

## 基于 SFLA-GA 混合算法求解时间最优的旅行商问题

张勇 高鑫鑫\* 王昱洁

(合肥工业大学计算机与信息学院 合肥 230009)

**摘要:** 该文以经典的对称旅行商问题(Symmetric Traveling Salesman Problem, STSP)为基础, 求解时间最优的旅行商问题(Time Optimal TSP, TOTSP), 将拟合函数引入到混合蛙跳遗传算法(SFLA-GA)的适应度函数来反映景点客流量随时间的变化, 旨在旅游旺季为游客提供一条游览时间最短的路径推送服务。实验结果表明: 相对于随机游览路径, SFLA-GA 混合算法得到的游览路径明显节省了游览时间; 与 SFLA 和混合粒子群遗传算法(PSO-GA)相比较, SFLA-GA 混合算法具有计算量少、收敛速度快、对初始种群依赖性低以及全局性更好等优点, 在求解 TOTSP 上搜索性能更强、时间更优。

**关键词:** 时间最优的旅行商问题; 混合蛙跳遗传算法; 适应度函数; 拟合函数; 游览时间

**中图分类号:** TP301.5

**文献标识码:** A

**文章编号:** 1009-5896(2018)02-0363-08

**DOI:** 10.11999/JEIT170484

## Solving the Time Optimal Traveling Salesman Problem Based on Hybrid Shuffled Frog Leaping Algorithm-genetic Algorithm

ZHANG Yong GAO Xinxin WANG Yujie

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

**Abstract:** In order to provide a recommended-path service for tourists with the shortest traveling time in the peak-season, the Time Optimal Traveling Salesman Problem (TOTSP) is further studied and the fit function is introduced into the fitness function of the hybrid Shuffled Frog Leaping Algorithm-Genetic Algorithm (SFLA-GA) to reflect the change of traffic over time, which is based on the classic and Symmetrical Traveling Salesman Problem (STSP). The experimental results show that compared with the random tour path, the tour path significantly saves the tour time which is obtained by the hybrid SFLA-GA. Compared with SFLA and hybrid Particle Swarm Optimization-Genetic Algorithm (PSO-GA), the hybrid SFLA-GA has some advantages, such as less amount of calculation, fast speed of convergence, low dependency on initial population, good global superiority and so on. The hybrid SFLA-GA has stronger search capability and less search time in solving the TOTSP.

**Key words:** Time optimal traveling salesman problem; Hybrid Shuffled Frog Leaping Algorithm-Genetic Algorithm (SFLA-GA); Fitness function; Fit function; Tour time

### 1 引言

随着我国旅游业的快速发展, 景区需要接纳的游客越来越多, 于是给景区的管理和服务带来了巨大的挑战, 如何更好地管理景区、服务游客成为当今旅游业发展的趋势所向。我国旅游业可分为淡季和旺季: 淡季情况下, 景区客流量在正常承载范围之内, 游览时间包括两部分: 景点之间行走的时间和景点游玩的时间; 旺季情况下, 景区客流量急剧膨胀, 景点出现排队等待现象, 游览时间是在淡季

游览时间的基础之上增加景点排队等待的时间。景区客流量是随着时间变化的, 通过对客流量随时间变化的分析, 可以给游客提供一条游览时间最短的路径推送服务, 使游客体验到更好的服务质量。

旅行商问题<sup>[1,2]</sup>(Traveling Salesman Problem, TSP)是一个著名的 NP 困难问题, 实质是一种求最短路径规划的问题。本文基于经典的对称 TSP (Symmetric TSP, STSP), 对时间最优的旅行商问题<sup>[3]</sup>(Time Optimal TSP, TOTSP)进行求解, 旨在旅游旺季为游客提供一条游览时间最短的路径推送服务。近几十年受自然界生物群体活动的启发, 学者们提出很多经典智能优化算法, 典型的智能算法包括模拟退火算法<sup>[4]</sup>、免疫算法<sup>[5]</sup>、蚁群算法<sup>[6]</sup>、遗传算法<sup>[7]</sup>和粒子群算法<sup>[8]</sup>等。此类算法大多数是模拟

收稿日期: 2017-05-18; 改回日期: 2017-11-08; 网络出版: 2017-12-04

\*通信作者: 高鑫鑫 13856064304@163.com

基金项目: 国家科技支撑计划项目(2013BAH52F01)

Foundation Item: The National Science and Technology Support Program of China (2013BAH52F01)

生物进化算法,不依赖于问题本身特征,具有较强的全局搜索能力。许多学者进一步研究了它们之间的混合算法或者其改进算法,提出了新型智能优化算法,比如:混合粒子群遗传算法<sup>[3]</sup>、改进遗传算法<sup>[7]</sup>及改进蚁群算法<sup>[9]</sup>等。

