# 求解 TSP 问题的离散型萤火虫群优化算法

周永权[1,2],黄正新[1,3],刘洪霞[1]
(1.广西民族大学信息科学与工程学院,广西南宁 530006;
2.广西混杂计算与集成电路设计分析重点实验室,广西南宁 530006;3.右江民族医学院网络中心,广西百色 533000)

**摘 要:** 基于求解 TSP 问题,提出一种离散型萤火虫群优化(DGSO)算法,该算法结合 TSP 问题特点,给出一种有效编码和解码方法,并定义适合编码的个体间距离计算公式和编码更新公式.同时,为增强算法求解 TSP 问题的局部搜索能力,加快算法的收敛速度,算法使用了操作简单的 2-Opt 优化算子.最后,通过对 10 个 TSP 问题进行仿真实验,实验结果表明本文提出的算法是在种群规模较小,迭代次数较少的情况下就可以收敛到已知最优解.在大规模 TSP 算例中算法获得的最优值与理论最优值的误差也在 1%以下.

**关键词:** 萤火虫群优化算法;离散萤火虫群算法;TSP 问题;2-Opt
**中图分类号:** TP183 **文献标识码:** A **文章编号:** 0372-2112 (2012)06-1164-07
**电子学报 URL:** http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.06.016

# Discrete Glowworm Swarm Optimization Algorithm for TSP Problem

ZHOU Yong-quan[1,2], HUANG Zheng-xin[1,3], LIU Hong-xia[1]
(1. *College of Information Science and Engineering, Guangxi University for Nationalities, Nanning, Guangxi* 530006, *China*;
2. *Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Nanning, Guangxi* 530006, *China*;
3. *Network Center, Youjiang Medical University for Nationalities, Baise, Guangxi* 533000, *China*)

**Abstract:** A discrete glowworm swarm optimization (DGSO) algorithm is designed to tackle the travelling salesman problem. A new encoding schema and decoding schema are given with the characteristics of the TSP problem, and a new distance formula and encoding update formula for the new algorithm are given. In order to enhance the capability of the algorithm local searching, and to speed up the algorithm convergence speed, the 2-opt local search scheme is integrated into the new algorithm for solving TSP problem. The proposed algorithm was evaluated on 10 TSP test problems. The numerical experiments show that the proposed algorithm can find the global optimal solution with less computation and evolving time. In case of large scale TSP algorithm can achieve optimal solution of the theory and the error of the optimal solution is also less than 1%.

**Key words:** discrete glowworm swarm optimization algorithm; glowworm swarm optimization; TSP; 2-Opt

## 1 引言

旅行商问题(Traveling Salesman Problem,TSP)是一个典型的 NP 难问题之一.它在计算机科学、运筹学及工程优化等领域都有着广泛地应用.近年来,人们从仿生学机理中受到启发,提出了许多种用于求解 TSP 问题的群智能优化算法,如人工神经网络(Artificial Neural Networks,ANN)、遗传算法(Genetic Algorithm,GA)、蚁群优化(Ant Colony Optimization,ACO)算法、粒子群优化(Particle Swarm Optimization,PSO)算法和人工免疫算法(Artificial Immune Algorithm,AIA)等.这些群智能算法虽不能保证在有限时间内获得 TSP 问题的最优解,但通过随机地搜索选择多个候选解"验证"后,错误概率会降到令人满意的地步,目前已成为人们求解 TSP 问题的最有效方法.

萤火虫群优化(Glowworm Swarm Optimization,GSO)算法是印度学者 K N Krishnanand 和 D Ghose 于 2005 年提出的一种新型仿生群智能优化算法[6].该算法思想源于自然界的萤火虫群中的萤火虫通过发光来吸引伴侣求偶或者觅食,并且越亮的萤火虫吸引力越大,最后,大多数萤火虫聚集在多个位置上的自然现象.GSO 算法与其它群智能算法一样,虽采用基于种群的多点并行全局随机搜索策略,无需复杂的进化操作,而是根据萤火虫个

收稿日期:2011-02-15;修回日期:2012-01-21
基金项目:国家自然科学基金(No.61165015);广西自然科学基金重点项目(No.2012GXNSFDA053028);智能感知与图像理解教育部重点实验室开放基金(No.IPIU012011001)

体的感知范围来决定搜索范围及路径.与目前已有的群智能算法,如 GA、PSO、AIA 等相比,GSO 算法的不同在于萤火虫种群中每一个萤火虫本身都携带一定数量的荧光素,且可随机地更新;适应度函数定义为当前萤火虫所处的位置函数的值来评价个体的优劣;萤火虫的移动在其感知决策半径范围内朝着荧光素大,即萤火虫亮度高的萤火虫移动;接着又对感知决策半径范围动态地修改,继续朝着荧光素大的萤火虫移动,最终实现所有萤火虫聚集到亮度最亮的萤火虫所处在的位置上.

