

# 多态蚁群算法

徐精明<sup>1</sup>, 曹先彬<sup>2</sup>, 王煦法<sup>2</sup>

(1. 安徽技术师范学院, 安徽蚌埠 233100; 2. 中国科学技术大学计算机科学技术系, 安徽合肥 230026)

**摘要:**在分析现有蚁群算法不足的基础上, 提出一种新的含多种蚁群、多种信息激素的多态蚁群算法。该算法通过引入不同种类的蚁群, 每一蚁群有不同的信息素调控机制, 将局部搜索与全局搜索相结合, 使搜索、收敛速度大幅度提高。针对 TSP 问题的仿真实验结果表明了该算法的有效性。

**关键词:**蚁群算法; 多态蚁群算法; TSP

**中图分类号:** TP18      **文献标识码:** A

## 0 引言

蚁群算法(ant colony algorithm)是一种基于种群的模拟进化、用于解决复杂优化问题的全新的启发式算法。它是在对自然界中真实蚁群的集体行为(蚂蚁依赖信息素进行通信而显示出的社会性行为)研究的基础上, 于二十世纪九十年代由意大利学者 Dorigo 等<sup>[1,2]</sup>首先提出。此后, 不断有学者对其完善, 提出了许多改进算法。如结合 Q-Learning 提出的 Ant-Q 算法<sup>[3]</sup>, MMAS 算法<sup>[4]</sup>等。但是, 不论 Dorigo 提出的基本蚁群算法, 还是后来学者提出的改进蚁群算法, 都是基于单种蚁群、单种信息素的算法, 主要模拟了实际蚁群信息系统的一部分。而实际上, 真实蚁群社会中的蚁群是有组织、有分工的<sup>[5]</sup>, 不同种类的蚁群有不同的信息素调控机制, 这种分工组织方式对蚁群完成复杂任务具有十分重要的地位。

我们认为, 由于蚁群算法的理论基础还很不完善, 因此对蚁群算法的改进有必要尽量忠实于蚁群的真实信息处理机制。以此为出发点, 提出了一种改进的蚁群算法-多态蚁群算法(polymorphic ant colony algorithm)。该算法更符合蚁群的真实信息处理机制, 使搜索、收敛速度大幅度提高。

## 1 基本蚁群算法模型

蚁群算法通常用于求解复杂的组合优化问题。在对不同性质的问题求解时, 蚁群算法模型的定义也有所不同。我们以平面上  $m$  个城市的 TSP<sup>[6]</sup>问题为例说明基本蚁群算法模型。 $m$  个城市的 TSP 问题就是寻找通过  $m$  个城市各一次且最后回到出发点的最短路径。

设  $n$  是蚁群中蚂蚁的数量,  $d_{ij}$  ( $i, j=1, 2, \dots, m$ ) 表示城市  $i$  和城市  $j$  之间的距离,  $\tau_{ij}(t)$

表示  $t$  时刻在城市  $i$  与城市  $j$  连线上信息素的浓度. 初始时刻, 各条路径上信息素的浓度相同, 设  $\tau_{ij}(0) = C$  ( $C$  为常数). 蚂蚁  $k$  ( $k=1, 2, \dots, n$ ) 在运动过程中, 根据各条路径上的信息素的浓度决定转移方向,  $P_{ij}^k(t)$  表示在  $t$  时刻蚂蚁  $k$  从城市  $i$  转移到城市  $j$  的概率, 其计算公式如(1)所示.

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{s \notin \text{tabu}_k} \tau_{ij}^\alpha(t) \eta_{is}^\beta(t)}, & j \notin \text{tabu}_k \\ 0, & j \in \text{tabu}_k \end{cases} \quad (1)$$

其中,  $\text{tabu}_k$  ( $k=1, 2, \dots, n$ ) 为蚂蚁  $k$  已走过城市的集合. 开始时  $\text{tabu}_k$  中只有一个元素, 即蚂蚁  $k$  的出发城市, 随着进化的进行,  $\text{tabu}_k$  中的元素不断增加. 随着时间的推移, 以前留在各条路径上的信息素逐渐消逝, 用参数  $(1-\rho)$  表示信息素的挥发程度, 所有蚂蚁完成一次循环, 各路径上信息素的浓度要根据式(2)作调整,

$$\left. \begin{aligned} \tau_{ij}(t+1) &= \rho * \tau_{ij}(t) + (1-\rho) * \Delta\tau_{ij} \quad \rho \in (0, 1) \\ \Delta\tau_{ij} &= \sum_{k=1}^n \Delta\tau_{ij}^k \end{aligned} \right\} \quad (2)$$

