

混沌烟花算法求解旅行商问题

蔡延光¹ 陈厚仁¹ 戚远航²

(广东工业大学自动化学院 广州 510006)¹ (电子科技大学中山学院 广东 中山 528402)²

摘 要 旅行商问题(Travelling Salesman Problem,TSP)是一种经典的组合优化问题,属于典型的 NP 难问题,具有重要的研究价值。文中提出了一种混沌烟花算法来求解 TSP。所提算法使用最大位置法定义离散域中的烟花算法,并加入混沌优化策略来增强算法的搜索能力。设计了 4 个参数实验来分析主要参数对 CFWA 的影响并确定了较优的参数设置。对比实验表明:相比于对比算法,混沌烟花算法求解旅行商问题时具有较好的收敛性和稳定性。

关键词 烟花算法,混沌搜索,旅行商问题,最大位置法,参数分析

中图法分类号 TP301 **文献标识码** A

Chaotic Fireworks Algorithm for Solving Travelling Salesman Problem

CAI Yan-guang¹ CHEN Hou-ren¹ QI Yuan-hang²

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

(Zhongshan Institute,University of Electronic Science and Technology of China,Zhongshan,Guangdong 528402,China)²

Abstract Travelling salesman problem (TSP) is a classic combinatorial optimization problem,which is a typical NP hard problem and has important research value. This paper presented a chaotic fireworks algorithm to solve the TSP problem. The algorithm uses LOV to define the discrete domain and adds chaos optimization strategy to enhance search capability of the algorithm. In this paper,four parametric experiments were designed to analyze the influence of the main parameters on CFWA and determine the optimal parameter setting. The comparison experiment shows that the chaotic fireworks algorithm has better convergence and stability than the comparison algorithm.

Keywords Fireworks algorithm,Chaotic searching,Travelling salesman problem,Largest order value,Parameter analysis

1 引言

旅行商问题(Traveling Salesman Problem,TSP)是一种经典的组合优化问题,属于典型的 NP 难问题,在物流配送与车辆路径规划等实际领域具有重要研究价值。TSP 问题^[1]是在一个网络中,寻找一条使之能够有且仅有一次通过网络中的每个点,而且最后需返回到出发点的最短路径。目前求解 TSP 问题的优化算法大致可分为两类:根据 TSP 问题本身特征设计的局部启发式搜索算法和独立于问题的启发式智能算法^[2]。局部启发式优化算法通常能仅用很短的时间计算出上百个城市点的 TSP 问题,如 2-opt^[3],3-opt^[4]和 Lin Kernighan(LKH)^[5]等。但是算法过于依赖 TSP 问题本身特征,所以其适应范围较窄且容易陷入局部最优解。目前现代启发式智能算法有遗传算法^[6]、粒子群算法^[7]、蚁群算法^[8]、蝙蝠算法^[9]、萤火虫群算法^[8]等。目前有很多学者致力于研究现代启发式智能算法及其优化算法,通过模拟某些自然界现象或者过程来解决 TSP 问题。文献[3]提出一种有效的编码与解码方法并结合离散型萤火虫群优化算法求解 TSP 问题;文

献[10]针对蚁群算法收敛慢等问题,提出一种基于自然选择策略的改进型蚁群算法;文献[11]针对传统遗传算法的不足,提出一种结合 OX 改良等算子的改进遗传算法;文献[12]结合遗传算法、蚁群算法和模拟退火算法的思想,提出用混合粒子群算法。

烟花算法(Firework Algorithm,FWA)^[13]是于 2010 年由 Tan 和 Zhu 所提出来的智能算法,该算法受启发于烟花在夜空中爆炸产生火花并照亮周围区域这一现象。目前烟花算法已被学者们应用到许多领域^[14-19];Zhu 等^[14]针传统烟花算法寻优过程中粒子间信息交流少等缺点,提出带引力搜索算子的烟花算法(FAGSO);Han 等^[15]结合模拟退火算法与高斯扰动提出烟花优化算法(SAFWA);Zhang 等^[16]提出将改进烟花算法运用在 Web 服务组合优化服务中。但是目前还没有学者不运用烟花算法及其相关算法来求解 TSP 问题。因此,本文提出了一种混沌烟花算法(CFWA)来求解 TSP 问题。CFWA 使用最大位置法^[17]定义离散域中的烟花算法,并加入混沌优化策略来增强算法搜索能力。本文设计了 4 个参数实验来分析主要参数对 CFWA 的影响并确定了较优的参