目前,GSO 算法已在多模态函数优化、多信号源追踪、多信号源定位、集群机器人学和有害气体泄漏定位等方面得到成功地应用,且表现出了良好的优化性能[7~11].GSO 算法逐渐成为群智能技术研究的热点方向之一.随着人们对 GSO 算法的研究的不断深入,无论是应用层面上研究成果,还是理论方面的探讨,作者通过分析发现目前人们所用的 GSO 算法仅适用于连续型论域中的目标优化问题.

基于此想法,以求解著名的 TSP 问题为例,提出一种离散型萤火虫群优化(Discrete Glowworm Swarm Optimization,DGSO)算法,该算法一方面结合 TSP 问题本身的特点,提出一种有效的新编码方法,给出其相应的解码方法,并定义适合编码的个体间距离计算公式和编码更新公式.另一方面,为了增加解的多样性,加快算法的收敛速度,在 DGSO 算法中,我们引入了简单的 2-Opt[12]算子作为局部优化方法来改进每次进化所得到的最优解,从而进一步扩大其解搜索范围,加快算法收敛速度,提高算法求解 TSP 问题的全部搜索能力.最后,为了验证算法的整体优化性能,我们对 10 个典型的 TSP 问题算例进行了计算机仿真实验,实验结果表明,在中小规模 TSP 问题中算法能够在较少的代数内找到最优解;在大规模 TSP 中计算出的最优值与已知最优值的误差精度在 1%以下,比目前同类文献算法给出结果更优.

## 2　萤火虫群优化算法

在自然界,萤火虫发光越亮越绚丽,越能吸引同伴求偶或觅食,GSO 算法是模拟基于这种自然现象设计出来的一种新型仿生群智能算法.GSO 算法最大的优势在于可同时捕获多模态函数的所有极值点[6~10].一般用 GSO 算法优化多模态函数时,首先,萤火虫群个体被随机分布到问题可行解空间,然后根据其邻域内其它个体发出的信号(荧光素)强度大小选择移动.一般地,GSO 优化算法包括以下 5 个步骤:

**步骤 1**　使用下面公式(1)把萤火虫 $i$ 在 $t$ 时刻的位置 $x_i(t)$ 对应的目标函数值 $J(x_i(t))$ 转化为荧光素值 $l_i(t)$;

**步骤 2**　每只萤火虫在其动态决策域半径 $r_d^i(t)$ 内,选择荧光素值比自己大的个体组成其邻域集 $N_i(t)$;

**步骤 3**　用公式(2)计算个体 $i$ 移向其邻域集内个体 $j\in N_i(t)$ 的概率 $p_{ij}(t)$;

**步骤 4**　选择移动对象,进行移动,根据公式(3)更新位置;

**步骤 5**　根据公式(4)更新动态决策域半径的值.

$$l_i(t)=(1-\rho)l_i(t-1)+\gamma J(x_i(t)) \tag{1}$$

$$p_{ij}(t)=\frac{l_j(t)-l_i(t)}{\sum_{k\in N_i(t)}l_k(t)-l_i(t)} \tag{2}$$

$$x_i(t+1)=x_i(t)+s*\left(\frac{x_j(t)-x_i(t)}{\|x_j(t)-x_i(t)\|}\right) \tag{3}$$

$$r_d^i(t+1)=\min\{r_s,\max\{0,r_d^i(t)+\beta(n_t-|N_i(t)|)\}\} \tag{4}$$

其中,参数 $t$ 为迭代次数;$\rho\in[0,1]$ 为荧光素挥发因子;$\gamma\in[0,1]$ 为荧光素更新率;$s$ 为移动步长;$r_s$ 为萤火虫个体的感知最大半径;$\beta$ 为动态决策域更新率;$n_t$ 为个体邻域集内包含的萤火虫种群数目的阈值.

目前有关 GSO 算法参数值的设置方法,还没有数学理论依据,Krishnanand 和 Ghose 通过大量的数值仿真实验对 GSO 算法中各参数取值范围做了较全面的数值仿真研究[7],从获得所测试的目标函数所有极值点因素考虑,最后给出了 GSO 算法中各参数参考取值分别为:$\rho=0.4,\gamma=0.6,\beta=0.08,n_t=5,s=0.03,l_0=5$.这些参数值为人们在应用 GSO 算法解决实际问题带来了方便.

## 3　求解 TSP 问题的离散型萤火虫群优化算法

鉴于目前 GSO 算法仅适用于连续论域中的优化问题,本文以 TSP 问题为例,提出离散的萤火虫群优化(DGSO)算法.在求解 TSP 问题的离散萤火虫群优化算法中,萤火虫个体 $i$ 在 $t$ 时刻的位置 $x_i(t)$ 表示一条可行路径的编码,位置移动对应于路径编码的一种变换.