$\Delta\tau_{ij}^k$  表示第  $k$  只蚂蚁在本次循环中留在路径  $ij$  上的信息素的浓度,  $\Delta\tau_{ij}$  表示本次循环所有蚂蚁在路径  $ij$  上所释放的信息素浓度之和.

Dorigo 曾给出 3 种不同模型, 分别称之为 ant cycle system, ant quantity system, ant density system, 它们的差别在于  $\Delta\tau_{ij}^k$  的计算表达式不同.

在 ant cycle system 模型中,

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k, & \text{若第 } k \text{ 只蚂蚁在时刻 } t \text{ 和 } t+1 \text{ 之间经过城市 } i, j \\ 0, & \text{否则} \end{cases} \quad (3)$$

其中  $L_k$  为第  $k$  只蚂蚁在本次循环中所走路径长度.

在 ant quantity system 和 ant density system 中,  $\Delta\tau_{ij}^k$  分别为:

$$\Delta\tau_{ij}^k = \begin{cases} Q/d_{ij}, & \text{若第 } k \text{ 只蚂蚁在时刻 } t \text{ 和 } t+1 \text{ 之间经过城市 } i, j \\ 0, & \text{否则} \end{cases} \quad (4)$$

$$\Delta\tau_{ij}^k = \begin{cases} Q, & \text{若第 } k \text{ 只蚂蚁在时刻 } t \text{ 和 } t+1 \text{ 之间经过城市 } i, j \\ 0, & \text{否则} \end{cases} \quad (5)$$

上述三种模型中, 后两者利用的是局部信息, 而前者利用的是整体信息. 所以, 一般都采用 ant quantity system 作为基本模型. 算法中参数  $\alpha, \beta, Q, \rho$ , 可以用实验方法确定其最优组合, 也可以用进化学习得到. 停止条件可以用固定进化代数或者当进化趋势不明显时便停止计算. 由算法复杂度分析理论可知, 该算法复杂度为  $O(nc \cdot m^3)$ , 其中  $nc$  表示循环次数,  $m$  表示规模(城市的个数).

## 2 多态蚁群算法设计

### 2.1 蚁群信息系统的多态性

蚂蚁是群居生活, 对蚁群社会的研究<sup>[5]</sup>表明, 真实蚁群社会中的蚁群是有组织、有分工的. 蚁群中有蚁后、侦察蚁、兵蚁和工蚁等, 它们各司其职, 又相互依赖协作, 形成一个有机整

体.在执行某项任务时,个体之间通过各自分泌的外激素相互联系.激素有报警类、吸引类、召集类、标记区域类等,不同激素代表不同的信息.虽然单个蚂蚁弱小,但多种蚂蚁群通过多种信息素通讯而组成的群体,就体现出高度组织性和社会性,使整个蚂蚁群体具有非凡的能力.这里的“多态性”指蚁群社会所具有的多种状态的蚁群及信息素.

## 2.2 现有蚁群算法的不足

蚁群算法本质上仍是一种随机搜索算法,它是通过对候选解组成的群体的进化来寻求最优解.算法由许多蚂蚁共同完成,每只蚂蚁在候选解的空间中独立地搜索解,并在所寻得的解经过的路径上留下一定的信息素,蚂蚁倾向于朝着该物质浓度高的方向移动.因此,由大量蚂蚁组成的蚁群的集体行为便表现出一种信息正反馈现象.随着算法的推进,较优解上的信息素将逐渐增多,算法渐渐趋于收敛.对于 TSP 这样的组合优化问题,其解是由一系列的子问题(城市到城市的路径)构成.从多态性角度分析,现有蚁群算法存在一些不足:

第一单种蚁群、单种信息素没有考虑蚂蚁群的种类差异和动作差异,使所有蚂蚁都做同一种任务,不能完全反映真实蚁群社会的复杂性,求解能力弱.如能在现有蚁群算法中,引入多种蚂蚁群,不同蚂蚁群的信息素调控不同,就能在蚂蚁搜索过程中,针对各具体路径选择合适的信息素的浓度,无疑会加快寻优收敛速度.