本文受国家自然科学基金(61074147),广东省自然科学基金(S2011010005059),广东省教育部产学研结合项目(2012B091000171,2011B090400460),广东省科技计划项目(2012B050600028,2014B010118004,2016A050502060),广州市花都区科技计划项目(HD14ZD001),广州市科技计划项目(201604016055),广州市天河区科技计划项目(2018CX005),广东省普通高校青年创新人才项目(2018KQNCX333)资助。
蔡延光(1963—),男,博士,教授,博士生导师,主要研究方向为网络控制与优化、组合优化、智能优化、智能交通系统等,E-mail:caiyg99@163.com(通信作者);陈厚仁(1994—),男,硕士生,主要研究方向为物联网智能信息处理技术;戚远航(1993—),男,博士,主要研究方向为复杂系统控制与优化、智能优化、物流运输优化调度。

数设置。对比实验表明:相对于对比算法,混沌烟花算法求解旅行商问题时具有较好的收敛性和稳定性。

2 TSP 问题模型

TSP 问题可描述为:已知 N 个城市的坐标位置或者两两之间的距离,一个销售员需要到该 N 个城市销售商品,从某个城市出发,必须有且仅有一次经过其他 $(N-1)$ 个城市,最后回到出发点城市,求如何安排访问次序使得行程路径最短^[2]。数学模型如下所示:

若访问次序为 $S=(P_1,P_2,\cdots,P_N)$ 可满足 $f(S)$ 的值最小,则访问次序 S 就为该 N 个城市条件下 TSP 问题的最优解,即为最优路径。

$$f(S)=\sum_{n=1}^{N-1}(d(P_i,P_{i+1}))+d(P_N,P_1)$$
 (1)

其中, P_i 为城市代号, $i\in(1,2,\cdots,N)$, $d(P_i,P_j)$ 为城市 P_i 与城市 P_j 之间的距离。若只已知城市坐标位置,则:

$$d(P_i,P_j)=\sqrt{(X_i-X_j)^2+(Y_i-Y_j)^2}$$
 (2)

其中, (X_i,Y_i) 为 P_i 的坐标位置。根据实际可知 $d(P_i,P_j)=d(P_j,P_i)$ 。

3 混沌烟花算法

3.1 烟花算法

烟花算法^[12]是受启发于在中国传统节日中燃放烟花的爆炸现象所提出来的智能优化算法。烟花的每一次爆炸相当于在可行解空间内的一次寻优过程,通过多次迭代中对烟花火花质量进行筛选,可较快地找到所设问题的最优解^[18]。

烟花算法由爆炸算子、高斯变异算子和选择策略 3 个部分组成^[19]。

3.1.1 爆炸算子

烟花算法中爆炸算子遵循“适应度好的烟花爆炸密而近,适应度差的烟花爆炸疏而远”原则^[20]。根据实际应用情况设计目标函数计算每个烟花的适应度值,并根据适应度值判断烟花质量。爆炸算子主要负责对烟花邻近搜索适应度更好的烟花,适应度较好的烟花在其较小的邻域产生较多的烟花,而适应度较差的烟花在较大的邻域产生较少的烟花。式(3)、式(4)分别为每个烟花爆炸火花数 S_i 与爆炸半径 R_i 的计算公式。

$$S_i=S_{sum} * \frac{f_{max}-f_i+\epsilon}{\sum_{i=1}^N(f_{max}-f_i)+\epsilon}$$
 (3)

$$R_i=A * \frac{f_i-f_{min}+\epsilon}{\sum_{i=1}^N(f_i-f_{min})+\epsilon}$$
 (4)

其中, S_{sum} 为预设的爆炸火花数, A 为基本爆炸半径, f_{max} 与 f_{min} 为当前烟花种群中的最大适应度值与最小适应度值, f_i 为烟花 i 的适应度值,加入 ϵ 避免除零操作。