### 3.1　编码与解码方法

#### 3.1.1　编码方法

对包含 $n$ 个城市的 TSP 问题,我们提出一种新的易理解的有效编码方法.一个可行编码是指整数 1 到 $n$ 的一个全排列所构成的序列$(a_1,\cdots,a_i,\cdots,a_n)$,其中 $a_i\in[1,n]$ 之间的整数.编码中 $a_i$ 表示编号为 $i$ 的城市排在路径上的第 $a_i$ 个位置.

**例 1**　设对于规模为 5 的 TSP 问题,一个可行编码为 3,1,2,5,4,表示编号为 1 的城市排在路径上的第 3

个位置,编号为 2 的城市排在路径上的第 1 个位置,编号为 3 的城市排在路径上的第 2 个位置,编号为 4 的城市排在路径上的第 5 个位置,编号为 5 的城市排在路径上的第 4 个位置.周游城市的顺序为:2→3→1→5→4.

一个条可行路径是整数 1 到 $n$ 的一个全排列.即任意一条可行路径 $p_1p_2\cdots p_n$,其中 $p_i\in[1,n]$之间的整数,对路径编号按升序排列,得到对应编号的下标所构成的序列 $a_1,\cdots,a_i,\cdots,a_n$ 就是该路径的编码.

**例 2** 设对于规模为 5 的 TSP 问题,一条可行路径为 15423,对路径编号进行升序排列得到 12345,其对应的下标构成的序列为(1,4,5,3,2),即序列(1,4,5,3,2)就是路径 15423 对应的编码.

### 3.1.2 解码方法

任意一个可行编码$(a_1,\cdots,a_i,\cdots,a_n)$,对编码数字进行升序排列,所得到对应数字下标构成的排列 $p_1p_2\cdots p_n$ 为该编码对应的路径.

**例 3** 设对于规模为 5 的 TSP 问题,一个可行编码为(3,1,2,4,5),对编码数字进行升序排列得到 12345,其对应的下标构成的排列是 23145,即 23145 就是可行编码(3,1,2,4,5)对应的路径.

**定理 1** 所有可行路径组成的集合与所有可行编码组成的集合之间存在着一一对应的关系.

证明:因为所有可行路径和可行编码都是整数 1 到 $n$ 的一个全排列,对整数 1 到 $n$ 的任意一个全排列,进行升序排列,得到的下标总是唯一的.也就是说,任意一条可行路径按编码方法存在唯一的一个可行编码与它相对应;任意一个可行编码按解码方法存在唯一的一条可行路径与它相对应.所以,所有可行路径组成的集合与所有可行编码组成的集合之间存在着一一对应的关系.

## 3.2 个体间距离计算公式

设个体 $i,j$ 在 $t$ 迭代表示的可行编码分别为序列:$x_i(t)=(x_{i1},x_{i2},\cdots,x_{in})$和 $x_j(t)=(x_{j1},x_{j2},\cdots,x_{jn})$,则个体 $i,j$ 在 $t$ 迭代的编码的差异度 $\delta_{ij}(t)$计算公式定义为:

$$\delta_{ij}(t)=\frac{\sum_{k=1}^{n}|d_{ij}(t,k)|}{M} \tag{5}$$

其中,$d_{ij}(t)=x_j(t)-x_i(t)$,$|*|$表示求绝对值运算,$M$ 是 $\sum_{k=1}^{n}|d_{ij}(t,k)|$取值的上限,当城市规模 $n$ 为奇数时,$M=(n-1)\times\frac{(n+1)}{2}$;当城市规模 $n$ 为偶数时,$M=(n-1)\times\frac{n}{2}$.由式(5)可知 $\delta_{ij}\in[0,1]$,且个体 $i,j$ 之间编码差异度满足对称性,即 $\delta_{ij}=\delta_{ji}$.

**例 4** 对于规模为 5 的 TSP 问题,设个体 $i,j$ 在 $t$ 迭代表示的可行编码序列分别为 $x_i(t)=(2,4,1,5,3)$和 $x_j(t)=(3,1,5,4,2)$,则个体 $i,j$ 在 $t$ 迭代表示编码的差异度 $\delta_{ij}(t)$计算如下:

$$d_{ij}(t)=x_j(t)-x_i(t)=(3,1,5,4,2)-(2,4,1,5,3)$$
$$=(1,-3,4,-1,-1)$$

因 5 为奇数,$M=(n-1)\times\frac{(n+1)}{2}=4\times\frac{6}{2}=12$.

$$\delta_{ij}(t)=\delta_{ji}(t)=\frac{\sum_{k=1}^{5}|d_{ij}(t,k)|}{M}$$
$$=\frac{|1|+|-3|+|4|+|-1|+|-1|}{12}=\frac{5}{6}.$$

所以,基于上述编码的差异度计算,可定义萤火虫个体 $i$ 与个体 $j$ 在 $t$ 迭代距离 $D_{ij}(t)$计算公式为:

$$D_{ij}(t)=c\times\delta_{ij}(t) \tag{6}$$

其中,$c$ 为常数,$\delta_{ij}(t)$为个体 $i,j$ 在 $t$ 迭代表示的编码的差异度.由式(6)知,$D_{ij}\in[0,c]$,且个体间距离满足对称性,即 $D_{ij}=D_{ji}$.参数 $c$ 取值太大或太小都不利于邻域集更新,结合实验,本文参数 $c$ 取值为 20.

