta t$ , 或者说,  $d\delta(t)/dt$  与  $\delta(t) - 1$  成正比。

可见,  $d\delta(t)/dt$  同时与  $\delta(t)$  和  $\delta(t) - 1$  成正比, 考虑到  $\delta(t)$  是时间  $t$  的严格单调增函数, 就可以建立以下的微分方程

$$\frac{d\delta}{dt} = \lambda\delta(1 - \delta) \quad (16)$$

上式中的  $\lambda > 0$ , 是一个常数。微分方程 (16) 的解为

$$\delta(t) = \frac{1}{1 + Ce^{-\lambda t}} \quad (17)$$

其中  $C$  是一个常数, 由初始条件确定。如果假设  $\delta(0) = \delta_0$ , 则有

$$\delta(t) = \frac{\gamma_0 e^{\lambda t}}{1 + \gamma_0 e^{\lambda t}} \quad (18)$$

这里  $\gamma_0 = \delta_0/(1 - \delta_0)$ 。 $\delta_0$  称为初始信息素残留率。由于  $\delta_0 \in (0, 1)$ , 于是  $\gamma_0 > 0$ 。对于 (18) 式, 下面的条件恒成立:  $0 < \delta(t) < 1$ ,  $d\delta(t)/dt > 0$ ,  $\inf \delta(t) = 0$ ,  $\sup \delta(t) = 1$ 。至此, 构造了一个严格单调增的信息素残留率函数  $\delta(t)$ 。

将 (18) 式写成离散时间序列的形式就是

$$\delta(t_m) = \frac{\gamma_0 e^{\lambda t_m}}{1 + \gamma_0 e^{\lambda t_m}}, \gamma_0 > 0, \lambda > 0 \quad (19)$$

### 1.2.3 信息素增量函数

根据 (12) 式,  $\mu(t)$  称为信息素增量函数。 $\mu(t)$  的引入是 IGACO 算法中实现正反馈机制的一个关键。在信息素更新阶段, 信息素更新标志函数值为 1 的蚂蚁会更新其路径上的信息素。但是, 信息素的更新必须满足信息素渐近平衡原理。

将 (12) 式代入 (10) 式得到

$$\begin{aligned} \tau(i, j, t_{m+1}) &= \delta(t_m)(\tau(i, j, t_m) + \mu(i, j, t_m)) \quad (20) \\ \tau(i, j, t_{m+1}) - \tau(i, j, t_m) &= \delta(t_m)(\tau(i, j, t_m) + \mu(i, j, t_m)) - \tau(i, j, t_m) \end{aligned} \quad (21)$$

在一些迭代周期, 如果某一路径  $(i, j)$  始终是最优路径, 则其信息素  $\tau(i, j, t_m)$  将会随着  $t_m$  而递增, 即满足

$$\tau(i, j, t_{m+1}) - \tau(i, j, t_m) > 0 \quad (22)$$

根据 (21) 式、(22) 式有

$$(\delta(t_m) - 1)\tau(i, j, t_m) + \delta(t_m)\mu(i, j, t_m) > 0 \quad (23)$$

解得

$$\mu(i, j, t_m) > \frac{(1 - \delta(t_m))\tau(i, j, t_m)}{\delta(t_m)} \quad (24)$$

令

$$\mu(i, j, t_m) = \frac{A(1 - \delta(t_m))\tau(i, j, t_m)}{\delta(t_m)}, A > 1 \quad (25)$$

其中  $A$  是大于 1 的常数。将 (25) 式代入 (20) 式得到信息素增量函数  $\mu(t)$  的表达式为

$$\begin{aligned} \tau(i, j, t_{m+1}) &= \delta(t_m)(\tau(i, j, t_m) + \mu(i, j, t_m)) \\ &= (1 + \frac{A(1 - \delta(t_m))}{\delta(t_m)})\delta(t_m)\tau(i, j, t_m) \\ &= (1 + (A - 1)\nu(t_m))(1 + (A - 1)\nu(t_{m-1}))\tau(i, j, t_{m-1}) \end{aligned}$$

$$= \cdots = \tau(i, j, t_0) \prod_{i=0}^m (1 + (A - 1)\nu(t_i)) \quad (26)$$

如果将时间  $t \rightarrow \infty$  时的信息素  $\tau(i, j, t_\infty)$  称为渐近信息素并记为  $\tau_\infty$ , 而将  $\tau(i, j, t_0)$  称为初始信息素并记为  $\tau_0$ , 则有

$$\begin{aligned} \tau_\infty &= \tau(i, j, t_\infty) = \lim_{m \rightarrow \infty} \tau(i, j, t_{m+1}) \\ &= \tau_0 \cdot \lim_{m \rightarrow \infty} \prod_{i=0}^m (1 + (A - 1)\nu(t_i)) \end{aligned} \quad (27)$$

