基于拥塞控制的多种群二元蚁群算法

严彬，熊伟清，程美英，叶青

1.宁波大学计算机科学与技术研究所 宁波 315211

摘　要: 二元蚁群算法在函数优化中有着良好的表现, 但仍存在易陷入局部最优和在多峰函数求解中无法同时得到多个解的缺陷。本文使用拥塞控制策略改善算法的全局寻优能力，同时引入多种群的思想，提出了基于拥塞控制多种群二元蚁群算法。通过对几个不同函数(包括单峰与多峰) 的测试, 实验结果表明该改进算法在保证较好的全局搜索能力的基础上，拥有很好的多目标求解能力。

关键词: 二元蚁群,拥塞控制策略,多种群,多峰函数

中图分类号：TP301.6　　**文献标识码** A

Multi-population Binary ant Colony Algorithm with Congestion Control Strategy

YAN Bin , XIONG Wei-qing , CHENG Mei-ying , Ye-qing

1. Institute of Computer Science and Technology, Ningbo University, Ningbo 315211

Abstract: Binary ant colony algorithm has good performance in the function optimization problem. However, the drawbacks that easy to fall into the local optimization and cannot get all the solutions at the same time still exist. In this paper, through introducing the congestion control strategy to improve the algorithm's globe optimization ability and the thought of multi-population, a novel binary ant colony algorithm based on congestion control strategy and multi-population is proposed. The tests of several different functions optimization (including single-modal and multi-modal function) prove that the improved algorithm not only ensure the good globe search ability, but also has better effect to the multi-objective problem.

Key Words: Binary ant Colony Algorithm, Congestion Control Strategy, Multi-population, Multi-modal Function

1　引言(Introduction)

蚁群优化算法是20世纪90年代才提出的一种新型的模拟进化算法，它是由意大利学者 M. Dorigo等人首先提出的，由于其在离散优化问题方面的一些优越性，以该算法求解TSP问题、分配问题、job-shop调度问题，取得了很好的结果[1,2]。在连续域上，如函数优化问题，蚁群算法也得到了广泛的应用，但是其缺点和局限性也很明显。作为改进，在文献[3]中，用二进制编码方式与蚁群算法相结合，提出的二元蚁群算法，并成功应用于离散优化问题和连续优化问题，如多维0/1背包问题及函数优化问题[4,5]。实验表明，与标准蚁群算法相比，二元蚁群算法有效率高，求解速度快的优点。然而二元蚁群算法虽然能够有效地求解单峰函数，但在处理多峰函数优化问题时，很难同时找到多个极值点。而事实上，无论是理论研究中还是实际的工程问题中，有很多问题在本质上属于多目标问题,因此有必要提高该算法求解多目标问题的能力。

基金项目：国家自然科学基金(60472099)；浙江省自然科学基金（Y106080）；宁波市自然科学基金项目（2007A610051）

本文引入多种群协同进化的思想，在算法迭代的过程中将单种群动态地发展为多种群，各个种群通过相互间的竞争以及外部环境的选择两种方式进行优胜劣汰的操作，被保留的种群通过协作和有限通信，并行地对问题进行求解；同时在各个种群内部加入拥塞控制策略，提高种群自身的搜索能力；从而达到多目标求解的目的。通过对几个不同的函数(包括多峰与单峰) 的测试, 结果证明了改进算法的有效。

2　基于拥塞控制策略的二元蚁群算法(Binary ant Colony Algorithm with Congestion Control Strategy)

二元蚁群算法[3]求解函数问题已获得了一定的进展,但是该算法也存在易陷入局部解的缺点。这是因为蚁群算法都存在“利用”与“探索”的冲突：即信息素的更新必须首先保证算法能够收敛，同时要保证算法收敛到全局最优。这对于信息素更新是个两难困境：不同路径的信息素落差大，保证算法的收敛和搜索的方向（这就是所谓的“利用”，即利用原有的信息）；同时信息素的落差又不能太过悬殊而使得算法无法探索新解（这就是所谓的“探索”，即开发新解）。

显式限定信息素上下界的Max-Min方法[6]是一个能兼顾两者的均衡策略，且在应用中取得了很好的效果。不过其缺点是对路径流量的控制不够灵活，不能即时地对蚂蚁进行分流，而且在一定程度上会放慢解的收敛。因此在使用Max-Min方法的基础上，本文引入了蚁群的拥塞控制。该策略的灵感来自于文献[7]中Audrey Dussutouretal.(2004)在对蚁群在高度拥挤的情况下的交通组织行为的研究中的发现。实验采用了一种黑园蚁，蚂蚁在巢和食物源之间的一个有两个对称分支的桥上移动，当桥的宽度改变时，蚂蚁的行为将发生变化：当宽度为10mm时，多数蚂蚁仅使用一条桥分支(不对称的交通)，而当宽度<=6mm时，将形成对称的交通状态(图1)。这说明蚁群在低密度的情况下,基于信息素吸引的机制只生成一条信息素轨迹，但当蚁群高度拥挤时，另一条信息素轨迹将在交通流量受到影响之前生成。

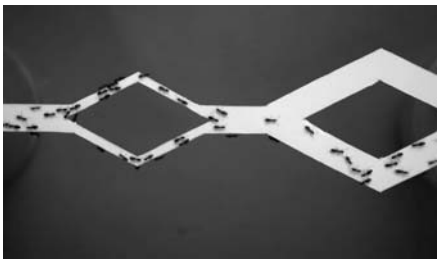

图 1: Dussutour的菱型桥实验
Fig.1 Dussutour's Diamond bridge experiment

把此模型转化为一个二元离散优化问题。在使用简化后的二进制网络[4]中(图2)，首先对于结点 v_j 引入流量和流量阈值 $N_{flow}(v_j,r)$，$r\in\{0,1\}$。$N_{flow}(v_j,r)$ 在每代开始时置0，在每个蚂蚁的解的构建中如果 v_j 被置为 r 则流量 $N_{flow}(v_j,r)$ 加1。