## 3.3 编码更新

在以下给出的 DGSO 算法中,个体编码更新时忽略步长的具体数字和固定的更新式子,而是以一定的概率选择一个更新式子进行更新.对于一个城市规模为 $n$ 的 TSP 问题,萤火虫个体表示的编码是一个 $n$ 维序列,在编码更新时,编码的每一维数据以一定的概率选择更新式子来进行更新,具体的更新公式为:

$$x_i(t+1,k)=\begin{cases}x_i(t,k), & \text{if} \quad r(k)<p_1\\ x_j(t,k), & \text{else if} \quad r(k)<p_2\\ x_j(t,k)+R, & \text{else}\end{cases} \tag{7}$$

其中,$r$ 是从 0 到 1 之间随机产生的 $n$ 维序列,且 $r(k)\in\{r_1,\cdots,r_k,\cdots,r_n\}$,$k\in\{1,2,\cdots,n\}$;$j$ 是个体 $i$ 在 $t$ 迭代选择的移动的对象;$p_1,p_2$ 为更新式子选择参数,且 $p_1,p_2\in[0,1]$;$R$ 为扰动项,是从 -1 到 1 之间随机产生的整数.由式(7)可知,个体 $i$ 在 $t$ 迭代编码更新时是以概率 $p_1$ 保留其编码第 $k$ 维上的数据,以概率$(p_2-p_1)$转移到个体 $j$ 编码的第 $k$ 维数据,以概率$(1-p_2)$向个体 $j$ 编码的第 $k$ 维数据临近值随机变化.结合实验,更新式子选择参数 $p_1$ 和 $p_2$ 取值分别为 0.85 和 0.9.

## 3.4 不可行编码处理

进行编码更新后,可能出现多个城市编号排在同一个位置或 $a_i$ 超出$[1,n]$之间整数的情况.对于这种情况,我们先对编码数字按 Matlab 默认排列方法进行升序排列,再对位置重叠的城市编号按它们的 $d_{ij}(t)$值升序重新排列,得到新的路径,最后对新路径按编码方法转化为编码.

**例 5** 对于包含 5 个城市的问题,假设个体 $i$ 在 $t$

迭代表示的编码为(1,2,4,5,3),它选择的移动对象个体 $j$ 表示的编码为(1,4,2,3,5),随机产生的序列 $r$ = (0.1,0.86,0.5,0.4,0.95),参数 $p_1$,$p_2$ 取值分别为 0.85 和 0.9,按公式(7)更新方式如下:

由 $r(1)=0.1<p_1$,所以 $x_i(t+1,1)=1$;

由 $p_1<r(2)=0.86<p_2$,所以 $x_i(t+1,2)=4$;

由 $r(3)=0.5<p_1$,所以 $x_i(t+1,3)=4$;

由 $r(4)=0.4<p_1$,所以 $x_i(t+1,4)=5$;

由 $r(5)=0.95>p_2$,设随机产生的整数 $R=1$,则 $x_i(t+1,5)=5+1=6$.更新得到的编码:$x_i(t+1)=(1,4,4,5,6)$,由编码知编号为 2、3 的城市都排在路径上的第 4 个位置,编号为 5 的城市都排在路径上的第 6 个位置是不可行的编码.对于这种情况,我们先对 $x_i(t+1)$ 按 Matlab 默认排列方法进行升序排列,得到的路径为 12345,再对编号为 2、3 的城市按 $d_{ij}(t)$ 值升序重新排列.因为 $d_{ij}(t)=(0,2,-2,-2,2)$,$d_{ij}(t,3)<d_{ij}(t,2)$,所以,先排城市 3,再排城市 2.处理后的最终路径为 13245.若 $d_{ij}(t)$ 值也相等,可随机选择一种排列作为最终路径.最后对路径 13245 按编码方法转化为编码,最终得 $x_i(t+1)=(1,3,2,4,5)$.更新前个体 $i$ 的编码与个体 $j$ 的编码的差异度为 $\frac{2}{3}$,更新后个体 $i$ 的编码与个体 $j$ 的编码的差异度为 $\frac{1}{6}$.

### 3.5 适应度函数

适应度函数 $f(x_i(t))$ 是对个体 $i$ 在 $t$ 迭代的编码 $x_i(t)$,按解码方法求出其对应路径的长度的倒数.