混合蛙跳算法<sup>[10,11]</sup>(Shuffled Frog Leaping Algorithm, SFLA)结合了模因演化算法和粒子群算法(Particle Swarm Optimization, PSO),主要包括2个部分:局部搜索和全局信息交换。SFLA具有实现简单、收敛速度快、算法参数少、寻优能力强等优点。但是,其在优化过程中受初始值影响较大且容易陷入局部最优,而遗传算法(Genetic Algorithm, GA)的算法简单、易于实现、能够并行化,具有强鲁棒性和全局搜索能力。因此,本文采用SFLA-GA混合算法,在局部搜索策略中引入遗传算法的交叉、变异机制,在全局信息交换过程中加入变异操作,从而保证了青蛙个体的多样性、缩短了算法搜索到全局最优解的时间。同时,将拟合函数引入到SFLA-GA混合算法的适应度函数来反映景点客流量随时间的变化。SFLA-GA混合算法结合了两算法各自的优势,相对于SFLA和混合粒子群遗传算法(PSO-GA)而言,更容易快速得到稳定的、最优的解,更好地解决了TOTSP。

## 2 求解TOTSP的SFLA-GA混合算法

**TOTSP描述:**假设景区中每一个景点都对应一个出入口,给定景点数目及其坐标,求一条经过各景点一次且仅一次、从起始景点进入、游览完从终止景点出来的时间最短的路径。

### 2.1 针对TOTSP的青蛙个体排序及分组

随机生成初始群体中的 $F$ 只青蛙,每只青蛙个体表示问题的一个解,比如: $U = (U_1, U_2, \dots, U_d)$ ,其中 $d$ 表示解的空间维度。根据指定的适应度函数,依据每一个可行解来计算初始群体中每只青蛙个体的适应度值 $f(i)$ ,按降序排列规则将 $f(i)$ 存储于 $X = \{U(i), f(i), i = 1, 2, \dots, F\}$ 。根据式(1)的划分原则将初始群体划分成 $m$ 个族群,即 $Y^1, Y^2, \dots, Y^m$ ,每个族群中包含 $n$ 只青蛙,关系如式(1)所示。

$$Y^k = \left[ U(i)^k, f(i)^k \mid U(i)^k = U(k + m(i-1)), \right. \\ \left. f(i)^k = f(k + m(i-1)) \right], \\ i = 1, 2, \dots, n; k = 1, 2, \dots, m; F = mn \quad (1)$$

### 2.2 针对TOTSP的青蛙个体更新策略

对于青蛙群体,具有全局最好适应度的解表示为 $U_a$ ;对于每一个族群,具有最好适应度的解表示为 $U_b$ ,最差适应度的解表示为 $U_w$ 。针对TOTSP,

青蛙个体更新策略为任意截取 $U_b$ 中的一段路径序列,替换 $U_w$ 与之相对应的位置, $U_w$ 其余位置的景点序号若出现在这段路径序列中,则将其去掉,反之保持不变,最后将没有出现的景点序号随机插入路径序列中,从而生成一个新的可行解 $U_c$ 。若 $U_c$ 优于 $U_w$ ,用 $U_c$ 更新 $U_w$ ,否则,用全局最优的 $U_a$ 代替 $U_b$ ,再进行上述相同操作,生成一个新的可行解 $U_c$ 。若 $U_c$ 优于 $U_w$ ,用 $U_c$ 更新 $U_w$ ,否则,随机产生一个新的可行解替换 $U_w$ 。

## 2.3 SFLA-GA混合算法的适应度函数

**2.3.1 客流量描述** 景区客流量是随着时间变化的,其日分布特征如图1所示<sup>[12]</sup>:早上到中午客流量上升,中午到下午客流量下降,中午客流量处于高峰。景区中不同景点客流压力是不同的。针对客流量随时间变化的日分布特征,本文构造了2.3.3节所述的适应度函数对TOTSP进行求解。

**2.3.2 Tanimoto系数概述** Tanimoto系数<sup>[13]</sup>由Jaccard系数<sup>[14]</sup>扩展而来,表示两个集合变量之间的相似程度。比如有两个集合分别为 $A, B$ ,其中 $A = \{1, 2, 3, 4\}, B = \{1, 2, 5\}$ ,求解过程如下:

(1)计算集合 $A$ 和 $B$ 的交集,即 $C = A \& B = \{1, 2\}$ ;

(2)计算集合 $A$ 和 $B$ 之间的相似度,即 $S = N_c / (N_a + N_b + N_c) = \text{len}(c) / (\text{len}(a) + \text{len}(b) - \text{len}(c)) = 2 / (4 + 3 - 2) = 0.4$ 。

**2.3.3 适应度函数** SFLA-GA混合算法中每一个可行解的排列组合是不相同的,可行解所对应的景点排队等待批数集合也是不相同的。为了反映客流量的变化情况,即计算任意时刻的景点排队等待批数,通过在适应度函数中引入拟合函数来计算。根据路径推送服务请求时间的不同,拟合函数的每一次调用都会产生不同的、在一定范围内波动的相应集合。通过2.3.2节所述的Tanimoto系数,对上述集合与相应的历史数据统计集合进行相似判断,其以当前路径推送服务请求时间为基准,对统计的历史数据模型做分析:首先,求得可行解中第1个景点的排队等待批数;其次,求得间隔第1个景点所需游览

