

申晓宁<sup>1</sup> 王森林<sup>1</sup> 吴俊潮<sup>1</sup> 仇友辉<sup>1</sup> 张磊<sup>1</sup> 李常峰<sup>1</sup> 王玉芳<sup>1</sup>

# 基于改进混合蛙跳算法的个性化旅游路线推荐

## 摘要

大众在旅游途中期望获得开销低、行程方便、舒适度高的旅游体验,同时还具有历史人文、自然景观、美食购物等不同游览需求.因此,本文提出了一种基于改进混合蛙跳算法的个性化旅游路线推荐方法.首先建立个性化旅游路线推荐问题的优化模型,并针对该模型的特点,设计改进混合蛙跳算法.通过调整可控精度,增加筛选准则和及时处理异常解等策略增强群体的多样性,降低遗漏最优解的风险,强化局部搜索能力,并提高算法的求解精度.以南京三日游个性化旅游路线推荐问题作为实例,收集南京市内知名景点的门票价格、开放时间、不同出行方式所需的时间和花费情况以及食宿费用等相关数据,基于改进混合蛙跳算法进行求解.实验结果表明,与改进前的方法相比,所提改进方法能够获取更优的路径解,推荐的路线能够更好地满足用户的个性需求.

## 关键词

个性化旅游;路线推荐;混合蛙跳算法;筛选准则;可控精度

中图分类号 TP182

文献标志码 A

收稿日期 2020-08-28

资助项目 国家自然科学基金(61502239,51705260);江苏省自然科学基金(BK20150924)

## 作者简介

申晓宁,女,博士,教授,主要研究方向为计算智能、多目标优化.sxnystyt@sina.com

## 0 引言

随着人们生活水平的不断提高,越来越多的人会在闲暇之余选择外出旅游,也因此带动了不少城市第三产业的发展.而外出旅游到一个陌生的地方,如果不提前做好旅游路线的规划,可能会导致在旅行过程中出现费时、费钱等体验差的问题.同时,不同的旅游人群对于历史人文、自然景观、美食购物等有不同的需求.因此,如何针对旅游者的个性需求,规划适合的旅游路线,并为其选择合适的出行方式以提高在旅游过程中的体验,对于民众生活质量的提升具有重要意义<sup>[1]</sup>.

旅游路线规划问题是基于经典的旅行商问题演化而来的,其主要思想是将候选城市的景点按照一定的规则进行组合优化.目前国内外对此问题的研究颇为广泛,针对此问题还提出了一些其他的改进算法,如动态规划算法<sup>[2]</sup>、模拟退火算法<sup>[3]</sup>、蚁群算法<sup>[4]</sup>等.杨萍<sup>[5]</sup>的研究表明,路线规划者最终考虑的目标主要分为两种,一是成本最低,二是收益最大;明勇等<sup>[6]</sup>以城市垃圾回收路线总路径、车辆总费用和惩罚成本为目标建立了城市生活垃圾回收路径规划的数学模型,并对基本混合蛙跳算法进行了改进,提出了一种能够有效实现城市垃圾回收的路线规划方法;黄于欣等<sup>[7]</sup>针对多景点景区路径规划问题提出了一种改进的蚁群算法,有效地避免了算法陷入局部最优,使得算法快速收敛;杨晓敏<sup>[8]</sup>通过对蚁群算法的参数调整和改进,成功实现了黄河金三角旅游路线的规划;陈春朝等<sup>[9]</sup>针对蛙跳算法迭代速度慢的问题提出了改进混合蛙跳算法,提高了算法的进化速度和精度,并将其用于优化人工势场的参数以提高路径规划能力;Shiripour等<sup>[10]</sup>根据旅行人口对链接时间的影响和在网络上的分布方式,提出了混合整数非线性规划模型,并提出了求解模型的遗传算法(GA),解决了规划出行时间灵活的人行道网络问题;Reyes-Rubiano等<sup>[11]</sup>为了更好地指导搜索过程,通过在可变邻域搜索框架内集成偏向随机化策略,提出了一种基于元启发式的方法,用于解决同时考虑了经济、环境和社会维度的多站点车辆路径问题;Sánchez-Oro等<sup>[12]</sup>提出了一种通用的变量邻域搜索算法,寻求使路线总数、总旅行成本和最长路线最小化的开放式车辆路径问题解决方案.本文在以上研究的基础上,建立个性化旅游路线推荐问题的优化模型,并针对该模型的特点,提出一种基于改进混合蛙跳算法的个性化旅游路线推荐方