### 3.6 路径初始化

通常人们对于求解 TSP 问题有这样的一个共识:最佳路径的选取,必然包括而且在很大程度上包括相邻城市间最短的路径[13].参照这些研究经验,本文据城市间的距离,我们采用轮盘赌法来初始化初始路径,这样产生的初始路径已经比较接近问题的解,因此可以节省搜索时间,提高算法收敛速度.

### 3.7 局部路径优化算子

分析目前求解 TSP 问题时,人们对路径优化策略,求解 TSP 问题的算法分为两种类型:局部启发式搜索算法和独立于问题的经典优化算法.常见局部启发式优化算法有 2-Opt,3-Opt,和 Lin-Kernighan(LK)等[12,14],其中用 2-Opt 优化路径对于寻找 TSP 的局部最优解显得非常奏效,但这些算法过于依靠问题本身特性,因而易陷入局部最优.独立于问题的优化算法,如蚁群算法、遗传算法等,如果时间不受限制的话,在理论上这类算法终将找到最优解.但对于稍大规模的 TSP 问题,时间代价开销大.研究结果说明将这两种算法结合起来可以有效地提高求解 TSP 问题的质量,这类算法在提高收敛速度和寻求全局最优解之间得到较好的平衡[13,15].所以,本文使用操作简单的 2-Opt 作为算法的局部优化算子,对运算得到的新路径进行局部优化.

### 3.8 Complete 2-Opt(C2Opt)算子

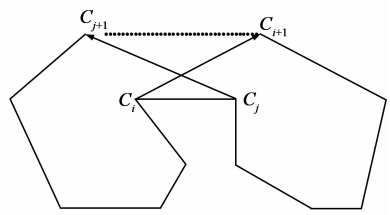

图1　使用C2Opt算子前

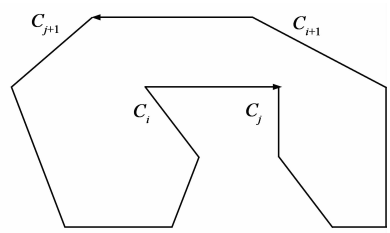

图2　使用C2Opt算子后

在遗传算法求解 TSP 问题中,通常随机选择路径上 20 ~ 30% 的城市用 2-Opt 进行局部优化,C2Opt 是对构成路径的所有城市进行 2-Opt 优化.下面以图 1 和图 2 为例,说明 C2Opt 算子的操作过程.

设 $c_k(k=0,1,\cdots,n-1)$ 表示城市所在顶点,$d(c_i,c_j)$ 表示任意两个城市 $c_i$ 和 $c_j$ 之间的距离.C2Opt 算子实现的步骤描述如下[12]:

**步骤 1**　选取一路径 $c=\{c_0,\cdots,c_i,c_{i+1},\cdots,c_i,c_{i+1},\cdots,c_{n-1}\}$(如图 1 所示).开始时,令 $i=j=0$;

**步骤 2**　选取一条边,标记为 No.1:$(c_i,c_{i+1})$,其中,$i<n$.

**步骤 3**　选取另一条边,标记为 No.2:$(c_j,c_{j+1})$,其中 $j<n$.

**步骤 4**　若 $|j-(i+1)|\geqslant 2$ 且 $d(c_i,c_j)+d(c_{i+1},c_{j+1})<d(c_i,c_{i+1})+d(c_j,c_{j+1})$,则用 2-Opt 算子删除边 $(c_i,c_{i+1})$ 和 $(c_j,c_{j+1})$;然后,分别连接边 $(c_i,c_j)$ 和边 $(c_{i+1},c_{j+1})$,且分别以相反箭头指向顶点 $c_{i+1}$ 和 $c_j$;

**步骤 5**　以顶点 $c_j$ 作为 No.2 边遍历开始的城市,置 $j=j+1$,重复执行步骤 3 和步骤 4,直到 $j=n-1$;若 $j=n-1$,则 $j+1=0$;

**步骤 6**　以顶点 $c_i$ 作为 No.1 边遍历开始的城市,置 $i=i+1$,重复执行步骤 2 ~ 步骤 5 直到 $i=n-1$;若 $i=n-1$,则 $i+1=0$;

**步骤 7**　重复执行步骤 2 ~ 步骤 6 直到 $N$ 次;使 $N$ 足够大,直到所选取路径无交叉边出现为止(如图 2 所示).

### 3.9 算法描述

综上所述,本文提出的求解 TSP 问题的 DGSO 算法可描述如下:

**步骤 1**　设置种群大小,最大迭代次数,初始萤光素值 $l_i(0)$,初始动态决策域半径 $r_d^i(0)$,感知最大半径 $r_s$,荧光素挥发因子 $\rho$,荧光素更新率 $\gamma$,动态决策域半

径更新率 $\beta$,个体邻域集内包含的萤火虫数目阈值 $n_t$,据轮盘赌法初始化萤火虫初始路径,更新式子选择参数 $p_1$,$p_2$;

