

旅行商问题的混沌混合离散蝙蝠算法

戚远航¹, 蔡延光¹, 蔡 颢², 汤雅连¹, 吕文祥¹

(1. 广东工业大学自动化学院, 广东广州 510006; 2. 奥尔堡大学健康科学与技术系, 丹麦奥尔堡 9220)

摘 要: 针对现有离散蝙蝠算法在求解旅行商问题时存在的收敛速度较慢、收敛率不高等问题, 提出了混沌混合离散蝙蝠算法. 该算法采用混沌初始化策略提高算法的寻优能力, 引入 2-Opt 技术增强算法的局部搜索能力、加快算法的收敛速度. 大量的仿真实验表明: 所提出的算法在求解小规模 TSP 时能快速收敛到已知最优解; 在求解大规模 TSP 时能在较短的时间内收敛到偏差 0.4% 以内的最优解.

关键词: 旅行商问题; 混沌初始化; 蝙蝠算法; 2-Opt

中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2016)10-2543-05

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2016.10.037

Chaotic Hybrid Discrete Bat Algorithm for Traveling Salesman Problem

QI Yuan-hang¹, CAI Yan-guang¹, CAI Hao², TANG Ya-lian¹, LÜ Wen-xiang¹

(1. School of Automation, Guangdong University of Technology, Guangzhou, Guangdong 510006, China;

2. Department of Health Science & Technology, Aalborg University, Aalborg 9220, Denmark)

Abstract: In view of some problems, like slow convergence speed and low constringency rate, arising during the process of applying discrete bat algorithms to solve travelling salesman problem, a chaotic hybrid discrete bat algorithm is proposed. The proposed algorithm adopts chaotic initialization strategy to improve the capability of optimization, and the 2-Opt to enhance the capability of local search and to speed up the convergence speed. A large amount of simulations show that the algorithm can achieve their solutions rapidly for some small scale traveling salesman problems, and obtain their solutions in a relatively short time with the error less than 0.4% for large ones.

Key words: traveling salesman problem; chaotic initialization; bat algorithm; 2-Opt

1 引言

旅行商问题(Traveling Salesman Problem, TSP)是组合优化中的一个著名难题. 其求解算法很多, 如遗传算法、蚁群算法、粒子群算法^[1].

蝙蝠算法(Bat Algorithm, BA)是 Xin-She YANG 在 2010 年提出的一种元启发式算法^[2]. 李枝勇等^[3]提出了离散型蝙蝠算法求解最小比率旅行商问题, A Rezaee Jordehi^[4]提出了混沌蝙蝠群算法. 但是, 现有离散蝙蝠算法在求解 TSP 时存在着收敛速度较慢、收敛率不高等问题.

本文提出了混沌混合离散蝙蝠算法(Chaotic Hybrid Discrete Bat Algorithm, CHDBA). CHDBA 采用混沌

初始化策略提高算法的寻优能力, 引入 2-Opt 技术增强算法的局部搜索能力、加快算法的收敛速度. 在大量的仿真实验中: 与离散型蝙蝠算法^[3](Discrete Bat Algorithm, DBA)、混合粒子群算法^[5](Hybrid Particle Swarm Optimization, HPSO)相比, CHDBA 在求解小规模 TSP 时能快速收敛到已知最优解; 与混合遗传算法^[6](Hybrid Genetic Algorithm, HGA)、离散型萤火虫群优化算法^[7](Discrete Glowworm Swarm Optimization, DGSO)相比, CHDBA 在求解大规模 TSP 时能在较短的时间内收敛到偏差 0.4% 以内的最优解.

2 TSP 的定义及其数学模型

给定 n 个城市以及各城市之间的距离, 要求找到一

收稿日期: 2015-03-24; 修回日期: 2015-06-04; 责任编辑: 李勇锋

基金项目: 国家自然科学基金(No. 61074147); 广东省自然科学基金(No. S2011010005059); 广东省教育部产学研结合项目(No. 2012B091000171, No. 2011B090400460); 广东省科技计划项目(No. 2012B050600028, No. 2014B010118004); 广州市花都区科技计划项目(No. HD14ZD001)

条遍历所有城市且每个城市只能被访问一次的路径,并使得总路径长度最短. 数学模型^[8]:对于 n 个城市,遍历所有城市且只能被访问一次的路径为 $C = (c_1, c_2, \dots, c_n)$, 使

$$\text{Min } L = \sum_{i=1}^{n-1} d(c_i, c_{i+1}) + d(c_n, c_1) \quad (1)$$

其中, $d(c_i, c_{i+1})$ 为城市 c_i, c_{i+1} 之间的距离, $i = 1, 2, \dots, n-1$, $d(c_n, c_1)$ 为 c_n, c_1 之间的距离.