第二在现有蚁群算法中,第  $k$  只蚂蚁第  $x$  步由城市  $i$  选择下一个城市时,需按概率  $P_{ij}^k$  ( $j \in \text{tabu}_k$ ),在剩下的  $(m-x)$  个城市中选择一个,因此搜索子空间的规模过大.若能引入多种蚂蚁群,则每种蚂蚁群只需在自己的搜索空间中查找,这不仅大大缩减了搜索子空间的规模,而且选择的有效率(位于最优路径上)也较高,从而使搜索、收敛速度大有提高.同时也易于引入一些较好的局部搜索方法,如我们在后面的实验中借鉴了[7]的结论,只需在剩下的  $\min\{m-x, \text{MAXPC}\}$  个城市中选择一个.这里 MAXPC 为最大可选城市数,MAXPC 的确定如表 1,数据来源于[7],它是对不同城市规模的最优解所作统计,  $PC_i$  是以最优解中每一城市  $C_i$  为中心,作半径  $R$  的圆,  $R$  从零不断扩大,直至取得  $C_i$  的近邻城市为止,所记录下的此时圆内的城市数;MAXPC 为  $PC_i$  的最大值.

表 1 最大可选城市数统计结果

Tab. 1 Statistic results of maximal selectable cities

城市规模	MAXPC	$PC_i = 1$ 的城市数
20	8	11
100	9	55
144	13	61
1000	19	545

由表 1 可知,当城市规模  $\leq 100$  时,  $\text{MAXPC} < 10$ ; 当  $100 < \text{城市规模} \leq 1\,000$  时,  $\text{MAXPC} < 20$ ; 且此时  $PC_i = 1$  的概率基本上为 50% 以上,也就是说对于每一个城市  $C_i$ , 选择由  $C_i$  为起点的最短边的概率很大. 如果我们对于每个城市, 从离其最近的 MAXPC 个城市中选择一个作为近邻城市, 而不是从剩下城市中任选一个, 也必然可以找到最优解. 虽然我们事先并不知最优解的结果, 为不失一般性, 可作适当放大. 如当城市规模  $\leq 50$  时, 取  $\text{MAXPC} = 10$ ; 当  $50 < \text{城市规模} \leq 500$  时, 取  $\text{MAXPC} = 20$ ; 当  $500 < \text{城市规模} \leq 1000$  时, 取  $\text{MAXPC} = 30$ .

第三:在 ant cycle system 模型中,

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k, & \text{若第 } k \text{ 只蚂蚁在时刻 } t \text{ 和 } (t+1) \text{ 之间经过城市 } i, j \\ 0, & \text{否则} \end{cases}$$

蚂蚁在所有经过的路径  $ij$  上都留下了信息素,增加了该路径上信息素浓度.我们认为,路径上信息素浓度的增量计算,是本算法的关键(因蚂蚁是根据路径上信息素浓度来选择下一城市).但现有蚁群算法的浓度调控机制是唯一的,全局的.如果只在可能是最优解组成部分(根据 MAXPC 判断)的路径上,增加适当( $Q$  变化)的信息素浓度,将会使寻优收敛速度大有提高.这就需要使不同蚂蚁群具有不同的信息素调控机制.

据此分析,我们提出了基于多态蚁群系统的多态蚁群算法.

### 2.3 多态蚁群算法

我们仍然以 TSP 问题为例来进行说明,与基本蚁群算法相同之处略叙.在多态蚁群算法中,将蚁群社会中的蚂蚁分为三类:侦察蚁、搜索蚁和工蚁.侦察蚁群的任务是以每个城市为中心,作局域侦察,并以侦察素来标记侦察结果,以便搜索蚁到达该城市选择下一站时,提供辅助信息;搜索蚁群的任务是作全局搜索,每到一站,根据侦察素和各出口的信息素等信息,来选择下一站,直至找到并标记最佳(最短)路线,以便工蚁从最佳路线取食回巢;工蚁群的任务是从已标记好的最佳路线取食回巢.在实际算法设计时,由于工蚁群与路径寻优无关,所以只需针对侦察蚁群和搜索蚁群设计各自的信息素调控机制.其中,侦察蚁群负责局部侦察,搜索蚁群负责全局搜索.