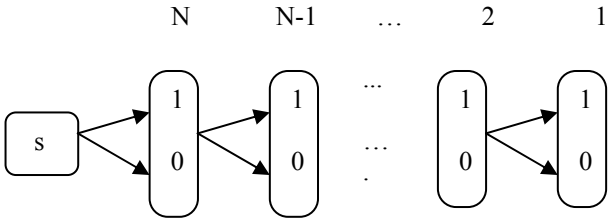

图 2:简化后的二进制网络

Fig.2 Simplified binary network

设θ为流量比例，流量阈值定义如下：

$$N_{th}(v_j,r)=Max(\frac{\theta}{\theta+1}\sum_{k\in\{0,1\}}N_{flow}(v_j,k),\theta+1)$$

$$r\in\{0,1\} \qquad (1)$$

由于二元蚁群算法中没用设置启发函数，因此信息启发因子α可以设为1，设 T 为最大迭代数，$c_1<c_2\in(0,1)$ 为迭代参数，结点 v_j 的状态选择如（2）式：

$$P(v_j=r\,|\,N_{flow},N_{th})=\begin{cases}\tau^{\alpha}_{i(r)}(t)\Big/\sum_{k\in\{0,1\}}\tau^{\alpha}_{i(k)}(t), if\ N_{flow}(v_j,r)<N_{th}(v_j,r)\\ \qquad\qquad\qquad\qquad \&\ t\ge c_2\cdot T\\ 0 \qquad\qquad\qquad\qquad ,otherwise\end{cases}$$

$$r\in\{0,1\} \qquad (2)$$

设Q为预置常数，为信息素初始值 τ_0 的两倍，$\tau_0>0$；s^{gbest} 表示至今最优解，s^{cbest} 表示本代最优解；$s^{gbest}_{(i)}$ 表示至今最优解在结点i的取值，$s^{cbest}_{(i)}$ 表示本代最优解在结点i的取值；ρ 为信息素挥发系数，$\rho\in(0,1)$ 为一非常小的数。在 $c_1\cdot T<t<c_2\cdot T$ 时，信息素的更新采用本代最优更新策略，而在算法初始（$t\le c_1\cdot T_1$）和末期($t\ge c_2\cdot T_1$)则采用至今最优更新策略：

$$\tau_{i(r)}(t+1)=(1-\rho)*\tau_{i(r)}(t)+\rho\Delta\tau \quad r\in\{0,1\} \qquad (3)$$

$$\Delta\tau=\begin{cases}\Delta\tau_1\ ,\ if\ \ c_1\cdot T<t<c_2\cdot T\\ \Delta\tau_2\ \ ,\ \text{else}\end{cases} \qquad (4)$$

其中 $\Delta\tau_1$ 为采用本代最优更新策略时的信息素增量：

$$\Delta\tau_1=\begin{cases}Q\ ,\ if\ \ s^{cbest}_{(i)}=r\\ 0\ \ ,\ \text{else}\end{cases}\quad r\in\{0,1\} \qquad (5)$$

而 $\Delta\tau_2$ 为采用至今最优更新策略时的信息素增量：

$$\Delta\tau_2=\begin{cases}Q\ ,\ if\ \ s^{gbest}_{(i)}=r\\ 0\ \ ,\ \text{else}\end{cases}\quad r\in\{0,1\} \qquad (6)$$

信息素的上下限通过式(7)限定：

$$\tau_{i(r)}=\begin{cases}\tau_{\max},\ if\ \ \tau_{i(r)}>\tau_{\max}\\ \tau_{\min},\ if\ \ \tau_{i(r)}<\tau_{\min}\end{cases}\quad r\in\{0,1\} \qquad (7)$$

接下来分析拥塞控制的作用和效果。当 $t<c_2\cdot T$ 时，拥塞控制起作用。假设流量比例θ为大于1的整数，则可以将算法每搜索θ+1次看作执行了一个拥塞控制周期。设在 $t'<c_2\cdot T$ 时刻，对于某个结点 v_j，算法执行了拥塞控制，则在 $[t'+1,t'+1+\theta]$ 这一拥塞控制周期内，拥塞控制至多在 $t'+1+\theta$ 时刻执行一次操作。假设此周期内该结点信息素选择概率 $P_\tau(v_j=r)=p$ 保持不变。设事件A为 v_j 在前θ次搜索中都选择了状态 r，事件B为一次都没有选择，事件C为包括除A、B外的所有情况，则根据全概率公式，可得在此周期的第θ+1步，算法将 $v_j=r$ 构建为解的一部分的概率为：

$$P(v_j = r \mid t = t'+1+\theta) = P_A P(v_j = r \mid P_A) + P_B P(v_j = r \mid P_B) + P_C P(v_j = r \mid P_C)$$
$$= p^{\theta} \cdot 0 + (1-p)^{\theta} \cdot 1 + (1 - p^{\theta} - (1-p)^{\theta}) \cdot p$$
$$= p + (1-p)^{\theta+1} - p^{\theta+1} \quad (8)$$

在此拥塞控制周期中的平均概率 $P(v_j = r \mid \theta)$ 可由式(8)及全概率公式得：

$$P(v_j = r \mid \theta) = \frac{\theta}{\theta+1} p + \frac{1}{\theta+1} P(v_j = r \mid t = t'+1+\theta)$$
$$= p + \frac{(1-p)^{\theta+1} - p^{\theta+1}}{\theta+1} \quad (9)$$