由于  $\delta(t_m) \in (0, 1)$  是严格单调增函数, 于是  $\nu(t_m) = 1 - \delta(t_m)$  是严格单调减函数, 并且有  $0 < \nu(t_m) < 1$ 。根据定理 1, 对于一个充分大的迭代次数  $m$ , IGACO 算法在  $m$  次迭代内至少找到一次最优解。在找到最优解之后, 最优路径上的信息素  $\tau(i, j, t_m)$  将会随着  $t_m$  而递增, 于是问题就归结为 (27) 式的无穷乘积的收敛性问题。

利用 (19) 式, (27) 式收敛的充分必要条件就使下面的级数收敛:

$$\sum_{i=0}^{\infty} (A - 1)\nu(t_i) = (A - 1) \sum_{i=0}^{\infty} \frac{1}{1 + \gamma_0 e^{\lambda t_i}} \quad (28)$$

由于  $\gamma_0 = \delta_0/(1 - \delta_0) > 0$  和  $\lambda > 0$ , (28) 式收敛, 因此 (27) 式也收敛。这意味着随着  $t_m \rightarrow \infty$ , 最优路径上的渐近信息素将会最终达到一个平衡值  $0 < \tau_\infty = \tau_0$

$\cdot \lim_{m \rightarrow \infty} \prod_{i=0}^m (1 + (A - 1)\nu(t_i)) < \infty$ , 信息素有上界, 记为  $\tau_{\sup}$ 。另一方面, 由于  $0 < \delta(t_m) < 1$ , 根据 (10) 式知道, 当  $t_m \rightarrow \infty$  时, 非最优路径上的渐近信息素  $\tau_\infty$  最终将会趋近于 0, 信息素有下界, 记为  $\tau_{\inf}$ 。因此, 所有路径上的信息素将都会最终达到平衡值。于是有:

定理 2: 如果信息素残留率函数和信息素增量函数分别如 (19) 和 (25) 式所示, 则信息素更新规则 (10) ~ (12) 满足信息素渐近平衡原理, 并且渐近信息素  $\tau_\infty$ 。

在最优路径上将会趋于一个正数,而在非最优路径上将会趋于0。

### 1.3 算法的实现

为了将算法具体实现,将概率选择函数(见(4)式)写成以下具体的表达式:

$$P(i \in r(t), j \in d(t)) = \frac{F(\tau(i, j, t))}{\sum_{i \in r(t), k \in d(t)} F(\tau(i, k, t))} \quad (29)$$

根据定理2,最优路径上的信息素将会大于非最优路径上的信息素,因此选择(29)式中的函数 $F$ 为信息素 $\tau$ 的严格单调增连续函数,且 $F>0$ ,这样就可以保证当前最优路径上的概率选择函数值大于非最优路径上的概率选择函数值。这是一种正反馈原理,它加强了当前最优路径被蚂蚁选中的概率。

由于信息素 $\tau$ 有上界 $\tau_{sup}$ 和下界 $\tau_{inf}$ ,于是在闭区间 $[\tau_{inf}, \tau_{sup}]$ 上,(29)式中的严格单调增连续函数 $F$ 有界,且 $F>0$ ,满足定理1的充分条件,因此按照(29)式给出的概率选择函数,算法是收敛的。

如果当前解仍然是部分可行解,蚂蚁就循环执行寻找下一个结点的过程:在第 $m$ 次迭代中,根据概率选择函数,蚂蚁随机地从当前结点 $i$ 选择下一个结点 $j$ 。蚂蚁构造解过程的伪代码如下:

Procedure IGACO\_Solution\_Construction

While (the next vertex belongs to the components of the problem) do;

at iteration  $m$ , for each source node  $I$ , select the next node  $j$  randomly following (29);

End Procedure

## 2 数值实验与分析

### 2.1 实验目的、平台及数据

为了便于和传统方法进行比较,采用TSP问题作为仿真实验对象,对它们进行了测试。实验的目的是对比IGACO算法和传统蚁群优化算法的收敛效果。

实验中的操作系统为Windows XP,编程环境为Matlab 7.0,TSP数据来自文献[11]。

### 2.2 实验方法

首先要具体给出概率选择函数如下:

$$P(i, j, t) = \frac{\eta^\beta(i, j) \cdot \tau^\alpha(i, j, t)}{\sum_{k \in D_i} \eta^\beta(i, k) \cdot \tau^\alpha(i, k, t)}, (i, j) \in D_i \quad (30)$$