针对侦察蚁群的做法是:将  $m$  个侦察蚁分别放置在  $m$  个城市中,每个侦察蚁以所在城市为中心,侦察其它  $(m-1)$  个城市,并将侦察结果(距离从小到大排序)与已有的先验知识(前文提到的 MAXPC)相结合,构成侦察素,记为  $s[i][j]$ ,标记在从城市  $i$  到城市  $j$  的路径上.  $s[i][j](i, j=0, 1, 2, \dots, m-1; i \neq j)$  的计算如下:

$$s[i][j] = \begin{cases} \tilde{d}_{ij}/d_{ij}, & \text{若城市 } j \text{ 在城市 } i \text{ 的 MAXPC 之内} \\ 0, & \text{否则} \end{cases} \quad (6)$$

其中,  $\tilde{d}_{ij}$  表示以城市  $i$  为中心,到其它  $(m-1)$  个城市的最小距离.并据此结果,首先置初始时刻各条路径上的信息量如下:

$$\tau_{ij}(0) = \begin{cases} C * s[i][j], & \text{若 } s[i][j] \neq 0 \\ C * \tilde{d}_{ij}/\hat{d}_{ij}, & \text{否则} \end{cases} \quad (7)$$

式中  $\hat{d}_{ij}$  表示以城市  $i$  为中心,到其它  $(m-1)$  个城市的最大距离,  $C$  为初始时刻,各条路径上信息素的浓度(常数).

如此针对不同情况,设置初始时刻各条路径上的信息量,较为合理,且对以后各代快速寻优起很重要作用.

其次,此侦察素为搜索蚁群在进行转移概率  $P_{ij}^k(t)$  的计算、各路径上信息素的浓度的调整提供辅助.

针对搜索蚁群的做法是:蚂蚁  $k(k=1, 2, \dots, n)$  在运动过程中的  $t$  时刻,从城市  $i$  转移到城市  $j$  的概率  $P_{ij}^k(t)$  的计算公式如下:

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^a(t) \eta_{ij}^\beta(t)}{\sum_{s \notin \text{tabu}_k} \tau_{ij}^a \eta_{ij}^\beta(t)}, & \text{若 } j \notin \text{tabu}_k, \text{ 且 } s[i][j] \neq 0 \\ 0, & \text{否则} \end{cases} \quad (8)$$

根据此式,搜索蚁每达一个城市,结合侦察素,只需在较小范围内搜索,大大减小了搜索规模.

所有蚂蚁完成一次循环,各路径上信息素的浓度要根据下式作调整,

$$\tau_{ij}(t+1) = \begin{cases} \rho * \tau_{ij}(t) + (1-\rho) * \Delta\tau_{ij}, & \text{若 } s[i][j] \neq 0 \\ \rho * \tau_{ij}(t), & \text{否则} \end{cases} \quad (9)$$

其中  $\Delta\tau_{ij}$  表示本次循环所有蚂蚁在路径  $ij$  上所释放的信息素浓度之和.  $\Delta\tau_{ij}^k$  表示第  $k$  只蚂蚁在本次循环中,在  $t$  到  $(t+1)$  之间留在路径  $ij$  上的信息素的浓度,  $\Delta\tau_{ij} = \sum_{k=1}^n \Delta\tau_{ij}^k$  的确定如下:

$$\Delta\tau_{ij}^k = \begin{cases} Q * (\tilde{d}_{ij}/d_{ij})/L_k, & \text{若蚂蚁 } k \text{ 经过 } ij, \text{ 且 } s[i][j] \neq 0 \\ 0, & \text{否则} \end{cases} \quad (10)$$

式中,每只搜索蚁根据侦察素,只在可能是最优解组成部分的路径上留下合适的信息素(局部信息:  $\tilde{d}_{ij}/d_{ij}$  与全局信息:  $L_k$  相结合);根据式(8),每只搜索蚁只在可能是最优解组成部分(按  $s[i][j]$  是否为零决定)的路径上追加信息素的浓度.