在 $t < c_2 \cdot T$ 时，式(9)的结论可推广到算法的每一次搜索。因此在引入拥塞控制后，每个结点 v_j 的实际选择概率由式(9)所决定。

从式(9)可知，在 $P_\tau(v_j = r) = p \in (0,1)$ 区间上，$P(v_j = r \mid \theta)$ 单调递增，$P(v_j = r \mid \theta) \in (\frac{1}{\theta+1}, \frac{\theta}{\theta+1})$，θ>1。另设有 $\theta_1 < \theta_2$，则有：

$$P(v_j = r \mid \theta_1) > P(v_j = r \mid \theta_2) > p, \quad p \in (0, \frac{1}{2}) \quad (10)$$

$$P(v_j = r \mid \theta_1) < P(v_j = r \mid \theta_2) < p, \quad p \in (\frac{1}{2}, 1) \quad (11)$$

θ值越大，$P(v_j = r \mid \theta)$ 曲线越逼近 $P(v_j = r) = p$；θ值越小，$P(v_j = r \mid \theta)$ 曲线越逼近直线 $P(v_j = r)$=0.5。

拥塞控制的引入，使得算法在拥塞控制起作用的任何时刻，无论信息素如何分布，能保留一定的探索新解的能力，这就保证了算法能够跳出局部最优。与不使用拥塞控制的原始二元蚁群算法相比，信息素上下限的比例及信息素更新的幅度，可以加以放大，这么做使得算法在后期单独使用信息素搜索时，解的收敛速度和求解质量得以提高。

3 基于拥塞控制策略的二元蚁群算法性能分析 (Performance analysis Binary ant Colony Algorithm with Congestion Control Strategy)

单种群的二元蚁群算法在多峰函数求解问题上，遇到的困难从本质上说，是其二元网络模型这一构造方式所不可避免的。用单种群的二元蚁群算法求解某个问题时，假设在一定的进化代数后，整个网络趋于收敛，因为每个节点只有2个状态可供选择，因此这意味着大多数节点的2个状态上的信息素分别达到τ_{min}或τ_{max}，这样解的搜索方向已被确定，这里的搜索方向指的是算法给出解的最可能出现的一个范围。假设 $\tau_{max} / \tau_{min} \to \infty$，解就被完全确定了。如果问题只有或者只要求一个最优解，那么这个解的信息可以轻易地从已收敛的二元网络中提取出来，但是如果问题有不止一个的最优解，会出现什么问题呢？

首先考虑 $ACO_{gb,\tau_{min}}$ 算法模型[8,9]的一些已有结论。

定理1[8] 设$P^{gbest}(t)$为t次迭代内算法首次发现最优解S^{gbest}的概率，则对于任意小的ε>0和充分大的迭代次数t，有：

$$P^{gbest}(t) \geq 1 - \varepsilon \quad (12)$$

$$\lim_{t \to \infty} P^{gbest} = 1 \quad (13)$$

定理1说明了算法是值收敛的。

定理2[8] 设t*为首次发现最优解S^{gbest}时的迭代次数，对于 $\forall (i,j) \in s^{gbest}, \forall (k,l) \in L \wedge (k,l) \notin s^{gbest}$，存在一个 t^0，$\forall t > t^* + t^0 = t^*[(1-\rho)/\rho]$，使得：

$$\tau_{i,j}(t) > \tau_{k,l}(t) \quad (14)$$

定理2表明算法在一定代数后能以优于平均的概率重现最优解。

定理3[8] 一旦搜索到最优路径，$\forall (i,j) \in s^{gbest}$，$\forall (k,l) \notin s^{gbest}$，有

$$\lim_{t \to \infty} \tau_{i,j}(t) = \tau_{max} = \frac{1}{\rho} g(s^{gbest}) \quad (15)$$

$$\lim_{t \to \infty} \tau_{k,l}(t) = \tau_{min} \quad (16)$$

由证明过程可知，定理1普遍适用于 $ACO_{\tau_{min}}$，即只要求算法满足以下条件：

$$0 < \tau_{min} < \tau_{max} < \infty \wedge \rho > 0 \wedge \Delta\tau < \infty \quad (17)$$

显然最常用的MMAS[6]和ACS[10]都属于 $ACO_{\tau_{min}}$，而本文提出的基于拥塞控制策略的二元蚁群算法（BACOCC）满足式(16)，因此也属于 $ACO_{\tau_{min}}$，定理1对BACOCC成立。

定理2和定理3则还要求算法执行至今最优解更新策略。BACOCC中，$t \geq c_2 \cdot T$ 时采用至今最优解更新策略，因此当 T 足够大时，定理3对BACOCC成立，将定理2中 $\forall t > t^* + t^0$ 改为 $\forall t > \max(t^*, c_2 \cdot T) + t^0$ 后也成立。

引理1 合理设置BACOCC 中的 τ_{max} 和 τ_{min}，

$$\forall t \geq 0, \sum_{k \in \{0,1\}} \tau_{i(k)}(t) = \sum_{k \in \{0,1\}} \tau_{j(k)}(t) = 2\tau_0, i \neq j.$$

证明：设信息素初始值为 τ_0，且 $\tau_{max} + \tau_{min} = 2\tau_0$，则

$$\sum_{k \in \{0,1\}} \tau_{i(k)}(0) = 2\tau_0 \quad (18)$$

假设 $\sum_{k\in\{0,1\}}\tau_{i(k)}(t)=2\tau_0$，由式(3)~式(6)及$\Delta\tau=2\tau_0$可得：

$$\sum_{k\in\{0,1\}}\tau_{i(k)}(t+1)=\sum_{k\in\{0,1\}}\tau_{i(k)}(t)+\rho\Delta\tau-\rho\tau_{i(0)}(t)$$

$$-\rho\tau_{i(1)}(t)=\sum_{k\in\{0,1\}}\tau_{i(k)}(t)+\rho(\Delta\tau-\sum_{k\in\{0,1\}}\tau_{i(k)}(t))$$

$$=\sum_{k\in\{0,1\}}\tau_{i(k)}(t)+\rho(2\tau_0-2\tau_0)=2\tau_0 \tag{19}$$

另若 $\min(\tau_{i(0)}(t+1),\tau_{i(1)}(t+1))<\tau_{\min}$，因 $\sum_{k\in\{0,1\}}\tau_{i(k)}(t)$ $=2\tau_0=\tau_{\max}+\tau_{\min}$，必有 $\max(\tau_{i(0)}(t),\tau_{i(1)}(t))>\tau_{\max}$，经过式(7)修整后，依然有：

$$\sum_{k\in\{0,1\}}\tau_{i(k)}(t+1)=\tau_{\max}+\tau_{\min}=2\tau_0 \tag{20}$$

同理可得 $\max(\tau_{i(0)}(t+1),\tau_{i(1)}(t+1))>\tau_{\max}$ 时，式(20)依然成立。由数学归纳法可得 $\sum_{k\in\{0,1\}}\tau_{i(k)}(t)=2\tau_0$ 成立。

