

## 求解 TSP 的离散人工蜂群算法

于宏涛<sup>1</sup>, 高立群<sup>1</sup>, 田卫华<sup>2</sup>

(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110819; 2. 沈阳工程学院 自动化学院, 辽宁 沈阳 110136)

**摘 要:** 针对旅行商问题, 提出了一种新型的离散人工蜂群算法. 根据该优化问题及离散量的特点, 对引领蜂、跟随蜂和侦查蜂角色转变机制和搜索策略进行了重新定义. 蜂群角色转变基于定义的收益比因子. 引领蜂邻域搜索采用 2-Opt 算子和学习操作来加速算法收敛速度; 跟随蜂搜索引入禁忌表来提高算法的局部求精能力; 侦查蜂搜索定义了排斥操作来保持种群的多样性, 从而较好地平衡了算法的探索及开采能力. 实验结果表明, 算法能够在较短时间内找到相对满意解, 提高了 TSP 的求解效率.

**关 键 词:** 离散人工蜂群算法; 旅行商问题; 2-Opt; 学习算子; 排斥算子

**中图分类号:** TP 18      **文献标志码:** A      **文章编号:** 1005-3026(2015)08-1074-06

## Discrete Artificial Bee Colony Algorithm for TSP

YU Hong-tao<sup>1</sup>, GAO Li-qun<sup>1</sup>, TIAN Wei-hua<sup>2</sup>

(1. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China; 2. College of Automation, Shenyang Institute of Engineering, Shenyang 110136, China. Corresponding author: YU Hong-tao, E-mail: neu970773@sohu.com)

**Abstract:** Aimed at traveling salesman problems, a novel discrete artificial bee colony algorithm is proposed. Based on the characteristics of such problems and discrete variables, the transforming mechanism and searching strategy of leader bees, follower bees and scout bees are redefined. The roles of bees are changed dynamically according to the values of profitability ratios. The 2-Opt operator and learning operator are used for leader bees to search the neighborhood of food sources so as to accelerate the convergence. A taboo list is introduced for follower bees to improve the algorithm's intensification ability, and a repulsion operator is designed for scout bees to maintain the diversity of bee colonies. The proposed algorithm can strike a good balance between exploration and exploitation by using these operators. The simulation results show that it can improve the efficiency of solving traveling salesman problems by finding relatively satisfactory solutions in a short time.

**Key words:** discrete artificial bee colony algorithm; TSP (traveling salesman problem); 2-Opt; learning operator; repulsion operator

旅行商问题 (traveling salesman problem, TSP) 是数学领域中一个重要的组合优化问题, 属于 NP 问题. 虽然目前有一些精确算法可用于求解该问题, 但精确算法求解原理复杂, 并且存在随城市数增加而产生的“组合爆炸”问题, 因此, 国内外学者一直努力寻求一种高效稳定的算法求解这一复杂问题.

人工蜂群算法 (artificial bee colony, ABC) 是

Karaboga<sup>[1]</sup> 于 2005 年提出的一种基于群体智能的启发式搜索算法. 该算法以蜂群的自组织模拟模型为基础, 具有设置参数少、易于实现、鲁棒性强等优点, 受到国内外学者的广泛关注, 并已在许多领域得到了应用<sup>[2-7]</sup>. 目前, ABC 算法主要应用于求解连续域优化问题, 而在离散域上的研究和应用相对较少. 虽然文献[6]和文献[7]提出了求解 TSP 的离散人工蜂群算法, 但文献[6]并未

收稿日期: 2014-06-24

基金项目: 国家自然科学基金资助项目(61273155); 辽宁省教育厅一般项目(L2014530).

作者简介: 于宏涛(1978-), 男, 辽宁鞍山人, 东北大学博士研究生, 沈阳工程学院讲师; 高立群(1949-), 男, 辽宁沈阳人, 东北大学教授, 博士生导师.

考虑从整体上提升人工蜂群算法求解性能,而是着重对比分析了产生邻域解的方法;而文献[7]又主要基于蚁群算法求解 TSP 思想,对每个城市逐一进行选择后生成解向量,导致算法求解速度较慢.为了提升 TSP 求解性能,本文提出了一种新的离散人工蜂群算法,该算法考虑了空间探索和局部求精间的平衡.通过对 TSPLIB 标准库中多个实例进行对比测试,结果表明,本文算法求解 TSP 具有较好的效果.