其中 $\tau$ 是信息素, $\eta$ 是一个参数,在TSP问题中,它是城市 $i$ 与城市 $j$ 之间距离的倒数。 $\eta$ 的引入使得距离短的城市之间的信息素更大。 $\alpha$ 和 $\beta$ 是两个参数。 $\eta > 0, \tau > 0, \alpha > 0$ 。

### 2.2.1 相关参数的设置

对于IGACO算法,初始化信息素为 $\tau_0 = 10$ , (30)式中的 $\alpha = 1, \beta = 5$ , (18)式中的 $\gamma_0 = 5, \lambda = 0.01$ ,信息素增量函数(25)式中的 $A = 1.01$ 。

对于传统蚁群优化算法,初始化信息素为 $\tau_0 = 10$ , (30)式中的 $\alpha = 1, \beta = 5$ ,信息素挥发参数(常数<sup>[7,8,12]</sup>) $\rho = 0.65$ 。经过多次实验验证,上面的参数能够使传统蚁群优化算法达到良好效果。

### 2.2.2 实验结果与分析

实验是对IGACO算法和传统蚁群算法分别执行20次并进行统计后得到的平均收敛速度。图1~图3中横坐标表示迭代周期数,纵坐标表示所求得的城市之间距离之和,实线表示IGACO算法的收敛过程,虚线表示传统蚁群优化算法的收敛过程。可见,不论是从收敛速度,还是从最终收敛到的最优解,IGACO算法都要优于传统蚁群优化算法。

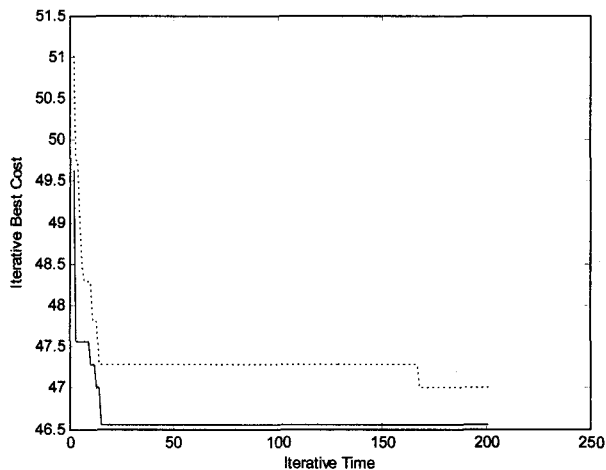


图1 10个城市收敛过程图

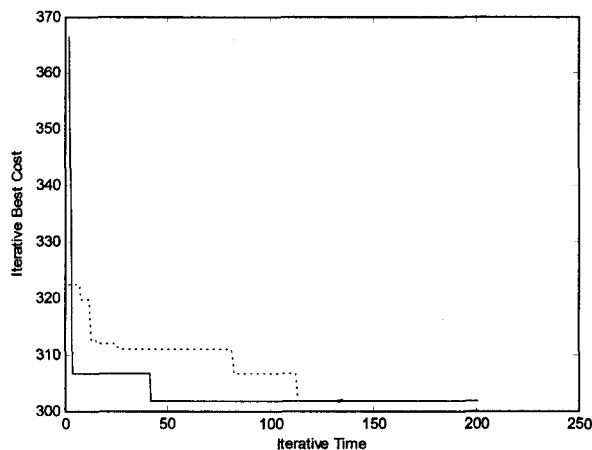


图2 30个城市收敛过程图

## 3 结束语

提出的IGACO算法放宽了概率选择函数的条件,给出了蚂蚁在某一源结点选择下一个结点的一般表达

式并证明了 IGACO 算法的收敛性。

对信息素残留率函数进行了建模,提出了新的信息素增量函数,证明了渐近信息素在最优路径上将会趋于一个正数,而在非最优路径上将会趋于 0。

提出了信息素渐近平衡原理。指出文中提出的 IGACO 算法满足信息素渐近平衡原理。

最后,计算机仿真实验说明文中提出的 IGACO 算法优于传统的蚁群优化算法。

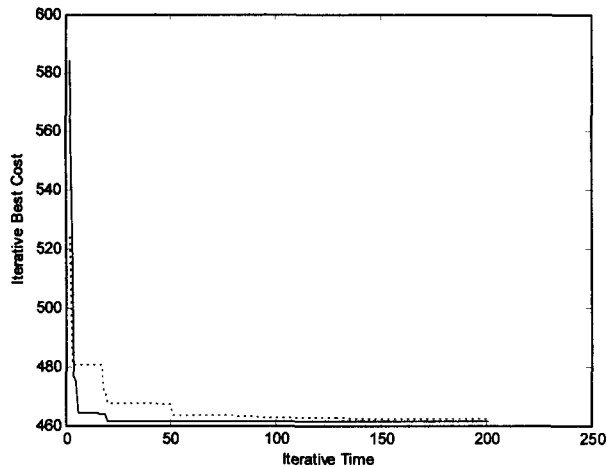


图 3 50 个城市收敛过程图

#### 参考文献:

- [1] Dorigo M, Bonabeau E, Theraulaz G. Ant algorithms and stigmergy[J]. Future Gener. Comput. Syst., 2000, 6(8): 851-871.
- [2] Dorigo M, Di Caro G D. The ant colony optimization meta-

heuristic[C]//New Ideas in Optimization. London: McGraw-Hill, 1999: 11-32.