为了避免在算法中适应度较好的烟花获得过多的爆炸,浪费运算资源,应确保爆炸火花数不能过多,同时适应度较差的烟花火花数也应该控制不能过少。用式(5)对每个烟花爆炸火花数进行调整:

$$S_i=\begin{cases} S_{max}, & S_i>S_{max} \\ S_i, & S_{min}<S_i<S_{max} \\ S_{min}, & S_i<S_{min} \end{cases}$$
 (5)

其中, S_{max} 与 S_{min} 皆为预设的最大爆炸火花数与最小爆炸火花

数,由式(5)可知 S_{max} 与 S_{min} 应小于爆炸火花数。

3.1.2 高斯变异算子

烟花算法中高斯变异火花的引入增强了种群多样性,避免算法陷入局部最优^[19]。高斯变异操作如下:首先在当前烟花种群中随机选择烟花 x_i ,再随机选择该烟花的一定数量的纬度 k 进行高斯变异运算。

$$\hat{x}_{ik}=x_{ik} * e$$
 (6)

其中, $e\sim N(1,1)$, $N(1,1)$ 代表均值为 1、方差为 1 的高斯分布。

3.1.3 选择策略

为了能使较优的种群信息传递到下一次迭代运算中,在爆炸火花与高斯变异火花运算完成后,在原本烟花、爆炸火花和高斯变异火花中选择最优 n 个个体作为下一代烟花,其余 $(N-n)$ 个烟花按照赌轮盘规则选出。每个烟花在赌轮盘规则中被选中的概率为:

$$p_i=\frac{\sum_{j=1}^Kd(N_i-N_j)}{\sum_{j=1}^KN_j}$$
 (7)

其中, $\sum_{j=1}^Kd(N_i-N_j)$ 表示候选烟花中个体与其他候选个体之间的距离。距离越大,该候选烟花被选中的概率就越大。

3.2 混沌优化策略

混沌是非线性系统所特有的一种非周期现象^[21]。混沌搜索的遍历性与随机性等特点有利于增强智能算法全局搜索能力^[7]。混沌优化的主要思路是:首先将烟花向量序列通过混沌映射规则转化为混沌搜索空间内的混沌变量,然后通过混沌变量进行遍历性寻优搜索,再将遍历到的混沌序列转回烟花向量序列。若通过适应度评估函数计算发现此次混沌搜索后得到了更优的烟花个体,则更新混沌搜索前的烟花个体。

为了保证烟花个体种群的多样性与随机性,本文在烟花算法初始化烟花个体时采用混沌生成策略,其主要思路为先利用系统随机函数生成烟花向量序列 x_i ,随后的初始化烟花向量序列 x_i 的生成规则为:

$$x_i=Chaos(x_{i-1}),i=2,3,\cdots,N$$

其中, N 为初始化烟花的个数。

在使用混沌自映射函数之前,每个烟花个体需要映射到混沌空间 $(-1,1)$ 中:

$$y_i=\frac{2(x_i-a_i)}{b_i-a_i}-1$$

本文采用具有更好遍历性的逻辑自映射函数来产生混沌序列:

$$y_{i+1}=1-2y_i^2,i=1,2,\cdots,y_i\in(-1,1)$$

逻辑自映射函数的定义域为 $(-1,1)$ 且不为 0 和 0.5,在逻辑定义域内,混沌就会生效。

在通过混沌搜索得到混沌变量序列后,需要将混沌遍历序列按照下式变换到原来的解空间中进行适应度值评估。

$$x_i'=\frac{1}{2}(b_i-a_i)*y_i+\frac{1}{2}(b_i+a_i)$$

其中, a_i 与 b_i 分别代表烟花 x_i 搜索的上边界与下边界。若在此过程中搜索到更优的解,则用更优的烟花向量序列 x_i' 代替原本的烟花 x_i 。

3.3 求解 TSP 的混沌烟花算法

因为 TSP 问题是离散空间的非数值优化问题,所以在运

用烟花算法求解 TSP 问题时需要在连续空间的粒子与离散空间的路径解之间建立对应关系,即将烟花算法离散化。本文设计了一种结合随机键与 TSP 路径的编码方式,结合最大位置法 LOV(Largest Order Value)^[17]对烟花内变量序列进行排序编码。先运用 n 个随机键编码进行快速排序得到每个随机键位置的大小顺序,再用 n 个位置值表示每个城市的位置矢量,随机键位大小顺序对应城市经过顺序的位置,即可得到经过 n 个城市先后顺序的一个排列。在这种城市顺序排列中每个城市均有且仅有出现一次,可以隐形地保证 TSP 问题的约束。例如:一个烟花个体可以表示为 $X=\{x_1,x_2,x_3,x_4,x_5\}$, x_1,x_2,x_3,x_4,x_5 皆为随机数,若经过快速排序可得随机键位置的大小顺序 $\hat{X}=\{5,2,1,4,3\}$,则其对应的 TSP 路径解为 $\{P_3,P_2,P_5,P_4,P_1\}$,其中 P_i 为城市代号, $i\in[1,5]$ 。转换出来的 TSP 路径解可用于计算 TSP 解的总路径与每个烟花的适应度值。

通过 3.2 节对于混沌优化策略的分析,在烟花算法的寻优过程中加入混沌优化策略,具体步骤如下。

Step1 初始化算法中烟花个数 N 、爆炸火花数、烟花爆炸半径 A 、高斯变异火花数 M 、烟花序列上下界等基本参数。

Step2 初始化一个烟花位置,结合混沌生成策略生成其余 $(N-1)$ 个烟花位置。根据上述排序编码的离散规则结合评估函数计算每个原始烟花的适应度值。

Step3 根据式(3)、式(4)计算爆炸火花数目和爆炸半径,通过 N 个原始烟花产生爆炸火花和高斯变异火花。

Step4 计算新产生的爆炸火花和高斯变异火花的适应度值,根据选择策略从原始烟花、爆炸火花和高斯变异火花中选择 N 个烟花作为下一代的预备原始烟花。

Step5 每个预备原始烟花利用 3.2 节中的混沌生成策略进行混沌搜索,若在搜索中发现更优的烟花,则更新该烟花。

Step6 判断是否满足迭代停止条件。若否,则进入 Step3;若是,则停止迭代并输出最佳烟花与其适应度值。

4 实验与分析

4.1 参数分析

为了提高烟花算法求解 TSP 问题的性能,本文选取国际通用的 TSP 数据库(The Library of TSP, TSPLIB)中的 Oliver30, Att48 和 Eil51 标准算例进行参数测试。本文算法测试采用 Visual Studio 2015 编程,在 CPU 为 Intel 3.00 GHz, RAM 为 4GB 的计算机上进行实验。

本文采取单因素分析方法来测定算法内部参数对于算法求解性能的影响,即选取一个参数因子作为变量,其他参数因子保持不变,多次测定记录平均值,从而测定该参数因子对于算法的影响。在烟花算法中内部可调的参数有初始化烟花数 N 、爆炸火花数 S_{sum} 、高斯变异火花数 M 和爆炸半径 A 。本文实验通过算法运算结果中最优值与平均值的质量来判断该参数的优劣,即在 TSP 问题中最优值与平均值越小,则表明该参数越准确。

参数测试实验结果如表 1—表 4 所列。由表 1 可见,随着初始化烟花 N 增加,实验中最优值与平均值均呈现单调上升的趋势,所以对比其他参数实验结果可得初始化烟花数 N 为 5 时最为合理。表 2 的数据表明,在 Oliver30 标准问题测

试中,当爆炸火花数 S_{sum} 为 200 时算法表现最佳;在 Att48 与 Eil51 的测试中,爆炸火花数 S_{sum} 为 250 时最优值与平均值仅略优于 S_{sum} 为 200 的情况。因此综合考虑,爆炸火花数 S_{sum} 取 200 时整体上是最优的。从表 3 可看出,当高斯变异火花数 M 增加时,实验中最优值与平均值均呈单调下降趋势,所以选择高斯变异火花数 M 为 70 时为最优参数。同样在表 4 中当爆炸半径 A 增加时,在 3 个标准集测试中最优值与平均值均呈现变优的趋势,所以在爆炸半径 A 取值为 2 000 时为合理。