显然 $\sum_{k\in\{0,1\}}\tau_{i(k)}(t)=\sum_{k\in\{0,1\}}\tau_{j(k)}(t)=2\tau_0, i\neq j$，证毕

定理4 $\forall t>\max(t^*,c_2\cdot T)+t^0$，$\exists x>0$，

若 $f_d(\mathrm{s},\mathrm{s}^{gbest})\geq x$，BACOCC下一搜索求得解s的概率将小于平均概率，其中 $f_d(\mathrm{s},\mathrm{s}^{gbest})$ 为两个解的海明距离。

证明：设 $(i,r)\in s^{gbest}$，$(i,k)\notin s^{gbest}$，解序列长度为n，则根据定理2

当 $t>\max(t^*,c_2\cdot T)+t^0$ 时，有：

$$\tau_{i(r)}(t)>\tau_{i(k)}(t) \tag{21}$$

另设 a_1^t，b_1^t，a_2^t，b_2^t 如下：

$$a_1^t=\min(\tau_{0(r)}(t),\tau_{1(r)}(t),...,\tau_{n-1(r)}(t))$$
$$b_1^t=\max(\tau_{0(k)}(t),\tau_{1(k)}(t),...,\tau_{n-1(k)}(t))$$
$$a_2^t=\max(\tau_{0(r)}(t),\tau_{1(r)}(t),...,\tau_{n-1(r)}(t))$$
$$b_2^t=\min(\tau_{0(k)}(t),\tau_{1(k)}(t),...,\tau_{n-1(k)}(t))$$

由引理1及式(21)可得：

$$a_1^t>\tau_0>b_1^t,\quad a_2^t>\tau_0>b_2^t \tag{22}$$

$$a_1^t+b_1^t=a_2^t+b_2^t=2\tau_0 \tag{23}$$

设 $\mathrm{P_s}(t)$ 为t代时个体下次搜索找到解s的概率，则有：

$$\mathrm{P_s}(t)=\prod_{i=0}^{n-1}\frac{\tau_{i(r)}(t)}{\sum_{k\in\{0,1\}}\tau_{i(k)}(t)}\leq\frac{b_1^t}{a_1^t+b_1^t}^x\ \frac{a_2^t}{a_2^t+b_2^t}^{n-x} \tag{24}$$

求解不等式：

$$\frac{b_1^t}{a_1^t+b_1^t}^x\ \frac{a_2^t}{a_2^t+b_2^t}^{n-x}=\frac{b_1^t}{2\tau_0}^x\ \frac{a_2^t}{2\tau_0}^{n-x}<\frac{1}{2}^n$$

$$\frac{b_1^t}{a_2^t}^x<\frac{\tau_0}{a_2^t}^n=\frac{a_2^t+b_2^t}{2a_2^t}^n\leq\frac{a_2^t+b_1^t}{2a_2^t}^n$$

对两边取以 b_1^t/a_2^t 为底的对数，得：

$$x>n\log_{\frac{b_1^t}{a_2^t}}^{\frac{\tau_0}{a_2^t}} \tag{25}$$

$$n\log_{\frac{b_1^t}{a_2^t}}^{\frac{\tau_0}{a_2^t}}\geq n\log_{\frac{b_1^t}{a_2^t}}^{\frac{1}{2}+\frac{1}{2}\frac{b_1^t}{a_2^t}} \tag{26}$$

当 $t>\max(t^*,c_2\cdot T)+t^0$，解s与至今最优解 s^{gbest} 的海明距离满足式(22) 时，算法下一搜索找到解s的概率将小于平均概率，证毕。

引理2 函数 $f(k_t)=n\log_{k_t}^{\frac{1}{2}+\frac{1}{2}k_t}$，在区间 $k_t\in(0,1)$ 上单调递增。设 $k_t=b_1^t/a_2^t$，很容易证明当 $t>\max(t^*,c_2\cdot T)+t^0$ 时，k_t 为t的单调非增函数，且 $\lim_{t\to\infty}k_t=\tau_{\min}/\tau_{\max}$，则此时 $f(k_t)$ 为t的单调非增函数。

引理3 当 $t>\max(t^*,c_2\cdot T)+t^0$ 时，设 $f_{b,a}(t)=(b_1^t-b_2^t)/a_2^t$，由定义可知 $f_{b,a}(t)\geq 0$，$\lim_{t\to\infty}f_{b,a}(t)=(\tau_{\min}-\tau_{\min})/\tau_{\max}=0$，是t的非增函数。

推论1 设 $x^\vee$ 为 x 取值的下界，由式(25)有：

$$x^\vee=f_x(t)=n\log_{\frac{b_1^t}{a_2^t}}^{\frac{\tau_0}{a_2^t}}=n\log_{\frac{b_1^t}{a_2^t}}^{\frac{1}{2}+\frac{1}{2}\frac{b_1^t}{a_2^t}-\frac{1}{2}\frac{(b_1^t-b_2^t)}{a_2^t}}$$

$$=n\log_{\frac{b_1^t}{a_2^t}}^{\frac{1}{2}+\frac{1}{2}\frac{b_1^t}{a_2^t}-\frac{1}{2}f_{b,a}(t)} \tag{27}$$