## 1 人工蜂群算法及其离散化定义

ABC 算法的寻优过程模拟蜜蜂寻找优质花蜜源的行为,按分工将人工蜂群分为引领蜂、跟随蜂和侦查蜂三种角色,角色之间基于一定条件进行转变.每个蜜源对应问题的一个可能解,蜜源的收益度代表问题解的质量,每个引领蜂对应一个确定的蜜源,并在迭代中对其邻域进行搜索.引领蜂完成搜索后将所搜索到的信息与跟随蜂共享,跟随蜂按照一定的概率选择蜜源,继续在其邻域进行搜索,若引领蜂在给定的搜索次数内没有获得更好的蜜源,便放弃该蜜源,并且引领蜂将转变为侦查蜂随机搜索可行的新蜜源.

ABC 算法是一种群智能优化算法,而探索和开采是决定群智能优化算法性能的两个主要因素.探索能力越好,个体在全局范围内搜索未知区域的能力越强,即全局寻优能力越强;开采能力越好,个体在局部范围内搜索更优解的能力越强,即局部求精的能力越强.因此,要保证 ABC 算法的求解质量,需要协调和平衡探索及开采的过程,这也是智能群体算法需要解决的一个核心问题<sup>[8]</sup>.

ABC 算法在解决连续变量问题时,缺乏对探索和开采能力的考虑,导致算法求解速度和求解质量有时难以得到保证,本文提出的改进离散人工蜂群算法通过对引领蜂、跟随蜂和侦查蜂的搜索策略进行重新定义,较好地协调和平衡了算法的探索及开采能力.为了便于描述,本文给出以下定义.

**定义 1** 收益比  $r_i$ : 每只蜜蜂搜索到的蜜源收益度  $\text{fit}_i$  与蜂群中最优个体蜜源收益度  $\text{fit}_{\text{best}}$  的比值.对于 TSP,  $\text{fit}_i$  与目标函数  $f(\mathbf{X}_i)$  的关系为

$$\text{fit}_i = \frac{1}{f(\mathbf{X}_i)}. \quad (1)$$

**定义 2** 相似度  $s_{i,j}$ : 是指任意两个解向量  $\mathbf{X}_i$  和  $\mathbf{X}_j$  的相似程度,定义为  $s_{i,j} = M/N$ , 式中  $N$  为城市的数目,  $M$  为两个解向量具有相同邻接边的个

数,可见  $s_{i,j} \in [0, 1]$ .

**例 1** 有一个 10 城市的 TSP, 其两个解向量分别为  $\mathbf{X}_1 = [1\ 5\ 6\ 2\ 4\ 8\ 3\ 10\ 7\ 9]$  和  $\mathbf{X}_2 = [1\ 3\ 6\ 2\ 4\ 8\ 10\ 5\ 7\ 9]$ , 则  $N = 10$ , 两个解向量具有 5 个相同邻接边:  $6 \rightarrow 2, 2 \rightarrow 4, 4 \rightarrow 8, 7 \rightarrow 9, 9 \rightarrow 1$ , 即  $M = 5$ , 所以  $s_{i,j} = 0.5$ .

**定义 3** 排斥操作: 是指对解向量  $\mathbf{X}_i$  和  $\mathbf{X}_j (i \neq j)$  的相同邻接边城市序列重新进行随机排序得到一个向量  $\mathbf{Y}$ , 再把该向量作用于  $\mathbf{X}_j$ , 从而使解向量  $\mathbf{X}_j$  和  $\mathbf{X}_i$  相似度下降.

**例 2** 解向量  $\mathbf{X}_1$  和  $\mathbf{X}_2$  同例 1, 可见相同邻接边城市序列为  $6, 2, 4, 8, 7, 9, 1$ , 若  $\mathbf{Y} = [2\ 0\ 8\ 6\ 1\ 4\ 0\ 0\ 7\ 9]$ , 经排斥操作后

$$\mathbf{X}_2 = [1\ 3\ 6\ 2\ 4\ 8\ 10\ 5\ 7\ 9] + [2\ 0\ 8\ 6\ 1\ 4\ 0\ 0\ 7\ 9] = [2\ 3\ 8\ 6\ 1\ 4\ 10\ 5\ 7\ 9].$$

此时, 解向量  $\mathbf{X}_2$  和  $\mathbf{X}_1$  具有 1 个相同邻接边  $7 \rightarrow 9$ , 则相似度为 0.1, 可见, 相比变换前的相似度 0.5 有所下降.