表 1 初始化烟花数 N 的参数实验对比结果

N	Oliver30		Att48		Eil51	
	Best	Avg	Best	Avg	Best	Avg
5	415	487.4	40237	44825.9	503	577.5
10	497	543.2	47514	59703.9	665	734.0
15	526	576.0	48486	59169.0	629	722.4
20	524	608.3	55790	61556.1	668	718.5

表 2 爆炸火花数 S_{sum} 的参数实验对比结果

S_{sum}	Oliver30		Att48		Eil51	
	Best	Avg	Best	Avg	Best	Avg
100	442	470.7	40900	45551.3	515	596.7
150	415	490.3	39353	44756.1	507	560.6
200	415	473.7	37889	42384.6	510	551.4
250	455	479.8	37412	41965.2	507	525.4

表 3 高斯变异火花数 M 的参数实验对比结果

M	Oliver30		Att48		Eil51	
	Best	Avg	Best	Avg	Best	Avg
10	437	503.3	40219	46063.3	504	597.0
30	442	497.4	43620	45225.6	514	577.2
50	460	504.4	42145	43718.0	517	543.0
70	425	478.4	41251	44220.9	505	546.3

表 4 爆炸半径 A 的参数实验对比结果

A	Oliver30		Att48		Eil51	
	Best	Avg	Best	Avg	Best	Avg
500	438	501.8	41258	47836.9	543	594.7
1 000	462	503.4	39426	47860.3	556	613.4
1 500	458	499.6	40794	47711.2	524	566.6
2 000	455	487.4	39996	46854.9	461	566.2

对比实验结果可知,基本烟花算法与混沌烟花算法的测试参数可设置为:初始化烟花数 $N=5$,爆炸火花数 $S_{sum}=200$,爆炸半径 $A=2000$,高斯变异火花数 $M=70$,迭代停止条件为最优烟花适应度值迭代不改变次数大于或等于 500,每组算例测试次数为 150。

4.2 对比实验与分析

为了验证算法性能,本文将 CFWA、粒子群算法(Particle Swarm Optimization, PSO)^[7,12]、遗传算法(Genetic Algorithm, GA)^[6,11]和基本烟花算法(Fireworks Algorithm, FWA)^[13]进行对比,测试混沌烟花算法的寻优性能。

表 5 给出了不同算法的在解决标准 TSPLIB 问题时性能的比较结果。遗传算法(GA)采用文献[21]的测试参数:染色体个数 $N=30$,交叉概率 $P_c=0.2$,变异概率 $P_m=0.5$;粒子群算法(PSO)参考文献[22]的设置,粒子数为 30。FWA 代表基本烟花算法。CFWA 是指加入混沌优化策略的烟花算法。从表 5 中可以看出,基本烟花算法在计算 3 个标准算例问题中最优解与平均解均优于基本粒子群算法所得出的解。整体而言,基本烟花要优于基本粒子群算法和遗传算法。相

比基本遗传算法,基本烟花算法在 Oliver30 上寻到了最优值,而且在 Oliver30 和 Att48 的标准问题中最优解得到了更好的结果。实验结果也证明了基本烟花算法具有较好的局部搜索与全局搜索能力。

表 5 不同算法的对比结果

算法	Oliver30		Att48		Eil51	
	Best	Avg	Best	Avg	Best	Avg
GA ^[22]	467.7	483.5	38541	38732	—	—
PSO ^[7,23-24]	527.5	571.1	49778	52377	571.6	601.2
FWA	415.0	483.5	35931	44615	452.0	548.4
CFWA	415.0	476.0	35108	43150	451.0	544.4

表 5 中的数据也显示了相比基本烟花算法,加入混沌优化策略后的烟花算法在解决 TSP 问题时的最优解与平均解都有不同程度的提高。混沌烟花算法相比基本烟花算法在 Att48 标准案例中最优值误差减少 2.4%,在平均值误差方面 3 个标准算例约减少 1%~4%。此测试结果验证了将混沌优化策略融合到基本烟花算法中可以一定程度地提高算法的收敛性、全局搜索能力和解的精确性。

在以上分析结果表明,烟花算法不仅能有效求解 TSP 问题,而且相对其他算法而言有着较高的鲁棒性。在基本烟花算法中加入混沌优化策略后,能有效提高算法求解精度以及全局搜索能力。因此本文提出的混沌烟花算法为求解 TSP 问题提供了一种较优的思路。