由引理2和引理3可知当 $t>\max(t^*,c_2\cdot T)+t^0$ 时，$n\log_{\frac{b_1^t}{a_2^t}}^{\frac{1}{2}+\frac{1}{2}\frac{b_1^t}{a_2^t}}$ 是t的非增函数，$f_{b,a}(t)$ 是t的非增函数，因此可得此时 $\log_{\frac{b_1^t}{a_2^t}}^{\frac{1}{2}+\frac{1}{2}\frac{b_1^t}{a_2^t}-\frac{1}{2}\frac{f_b(t)}{a_2^t}}=n\log_{\frac{b_1^t}{a_2^t}}^{\frac{\tau_0}{a_2^t}}$ 为t的非增函数。

$x^\vee=f_x(t)$ 随着t的增大而减小，这意味着下次搜索能够以高于或等于平均概率求得的解的范围越来越小，越来越接近 s^{gbest}。

引理4 $\lim_{t\to\infty} n\log_{\frac{b_1^t}{a_2^t}}^{\frac{\tau_0}{a_2^t}} = n\log_{\frac{\tau_{\min}}{\tau_{\max}}}^{\frac{\tau_0}{\tau_{\max}}}$，设 $f_p(x,t)=\left(\frac{b_1^t}{a_1^t+b_1^t}\right)^x\left(\frac{a_2^t}{a_2^t+b_2^t}\right)^{n-x}$，由推论1可知，$\forall x > n\log_{\frac{\tau_{\min}}{\tau_{\max}}}^{\frac{\tau_0}{\tau_{\max}}}$ $\exists t^x > \max(t^*, c_2\cdot T)+t^0$，使得 $f_p(x,t^x) < \frac{1}{2}^n$ 成立。

推论2 设解s'与至今最优解s^{gbest}的海明距离为x'，$x' > n\log_{\frac{\tau_{\min}}{\tau_{\max}}}^{\frac{\tau_0}{\tau_{\max}}}$，由引理4得，$\exists t_1 > \max(t^*, c_2\cdot T)+t^0$，使得 $f_p(x',t_1)=\lambda\frac{1}{2}^n$，$0<\lambda\le 1$，即：

$$\left(\frac{b_1^{t_1}}{a_1^{t_1}+b_1^{t_1}}\right)^{x'}\left(\frac{a_2^{t_1}}{a_1^{t_1}+b_1^{t_1}}\right)^{n-x'} = \lambda\frac{1}{2}^n \tag{28}$$

解得 $x' = \log_{\frac{b_1^{t_1}}{a_2^{t_1}}}^{\lambda} + n\log_{\frac{b_1^{t_1}}{a_2^{t_1}}}^{\frac{\tau_0}{a_2^{t_1}}}$。

$\forall t_2 > t_1$，求解满足式(28)的x''

$$\left(\frac{b_1^{t_2}}{a_1^{t_2}+b_1^{t_2}}\right)^{x''}\left(\frac{a_2^{t_2}}{a_1^{t_2}+b_1^{t_2}}\right)^{n-x''} = \lambda\frac{1}{2}^n \tag{29}$$

解得 $x'' = \log_{\frac{b_1^{t_2}}{a_2^{t_2}}}^{\lambda} + n\log_{\frac{b_1^{t_2}}{a_2^{t_2}}}^{\frac{\tau_0}{a_2^{t_2}}}$。

由$\log_{\frac{b_1^t}{a_2^t}}^{\lambda}$为t的递减函数，$n\log_{\frac{b_1^t}{a_2^t}}^{\frac{\tau_0}{a_2^t}}$为t的非增函数，可得$x'' < x'$。

由式(28)及(29)可得：

$$\left(\frac{b_1^{t_1}}{a_1^{t_1}+b_1^{t_1}}\right)^{x'}\left(\frac{a_2^{t_1}}{a_1^{t_1}+b_1^{t_1}}\right)^{n-x'} = \left(\frac{b_1^{t_2}}{a_1^{t_2}+b_1^{t_2}}\right)^{x''}\left(\frac{a_2^{t_2}}{a_1^{t_2}+b_1^{t_2}}\right)^{n-x''} > \left(\frac{b_1^{t_2}}{a_1^{t_2}+b_1^{t_2}}\right)^{x'}\left(\frac{a_2^{t_2}}{a_1^{t_2}+b_1^{t_2}}\right)^{n-x'} \tag{30}$$

即 $f_p(x',t_1) > f_p(x',t_2)$。

由定义可知$\lim_{t\to\infty} P_{s'}(t) = f_p(x',t)$，且由式(24)可知$P_{s'}(t) \le f_p(x',t)$，这意味着与至今最优解$s^{gbest}$的海明距离为$x'$的解s'，随着时间t的增大，被下次搜索找到的几率将越来越小。

以上结论证明了BACOCC是值收敛的，算法在运行足够长的时间后能够以任意接近1的概率找到问题的一个最优解。但是如果问题的最优解集S_{best}中的元素大于1，则在算法找到其中一个最优解s^{gbest}后，对于属于子集：

$$\{s^* | s^* \in S_{best} \wedge s^* \ne s^{gbest} \wedge f_d(s^*, s^{gbest}) > n\log_{\frac{\tau_{\min}}{\tau_{\max}}}^{\frac{\tau_0}{\tau_{\max}}}\}$$

中的最优解s^*，在时刻t''($f_p(f_d(s^*, s^{gbest}), t'') < \frac{1}{2}^n$)之后，被下次搜索找到的概率将小于平均概率，且随着时间t的增大，这个概率将越来越小。因此BACOCC每次运行只能保证求解到问题的一个最优值，而很难求得全部最优解。