**步骤 2** 计算各路径适应度值,并根据式(1)将适应度转化为萤火虫荧光素值;

**步骤 3** 按编码方法将路径转化为编码,根据式(5)和(6)计算个体间距离,在其动态决策域半径 $r_d^i(t)$ 内,选择荧光素值比自己高的个体组成其邻域集 $N_i(t)$;

**步骤 4** 根据式(2)计算个体移向其邻域集内个体 $j\in N_i(t)$ 的概率 $p_{ij}(t)$,依概率大小采用轮盘赌法选择移动对象;

**步骤 5** 从 0 到 1 之间随机产生的一个 $n$ 维的序列 $r=(r_1,\cdots,r_k,\cdots,r_n)$,由 $r_k$ 大小,根据式(7)更新编码 $x_i(t)$ 每一维的数据;

**步骤 6** 对更新后的不可行编码 3.4 节方法处理;

**步骤 7** 对得到的新路径用 2-Opt 算子进行局部优化;

**步骤 8** 根据式(4)更新萤火虫个体动态决策域半径;

**步骤 9** 判断当前运行代数是否达到设定运行代数,如未达到,跳转到步骤 2,继续执行;否则,终止寻优,输出全局最优值和全局最优路径.

## 4 仿真实验及结果

为了验证本文算法的有效性和正确性,我们选用 10 个 TSP 问题算例(城市规模从 14 到 200)进行仿真实验.实验软件 Matlab 2008a, CPU 为 Intel2.20GHz, 内存 1024MB, Windows XP 操作系统.

### 4.1 算法参数取值

在计算机仿真实验中,所有算例的种群规模设置为 100,最大迭代次数为 200;萤火虫个体的初始荧光素值为 5;萤火虫个体的初始动态决策域半径为 4;萤火虫个体的感知最大半径 $r_s=20$;荧光素挥发因子 $\rho=0.4$;荧光素更新率 $\gamma=0.6$;动态决策域半径更新率 $\beta=0.08$;萤火虫个体邻域集内包含的萤火虫数目阈值 $n_t=5$,更新式子选择参数 $p_1$ 和 $p_2$ 取值分别为 0.85 和 0.9.

### 4.2 问题解与对比分析

首先测试文献[16]实验仿真部分采用的 Burma14、Oliver30、Eil51 三个算例.表 1 是两种算法在各测试集上分别计算 20 次得到的最优值,最优值的平均值的比较结果,表中"TSPLIB 路径长度"项是目前 TSPLB 标准库提供的最优路径的长度,"已知最优值"项是目前问题已知最优值,"——"表示该文献没有提供该项数据.

由表 1 数据可知,在求解结果上 DGSO 算法明显优于文献[15]的自适应离散粒子群(SAPSO)算法.对于 Burma14 算例,DGSO 算法与 SAPSO 算法一样在每次运行都能得到问题最优解;对于 Oliver30 算例,DGSO 算法每次运行都得到已知最优值 423.7406,计算出的最优值和平均值分别比 SAPSO 算法小 0.004 和 1.0861;对于 Eil51 算例,虽然两种算法都没有得到已知最优值,但是 DGSO 算法计算得到的最优值和平均值分别比 SAPSO 算法小 7.9012 和 11.3080,找到的最优值为 428.8718,比目前 TSPLB 标准库提供的最优路径长度 429.9833 小 1.1215.总体上,本文算法比 SAPSO 算法求解结果更好,具有更强的稳定性.由于算例 Oliver30 和 Eil51 被广泛用于各种算法的测试中,为了便于比较,图 3 和图 4 分别给出了 DGSO 算法优化两个算例的最优路径图.

**表 1** DGSO 与 SAPSO 求解 Burma14, Oliver30 和 Eil51 算例结果比较

| TSP 问题 | 算法 | 最优值 | 平均值 | 已知最优值(TSPLIB 路径长度) |
|---|---|---|---|---|
| Burma14 | SADPSO | 30.8785 | 30.8785 | 30.8785(30.8785) |
| | DGSO | 30.8785 | 30.8785 | |
| Oliver30 | SADPSO | 423.7410 | 424.8267 | 423.7406(——) |
| | DGSO | 423.7406 | 423.7406 | |
| Eil51 | SADPSO | 436.7730 | 440.7810 | 426(429.9833) |
| | DGSO | 428.8718 | 429.4730 | |