结束语 本文提出了一种混沌烟花算法(CFWA)来求解 TSP 问题。结合 TSP 的特点,CFWA 使用 LOV 定义离散域中的烟花算法,并加入混沌优化策略来增强算法搜索能力。本文设计了 4 个参数实验来分析主要参数对 CFWA 的影响并确定了较优的参数设置。对比实验表明:相比于对比算法,混沌烟花算法求解旅行商问题时具有较好的收敛性和稳定性。

参 考 文 献

[1] 孙惠文. 遗传算法求解旅行商问题[J]. 西南交通大学学报, 1996,31(5):550-554.

[2] 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.

[3] 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.

[4] 张涛,王梦光. 遗传算法和 3-opt 结合求解带有能力约束的 VRP [J]. 东北大学学报(自然科学版),1999,20(3):254-256.

[5] LIN S,KERNIGHAN B W. An Effective Heuristic Algorithm for the Traveling-Salesman Problem[M]. INFORMS,1973.

[6] 孟繁桢,胡云昌,徐慧,等. 旅行商问题的遗传算法[J]. 系统工程理论与实践,1997,17(9):15-21.

[7] 程毕芸,鲁海燕,徐向平,等. 求解旅行商问题的改进局部搜索混沌离散粒子群优化算法[J]. 计算机应用,2016,36(1):138-142.

[8] 李擎,张超,陈鹏,等. 一种基于粒子群参数优化的改进蚁群算法[J]. 控制与决策,2013(6):873-878.

[9] 戚远航,蔡延光,蔡颢,等. 旅行商问题的混沌混合离散蝙蝠算法[J]. 电子学报,2016,44(10):2543-2547.

[10] 吴华锋,陈信强,毛奇凰,等. 基于自然选择策略的蚁群算法求解 TSP 问题[J]. 通信学报,2013(4):165-170.

[11] 冯春松,王军宇,周松盛,等. TSP 问题的一种改进遗传算法[J]. 武汉理工大学学报,2006,28(4):116-118.

[12] 高尚,韩斌,吴小俊,等. 求解旅行商问题的混合粒子群优化算法[J]. 控制与决策,2004,19(11):1286-1289.

[13] 谭营,郑少秋. 烟花算法研究进展[J]. 智能系统学报,2014(5):515-528.

[14] 朱启兵,王震宇,黄敏. 带有引力搜索算子的烟花算法[J]. 控制与决策,2016,31(10):1853-1859.

[15] 韩守飞,李席广,拱长青. 基于模拟退火与高斯扰动的烟花优化算法[J]. 计算机科学,2017,44(5):257-262.

[16] 张以文,吴金涛,赵姝,等. 基于改进烟花算法的 Web 服务组合优化[J]. 计算机集成制造系统,2016,22(2):422-432.

[17] QIAN B,WANG L,HUANG D X,et al. An effective hybrid DE-based algorithm for flow shop scheduling with limited buffers[J]. Computers & Operations Research,2008,35(9):2791-2806.

[18] ZHENG S,JANECEK A,LI J,et al. Dynamic search in fireworks algorithm[C]// Evolutionary Computation. IEEE,2014:3222-3229.

[19] 曹磊,叶春明,黄霞. 应用混沌烟花算法求解置换流水车间问题[J]. 计算机应用与软件,2016,33(11):188-192.

[20] 包晓晓,叶春明,黄霞. 烟花算法求解 JSP 问题的研究[J]. 计算机工程与应用,2017,53(3):247-252.

[21] 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.

[22] 江新姿,高尚,陈建忠. 求解旅行商问题的模拟退火蚁群算法[J]. 计算机工程与设计,2008,29(6):1491-1493.

[23] WANG K P,HUANG L,ZHOU C G,et al. Particle swarm optimization for traveling salesman problem [C] // International Conference on Machine Learning and Cybernetics. IEEE,2003:1583-1585.

[24] 朱小平,赵曦. 一种改进的离散粒子群优化算法在 TSP 问题中的应用[J]. 江西师范大学学报(自然版),2010,34(4):369-373.