- [3] Dorigo M, Di Caro G, Gambardella L M. Ant algorithms for discrete optimization[J]. Artif. Life, 1999, 5(2): 137-172.
- [4] Meuleau N, Dorigo M. Ant colony optimization and stochastic gradient descent[J]. Artif. Life, 2002, 8(2): 103-121.
- [5] Badr A, Fahmy A. A proof of convergence for ant algorithms[J]. International Journal of Intelligent Computing and Information, 2003, 3(1): 22-33.
- [6] Gutjahr W J. A graph-based ant system and its convergence[J]. Future Gener. Comput. Syst., 2000, 16(8): 873-888.
- [7] Stützle T, Dorigo M. A Short Convergence Proof for a Class of Ant Colony Optimization Algorithms[J]. IEEE Trans. on Evolutionary Computation, 2002, 6(4): 358-365.
- [8] Dorigo M, Stützle T. 蚁群优化[M]. 张军, 胡晓敏, 罗旭耀, 等译. 北京: 清华大学出版社, 2007.
- [9] Zhang D. Convergence Analysis for Generalized Ant Colony Optimization Algorithm[C]//Proceedings of the 11th Joint Conference on Information Sciences. [s. l.]: Atlantis Press, 2008.
- [10] Zhang D, Liu Y. Generalized Ant Colony Optimization Algorithm and Its Applications on P2P Searching[C]//2011 International Conference on Information Science and Engineering (ICISE). Yangzhou: IEEE Press, 2011: 1013-1016.
- [11] TSPLib[DB/OL]. 2008. <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/>.
- [12] Drigo M, Gambardella L M. Ant colony system; a cooperation learning approach to the traveling salesman problem[J]. IEEE Trans. on Evolution Comput., 1997, 1(4): 53-66.

(上接第 38 页)

诊断知识共享系统进行了初步应用。但是本框架还不是很成熟,比如尚不支持按功能服务划分服务器、不支持缓存集群等缺点,这将在后续的版本中得到扩展和改进。

#### 参考文献:

- [1] 郝彬, 陈翔鹰. 利用框架技术构建 Web 应用[J]. 计算机工程与设计, 2007, 28(1): 8-13.
- [2] 张宇, 王映辉, 张翔南. 基于 Spring 的 MVC 框架设计与实现[J]. 计算机工程, 2010, 36(4): 59-62.
- [3] 魏学松, 张育平. IOC 框架的研究与设计[J]. 计算机技术与发展, 2006, 16(3): 213-216.
- [4] Gao Songfeng, Liu Hongguang. Management information system of RIA constructed based on open source frame[C]//IE&EM 09 16th International Conference. [s. l.]: [s. n.], 2009: 651-653.
- [5] 陈洪磊. 面向 RIA 的 Web 应用程序框架研究[D]. 西安:

西北工业大学, 2007.

- [6] 钱钰, 陈志云. 基于 Flex 的 RIA 技术在教学软件中的应用[J]. 计算机与数字工程, 2009, 37(5): 160-163.
- [7] 石永革, 许建林, 石峰. 富客户端技术应用研究与实现[J]. 计算机工程与设计, 2008(3): 639-641.
- [8] 符培炯, 杜忠军. Spring 在实现 MVC 构架中的应用[J]. 计算机技术与发展, 2006, 16(6): 236-238.
- [9] Li Fan, Pei Cao. Summary cache: a scalable wide-area web cache sharing protocol[J]. IEEE/ACM Trans. on Netw., 2000, 8(3): 281-293.
- [10] Fielding R T. Architectural Styles and the Design of Network-based Software Architectures[D]. Berkeley: University of California, 2000.
- [11] 刘宁, 陆荣国, 缪万胜. MVC 体系架构从模式到框架的持续抽象进化[J]. 计算机工程, 2008, 34(4): 107-110.
- [12] 古全友, 王恩波, 胥昌胜. AOP 技术在 J2EE 系统构建中的应用[J]. 计算机技术与发展, 2006, 16(4): 150-152.