**定义 4** 学习操作: 是指对解向量  $\mathbf{X}_i$  和  $\mathbf{X}_j (i \neq j)$  的不同邻接边城市序列重新进行随机排序得到一个向量  $\mathbf{Y}$ , 再把该向量作用于  $\mathbf{X}_j$ , 从而使解向量  $\mathbf{X}_j$  和  $\mathbf{X}_i$  相似度上升, 即学习操作为排斥操作的反向操作.

## 2 求解 TSP 的离散人工蜂群算法

在明确定义 1 ~ 定义 4 的基础上, 本文提出了离散人工蜂群算法. 该算法的蜂群同样分引领蜂、跟随蜂和侦查蜂三种角色, 但角色之间不再基于设定的搜索次数进行转换, 而是基于定义的收益比, 并且为了平衡算法探索及开采能力, 对每种角色的搜索策略重新进行了制定.

### 2.1 引领蜂

离散人工蜂群算法中引领蜂的作用与 ABC 相同, 同样是在每个蜜源的邻域内寻找更优蜜源, 但由于求解的 TSP 是离散变量问题, 邻域解的产生机制需要重新定义, 本文寻优初期采用 2-opt 邻域结构<sup>[9]</sup>对每个蜜源进行邻域搜索. 2-opt 邻域操作时, 首先在当前解中分别移走两条不相邻的边, 然后针对这两条边对应的 4 个城市重新进行操作: 将第 1 条边上的第 1 个城市与第 2 条边上的第 1 个城市相连, 第 1 条边上的第 2 个城市与第 2 条边上的第 2 个城市相连; 由此得到的新解作为邻域解.

**例 3** 有一个 5 节点的 TSP, 解为  $\mathbf{X}_i = [1\ 2$

3 4 5], 见图 1a; 对边 (1, 2) 和 (4, 5) 进行 2-opt 变换, 则得到邻域解  $X'_i = [1 \ 4 \ 3 \ 2 \ 5]$ , 见图 1b.

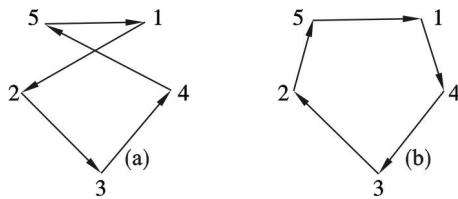


图 1 2-opt 邻域操作示意图

Fig. 1 Neighborhood operation of 2-opt

(a)  $\rightarrow X_i = [1 \ 2 \ 3 \ 4 \ 5]$ ;

(b)  $\rightarrow X'_i = [1 \ 4 \ 3 \ 2 \ 5]$ .

对比解  $X_i$  和  $X'_i$  可以看出, 为了保证经过 2-opt 变换的解为 TSP 的可行解, 实质上 2-opt 邻域操作是倒置  $X_i$  第 1 条边的第 2 个城市和第 2 条边第 1 个城市之间的城市序列.

## 2.2 跟随蜂

离散人工蜂群算法中跟随蜂的作用与 ABC 相同, 是基于轮盘赌方法按照式 (2) 计算出的概率选择某一蜜源, 在其邻域内作局部搜索.

$$p_i = \frac{\text{fit}_i}{\sum_{i=1}^{\text{SN}} \text{fit}_i}. \quad (2)$$

式中 SN 为蜜源总数.

离散人工蜂群算法中跟随蜂搜索与 ABC 不同的是引入了禁忌表  $\text{tabu}_i$  ( $i = 1, 2, \dots, \text{SN}$ ), 用来记录蜜蜂对于蜜源  $X_i$  的最近邻域搜索信息, 如例 3 中蜜源  $X_i$  的最近邻域解  $X'_i$  是通过倒置其城市 2 和城市 4 之间的序列得出的, 对于蜜源  $X'_i$  如果再对以这两个城市分别为首和尾的边 (2, 5) 和 (1, 4) 进行 2-opt 变换, 得到的邻域解又将是  $X_i$ ; 这样的搜索是毫无意义的, 为了避免在蜜源  $X_i$  同一邻域内反复搜索, 提高算法的开采能力, 对于例 3 禁忌表  $\text{tabu}_i$  只需记录 2, 4 两点, 即记录第 1 条边的第 2 个城市和第 2 条边第 1 个城市.