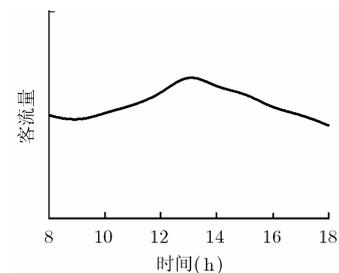


图1 景区客流量的日分布特征

时间之后的可行解中下一个景点的排队等待批数; 以此类推, 求得与可行解所对应的景点排队等待批数集合。当上述两个集合的相似度达到 80%及以上时, 表示通过拟合函数生成的相应集合符合要求, 可以将其应用到对应的可行解的适应度值求解过程中, 即游览时间的求解。

SFLA-GA 混合算法的适应度函数如式(2)所示。

$$\text{indiFit} = \text{fitness}(\text{individual}, \text{spotCoor}, \text{spotTime}, \text{wait}, \text{tp}) \quad (2)$$

其中, **individual** 表示初始化可行解; **spotCoor** 表示景点的坐标; **spotTime** 表示景点之间行走的时间; **tp** 表示景点游玩的时间; **wait** 为拟合函数, 表示与初始化可行解相对应的初始化景点排队等待批数集合的可行解, 可求得景点排队等待的时间; **indiFit** 表示适应度值。

$$\text{individual}(i, :) = \text{randperm}(\text{totals}) \quad (3)$$

$$\text{wait}(i, :) = \text{randi}([1, n], 1, \text{totals}) \quad (4)$$

其中,  $i$  表示第  $i$  个初始化可行解; **totals** 表示景点的数目;  $n$  表示景点排队等待批数的上限。假定景点的数目 **totals** 是 10, 景点排队等待批数的上限  $n$  是 4、下限是 1。式(3)表示随机生成第  $i$  个初始化可行解, 比如: **individual**(9, :) = {5-3-4-2-1-6-8-9-7-10} 即为第 9 个初始化可行解。式(4)表示生成第  $i$  个初始化可行解所对应的景点排队等待批数集合, 比如: **wait**(9, :) = {1111211121} 即为第 9 个初始化可行解所对应的景点排队等待批数集合, 判断该集合是否满足条件。若满足要求, 假定 10 个景点游玩的时间均为 25 min, 则可求得第 9 个初始化可行解的适应度值中景点排队等待的时间为:  $25 \times (1+1+1+1+2+1+1+1+2+1) = 300 \text{ min}$ , 再加上景点之间行走的时间和景点游玩的时间就是所需的游览时间, 即适应度值。

## 2.4 算法流程

利用 SFLA-GA 混合算法求解 TOTSP 的基本步骤如下:

(1)加载实验数据, 包括景点的坐标 **spotCoor**、景点之间行走的时间 **spotTime** 及景点游玩的时间 **tp**;

(2)初始化参数, 包括青蛙族群数目  $m$ , 族群中青蛙个数  $n$ , 因而总青蛙数  $F = mn$ , 族群内更新迭代次数  $N$ , 种群进化代数  $\text{MAX}$  以及 GA 的交叉算子  $P_c$ 、变异算子  $P_m$ ;

(3)设定阈值  $T$ , 随机生成一定数目的青蛙个

体, 若其适应度值小于阈值就添加到初始群体中, 否则不添加。如此反复, 直到初始群体的规模达到  $F$  个初始可行解。同时利用式(4)所述的拟合函数生成与  $F$  个初始可行解一一对应的  $F$  个初始景点排队等待批数集合, 判断集合是否满足条件。若满足要求, 根据式(2)计算初始群体中每一个青蛙个体的适应度值;

(4)按照适应度值的大小将青蛙个体升序排列, 并划分成  $m$  个族群, 且表示出种群的全局最优解及每个族群中的局部最优解;

(5)在族群的局部搜索策略中引入遗传算法的交叉、变异机制, 对除去族群内局部最优解外的其它解进行更新。该程序中采用 GA 的多点交叉, 首先随机截取最优解的一部分, 其次删除待更新解中与截取部分相同的元素, 最后将截取部分插入到待更新解中, 从而产生一个新解。对于 GA 的变异操作, 该程序中首先随机产生待更新解的两个变异位, 其次交换待更新解中两个变异位的元素, 从而产生一个新解。以下给出具体步骤:

(a)产生随机数  $\text{rand} \in [0, 1)$ , 若  $\text{rand} < P_c$  则利用族群中的局部最优解进行交叉操作产生一个新解, 并调用式(4)所述的拟合函数产生一个与新解相对应的景点排队等待批数集合, 判断该集合是否满足条件。若满足要求, 根据式(2)来计算新解的适应度值, 如果新解的适应度值更小, 那么就用新解代替原来的旧解, 否则保留旧解。同理, 用全局最优解代替局部最优解, 然后进行上述相同操作。

(b)以概率  $P_m$  保留待更新解中的元素, 未保留的元素利用变异操作产生一个新解, 同理执行步骤(5)中的操作;

(6)所有族群混合并求出新的全局最优解, 以概率  $P_m$  保留全局最优解中的元素, 未保留的元素利用变异操作产生一个新解, 同理执行步骤(5)中的操作;