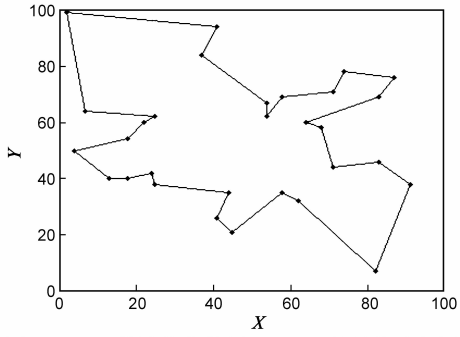


**图3 DGSO算法优化算例Oliver30的最优路径图**

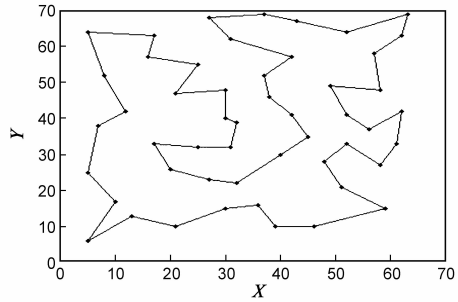


**图4 DGSO算法优化算例Eil51的最优路径图**

表 2 是 DGSO 算法与文献[16]的混合遗传算法的粒子群(HPSO)算法求解更多 TSP 问题实验结果比较.每种算法连续运算各算例 20 次,表中已知最优值项和计算最优值项表示意义与表 1 相同,偏差项是计算最优值与已知最优值的偏差.

观察表 2 数据,从所有算例的计算最优值、偏差和迭代次数上比较,DGSO 算法都比文献[17]的 HPSO 算

法更优.在 7 个算例中,除了 Bays29 算例外,DGSO 算法都比 HPSO 算法找到更优的路径.在城市规模大于等于 48 的 6 个算例中,DGSO 算法优化结果的偏差比 HPSO 算法分别降低了 0.78、1.8、1.7、5.55、6.31 和 8.13,随着问题规模增大,偏差降低更加明显.DGSO 算法使用的迭代次数为 200,仅是 HPSO 算法使用的迭代次数的$\frac{1}{3}$.

**表 2　DGSO 算法与 HPSO 算法求解 TSP 问题结果比较**

| TSP | 已知最优值 | 计算最优值 | | 偏差(%) | | 迭代次数 | |
|---|---|---|---|---|---|---|---|
| 问题 | (TSPLIB 路径长度) | HPSO | DGSO | HPSO | DGSO | HPSO | DGSO |
| Bays29 | 9074.15(9291.35) | 9074 | 9074.15 | 0.00 | 0.00 | 600 | 200 |
| Att48 | 33523(33523.71) | 33784 | 33523.71 | 0.78 | 0.00 | 600 | 200 |
| Pr76 | 108159(108159.44) | 110140 | 108159.44 | 1.80 | 0.00 | 600 | 200 |
| Rrob100 | 22141(—) | 22508 | 22139.07 | 1.70 | 0.00 | 600 | 200 |
| Ch130 | 6110(6110.86) | 6463 | 6125.07 | 5.80 | 0.25 | 600 | 200 |
| Krob150 | 26130(—) | 27865 | 26206.69 | 6.60 | 0.29 | 600 | 200 |
| Krob200 | 29437(—) | 31960 | 29605.13 | 8.60 | 0.57 | 600 | 200 |

同时,由表 2 数据看出,在求解城市规模小于等于 100 的问题中,DGSO 算法均能搜索到目前 TSPLB 标准库提供的最优路径或更好的路径.特别地,在算例 Bays29 和 Krob100 上,DGSO 算法得到的最优值分别为 9074.15 和 22139.07.比算例 Bays29 目前 TSPLB 标准库提供的最优路径长度 9291.35 小 217.20;比算例 Krob100 目前已知最优值 22141 小 1.9300.图 5 是 DGSO 优化算例 Krob100 的最优路径图.在城市规模大于 100 的算例中,虽然 DGSO 算法未必能搜索到已知最优值,但其所得的最优值与已知最优值的误差也在 1% 以下,比 HPSO 算法具有更高的稳定性.由此可见,DGSO 算法具有较好的鲁棒性,可有效的求解 TSP 问题.

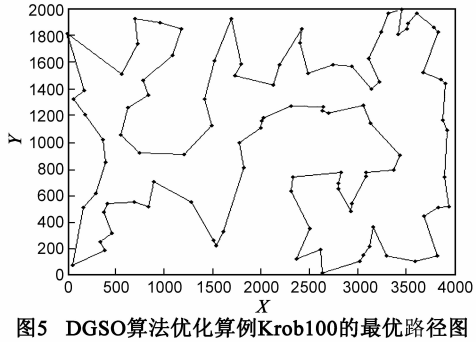


**图5　DGSO算法优化算例Krob100的最优路径图**

## 5　结论

本文以求解 TSP 问题为例,提出了一种离散型萤火虫群优化算法(DGSO),将连续型萤火虫群优化算法推广到离散情形.通过对 10 个 TSP 问题进行了计算机数值仿真实验,实验结果表明,在中小规模算例中算法都能够在较少的代数内找到满意解,在大规模算例中算法计算出的最优值与理论最优值的误差也在 1% 以下,表明本文提出的 DGSO 算法是有效和正确的.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;同时将本文提出的离散萤火虫群优化算法也可应用于求解离散论域的其它复杂优化问题.