## 2.3 侦查蜂

ABC 算法中侦查蜂是由引领蜂转变而来的, 其作用是负责发现可能陷入局部最优的个体, 并对其进行更新, 从而降低算法出现早熟的概率. 引领蜂转变为侦查蜂的条件是其在设定的搜索次数内没有获得更好的蜜源. 在 ABC 算法中, 侦查蜂转变机制及搜索行为存在以下 3 个问题:

1) 侦查蜂的角色转变依赖于设定的搜索次数, 求解不同实际问题时该参数并不相同, 难以进行设定, 且如设置不当会影响算法跳出局部最优的能力.

2) 侦查蜂是在发现个体已经早熟收敛时才被转变, 并探索新的蜜源, 这样导致算法找到全局最优的速度较慢.

3) ABC 算法中侦查蜂随机地搜索新蜜源, 并未考虑当前处于主导搜索的引领蜂所处的位置, 无法保证尽可能在全局范围内搜索到更多的新蜜源, 从而抑制算法的探索能力.

为了解决以上不足, 本文定义了收益比这一指标, 引领蜂基于该指标转变为侦查蜂. 其优点一方面在于解决不同实际问题时该指标更易被设定; 另一方面在于蜂群角色及搜索机制依赖搜索次数的动态变化. 寻优初期发现个体收益比相对较差时, 引领蜂立刻转变为侦查蜂, 并基于轮盘赌方法执行排斥操作以便基于当前引领蜂位置信息尽可能在全局范围内寻找更优的蜜源; 在寻优后期发现个体收益比都较好时, 引领蜂则不再转变为侦查蜂, 且搜索机制从 2-opt 操作转变为基于轮盘赌方法的学习操作. 即寻优初期充分考虑种群个体多样性, 提高算法的探索能力; 寻优后期充分考虑种群信息共享, 提高算法的开采能力和寻优速度.

归纳起来, 应用离散人工蜂群算法求解 TSP 步骤如下:

步骤 1 初始化蜂群. 随机初始化蜜源, 即随机生成种群初始解, 并按式 (1) 计算各蜜源的收益度.

步骤 2 引领蜂搜索阶段. 计算收益比, 若收益比不都大于或等于设定值  $r$ , 则引领蜂执行 2-opt 操作, 搜索新蜜源, 得到候选邻域解  $X'_i$ ; 否则采用轮盘赌方法按照式 (2) 计算的概率选择蜜源  $X_j$ , 执行学习操作, 搜索新蜜源, 得到候选邻域解  $X'_i$ , 若候选解收益度  $\text{fit}_{X'_i}$  大于  $\text{fit}_{X_i}$ , 则用  $X'_i$  代替  $X_i$ , 反之保留  $X_i$  不变.

步骤 3 跟随蜂搜索阶段. 跟随蜂采用轮盘赌方法按照式 (2) 计算的概率选择蜜源  $X_i$ , 并在可允许操作范围内 (除  $\text{tabu}_i$  外的邻域) 执行 2-opt 操作, 搜索新蜜源, 得到候选邻域解  $X'_i$ , 若候选解收益度  $\text{fit}_{X'_i}$  大于  $\text{fit}_{X_i}$ , 则用  $X'_i$  代替  $X_i$ , 反之保留  $X_i$  不变.

步骤 4 侦查蜂搜索阶段. 计算收益比, 若收益比不都大于或等于设定值  $r$ , 则放弃收益比小于设定值  $r$  的蜜源  $X_i$ , 并采用轮盘赌方法按照式 (2) 计算的概率选择蜜源  $X_j$ , 执行排斥操作搜索新蜜源  $X'_i$ , 否则转步骤 5.

步骤 5 记录迄今为止搜索到的最好解.