(7)判断是否满足算法收敛条件, 若满足就输出全局最优的青蛙个体, 即最优的路径序列, 并画出游览路径图及输出游览时间; 否则, 返回到步骤(4)。

## 3 仿真实验

仿真实验假设某景区内 10 个景点: 景点的坐标 **spotCoor** 如表 1 所示; 景点游玩的时间 **tp** 如表 2 所示; 景点之间行走的时间 **spotTime** 如表 3 所示; 景区开放的时间是 8:00~18:00; 景点每批同时接待的人数不超过 30; 式(4)中景点排队等待批数的上限是 4、下限是 1。假设旅游旺季某日景点 1~景点 10 的历史数据统计模型均服从正态分布, 函数表达式

表1 景点坐标(m)

| 景点        | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     |
|-----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 景点坐标: $x$ | 1535.3 | 1979.0 | 2473.0 | 2191.1 | 2335.8 | 2458.4 | 2477.3 | 1930.6 | 1833.1 | 1516.9 |
| $y$       | 706.1  | 369.6  | 587.2  | 515.1  | 619.8  | 802.1  | 1019.8 | 870.5  | 753.7  | 1030.0 |

表2 景点游玩的时间(min)

| 景点     | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|--------|----|----|----|----|----|----|----|----|----|----|
| 景点游玩时间 | 20 | 20 | 15 | 15 | 25 | 30 | 15 | 25 | 20 | 15 |

表3 景点之间行走的时间(min)

| 景点 | 1 | 2   | 3    | 4   | 5    | 6    | 7    | 8   | 9    | 10   |
|----|---|-----|------|-----|------|------|------|-----|------|------|
| 1  | 0 | 7.8 | 12.6 | 8.2 | 9.7  | 12.3 | 10.5 | 4.0 | 2.8  | 8.5  |
| 2  | - | 0   | 7.3  | 4.0 | 10.4 | 8.5  | 10.0 | 5.7 | 6.0  | 11.8 |
| 3  | - | -   | 0    | 5.6 | 8.3  | 5.1  | 6.5  | 5.0 | 13.0 | 11.6 |
| 4  | - | -   | -    | 0   | 1.5  | 9.6  | 6.0  | 4.2 | 6.8  | 12.0 |
| 5  | - | -   | -    | -   | 0    | 3.1  | 4.5  | 3.8 | 8.3  | 10.0 |
| 6  | - | -   | -    | -   | -    | 0    | 3.0  | 2.0 | 7.8  | 8.5  |
| 7  | - | -   | -    | -   | -    | -    | 0    | 2.5 | 7.0  | 11.0 |
| 8  | - | -   | -    | -   | -    | -    | -    | 0   | 2.0  | 4.5  |
| 9  | - | -   | -    | -   | -    | -    | -    | -   | 0    | 5.5  |
| 10 | - | -   | -    | -   | -    | -    | -    | -   | -    | 0    |

表4 式(5)中的参数A和B

| 景点         | 1  | 2   | 3   | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|------------|----|-----|-----|----|----|----|----|----|----|----|
| 式(5)中参数: A | 70 | 80  | 70  | 40 | 45 | 40 | 30 | 60 | 30 | 60 |
| B          | 0  | -10 | -15 | 0  | -5 | 5  | 0  | -5 | 0  | 5  |

如式(5)所示, 其中: 方差 $\sigma^2$ 均为9; 数学期望 $\mu$ 分别为9, 12, 13, 10, 14, 10, 9, 13, 16和15; 参数A和B如表4所示。由图2中景点1~景点10的客流量正态分布变化情况可知, 景点1, 2和景点10的排队等待批数范围是[1,3], 景点3, 4, 5, 6和景点8的排队等待批数范围是[1,2], 景点7和景点9的排队等待批数始终为1。

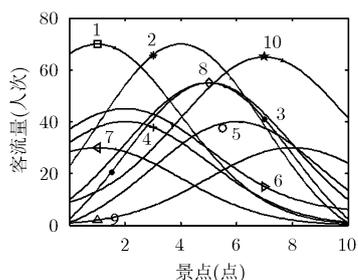


图2 旅游旺季某日不同景点客流量的正态分布

$$f(x) = \frac{A}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) + B \quad (5)$$

仿真实验环境: 软件是 MATLAB7.11.0 (R2010b); 硬件为 Intel(R) Pentium(R) CPU P6200 @2.13 GHz, 4.00 GB 内存的笔记本电脑。对于一些比较复杂的问题或多目标优化等比较难以解决的问题, 群体规模一般为 100 或 200, 进化代数数为 100。由文献[15]可知, SFLA 的参数选择建议是: 青蛙族群数目  $m$  为 20, 族群中青蛙个数  $n$  为 10, 族群内更新迭代次数  $N$  为 10。并且, 对比利用 SFLA-GA 混合算法对  $F$  为 100 和 200 时分别求得的最优解为 8.07 h 和 7.59 h, 其收敛过程分别如图 3 和图 4 所示, 可知当  $F$  为 200 时 SFLA-GA 混合算法能找到更优的解。因此, 仿真实验中 SFLA-GA 混合算法的参数设置如下: 每次实验中的进化代数数 MAX 均为