3 混沌混合离散蝙蝠算法

3.1 参数定义与算子设计

①蝙蝠位置:第 i 个蝙蝠的位置定义为 $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$, 其中 n 为城市的个数, $i = 1, 2, \dots, Q$ ($Q \in N^+$ 为种群规模), $(x_{i1}, x_{i2}, \dots, x_{in})$ 是 $(1, 2, \dots, n)$ 的一个置换. x_i 表示第 i 个蝙蝠的城市遍历路径为 $x_{i1} \rightarrow x_{i2} \rightarrow \dots \rightarrow x_{in} \rightarrow x_{i1}$.

②蝙蝠速度:第 i 个蝙蝠的速度定义为 $v_i = \{(s_{i1}, t_{i1}), (s_{i2}, t_{i2}), \dots, (s_{in}, t_{in})\}$, 其中 $s_{im}, t_{im} \in \{1, 2, \dots, n\}$, $m = 1, 2, \dots, n$.

③蝙蝠频率:第 i 个蝙蝠的频率定义为 $f_i \in [0, 1]$.

④蝙蝠发射频度和脉冲响度:定义第 i 个蝙蝠在 t 时刻的发射频度为 R_i^t 、脉冲响度为 A_i^t . 初始 $t = 0$ 时, R_i^0, A_i^0 为预给常数 ($0 < R_i^0, A_i^0 \leq 1$); 在 t 时刻其则在 $t + 1$ 时刻蝙蝠的脉冲响度 A_i^{t+1} 和发射频度 R_i^{t+1} 按式(2)、(3)计算:

$$A_i^{t+1} = \alpha A_i^t \quad (2)$$

$$R_i^{t+1} = R_i^0 \times [1 - \exp(-\gamma t)] \quad (3)$$

其中: α, γ 均为常数, $0 < \alpha < 1, \gamma > 0; t = 1, 2, \dots$.

⑤置换操作:设第 i 个蝙蝠的位置为 $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$, $w_i = (k_1, k_2)$ 为置换序列, 其中 $k_1, k_2 = 1, 2, \dots, n$. x_i 的基于 w_i 的置换操作是指 x_i 中第 k_1 分量和第 k_2 分量互换位置.

⑥蝙蝠位置与位置的减法操作:设第 i 个蝙蝠的位置为 $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$, 当前全局最优蝙蝠的位置为 $x_* = (x_{*1}, x_{*2}, \dots, x_{*n})$. 第 i 个蝙蝠的位置与位置的减法操作:初始 $v_i^{\text{diff}} = \emptyset$, 从第 1 个分量开始, 依次检查 x_i 、 x_* 的每个分量, 如果 $x_{im} - x_{*m} \neq 0$, 则 $v_i^{\text{diff}} = v_i^{\text{diff}} \cup (x_{im}, x_{*m})$, $m = 1, 2, \dots, n$. 称 v_i^{diff} 为第 i 个蝙蝠的位置差.

⑦蝙蝠速度与位置差的加法操作:设第 i 个蝙蝠的速度为 $v_i = \{(s_{i1}, t_{i1}), (s_{i2}, t_{i2}), \dots, (s_{in}, t_{in})\}$, 频率为 f_i , 位置差为 $v_i^{\text{diff}} = \{(s_{i1}^{\text{diff}}, t_{i1}^{\text{diff}}), (s_{i2}^{\text{diff}}, t_{i2}^{\text{diff}}), \dots, (s_{in}^{\text{diff}}, t_{in}^{\text{diff}})\}$. 第 i 个蝙蝠的速度与位置差的加法操作:初始 $v_i^{\text{all}} = v_i \cup v_i^{\text{diff}}$, $l = \lfloor f_i \times n \rfloor + 1$; 然后在 v_i^{all} 中随机抽取 l 个元素得 $v_i^{\text{add}} = \{(s_{i1}^{\text{add}}, t_{i1}^{\text{add}}), (s_{i2}^{\text{add}}, t_{i2}^{\text{add}}), \dots, (s_{il}^{\text{add}}, t_{il}^{\text{add}})\}$. 称 v_i^{add} 为第 i 个蝙蝠的速度和.