步骤 6 检查是否达到算法终止条件,若是,则结束;否则转步骤 2.

### 3 仿真结果与分析

#### 3.1 收益比设定值对算法的影响

收益比设定值  $r$  是离散人工蜂群算法主要参数,下面通过仿真实验分析该参数对算法求解性能的影响. 实验环境为 CPU AMD Athlon 64 X2 Dual Core Processor 4 400 + 2.29 GHz, 内存 2.00 GB, 操作系统为 Windows XP, 编程软件为 MATLAB 7.1, 在给定其他参数的条件下, 针对 Att 48 TSP, 采用不同  $r$  值进行实验. 实验中其他参数设置: 最大迭代次数  $T_{\max} = 2\,000$ , 蜜蜂数目  $m = 48$ , 算法独立运行 20 次, 表 1 记录了不同  $r$  值的运行结果.

表 1 $r$ 对离散人工蜂群算法性能的影响					
Table 1 Effect of $r$ on the performance of discrete artificial bee colony algorithm					
$r$	最好值 $\times 10^{-4}$	最差值 $\times 10^{-4}$	平均值 $\times 10^{-4}$	标准差 $\times 10^{-2}$	平均运行时间 /s
1	3.41	3.61	3.52	5.90	9.922 4
0.9	3.37	3.59	3.49	4.96	9.037 0
0.8	3.36	3.51	3.45	3.60	9.034 3
0.7	3.40	3.54	3.46	3.52	9.046 4
0.6	3.46	3.62	3.56	4.82	9.688 1
0.5	3.49	3.68	3.57	4.87	9.643 7
0.4	3.46	3.67	3.58	5.45	9.623 7
0.3	3.50	3.67	3.59	4.49	9.588 0
0.2	3.47	3.64	3.58	4.35	9.621 3
0.1	3.47	3.66	3.58	5.86	9.602 1
0	3.45	3.68	3.58	6.30	9.622 0

从表 1 可以看出,在求解时间方面,当  $r$  逐渐减小时,算法平均运行时间并没有明显的变化规律,在  $r$  为 0.9,0.8 和 0.7 时,算法平均运行时间相对较短,其中  $r$  为 0.8 时的平均运行时间 9.034 3 最短;但从整体看,算法平均运行时间变化较小,这主要是由于无论  $r$  取何值,算法迭代过程中引领蜂、跟随蜂和侦查蜂的数量之和始终不变,即本次迭代转变为侦查蜂的引领蜂下次迭代时不再执行引领蜂操作,并且基于收益比的蜂群在不同阶段的搜索策略复杂度相差无几. 在求解

质量方面,当  $r$  逐渐减小时,求解性能逐渐变好后又逐渐变差,在  $r$  为 1,0.6,0.5,0.4,0.3,0.2,0.1 和 0 时,离散人工蜂群算法所得到的平均值都在  $3.50\text{E}+04$  以上,最差值都在  $3.60\text{E}+04$  以上,而在  $r$  为 0.9,0.8 和 0.7 时算法得到的平均值都在  $3.50\text{E}+04$  以下,最差值都在  $3.60\text{E}+04$  以下,其中  $r$  为 0.8 时,算法所得到的平均值  $3.45\text{E}+04$ 、最差值  $3.51\text{E}+04$ 、最好值  $3.36\text{E}+04$  均为最小,并且此时得到的标准差  $3.60\text{E}+02$  仅大于  $r$  为 0.7 的标准差  $3.52\text{E}+02$ ,也相对较小;因此可以认为:对此问题, $r$  为 0.8 时离散人工蜂群算法具有较好的求解性能.

#### 3.2 算法性能对比实验

为了进一步验证本文算法的可行性和有效性,选用国际上通用的 TSPLIB 测试库中多个实例进行测试,并将测试结果与相关文献中的 RRS ABC<sup>[6]</sup>,ABC<sup>[7]</sup> 和求解 TSP 性能较好的蚁群优化 (ant colony optimization, ACO) 算法进行对比分析. 文献[6]对多种邻域结构的 ABC 算法进行了仿真实验,RRS 邻域结构与本文最相似,因此选用该邻域结构的 ABC 算法进行对比.