4 多种群蚁群算法(Multi-population Binary ant Colony Algorithm)

4.1 种群的动态生灭策略 (Dynamic creation and annihilation Strategy for Population)

由第3节的证明可知，基于拥塞控制的二元蚁群算法(BACOCC)处理多目标问题时，使用多个种群是必须的。种群的数量显然应该与问题相关。一般来说，问题有多少个最优解，常规的想法就是设置相应数量的种群，但遗憾的是，在大多数实际问题中，事先是不知道解的数量和分布的。那么种群的设置就需要通过一些策略来合理地实现。

使用隔离小生境技术[11], 将算法的初始群体分为几个子群体，子群体之间独立进化,隔离后的子群体彼此独立，可以视作有多个单种群二元蚁群算法在同时运行。如同在生物界中，竞争不仅存在于个体之间，种群作为整体同样存在着竞争，适者生存的法则在种群这一层次上同样适用。

子群体历史越短，在历史上产生的解越好，其被保留的概率就越高。在群体进化过程中，如果某一子群体在规定的代数内，持续表现很差，应该使这个子群体灭绝，释放资源给新生种群以搜索空间的新解。子群体在进化过程中,如果出现两个子群体相似或相同的现象，则去掉其中的一个，避免无意义地重复搜索，这种策略称为同种互斥(intraspecific competition)或种内竞争。

为了保持群体的多样性，有时需要有意识地保护某些子群体，使之不会过早地被淘汰。种群中新出现的子群体，在进化的初期往往无法同已经得到进化的其它子群体相竞争，如果不对此施加保护，这些新群体往往在进化的初期就被淘汰掉,这显然是算法所不希望的。为了解决这个问题，必须对新产生的群体加以保护。这种保护新群体的策略被称为幼弱保护(immature protection)。

子群体在进化过程中，如果收敛到或接近局部最优解,会出现进化停滞的现象，此时，应当以某种概率将该子群体去掉，释放资源给新生种群，此种策略称为新老更替(the new superseding the old)。

在进化过程中，将各群体的最优解以表现好坏进行排序，拥有表现排入前k位（k的大小一般为先验的最优解数或期望得到解的数量）的几个个体的子群体进化出最优解的概率最大，应当使它们充分进化，故新老更替策略不应当用于拥有这些个体的子群体，这种做法称为优种保留(the best live)。

根据以上思想，设计出多种群动态生灭的基于拥塞控制的多种群二元蚁群算法（MPBACOCC）。

4.2 多种群蚁群算法的基本概念 (Conception of Multi-population ant Colony Algorithm)

定义1 母群 算法初始阶段建立的由N_0个蚂蚁个体组成的群体,其个体采用随机搜索策略。该群体存在于整个算法周期中，为优秀解信息库提供其搜索到的解。

定义2 优秀解信息库 由一系列存储单元和方法函数组成。信息库接收来自各个种群提供的解，进行库存解资格判断后，将符合要求的解存入其中。信息库中的解的个数容量上限设为N_{max}（$N_{max} \geq N_{exp}$，N_{exp}为期望得到的解的个数）。

定义3 种子 当优秀信息库中的解被保留超过T_a代后，即被标记为种子。种子用于生成并初始化新的子种群。

定义4 库存解资格判断 信息库中的解将作为种子，基于小生境隔离的思想，各种子应保持一定的多样性。因此解被信息库保留的条件为，在该解的解空间上的范围η_0内，信息库内没有比之更优的解存在。

定义5 子种群 子种群由种子生成，由N_1个蚂蚁个体组成。其个体采用式(2)所确定的搜索策略，并将所得解提供给优秀解信息库。种群的信息素更新策略由式(3)和(4)决定。子种群的最大生命周期（迭代数）为T_1，另有销毁阈值T_2。

定义6 子种群销毁 当子种群存在代数超过生命周期T_1，或在销毁阈值T_2内没有搜索到比其种子更优的解，即实行对该子种群的销毁程序。另外所有历史超过T_2代的将参与竞争，以一定概率P_{delete}销毁表现较差的子群体。

4.3 **多种群蚁群算法的实现** (Procedure of Multi-population ant Colony Algorithm)

多种群蚁群算法的实现可由下面的伪代码表示:

```
procedure MPACO
  InitializeOriginalPopulation A0;//生成初始种群A0
  InitializeInfStack ; //生成优秀解信息库
  while (termination condition not met) do
    OriginalPopulationSearch A0;
     //母群A0执行随机搜索
    InfStackCreateSeed;//在优秀解信息库中标记种子
    while (Sub-PopulationAmount<Nmax and
          SeedAmount >0) do
      InitializeSub-Population; //生成子种群
      Sub-PopulationAmount++;//更新子种群数
      Seed Amount--;//更新种子数
    end-while
    while (any Sub-Population hasn't worked) do
      Sub-PopulationSearch Ai; //子种群Ai执行搜索
    end-while
    while (any Sub-Population hasn't checked) do
      if (termination condition of Sub-Population Ai
          met ) then
        DeleteSub-Population Ai;
        //销毁符合销毁条件的子种群Ai
      end-if
    end-while
  end-while
  OutputInfStack;//输出优秀解信息库中的解
end-procedure
```

母群搜索过程的伪代码:

```
procedure OriginalPopulationSearch
 for i= 1 to N0 do
    s^i ←randomsearch;//随机搜索产生解s^i
    if (qualification of s^i was accredited) then
      add s^i into InitializeInfStack;
      //将s^i存入优秀解信息库
    end-if
 end-for
end-procedure
```