<sup>1</sup> 南京信息工程大学 自动化学院/大气环境与装备技术协同创新中心/江苏省大数据分析技术重点实验室,南京,210044

法.通过调整可控精度在算法接近结束时扩大搜索范围,以降低遗漏更优解的风险;通过在劣解的改进过程中增加筛选准则以强化局部搜索能力,从而在保持精度的情况下尽可能加快收敛速度;通过引入惩罚因子大幅度降低异常解的适应度,使其在子群的更新中被淘汰,以及时处理异常解,提高算法的运算速度.以南京三日游为旅游路线优化问题的实例,对所提模型和方法进行了仿真实验验证.

## 1 基本蛙跳算法原理

混合蛙跳算法(Shuffled Frog Leaping Algorithm, SFLA)是由 Eusuff 和 Lansey 为解决组合优化问题于 2003 年提出的一种全新的元启发式群体进化算法,具有高效的计算性能和优良的全局搜索能力<sup>[13]</sup>. SFLA 的思想是:在一片湿地中生活着一群青蛙,湿地内离散地分布着许多石头,青蛙通过在不同的石头上跳跃以找到食物较多的地方.每只青蛙可以定义为问题的一个解.将湿地的整个青蛙群体分为不同的子群体.在子群体中的每个个体有自己的文化,既影响其他个体,也受其他个体的影响,在子群内个体之间通过文化的交流实现信息的交换,执行局部搜索策略.每个个体随着子群体的进化而进化.每个子群体也有着各自的文化,当子群体进化到一定阶段以后,各个子群体再进行混洗,实现子群体间思想的交流(全局信息交换),直到满足所设置的终止条件后算法停止.基本混合蛙跳算法 SFLA 的伪代码如下算法 1 所示:

### 算法 1 基本蛙跳算法 SFLA

输入:子群个数  $M$ ,子群内个体数量  $I$ ,种群个体总数量  $I \times M$ ,子群更新次数  $N$ ,混合迭代次数  $G$ ,当前迭代次数  $g$ .

输出:全局最优解.

a) 初始化种群,计算个体目标函数值;

- b) for  $g = 1$  to  $G$  do
- c) 对所有个体按目标函数值从小到大排序并分成  $M$  个子群; // 假设为最小化问题,目标值越小,则个体越优;
- d) 根据排序与分组结果记录当前全局最优解  $X_g$ 、各子群局部最优解  $X_b$ 、局部最差解  $X_w$ ;
- e) for  $i = 1$  to  $N$  do
- f)  $X_{w_{new}} = X_w + \lambda(X_b - X_w)$  //  $\lambda$  为  $0 \sim 1$  的随机数;
- g) if  $F(X_{w_{new}}) < F(X_w)$  then //  $F$  表示个体的目标值;
- h)  $X_w = X_{w_{new}}$
- i) else  $X'_{w_{new}} = X_w + \lambda(X_g - X_w)$  //  $\lambda$  为  $0 \sim 1$  的随机数;
- j) end if
- k) if  $F(X'_{w_{new}}) < F(X_w)$  then
- l)  $X_w = X'_{w_{new}}$
- m) else
- n) 随机产生新解代替  $X_w$ ;
- o) end if
- p) 对组内个体按目标值从小到大排序,重新确定局部最优解  $X_b$ 、局部最差解  $X_w$ ;
- q) end for
- r) 将所有个体混洗,记录全局最优个体  $X_g$ ;
- s) end for

其中,分组是将所有个体按目标值大小进行升序排序后,将种群分成了  $M$  个子群,每个子群中个体数量为  $I$ .按照混合蛙跳算法的分组规则将每个个体分配至各自的子群.分配规则如式(1)与表 1 所示.

$$\text{Race}(k) = [\text{rank}(k), \text{rank}(M+k),$$

$$\text{rank}(2M+k), \dots, \text{rank}((I-1)M+k)], \quad (1)$$

式中,  $\text{Race}(k)$  表示编号为  $k$  的子群,  $\text{rank}(k)$  表示排序在第  $k$  位的个体,  $k = 1, 2, \dots, M$ .