算例中的数字表示城市数目,括号内的数是已知最优解,计算结果“—”表示对比文献未给出该项指标计算结果. 实验环境与前面相同,离散人工蜂群算法参数设置为:最大迭代次数  $T_{\max} = 2\,000$ ,  $r = 0.8$ , 蜜蜂的数目  $m = n$ , 其中  $n$  是城市数目;ACO 算法参数设置为:最大迭代次数  $T_{\max} = 2\,000$ , 蚂蚁的数目  $m = n$ ,  $\alpha = 1$ ,  $\beta = 5$ ,  $\rho = 0.9$ , 本文算法和 ACO 算法对于每个算例独立运行 20 次,计算结果见表 2.

从表 2 可以看出,本文算法与文献[6]中的 RRS ABC 算法相比,在实例 Oliver30, Eil51 和 Berlin52 中,RRS ABC 算法均得到了不差于已知最优解的结果,本文算法虽然仅在 Oliver30 中得到了与已知最优解相同的结果,但对于所有对比实例,本文算法比 RRS ABC 算法迭代次数都减少了 98%,且本文算法求解步骤也并不复杂,而计算结果平均值却仅增加了不大于 0.5%,计算结果最好值增加不大于 1.8%.

本文算法与 ACO 算法相比,对 5 个实例,本文算法计算所得的最好值、平均值都相对更小,且对于 Bays29, Oliver30 和 Dantzig42,其标准差也更小;更重要的是,对 5 个实例,本文算法计算时间分别节省了 98.21%, 98.31%, 98.67%, 98.97% 和 98.98%.

表 2 标准测试问题的计算结果  
Table 2 Results for benchmark test problems

实例	计算指标	本文算法	RRS ABC <sup>[6]</sup>	ABC <sup>[7]</sup>	ACO
Bays29 (2 020)	最好值	2 020	—	2 020	2 040
	平均值	2 027.8	—	2 031.4	2 061.7
	迭代次数	2 000	—	50	2 000
	运行时间	5.11	—	—	285.98
Oliver30 (423.74)	最好值	423.74	423.74	423.5	425.82
	平均值	425.00	423.74	432.5	428.76
	迭代次数	2 000	10 <sup>5</sup>	50	2 000
	运行时间	5.36	—	—	317.73
Dantzig42 (699)	最好值	679.20	—	699	696.12
	平均值	692.05	—	710.2	716.46
	迭代次数	2 000	—	100	2 000
	运行时间	8.95	—	—	672.48
Eil51 (429.98)	最好值	431.24	428.87	426	457.79
	平均值	444.09	431.11	427.6	468.51
	迭代次数	2 000	10 <sup>5</sup>	50	2 000
	运行时间	10.36	—	—	1 001.04
Berlin52 (7 544.37)	最好值	7 680.78	7 544.37	7 542	8 048.07
	平均值	7 941.27	7 562.80	7 573.48	131.51
	迭代次数	2 000	10 <sup>5</sup>	150	2 000
	运行时间	10.76	—	—	1 055.01

本文算法与文献[7]中的 ABC 算法相比, ABC 算法对于所有实例均得到了不差于已知最优解的结果;本文算法对于实例 Bays29, Oliver30 得到了与已知最优解相同的结果;对于实例 Dantzig42, 本文算法计算结果比已知最优解及 ABC 算法计算所得的最优解均少了 19.8。图 2 为 Dantzig42 实例的实验仿真图, 对应的最优路径为 [33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 25, 26, 27, 24, 11, 12, 23, 22, 17, 16, 13, 14, 15, 18, 19, 20, 21, 28, 29, 30, 31, 32]。另外, 对于这 3 个实例, 本文算法的平均值都相对更小。对于实例 Eil51 和 Berlin52, 本文算法未得到已知最优解, 比文献[7]中的 ABC 算法计算结果略差;但 ABC 算法求解步骤主要以 ACO 算法思想为基础并融合了其他操作, 在种群数相等时, 其求解时间至少应与 ACO 算法相当。从本文 ACO 算法仿真结果及文献[10]中的 ACO 算法仿真结果看, ACO 算法运行的确需要较长时间, 因此, 文献[7]中的 ABC 算法计算时间相对较长。