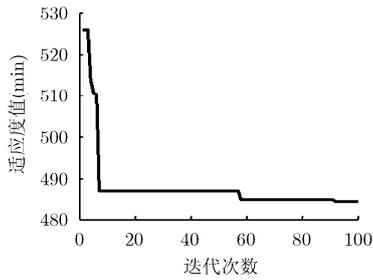


图3 SFLA-GA混合算法的收敛过程(F=100)

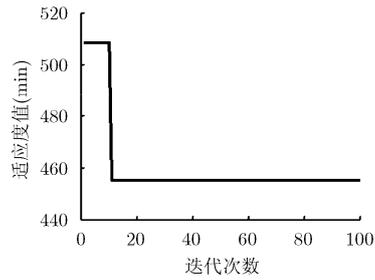


图4 SFLA-GA混合算法的收敛过程(F=200)

100, 青蛙族群数目  $m$  为 20, 族群中青蛙个数  $n$  为 10, 因而总青蛙数  $F$  为 200, 族群内更新迭代次数  $N$  为 10; 由于景区开放的时间是 8:00~18:00, 因而设置阈值  $T$  为 10 h; 在 TSP 中兼顾 GA 的寻优质量和效率时的最佳交叉算子  $P_c$  和变异算子  $P_m$  [7] 分别为  $P_c = 0.8$  和  $P_m = 0.9$ 。

### 3.1 SFLA-GA 混合算法的仿真实验结果

旺季情况下, SFLA-GA 混合算法求解时间最优的游览路径时, 算法的收敛过程如图 4 所示。最优解对应的 CPU 运行时间为 10.228 s、游览时间为 7.59 h、景点之间行走的时间为 0.98 h、游览路径如图 5 所示, 即 5-3-8-1-9-2-4-6-10-7。图 5 中 S 表示起点、E 表示终点, 游览路径是从起点 S 进入、依次游览各个景点直到到达终点 E, 最后从终点 E 出去。

淡季情况下, 游览时间只包含景点之间行走的时间和景点游玩的时间。SFLA-GA 混合算法求解时间最优的游览路径时, 最优解对应的 CPU 运行时间为 4.052 s、游览时间为 4.33 h、景点之间行走的时间为 1 h、游览路径如图 6 所示, 即 1-2-5-10-8-3-6-7-9-4, 与图 5 同理。

对比旺季和淡季的最优解得出结论: 旺季情况下, SFLA-GA 混合算法求得的景点排队等待时间最少, 即保证到达游览路径中的下一个景点时只有 1 批人在排队等待, 帮助游客节省了游览时间。旺季最优解景点人数的变化情况如表 5 所示。旺季情况下, 假定游客随机游览的景点顺序是: 5-3-4-2-1-6-8-9-7-10, 相应的景点排队等待批数集合的一种可能是 {1 1 1 1 2 1 1 1 2 1}, 可得景点之间行走的时间为 1 h, 景点排队等待的时间为 3.92 h, 景点游玩的时间为 3.33 h。与 SFLA-GA 混合算法求得的旺季最优解对比: 景点之间行走的时间变长、景点排队等待批数至少为 1, 总时间明显大于 7.59 h, 旺季最优解与随机游览路径对比结果如表 6 所示。

### 3.2 对比 SFLA, SFLA-GA, PSO-GA 3 种算法的仿真实验

旺季情况下, 为了进一步验证 SFLA-GA

表 5 旺季最优解景点人数的变化

| 景点游览顺序               | 景点 1 至 10 的排队等待批数变化情况 |
|----------------------|-----------------------|
| 5                    | {3 2 1 1 1 2 1 1 1 1} |
| 5-3                  | {3 2 1 2 1 2 1 1 1 1} |
| 5-3-8                | {2 3 1 2 1 2 1 1 1 1} |
| 5-3-8-1              | {1 3 2 2 1 2 1 2 1 1} |
| 5-3-8-1-9            | {1 2 2 1 1 1 1 2 1 2} |
| 5-3-8-1-9-2          | {1 1 2 1 2 1 1 2 1 2} |
| 5-3-8-1-9-2-4        | {1 1 2 1 2 1 1 2 1 2} |
| 5-3-8-1-9-2-4-6      | {1 1 2 1 2 1 1 2 1 2} |
| 5-3-8-1-9-2-4-6-10   | {1 1 2 1 2 1 1 2 1 1} |
| 5-3-8-1-9-2-4-6-10-7 | {1 1 1 1 1 1 1 1 1 1} |

表 6 旺季最优解与随机游览路径对比

| 景点游览顺序               | 景点之间行走的时间(h) | 景点排队等待批数              | 游览时间(h) |
|----------------------|--------------|-----------------------|---------|
| 最优解:                 | 0.98         | {1 1 1 1 1 1 1 1 1 1} | 7.59    |
| 5-3-8-1-9-2-4-6-10-7 |              |                       |         |
| 随机游览路径:              | 1.00         | {1 1 1 1 2 1 1 1 2 1} | 8.25    |
| 5-3-4-2-1-6-8-9-7-10 |              |                       |         |