至此,多态蚁群算法的流程可以描述如下:

- 1) 初始化  $Q$ 、 $C$  和最大进化代数;
- 2) 将  $m$  只侦察蚁分别放在  $m$  个城市中,每个侦察蚁以所在城市  $i$  为中心,侦察其它  $(m-1)$  个城市,按式(6)计算侦察素,并将结果放入  $S[i][j]$  中;
- 3) 按公式(7)置初始时刻各条路径上的信息量;
- 4) 置进化代数  $NC$  初值为 0;
- 5) 随机选择每只搜索蚁的初始位置,并将该位置放入每个搜索蚁对应的  $\text{tabu}$  表中;
- 6) 按公式(8)计算每只搜索蚁  $k$  将要转移的位置,假设为  $j$ ,上一个位置假设为  $i$ ,并将  $j$  放入搜索蚁  $k$  对应的  $\text{tabu}$  表中,直至每只搜索蚁完成一个循环,得到一个解;
- 7) 计算各搜索蚁的目标函数值  $L_k (k=1, 2, \dots, n)$ ,并记录当前最好解;
- 8) 若达到指定的进化代数或者所求得解在最近若干代中无明显改进,转 11); 否则;
- 9) 按公式(9)修改各路径上信息素浓度;
- 10) 置  $\Delta\tau_{ij}$  为 0,置  $\text{tabu}$  表为空,  $NC \leftarrow NC+1$ , 转 5);
- 11) 输出最优解.

在上述算法中,引入了多种状态的蚁群及信息素,优化了初始时刻各条路径上信息素的浓度和每只蚂蚁在循环中留在路径上的信息素的浓度,缩小了搜索子空间的规模,与基本蚁群算法相比,搜索收敛速度、寻优效果和算法效率有较大提高.

## 2.4 多态蚁群算法复杂性分析

本算法的复杂性主要体现在侦察蚁群的侦察过程复杂性和搜索蚁群的搜索过程复杂性. 由于侦察过程是一次性的,与进化代数无关,该过程的复杂度为  $O(n \cdot m^2)$ ,其中  $n$  为侦察蚁数,  $m$  为城市数;对于搜索过程,它与进化代数  $NC$ 、最大可选城市数  $\text{MAXPC}$ 、搜索蚁数

$n$  及城市数  $m$  有关,该过程的复杂度为  $O(NC \cdot \text{MAXPC} \cdot n \cdot m)$ . 一般情况下  $NC \cdot \text{MAXPC}$  大于  $m$ , 所以本算法的总复杂度为  $O(NC \cdot \text{MAXPC} \cdot n \cdot m)$ .

### 3 实验结果与分析

我们选用 Oliver30 作为实验例子进行实验. 之所以选用这个例子, 是因为 TSP 问题是典型的 NP-hard 问题, 常常用来验证某一算法的有效性. 我们利用 PC 机 (p III), 在 Visual C++ 语言编程环境下, 分别用基本蚁群算法和多态蚁群算法进行了仿真实验. 实验中, 取  $Q=100, C=3, \alpha=1, \beta=3, \rho=0.3, \text{MAXPC}=10$ . 实验结果如表 2、图 1 所示. 表 2 中最短路径长度为最优路径的长度, 进化代数是取 10 次实验的平均而得到的. 图 1 是本文算法所找到的最优路径.

表 2 基本蚁群算法、多态蚁群算法的实验结果及比较

Tab. 2 Experiment results and their comparison between standard ant colony algorithms and poly morphic ant colony algorithms

最短路径长度	进化代数(多态算法)	进化代数(基本算法)	时间比(多态算法/基本算法)
424.635 376	27	52	1:3
424.464 270	41	126	1:4
423.911 743	116	362	1:4
423.740 601	137	>400	1:?