⑧蝙蝠位置与速度和的加法操作:设第 i 个蝙蝠的

位置为 $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$, 速度和为 $v_i^{\text{add}} = \{(s_{i1}^{\text{add}}, t_{i1}^{\text{add}}), (s_{i2}^{\text{add}}, t_{i2}^{\text{add}}), \dots, (s_{il}^{\text{add}}, t_{il}^{\text{add}})\}$. 第 i 个蝙蝠的位置与速度和的加法操作:从 v_i^{add} 的第 1 个分量开始, 依次进行 x_i 的基于 $(s_{im}^{\text{add}}, t_{im}^{\text{add}}) \in v_i^{\text{add}}$ ($m = 1, 2, \dots, l$) 的置换操作, 得到第 i 个蝙蝠的新位置为 x_i^{new} ; 最后随机生成 $v_i^{\text{rand}} = \{(s_{i1}^{\text{rand}}, t_{i1}^{\text{rand}}), (s_{i2}^{\text{rand}}, t_{i2}^{\text{rand}}), \dots, (s_{iw}^{\text{rand}}, t_{iw}^{\text{rand}})\}$, 其中 $w = n - l$, $s_{im}^{\text{rand}}, t_{im}^{\text{rand}} \in \{1, 2, \dots, n\}$, $m = 1, 2, \dots, w$, 则第 i 个蝙蝠的新速度为 $v_i^{\text{new}} = v_i^{\text{add}} \cup v_i^{\text{rand}}$.

3.2 混沌初始化

3.2.1 Logistic 映射

选用 Logistic 映射来产生混沌序列:

$$z_{i+1} = \mu z_i (1 - z_i) \quad (4)$$

其中 $0 \leq z_i \leq 1$ 且 $z_i \neq 0.25, 0.5, 0.75, i = 0, 1, 2, \dots, \mu = 4$.

3.2.2 混沌量与路径的对应关系

应用全排列构造理论^[9]建立序号 D ($D \in \{1, \dots, n!\}$) 与路径(即 $1, 2, \dots, n$ 的全排列)的对应关系.

以 $(1, 2, 3)$ 的全排列为例, 序号 D 、向量 V 和构造 C (即路径)组成了 DVC 表, 如表 1 所示.

表 1 (1, 2, 3) 全排列的 DVC 表

D	V	C	D	V	C	D	V	C
1	(1, 1)	(1, 2, 3)	3	(2, 1)	(2, 1, 3)	5	(3, 1)	(3, 1, 2)
2	(1, 2)	(1, 3, 2)	4	(2, 2)	(2, 3, 1)	6	(3, 2)	(3, 2, 1)

① D/V 转换公式:

$$\begin{cases} D_0 = D \\ v_i = \left\lceil \frac{D_{i-1}}{(n-i)!} \right\rceil \\ D_i = D_{i-1} - (v_i - 1)(n-i)! \end{cases} \quad (5)$$

其中 $i = 1, 2, \dots, n-1$.

以 $(1, 2, 3)$ 的全排列为例, $n = 3$; 取序号 $D = 3$, 根据式(5)可得 $V = (v_1, v_2) = (2, 2)$.

② V/C 转换

通过向量 V 的指针功能来确定构造 C .

以 $V = (2, 2)$ 为例, 转换步骤如表 2 所示.

表 1 V/C 转换步骤

步骤	V 的取值	V 指针指向的位置	C 的取值
1	$v_1 = 2$	(1, <u>2</u> , 3)	$c_1 = 2$
2	$v_2 = 2$	(1, <u>3</u>)	$c_2 = 3, c_3 = 1$

从表 2 可看出, $C = (c_1, c_2, c_3) = (2, 3, 1)$.