子种群搜索过程的伪代码:

```
procedure Sub-PopulationSearch
 for i= 1 to N1 do
    s^i ← PheromonesearchwithCongestionControl;
    //利用式(1)及(2)所描述的搜索策略生成解s^i
    if (qualification of s^i was accredited) then
      add s^i into InitializeInfStack;
    //将s^i存入优秀解信息库
    end-if
    if f(s^i)< f(s^currentbest) then
      s^currentbest ← s^i;//更新本代最优解s^currentbest
    end-if
  end-for
  if f(s^currentbest) < f(s^gbest) then
    s^gbest ← s^currentbest;//更新至今最优解s^gbest
  end-if
  foreach (i,r)∈L do
    τi(r) ←(1-ρ)τi(r) ;//信息素全局挥发
    τi(r) ←max {τi(r) , τmin};//限定信息素下限
  end- foreach
  if c1T1<t<c2T1 do
    foreach (i,r)∈ s^currentbest do
      τi(r) ←τi(r) +ρΔτ;//本代最优解释放信息素
      τi(r) ←min {τi(r) , τmax};//限定信息素上限
    end- foreach
  else
    foreach (i,r)∈ s^gbest do
```

$\tau_{i(r)} \leftarrow \tau_{i(r)} + \rho\Delta\tau$;//至今最优解释放信息素

$\tau_{i(r)} \leftarrow \min\{\tau_{i(r)}, \tau_{max}\}$;//限定信息素上限

end- foreach

end-if

end-procedure

5 实验与结果（Experiment and results）

以下实验测试函数均来自文献[11]。

5.1测试函数 Rosenbrock (Rosenbrock Function)

$F = 100 \cdot (x_1^2 + x_2)^2 + (1 - x_1)^2$，其中 $x_i \in (-2.048, 2.048)$，$i \in \{0,1\}$

在测试基于拥塞控制的多种群二元蚁群算法(MPBACOCC)的多目标求解能力之前，有必要对BACOCC的基本计算能力进行测试。Rosenbrock函数具有一个全局也是唯一的极小点f(1, 1) =0。虽然在求极小值时它是单峰值的函数, 但它却是病态的, 难以进行全局极小化。

本实验中采用BACOCC与BACO的进行对比。其中BACOCC采用单种群，蚂蚁个体数N_1=200，最大迭代数T=300，迭代参数c_1 =0.2，c_2=0.8，ρ=0.01，流量比例θ=3；BACO中蚂蚁数N_1=200，最大迭代数T=300，其余参数与文献[3]中设置相同。未知数x_i的二进制编码长度为30。实验结果见表1。

表 1. Rosenbrock 函数对比实验结果

Table 1. Results of Rosenbrock Function's opposite experiment

50次实验的对比结果	BACOCC		
	x1	x2	f(x1,x2)
最好结果	1.000000	1.000000	8.76035E-17
最差结果	0.999999	0.999998	1.60153E-12
搜索到最优解的次数	50		
50次实验的对比结果	BACO		
	x1	x2	f(x1,x2)
最好结果	0.999998	0.999997	4.45801E-10
最差结果	0.997678	0.995222	7.32673E-06
搜索到最优解的次数	50		

实验证明，BACOCC有优秀的全局最优求解能力。

5.2测试函数Six- hump Camel Back Function (Six-hump Camel Back Function)

$F = (4 - 2.1 \cdot x_1^2 + x_1^4/3) \cdot x_1^2 + x_1 \cdot x_2 + (-4 + 4 \cdot x_2^2) \cdot x_2^2$，其中 $x_1 \in [-3,3], x_2 \in [-2,2]$。

该函数在定义域内有两个全局最小点,(x1, x2) =(-0. 0898, 0. 7126)，(0. 0898, -0. 7126)，全局最小值为-1. 0316。

本试验中采用MPBACOCC与BACO的进行对比。其中MPBACOCC设置母群个体数N_0=400，子种群个体数N_1=200，子种群最大生命周期T_1=300，子种群销毁阈值T_2=60，迭代参数c_1 =0.2，c_2=0.8，欧氏距离η_0=2，种子阈值T_a=5，ρ=0. 01，流量比例θ=3，信息库容量上限N_{max}=5，期望解数N_{exp}=2，MPBACOCC的结束条件为子群体全部销毁，销毁概率P_{delete}=0.25。BACO中蚂蚁数N_1=200，最大迭代数T=300，其余参数与文献[3]中设置相同。未知数x_i的二进制编码长度为30。实验结果见表2和表3。

实验证明，MPBACOCC可以同时搜索到多峰函数的多个最优解,而BACO则很难保证每次都能搜索到多个最优解。

表 2. Six- hump Camel Back Function 函数对比实验结果

Table 2. Results of Six- hump Camel Back Function's opposite experiment

50次实验的对比结果	MPBACOCC	BACO
搜索到全部最优解的次数	50	9
仅搜索到此最优解的次数 f(-0.0898,0.7126)=-1.0316	0	17
仅搜索到此最优解的次数 f(0.0898,-0.7126)=-1.0316	0	24

表 3. MPBACOCC 50 次实验其中一次实验结果

Table 3 Results of Six- hump Camel Back Function in MPBACOCC

x1	x2	x1的30位编码长度的二进制形式	x2的30位编码长度的二进制形式	函数的实验值
-0.0898352	0.712656	011111000010101100010101100	101011011001110000010010101011	-1. 03163E+00
0.0898385	-0.712664	100000111101010100010100011	010100100110001110110101101	-1. 03163E+00

5.3测试函数Shubert (Shubert Function)

$$F = \sum_{i=i}^{5} i \cdot \cos[(i+1) \cdot x_1 + i] \cdot \sum_{i=i}^{5} i \cdot \cos[(i+1) \cdot x_2 + i],$$

其中 $x_i \in [-10,10]$，$i \in \{0,1\}$。

该测试函数有760 个局部最小解，其中18 个是全局最小，其值为- 186.73。

本试验中采用MPBACOCC与BACO的进行对比。其中MPBACOCC设置母群个体数N_0=400，子种群个体数N_1=200，子种群最大生命周期T_1=300，子种