## 参考文献


[1] Yong-Hyun Cho. An efficient solving the travelling salesman problem:Global optimization of neural networks by using hybrid method [A]. Traveling Salesman Problem, Theory and Applications [C]. Switzerland: Intech Press, 2010. 155 – 176.

[2] Hirotaka Itoh. The method of solving for travelling salesman problem using genetic algorithm with immune adjustment mechanism [A]. Traveling Salesman Problem, Theory and Applications[C]. Switzerland: Intech Press, 2010. 97 – 112

[3] Kaji T. Approach by ant tabu agents for ttraveling salesman problem[A]. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics [C]. Tucson, AZ, USA: IEEE Press, 2001. 3429 – 3434.

[4] X H Shi, Y C Liang, H P Lee, C Lu, Q X Wang. Particle swarm optimization-based algorithms for TSP and generalized TSP [J]. Information Processing Letters, 2007, 103(5): 169 – 176.

[5] 戚玉涛,焦李成,刘芳.基于并行人工免疫算法的大规模 TSP 问题求解[J].电子学报,2008,36(8):1552 – 1558.
Qi Yutao, Jiao Licheng, Liu Fang. Parallel artificial immune algorithm for large-scale TSP [J]. Acta Electronica Sinica, 2008, 36(8): 1552 – 1558. (in Chinese).

[6] K N Krishnand, D Ghose. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics [A]. Proceedings of IEEE Swarm Intelligence Symposium. Pisatway [C]. Pasadena California: IEEE Press, 2005. 84 – 91.

[7] K N Krishnand, D Ghose. Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications[J]. Multiagent and Grid Systems, 2006, 2(3): 209 – 222.

[8] K N Krishnanad, D Ghose. Theoretical foundations for rendezvous of glowworm-inspired agent swarms at multiple locations [J]. Robotics and Autonomous Systems, 2008, 56(7): 549 – 569.

[9] K N Krishnand, D Ghose. Glowworm swarm optimisation: a new method for optimizing multi-modal functions [J]. Int J Computational Intelligence Studies, 2009, 1(1): 93 – 119.

[10] K N Krishnand, D Ghose. Glowworm swarm optimisation for simultaneous capture of multiple local optima of multimodal functions [J]. Swarm Intelligence, 2009(3): 87 – 124.

[11] K McGill, S Taylor. Robot algorithms for localization of multiple emission sources [J]. ACM Computing Surveys, 2011, 43(3): 15:1 – 15:25.

[12] Lin S, Kernigham B W. An effective heuristic algorithm for the travelling salesman problem [J]. Operations Research, 1973, 21(2): 4982 – 5161.

[13] 杨辉,康立山,陈毓屏.一种基于构建基因库求解 TSP 问题的遗传算法[J].计算机学报,2003,26(12):1753 – 1758.
Yang Hui, Kang Lishan, Chen Yuping. A gene-based genetic algorithm for TSP [J]. Chinese Journal of Computers, 2003, 26(12): 1753 – 1758. (in Chinese)

[14] 祝崇隽,刘民,吴澄,吴晓冰.针对模糊需求的 VRP 的两种 2-OPT 算法[J].电子学报,2001,29(8):1035 – 1037.
Zhu Chongjun, Liu Min, Wu Cheng, Wu Xiaobing. Two kinds of 2-OPT algorithm for VRP with fuzzy demand [J]. Acta Electronica Sinica, 2001, 29(8): 1035 – 1037. (in Chinese)

[15] 韩丽霞,王宇平.解旅行商问题的一个新的遗传算法[J].系统工程理论与实践,2007,(12):145 – 150.
Han Lixia, Wang Yuping. A novel genetic algorithm for traveling salesman problem. systems engineering [J]. Theory & Practice, 2007, (12): 145 – 150. (in Chinese)

[16] 张长胜,孙吉贵,欧阳丹彤.一种自适应离散粒子群算法及其应用研究[J].电子学报,2009,37(2):299 – 304.
Zhang Chang sheng, Sun Ji gui, OUANG Dantong. A self-adaptive discrete particle swarm optimization algorithm [J]. Acta Electronica Sinica, 2009, 37(2): 299 – 304. (in Chinese)

[17] 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 – 187.
Yu Liang liang, Wang Wan liang, Jie Jing. Solution of travel salesman problem based on hybrid particle swarm optimization olgorithm [J]. Computer Engineering, 2010, 36(11): 183 – 187. (in Chinese)

## 作者简介

**周永权** 男,1962 年出生于陕西咸阳.1993 年和 2006 年分别在兰州大学、西安电子科技大学获理学硕士和工学博士学位.现为广西民族大学教授,主要从事计算智能,神经网络及其应用等方面的研究.
E-mail: yongquanzhou@126.com

**黄正新** 男,1985 年出生于广西百色,硕士,右江民族医学院教师,主要从事群智能算法及其应用方面的研究.

**刘洪霞** 女,1984 生出生于山东淄博,硕士,主要从事计算智能及其应用方面的研究.