③混沌量可以与向量 V 的转换公式

由式(4)产生混沌量 z_i , 则 $D_0 = n!$, z_i , 代入式(5), 令 $d_i = nz_i$, 得

$$\begin{cases} v_1 = \lceil d_1 \rceil \\ d_i = (n-i+1)(d_{i-1} - v_{i-1} + 1) \\ v_i = \lceil d_i \rceil \end{cases} \quad (6)$$

其中 $i = 2, 3, \dots, n - 1$.

混沌量 z_i 与向量 V 对应,再通过 V/C 转换,混沌量 z_i 便可与构造 C (即路径)对应.

3.2.3 混沌初始化策略

当蝙蝠种群初始化时,通过式(4)和式(6)生成一定数量的可选择蝙蝠位置,择优初始化蝙蝠种群,以此加快算法的收敛速度.

3.3 2-Opt 算法

2-Opt 算法示意图如图 1 所示,其中没有标号的顶点代表两个或者两个以上顶点间一系列的边. 如果 $ab + cd > ac + bd$ 成立,则删除边 ab 和 cd ,同时增加边 ac 和 bd ,并把顶点 b, c 之间的边反向^[10].

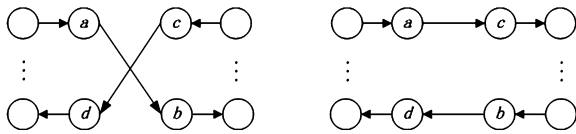


图1 2-Opt算法示意图

3.4 算法步骤

第 1 步:初始化. 最大迭代次数为 N ,蝙蝠的种群规模为 Q ,当前迭代次数为 $N_{\text{now}} = 0$,当前时刻为 $t = 0$,第 i 只蝙蝠的位置为 x_i 、速度为 v_i 、脉冲发射频率为 $R_i^t = R_i^0$ 、脉冲响度为 $A_i^t = A_i^0$ 、脉冲频率 f_i . 其中 x_i 由 3.2 节方法生成, $i = 1, 2, \dots, Q$.

第 2 步:根据初始蝙蝠种群中每个蝙蝠的 x_i 计算函数适应值 $fitness_i$,初始化全局最优解 x_* 和 $fitness_*$.

第 3 步:按照 3.1 节⑤~⑧定义的算子更新每个蝙蝠的位置 x_i^{new} ,速度 v_i^{new} .

第 4 步:如果 $rand > R_i^t$,则使用 3.2 节方法生成新蝙蝠位置 x_i^{new} ,覆盖原来的 x_i^{new} ,否则,执行下一步.

第 5 步:通过 2-Opt 优化技术找出 x_i^{new} 邻域中最好的蝙蝠位置 x_i^{pbest} 及其 $fitness_i^{\text{new}}$.

第 6 步:如果 $fitness_i^{\text{new}} < fitness_i^{\text{old}}$ 且 $rand < A_i^t$,则 $x_i = x_i^{\text{pbest}}$, $t = t + 1$,并分别按公式(2)、(3)计算更新 A_i^t 和 R_i^t ,否则,执行下一步.

第 7 步:如果 $fitness_i^{\text{new}} < fitness_*$,则更新全局最优解 x_* 和 $fitness_*$,否则,执行下一步.

第 8 步:如果 $N_{\text{now}} < N$,则 $N_{\text{now}} = N_{\text{now}} + 1$,并转至第 3 步,否则,执行下一步.

第 9 步:输出全局最优解 x_* .

4 算法测试

4.1 实验环境与算法参数设置

实验软件为 VS2008, CPU 为 Intel Core i7 2.30GHz, 内存为 4GB, Window 7 操作系统.

在仿真实验中,每个 TSP 算例分别单独优化 40 次. 种群规模 Q 和最大迭代次数 N 的取值如表 3 所示;第 i 个蝙蝠的频率 $f_i \in [0, 1]$,响度 $A_i^0 \in [0, 1]$,发射频率 $R_i^0 \in [0, 1]$,响度因子影响 $\alpha = 0.9$,发射频率影响因子 $\gamma = 0.9, i = 1, 2, \dots, Q$.