综上所述, 本文算法与文献[6]中的 RRS ABC、文献[7]中的 ABC 和 ACO 相比, 综合权衡了求解时间和求解质量两项指标, 具有较好的求解性能。

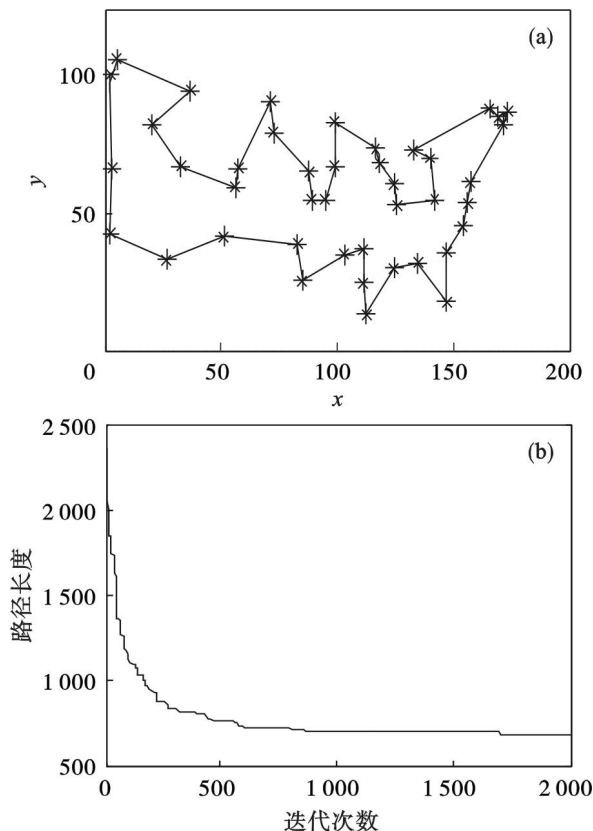


图 2 实例 Dantzig42 的实验仿真图  
Fig. 2 Simulation diagram of Dantzig42. tsp  
(a)—最优路径; (b)—最好值进化曲线。

## 4 结 论

以求解 TSP 为例, 本文提出了一种离散人工蜂群算法, 将连续型人工蜂群优化算法推广到离散域。算法基于定义的收益比指标进行蜂群角色及搜索机制的转变, 定义排斥算子以保持种群的多样性, 引入禁忌表并定义了学习算子以提高算法的局部求精能力和寻优速度, 从而在较好地协调算法空间探索和局部求精能力的同时, 加快了算法的收敛速度。实验结果表明, 本文算法能够在较短时间内找到相对满意解, 提高了 TSP 的求解效率。

### 参考文献:

- [1] Karaboga D. An idea based on honey bee swarm for numerical optimization [R/OL]. [2014-05-12]. [http://mf.erciyes.edu.tr/abc/pub/tr06\\_2005.pdf](http://mf.erciyes.edu.tr/abc/pub/tr06_2005.pdf).
- [2] Xu C, Duan H. Artificial bee colony optimized edge potential function approach to target recognition for low-altitude aircraft [J]. *Pattern Recognition Letters*, 2010, 31(13): 1759-1772.
- [3] Ozturk C, Karaboga D, Gorkemli B. Probabilistic dynamic deployment of wireless sensor networks by artificial bee

- colony algorithm[J]. *Sensors*, 2011, 11(6): 6056–6065.
- [4] Horng M H. Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation[J]. *Expert Systems with Applications*, 2011, 38(11): 13785–13791.
- [5] Omkar S N, Senthilnath J, Khandelwal R, et al. Artificial bee colony (ABC) for multi-objective design optimization of composite structures[J]. *Applied Soft Computing*, 2011, 11(1): 489–499.
- [6] Kiran M S, Iscan H, Gunduz M. The analysis of discrete artificial bee colony algorithm with neighborhood operator on traveling salesman problem[J]. *Neural Computing and Applications*, 2013, 23(1): 9–21.
- [7] 胡中华, 赵敏. 基于人工蜂群算法的 TSP 仿真[J]. 北京理工大学学报, 2009, 29(11): 978–982.
- (Hu Zhong-hua, Zhao Min. Simulation on traveling salesman problem(TSP) based on artificial bees colony algorithm[J]. *Transactions of Beijing Institute of Technology*, 2009, 29(11): 978–982.)
- [8] 林小军, 叶东毅. 一种带规范知识引导的改进人工蜂群算法[J]. 模式识别与人工智能, 2013, 26(3): 307–314.
- (Lin Xiao-jun, Ye Dong-yi. An improved artificial bee colony algorithm with guided normative knowledge[J]. *Pattern Recognition and Artificial Intelligence*, 2013, 26(3): 307–314.)
- [9] Croes G A. A method for solving traveling-salesman problems[J]. *Operations Research*, 1958, 6(6): 791–812.
- [10] Yang C H, Tang X L, Zhou X J, et al. A discrete state transition algorithm for traveling salesman problem[J]. *Control Theory & Applications*, 2013, 30(8): 1040–1046.