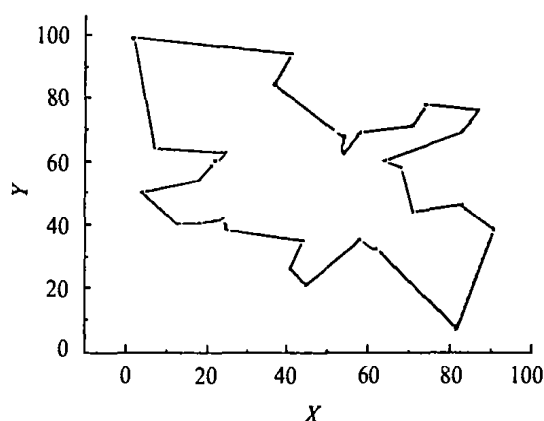


图 1 最优路径(423.740 601)线路

Fig. 1 Optimum approach routine found

从表 2 中可以看出, 用多态蚁群算法, 平均 137 代能得目前最好解 (423.740 601), 而用基本蚁群算法在 400 代内也得不到此解 (实际上我们实验时在 1 000 代内, 也未能搜索到过此解). 就次优解而言, 相同的结果, 基本蚁群算法所需代数比多态蚁群算法大得多, 这主要是因为多态蚁群算法中, 根据侦察蚁提供的侦察素, 预置了初始时刻各条路径上的信息量, 将局部信息和全局信息相结合, 来调整每只蚂蚁在循环中留在路径上的信息素的浓度.

从表中寻优时间比可以看出, 基本蚁群算法所花时间比多态蚁群算法大得多, 这主要是因为多态蚁群算法中, 搜索蚁每达一个城市, 根据侦察素, 只需在较小范围内搜索, 大大减小了搜索规模. 如以 400 个城市为例, 在基本蚁群算法中, 蚂蚁从一个城市选择下一个城市时, 平均需搜索 200 次; 而在本文算法中, 平均不超过 20 次.

另从仿真实验中可观察到, 多态蚁群算法中每代平均解值的下降率, 比基本蚁群算法的每代平均解值的下降率也要大.

我们还就其它多组参数值进行了仿真实验, 结果也大致如此.

## 4 结论

本文在分析现有蚁群算法的基础上,提出了一种更加忠实于真实蚁群行为的多态蚁群算法. 并对 TSP 问题(30 个城市)进行了仿真实验,从实验结果可以看出,多态蚁群算法比基本蚁群算法的寻优收敛速度有很大的提高,且比交换遗传算法<sup>[8]</sup>求解 TSP 问题的收敛速度度快. 如果对更大规模的 TSP 问题,本算法的优越性更加明显.

## 参 考 文 献

- [1] Dorigo M, Maniezzo V, Colormi 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.
- [2] 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.
- [3] Dorigo M, Gambardella L M, Middendorf M, Stutzle T. Guest editorial: special section on ant colony optimization [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(4): 317-319.
- [4] Stutzle T, Hoos H. MAX-MIN Ant System [J]. Future Generation Computer Systems, 2000, 16(8): 889-914.
- [5] 吴坚, 王常禄. 中国蚂蚁[M]. 北京: 中国林业出版社, 1995. 1-12.
- [6] 杜端甫. 运筹图论[M]. 北京: 北航出版社, 1990. 240-254.
- [7] 全惠云, 文高进. 求解 TSP 的子空间遗传算法[J]. 数学理论与应用, 2002, 22(1): 36-39.
- [8] 胡静, 陈思红, 王上飞, 等. 交换遗传算法收敛性及用户评估质量的提高[J]. 中国科学技术大学学报, 2002, 32(2): 210-216.

# Polymorphic Ant Colony Algorithm

XU Jing-ming<sup>1</sup>, CAO Xian-bin<sup>2</sup>, WANG Xu-fa<sup>2</sup>

(1. Anhui Technical Teachers College, Bangbu 233100, China)

(2. Dept. of Computer Science and Technology, USTC, Hefei 230026, China)

**Abstract:** Standard and improved Ant Colony Algorithms have so far been so far novel optimization algorithms based on single ant population and single pheromone. In this paper, a Polymorphic Ant Colony Algorithm is proposed considering of the diversity of ant populations and pheromones, which is more accordant with the ants' information processing mechanism. The algorithm combines the local search with the global search to improve its convergence and searching ability. The simulation results on TSP problem show the validity of this algorithm.

**Key words:** ant colony algorithm; polymorphic ant colony algorithm; TSP problem