在仿真实验的分析中,收敛率、偏差分别由式(7)、(8)定义,“已知最优解”是指 TSPLIB 标准库提供的最优解,“—”是指所参考的文献并未提供该相关数据.

$$\text{收敛率} = \frac{\text{最优解小于或等于已知最优解的次数}}{\text{测试次数}} \% \quad (7)$$

$$\text{偏差} = \frac{\text{最优解} - \text{已知最优解}}{\text{已知最优解}} \% \quad (8)$$

表 3 CHDBA 在不同 TSP 算例中种群规模和最大迭代次数的取值

TSP 算例	已知最优解	种群规模	最大迭代次数
Burma14	30.8785	2	6
Oliver30	423.7406	3	25
Eil51	429.9833	80	150
Pr76	108159.44	3	350
Ch130	6110.72	6	1000
Krob150	26127.36	10	1000
Krob200	29440.41	15	1500

4.2 小规模 TSP 实验与分析

小规模 TSP 实验中,将 CHDBA 和 DBA、HPSO 进行比较,实验结果如表 4 所示.

从表 4 可以看出,针对小规模 TSP,CHDBA 具有较好的寻优能力(所有算例均收敛到已知最优解),偏差为 0,收敛率为 100%,较少的时间耗费. 图 2(a) 为 CHDBA 优化 Eil51 的最优解.

表 4 小规模 TSP 实验结果

TSP 算例	算法	最优解	收敛率 (%)	偏差 (%)	时间 (s)
Burma14	DBA	30.8785	35	0.00	0.853
	HPSO	30.8785	100	0.00	—
	CHDBA	30.8785	100	0.00	0.001
Oliver30	DBA	423.7406	10	0.00	6.945
	HPSOA	—	—	—	—
Eil51	CHDBA	423.7406	100	0.00	0.019
	DBA	814.5289	0	89.43	49.927
	HPSO	431	—	2.1	—
	CHDBA	428.8718	100	0.00	5.375

4.3 大规模 TSP 实验与分析

大规模 TSP 实验中,将 CHDBA 和 HGA、DGSO 进行

比较,实验结果如表 5 所示.

从表 5 可以看出,针对大规模 TSP,CHDBA 能收敛到偏差 0.4% 以内的最优解,寻优能力比 DGSO、HGA 更强,而相对于 HGA,CHDBA 耗费的时间更少.尤其在 Krob200 中,CHDBA 虽然也没有收敛到已知最优解,但是收敛结果为 29554.13,偏差仅为 0.39%,比 DGSO 降低了 0.18%,比 HGA 降低了 0.02%,耗费的时间却仅为 HGA 的 1/4. 图 2(b)、2(c)、2(d)分别为 CHDBA 优化 Ch130、Krob150 和 Krob200 的最优解.

表 5 大规模 TSP 实验结果

TSP 算例	算法	最优解	偏差 (%)	时间(s)
Pr76	DGSO	108159.42	0.00	—
	HGA	108159.44	0.00	110.781
	CHDBA	108159.44	0.00	2.053
Ch130	DGSO	6125.07	0.25	—
	HGA	6110.71	0.00	1475.906
	CHDBA	6110.72	0.00	74.694
Krob150	DGSO	26206.69	0.29	—
	HGA	26127.34	0.00	2613.016
	CHDBA	26127.36	0.00	201.855
Krob200	DGSO	29605.1	0.57	—
	HGA	29557.35	0.41	7443.150
	CHDBA	29554.13	0.39	1734.270

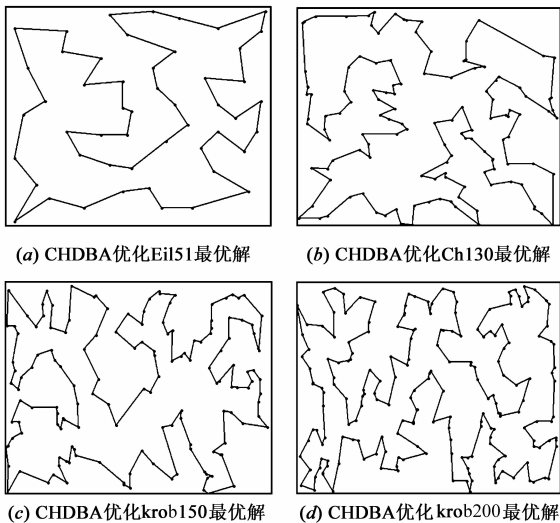


图 2 CHDBA 优化不同 TSP 的最优解

5 结语

本文提出的 CHDBA 采用混沌初始化策略提高算法的寻优能力,引入 2-Opt 技术增强算法的局部搜索能力,加快算法的收敛速度.大量的仿真实验表明:

CHDBA 在求解小规模 TSP 时能快速收敛到已知最优解,在求解大规模 TSP 时能在较短的时间内收敛到偏差 0.4% 以内的最优解.今后工作仍需进行更多的数值实验和对算法的参数取值做进一步的研究.

参考文献

- [1] 高海昌,冯博琴,朱利.智能优化算法求解 TSP 问题[J].控制与决策,2006,21(3):241-247.
GAO Hai-chang, FENG Bo-qin, ZHU Li. Reviews of the meta-heuristic algorithms for TSP [J]. Control and Decision, 2006, 21(3): 241-247. (in Chinese)
- [2] Xin-She YANG. A new metaheuristic bat-inspired algorithm [J]. Nature Inspired Cooperative Strategies for Optimization, 2010, 284: 65-74.
- [3] 李枝勇,马良,张慧珍.求解最小比率旅行商问题的离散蝙蝠算法[J].计算机应用研究,2015,32(2):356-359.
LI Zhi-yong, MA Liang, ZHANG Hui-zhen. Discretebat algorithm for solving minimum ratio traveling salesman problem [J]. Application Research of Computers, 2015, 32(2): 356-359. (in Chinese)
- [4] A Rezaee Jordehi. Chaotic bat swarm optimization (CB-SO) [J]. Applied Soft Computing, 2015, 26: 523-530.
- [5] 俞靓亮,王万良,介婧.基于混合粒子群优化算法的旅行商问题求解[J].计算机工程,2010,36(11):183-184.
YU Liang-liang, WANG Wan-liang, JIE Jing. Solution of travel salesman problem based on hybrid particle swarm optimization algorithm [J]. Computer Engineering, 2010, 36(11): 183-184. (in Chinese)
- [6] Yong Wang. The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem [J]. Computers & Industrial Engineering, 2014, 70: 124-133.
- [7] 周永权,黄正新,刘洪霞.求解 TSP 问题的离散型萤火虫群优化算法[J].电子学报,2012,40(6):1164-1170.
ZHOU Yong-quan, HUANG Zheng-xin, LIU Hong-xia. Discrete glowworm swarm optimization algorithm for TSP problem [J]. Acta Electronica Sinica, 2012, 40(6): 1164-1170. (in Chinese)
- [8] Marjan Kuchaki Rafsanjani, Sadegh Eskandari, Arsham Borumand Saeid. A similarity-based mechanism to control genetic algorithm and local search hybridization to solve traveling salesman problem [J]. Neural Computing and Applications, 2015, 26(1): 213-222.
- [9] 高尚.解旅行商问题的混沌蚁群算法[J].系统工程理论与实践,2005,25(9):100-104.
GAO Shang. Solving traveling salesman problem by chaotic colony optimization algorithm [J]. System Engineering Theory and Practice, 2005, 25(9): 100-104. (in Chinese)
- [10] 姜昌华,戴树贵,胡幼华.求解车辆路径问题的混合遗传算法[J].计算机集成制造系统,2007,13(10):2047

-2052.
JIANG Chang-hua, DAI Shu-gui, HU You-hua. Hybrid genetic algorithm for capacitated vehicle routing problem

[J]. Computer Integrated Manufacturing Systems, 2007, 13(10):2047-2052. (in Chinese)

作者简介



戚远航 男,1993年6月出生,广东湛江人.现为广东工业大学硕博连读生,从事供应链物流及智能算法的研究.
E-mail: qiyuanhang77@163.com



蔡延光 男,1963年2月出生,湖北咸宁人.1988年和1996年分别在重庆大学和浙江大学获理学硕士和工学博士学位.现为广东工业大学教授,博士生导师,从事复杂网络系统建模、控制与优化、物流控制与优化、智能交通系统、组合优化、智能优化、物联网信息处理与优化控制等方面的研究.