(上接第 1068 页)

从表 2 可以清楚地看出:在不同的天气条件下,光伏系统始终实现了 MPPT 控制,但由于阈值  $\varepsilon$ 、跟踪步长以及测量误差等原因,两种方法的输出功率与理想值之间均存在微小误差。同时表 2 也显示,采用本文方法的输出功率值始终比 P&O 法更接近于输出功率的理想值。

综合上述所有实验可以得出结论:当采用 VWP 模糊 MPPT 控制策略时,输出功率特性比传统扰动观察法具有更好的快速性、准确性和平稳性。

## 5 结 语

本文提出的 VWP 分段模糊 MPPT 控制方法采用了两条 VWP 直线将输出功率曲线划分为三个跟踪区域,每个区域采用了不同的模糊控制技术。该控制方法既最大限度地提高了跟踪的快速性,又兼顾了跟踪的准确性和平稳性。仿真实验显示了该控制策略运行的良好性能,验证了该控制策略相对于传统的扰动观察法在跟踪最大功率点时快速性、准确性和平稳性方面的优势。

### 参考文献:

- [1] Abdelsalam A K, Massoud A M, Ahmed S, et al. High-performance adaptive perturb and observe MPPT technique for photovoltaic-based microgrids[J]. *IEEE Transactions on Power Electronics*, 2011, 26(4): 1010–1021.
- [2] Zegaoui A, Aillerie M, Petit P, et al. Comparison of two common maximum power point trackers by simulating of PV generators[J]. *Energy Procedia*, 2011(6): 678–687.
- [3] Messai A, Mellit A, Guessoum A, et al. Maximum power point tracking using a GA optimized fuzzy logic controller and its FPGA implementation[J]. *Solar Energy*, 2011, 85: 265–277.
- [4] Lin W M, Hong C M, Chen C H. Neural-network-based MPPT control of a stand-alone hybrid power generation system[J]. *IEEE Transactions on Power Electronics*, 2011, 26(12): 3571–3581.
- [5] Gao X W, Li S W, Gong R F. Maximum power point tracking control strategies with variable weather parameters for photovoltaic generation systems[J]. *Solar Energy*, 2013, 93: 357–367.
- [6] Li S W, Gao X W, Wang L N, et al. A novel maximum power point tracking control method with variable weather parameters for photovoltaic systems[J]. *Solar Energy*, 2013, 97: 529–536.
- [7] 张明君, 张化光. 一种新的模糊自适应控制方法[J]. 东北大学学报: 自然科学版, 2004, 25(7): 633–636.
- (Zhang Ming-jun, Zhang Hua-guang. New fuzzy adaptive control scheme[J]. *Journal of Northeastern University: Natural Science*, 2004, 25(7): 633–636.)
- [8] 刘毅, 赵军. 基于输出反馈的切换模糊控制设计[J]. 东北大学学报: 自然科学版, 2008, 29(1): 9–12.
- (Liu Yi, Zhao Jun. Output feedback based design of switching fuzzy control[J]. *Journal of Northeastern University: Natural Science*, 2008, 29(1): 9–12.)
- [9] Moura S. A switched extreme seeking approach to maximum power point tracking in photovoltaic systems [EB/OL]. [2014-02-05]. <http://www-personal.umich.edu/~sjmoura/files/EECS498.pdf>.
- [10] Mutoh N, Ohno M, Inoue T. A method for MPPT control while searching for parameters corresponding to weather conditions for PV generation systems[J]. *IEEE Transactions on Industrial Electronics*, 2006, 53(4): 1055–1065.