混合算法的性能, 对比 SFLA, SFLA-GA, PSO-GA 3 种算法求得的最优游览时间、平均游览时间和 CPU 平均运行时间。平均游览时间和 CPU 平均运行时间都是 50 次独立实验的统计平均。

**3.2.1 3 种算法最优解** SFLA-GA 混合算法的最优解如 3.1 节所示。对于 SFLA 的实验参数设置与 SFLA-GA 混合算法一致。SFLA 在求解时间最优的游览路径时, 算法的收敛过程如图 7 所示, 最优解对应的 CPU 运行时间为 28.456 s、游览时间为 7.75 h、游览路径如图 8 所示, 即 8-7-5-6-1-3-10-9-2-4, 与图 5 同理。

对于 PSO-GA 混合算法, 由文献[16]可知, 当  $C_1 = 0.1, C_2 = 2$  时, 粒子向全局最优解快速群聚, 发生早熟收敛; 当  $C_1 = 2, C_2 = 0.1$  时, 粒子进化速度从开始一直在减速; 当  $C_1 = 0, C_2 = 0$  时, PSO“智

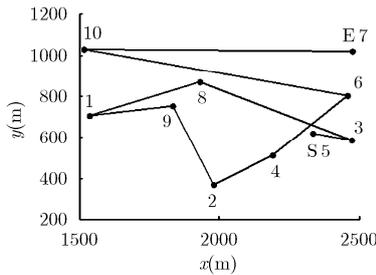


图 5 SFLA-GA 混合算法的游览路径(旺季)

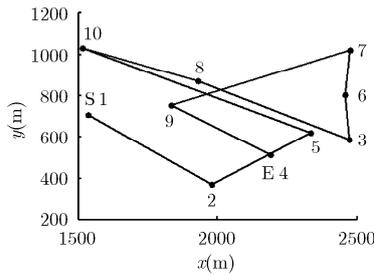


图 6 SFLA-GA 混合算法的游览路径(淡季)

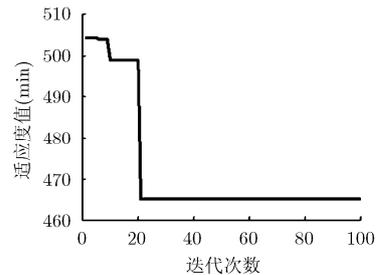


图 7 SFLA 的收敛过程

力”为零。本文通过对：(1)  $C_1 = 0.1, C_2 = 2$ ；(2)  $C_1 = 2, C_2 = 0.1$ ；(3)  $C_1 = C_2 = 1$  这 3 种情况进行对比实验得知： $C_1 = C_2 = 1$  时 PSO-GA 混合算法能找到更优的解，其最优解分别为 8.52 h, 8.07 h 及 7.83 h，情况(3)的收敛过程如图 9 所示。因此，算法参数设置如下：粒子个数与  $F$  保持一致为 200,  $C_1 = C_2 = 1$ 。PSO-GA 混合算法在求解时间最优的游览路径时，算法的收敛过程如图 9 所示，最优解对应的 CPU 运行时间为 1.86 s、游览时间为 7.83 h、游览路径如图 10 所示，即 2-4-5-3-9-10-6-7-1-8，与图 5 同理。

通过对比图 4，图 7 和图 9 的算法收敛过程可知：SFLA-GA 混合算法的收敛速度更快，即同等条件下，SFLA-GA 混合算法找到最优游览路径所需花费的时间最少。由于算法初始化时种群是随机产生的，所以输出的游览路径都具有随机性。图 5、图 6、图 8 及图 10 中给出的游览路径只是仿真实验结果之一。

**3.2.2 3 种算法综合对比** 将 SFLA, SFLA-GA 和 PSO-GA 3 种算法各独立运行 50 次，得到最优游览时间、平均游览时间和 CPU 平均运行时间如表 7 所示。

从表 7 得出结论：SFLA-GA 混合算法在最优游览时间和平均游览时间两方面优于其它算法；SFLA-GA 混合算法的平均运行时间优于 SFLA；SFLA-GA 混合算法与 PSO-GA 混合算法相比，虽然运行时间较长，但是算法结果有明显的优势，且

表 7 SFLA, SFLA-GA, PSO-GA 3 种算法的实验结果

| 算法      | 最优游览时间<br>(h) | 平均游览时间<br>(h) | 平均运行时间<br>(s) |
|---------|---------------|---------------|---------------|
| SFLA    | 7.75          | 8.00          | 28.817        |
| SFLA-GA | 7.59          | 7.86          | 10.181        |
| PSO-GA  | 7.83          | 7.91          | 1.917         |

运行时间也在正常范围之内。可见，SFLA-GA 混合算法比 SFLA 和 PSO-GA 混合算法具有更好的性能。

**3.3 SFLA-GA 混合算法对于景点数目的敏感性分析**

旺季情况下，为了验证 SFLA-GA 混合算法对于景点数目的敏感性，分别选取本文中 spotCoor 设定的前 6 个景点和前 8 个景点，利用 SFLA 和 SFLA-GA 混合算法分别求得最优解、平均解和 CPU 平均运行时间。平均解和 CPU 平均运行时间是 50 次独立实验的统计平均，实验结果如表 8 所示。