群销毁阈值T_2=60，迭代参数c_1 =0.2，c_2=0.8，欧氏距离η_0=0.5，种子阈值T_a=5，ρ=0.01，流量比例θ=3，信息库容量上限N_{max}=20，期望解数N_{exp}=18，MPBACOCC的结束条件为子群体全部销毁，销毁概率P_{delete}=0.25。BACO中蚂蚁数N_1=200，最大迭代数T=300，其余参数与文献[3]中设置相同。未知数x_i的二进制编码长度为30。实验结果见表4和表5。

实验证明，MPBACOCC具有BACO所没有的多目标问题求解能力。

表 4. Shubert 函数对比实验结果

Table 4. Results of Shubert Function's opposite experiment

50次实验的对比结果	MPBACOCC	BACO
搜索到全部18个最优解的次数	49	0
搜索到不同最优解的最大个数	18	7
搜索到不同最优解的最小个数	17	3

表 5. MPBACOCC 50 次实验其中一次实验结果

Table 5. Results of Shubert Function in MPBACOCC

x1	x2	x1的30位编码长度的二进制形式	x2的30位编码长度的二进制形式	函数实验值	理论值
5.48287	-1.42513	110001100010111001000000100101	011011011100000100100100110100	-186.731	-186.73
-1.42513	-0.80032	011011011100000100100001000010	011101011100000110000100000110	-186.731	-186.73
-7.08351	4.85807	001001010101010011000101000100	001001010101010011000101000100	-186.731	-186.73
-1.42511	5.48286	011011011100000100110001111101	110001100010111001101010000101	-186.731	-186.73
5.48285	-7.7082 9	110001100010111001100100001011	000111010101010111000101101111	-186.731	-186.73
-7.7083 4	5.48286	000111010101010100111110101011	110001100010111001101100001	-186.731	-186.73
-0.8003	4.85802	011101011100000110010110110	101111100010111011011101110	-186.731	-186.73
-0.8003 5	-7.7082 7	011101011100000101101111110	000111010101010110001011100110	-186.731	-186.73
-7.7083 7	-7.08343	000111010101010100111010100101	001001010101010000010000000	-186.731	-186.73
-7.08342	-1.42508	001001010101010100001111001011	011011011100000100101010101011	-186.731	-186.73
-7.0836 2	-7.7082 7	001001010101010001101100110	000111010101010110001011000 1001010010	-186.731	-186.73
5.48267	4.85823	110001100010111001001110101110	101111100010111011100111 0110101011	-186.731	-186.73
4.8584	-0.80028 9	101111100011000000000100001110	011101011100000011001 1101000110	-186.731	-186.73
4.85813	5.48341	101111100010111001000001000000	110001100011000000000 0110111101	-186.73	-186.73
4.85806	-7.08351	101111100010111011100000001000	001001010101001100 0000101101	-186.731	-186.73
-7.7083 3	-0.80031 7	000111010101010101010111001100	011101011100000011000 0101110100	-186.731	-186.73
-0.8000 7	-1.42585	011101011100001001011100000011	011011011011111110010011111	-186.73	-186.73
-1.4258	-7.08359	011011011011111111100010011011	001001010101001000 0000000100	-186.73	-186.73

6 小结(Summary)

本文将拥塞策略引入二元蚁群算法，提高了算法的搜索质量，也扩大了搜索范围，进一步地抑制了算法的“早熟”现象。同时证明了单种群的二元蚁群算法不具有多目标问题处理能力。在此基础上，提出了多种群动态生灭的二元蚁群算法。该算法继承了原算法简洁高效的优点，同时有效克服了二元蚁群固有的单目标求解局限性，拥有很强的多目标问题处理能力。实验结果有力地证明了，在求解单峰函数和多峰函数的问题上，该算法都有很好的表现。

参考文献(References)

[1] M. Dorigo, Caro GD.Ant colony optimization: A new meta-heuristic[C].In: Proc of the 1999 Congress on Evolutionary Computation,Washington,DC, USA: IEEE Press, 1999: 1470～1477.

[2] Merkle D,Middendorf M, Schmeck H.Ant colony optimization for resource-constrained project scheduling. IEEE Trans. Evolutionary Comput.2002:6(4)333-46

[3] Weiqing Xiong, Chenyang Yan, Liuyi Wang. Binary Ant Colony Evolutionary Algorithm[C].In: International Conference on Intelligent Computing,HeFei, China, 2005: 1341～1350.

[4] 熊伟清 魏平,二进制蚁群进化算法[J],自动化学报,2007; 33(3):259-264

[5] 熊伟清,魏平,赵杰煜,信号传递的二元蚁群算法[J],模式识别与人工智能,2007;20(1):15-20

[6] T. G. Stützle and H.H. Hoos. Improvements on the Ant System: Introducing the MAX－MIN Ant System. In R.F. Albrecht G.D. Smith, N.C. Steele, editor, Artificial Neural Networks and Genetic Algorithms, 1998:245-249

[7] A. Dussutour, V. Fourcassié et.al., Optimal traffic organization in ants under crowded conditions[J].Nature 428, 70-73 ,2004.

[8] T. G. Stützle and M. Dorigo, short convergence proof for a class of Ant Colony Optimization algorithms. IEEE Transactions on Evolutionary Computation, 2002, 6(4):358-365.

[9] N. Meuleau and M. Dorigo. Ant Colony Optimization and Stochastic Gradient Descent . Artificial Life,8(2),2002

[10] M. Dorigo and Gambardella L.M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation,1997:1(1):53－66

[11] 米凯利维茨,演化程序—遗传算法和数据编码的结合[M]. 北京: 科学出版社, 2000

附录

作者简介：

严彬（1981-），男，硕士研究生，研究方向为进化计算；
熊伟清（1966-），男，教授，研究方向为进化计算，人工智能；
程美英，女，硕士研究生，研究方向为智能计算；
叶青，男，硕士研究生，研究方向为进化计算；