从表 8 得出结论：虽然景点数目不同，但 SFLA-GA 混合算法在最优解、平均解和 CPU 平均运行时间 3 方面均要优于 SFLA；随着景点数目的增加，SFLA-GA 混合算法的 CPU 平均运行时间的增加很小，运行时间在正常范围之内，即景点数目的不同不影响 SFLA-GA 混合算法快速寻找最优解。因此，SFLA-GA 混合算法适用于景点数目不等的不同景区中，能在最短的时间内为游客提供一条游览时间最短的路径推送服务。

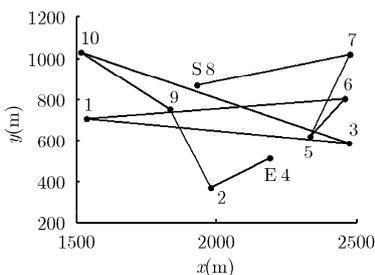


图 8 SFLA 的游览路径

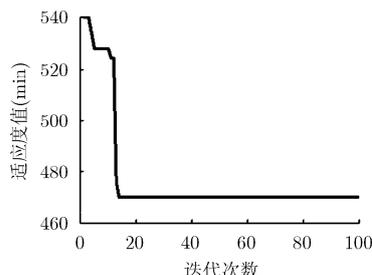


图 9 PSO-GA 混合算法的收敛过程 ( $C_1 = C_2 = 1$ )

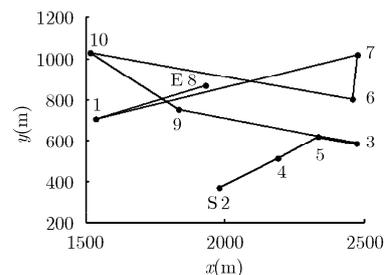


图 10 PSO-GA 混合算法的游览路径

表 8 利用 SFLA 和 SFLA-GA 混合算法对比 6、8 和 10 个景点时的实验结果

| 算法      | 景点数目      |        |        |        |
|---------|-----------|--------|--------|--------|
|         | 6         | 8      | 10     |        |
| SFLA    | 最优解(h)    | 5.32   | 6.65   | 7.75   |
|         | 平均解(h)    | 5.50   | 6.94   | 8.00   |
|         | 平均运行时间(s) | 22.787 | 25.575 | 28.817 |
| SFLA-GA | 最优解(h)    | 5.14   | 6.47   | 7.59   |
|         | 平均解(h)    | 5.45   | 6.72   | 7.86   |
|         | 平均运行时间(s) | 9.297  | 9.600  | 10.181 |

#### 4 结束语

针对旅游旺季客流量随时间变化的日分布特征, 本文采用 SFLA-GA 混合算法求解 TOTSP。首先介绍 SFLA-GA 混合算法; 其次构造适应度函数, 函数值即游览时间; 然后引入拟合函数反映客流量随时间的变化; 最后对适应度函数进行理论验证。仿真实验证实, SFLA-GA 混合算法具有计算量少、收敛速度快、对初始种群依赖性低以及全局性更好等优点, 其综合能力要明显优于 SFLA 和 PSO-GA 混合算法, 并且相对于 PSO 和 GA 等典型智能算法也有较多的优势, 能为游客提供一条时间最短的路径推送服务。目前, 旅游旺季客流量非常容易爆满, 且景区实际出入口数目肯定少于景点数目。因此, 下一步工作是对旅游旺季景区的拥堵程度做进一步的研究, 减少出入口数目, 对 SFLA-GA 混合算法中的适应度函数做更进一步的改进。

#### 参 考 文 献

- [1] ALFONSAS M, ANDRIUS B, and ANTANAS L. Modified local search heuristics for the symmetric traveling salesman problem[J]. *Information Technology and Control*, 2013, 42(3): 217-230. doi: 10.5755/j01.itc.42.3.1301.
- [2] IACOPO G, FRANCOIS M, and KENJI S. The traveling salesman problem with neighborhoods: MINLP solution[J]. *Optimization Methods and Software*, 2013, 28(2): 364-378. doi: 10.1080/10556788.2011.648932.
- [3] 张勇, 陈玲, 徐小龙, 等. 基于 PSO-GA 混合算法时间优化的旅行商问题研究[J]. *计算机应用研究*, 2015, 32(12): 3613-3617. doi: 10.3969/j.issn.1001-3695.2015.12.019.  
ZHANG Yong, CHEN Ling, XU Xiaolong, et al. Research on time optimal TSP based on hybrid PSO-GA[J]. *Application Research of Computers*, 2015, 32(12): 3613-3617. doi: 10.3969/j.issn.1001-3695.2015.12.019.
- [4] ZHAN Shihua, LIN Juan, ZHANG Zejun, et al. List-based simulated annealing algorithm for traveling salesman problem[J]. *Computational Intelligence and Neuroscience*, 2016, (5): 1-12. doi: 10.1155/2016/1712630.
- [5] PAN G, LI K, OUYANG A, et al. Hybrid immune algorithm based on greedy algorithm and delete-cross operator for solving TSP[J]. *Soft Computing*, 2016, 20(2): 1-12. doi: 10.1007/s00500-014-1522-3.
- [6] YANG Jianyi, DING Ruifeng, ZHANG Yuan, et al. An improved ant colony optimization(I-ACO) method for the quasi travelling salesman problem(Quasi-TSP)[J]. *International Journal of Geographical Information Science*, 2015, 29(9): 1534-1551. doi: 10.1080/13658816.2015.1013960.
- [7] 王勇臻, 陈燕, 于莹莹. 求解多旅行商问题的改进分组遗传算法[J]. *电子与信息学报*, 2017, 39(1): 198-205. doi: 10.11999/JEIT160211.  
WANG Yongzhen, CHEN Yan, and YU Yingying. Improved grouping genetic algorithm for solving multiple traveling salesman problem[J]. *Journal of Electronics & Information Technology*, 2017, 39(1): 198-205. doi: 10.11999/JEIT160211.
- [8] 胡颖, 庄雷, 兰巨龙, 等. 基于自适应协同进化粒子群算法的虚拟网节能映射研究[J]. *电子与信息学报*, 2016, 38(10): 2660-2666. doi: 10.11999/JEIT151434.  
HU Ying, ZHUANG Lei, LAN Julong, et al. Energy aware virtual network embedding using particle swarm optimization algorithm based on adaptive[J]. *Journal of Electronics & Information Technology*, 2016, 38(10): 2660-2666. doi: 10.11999/JEIT151434.
- [9] 程希, 沈建华. 一种基于改进蚁群算法的光网络波长路由分配算法[J]. *电子与信息学报*, 2012, 34(3): 710-715. doi: 10.3724/SP.J.1146.2011.01032.  
CHENG Xi and SHEN Jianhua. An improved ant colony algorithm for routing and wavelength assignment in optical networks[J]. *Journal of Electronics & Information Technology*, 2012, 34(3): 710-715. doi: 10.3724/SP.J.1146.2011.01032.
- [10] HASANIEN H M. Shuffled frog leaping algorithm-based static synchronous compensator for transient stability improvement of a grid-connected wind farm[J]. *Iet Renewable Power Generation*, 2014, 8(6): 722-730. doi: 10.1049/iet-rpg.2013.0277.
- [11] 骆剑平, 李霞, 陈泯融. 基于改进混合蛙跳算法的 CVRP 求解[J]. *电子与信息学报*, 2011, 33(2): 429-434. doi: 10.3724/SP.J.1146.2010.00328.  
LUO Jianping, LI Xia, and CHEN Minrong. Improved shuffled frog leaping algorithm for solving CVRP[J]. *Journal of Electronics & Information Technology*, 2011, 33(2): 429-434. doi: 10.3724/SP.J.1146.2010.00328.
- [12] 马乐, 叶见新, 王晖. 旅游景区智能客流统计系统应用研究——以玄武湖为例[J]. *中国科技信息*, 2013, (2): 88-89. doi: 10.3969/j.issn.1001-8972.2013.02.040.  
MA Le, YE Jianxin, and WANG Hui. Application research of

- intelligent statistical system for scenic area passenger flow-take xuanwu lake as an example[J]. *China Science and Technology Information*, 2013, (2): 88-89. doi: 10.3969/j.issn.1001-8972.2013.02.040.
- [13] 文俊浩, 舒珊. 一种改进相似性度量的协同过滤推荐算法[J]. *计算机科学*, 2014, 41(5): 68-71. doi: 10.3969/j.issn.1002-137X.2014.05.015.
- WEN Junhao and SHU Shan. Improved collaborative filtering recommendation algorithm of similarity measure[J]. *Computer Science*, 2014, 41(5): 68-71. doi: 10.3969/j.issn.1002-137X.2014.05.015.
- [14] HWANG Chaoming and YANG Miinshen. New similarity measures between generalized trapezoidal fuzzy numbers using the jaccard index[J]. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 2014, 22(6): 831-844. doi: 10.1142/S0218488514500445.
- [15] 孙冲. 混合蛙跳算法改进及控制参数优化仿真研究[D]. [硕士论文], 哈尔滨工业大学, 2011.
- SUN Chong. Shuffled frog leaping algorithm improvement and simulation research for optimization of control parameters[D]. [Master dissertation], Harbin Institute of Technology, 2011.
- [16] 麻荣永, 杨磊磊, 张智超. 基于粒子迭代位移和轨迹的粒子群算法C1、C2 参数特性分析[J]. *数学计算*, 2013, 2(4): 109-115.
- MA Rongyong, YANG Leilei, and ZHANG Zhichao. Analysis the characteristic of C1, C2 based on the PSO of iterative shift and trajectory of particle[J]. *Mathematical Computation*, 2013, 2(4): 109-115.
- 张 勇: 男, 1973 年生, 副教授, 研究方向为智能信息处理、智能优化算法、信号与信息处理.
- 高鑫鑫: 男, 1992 年生, 硕士生, 研究方向为智能信息处理、智能优化算法、信号与信息处理.
- 王昱洁: 女, 1980 年生, 讲师, 研究方向为智能优化算法、模式识别.