表 1 子群分配表

Table 1 Subgroup assignment

		子群编号				
		1	2	3	...	$M$
个体		$\text{rank}(1)$	$\text{rank}(2)$	$\text{rank}(3)$	...	$\text{rank}(M)$
		$\text{rank}(M+1)$	$\text{rank}(M+2)$	$\text{rank}(M+3)$	...	$\text{rank}(2M)$
		$\text{rank}(2M+1)$	$\text{rank}(2M+2)$	$\text{rank}(2M+3)$	...	$\text{rank}(3M)$
		$\vdots$	$\vdots$	$\vdots$		$\vdots$
		$\text{rank}((I-1)M+1)$	$\text{rank}((I-1)M+2)$	$\text{rank}((I-1)M+3)$	...	$\text{rank}(IM)$

注:  $\text{rank}(\cdot)$  为各排名对应的个体.

## 2 数学模型建立

### 2.1 数学模型假设

为了便于模型描述,本文做出如下假设:

1) 假设旅游者从自己所在地到达旅游目的城市的相关影响因素不在路线规划考虑之内,模型选取的出发点为旅游者所在旅游目的城市的任意景点或者酒店。

2) 根据文献[14],假设旅游者在旅游过程中只能选择步行、公交、地铁、驾车4种出行方式,且若选中某个景点,该景点只能被游览一次。

3) 假设以旅游期间内的日平均费用最少作为优化目标。

4) 根据文献[15],假设所有景点开放时间为8:00—18:00,共计10h,则每日耗时 $T$ 应满足 $T \leq 10$ 。

5) 假设游玩的最大天数为 $d_{\max}$ ,每天最多游玩 $n_d$ 个景点,其中 $d=1,2,\dots,d_{\max}$ 。

6) 在选定某一个酒店作为起点后,每天都以该酒店作为旅游路线的起点与终点。

7) 假设用户的个性化设置体现在选择偏好的旅游类型,如自然景观游、历史人文游、美食购物游等。设共有 $n$ 种可选旅游类型,每种旅游类型对应的景点集合为 $O_1, O_2, \dots, O_n$ 。满足用户的个性化需求是指:为其规划出的旅游路线中,至少包含一个符合该用户偏好旅游类型的景点。

### 2.2 个性化旅游路线推荐问题的数学模型

个性化旅游路线推荐问题的数学模型可以描述为:在用户的个性化设置下,找出最优决策向量 $\mathbf{X} = [\mathbf{r}, \mathbf{P}_b]$ ,其中 $\mathbf{r}$ 为地点标号组成的向量,表示旅游路径, $\mathbf{P}_b$ 为交通方式标号组成的向量,表示交通方式的选择,以最小化如下目标:

$$\min F = f(\mathbf{X}), \quad (2)$$

$$f(\mathbf{X}) = \text{con}(\mathbf{r}) + \text{con}(\mathbf{P}_b), \quad (3)$$

$$\text{con}(\mathbf{r}) = \frac{1}{d_{\max}} \sum_{d=1}^{d_{\max}} \text{con}(\mathbf{r}(d)), \quad (4)$$

$$\text{con}(\mathbf{r}(d)) = \sum_{i=1}^{n_d} P_m(p_i^d) + P_h, \quad (5)$$

$$\text{con}(\mathbf{P}_b) = \frac{1}{d_{\max}} \sum_{d=1}^{d_{\max}} \text{con}(\mathbf{P}_b(d)), \quad (6)$$

$$\text{con}(\mathbf{P}_b(d)) = m_{\text{cost}}(p_{n_d}^d, p_0^d, p_{b_{n_d}}^d) + \sum_{i=0}^{n_d-1} m_{\text{cost}}(p_i^d, p_{i+1}^d, p_{b_i}^d), \quad (7)$$

$$T(d) = t_{\text{cost}}(p_{n_d}^d, p_0^d, p_{b_{n_d}}^d) + \sum_{i=1}^{n_d} P_i(p_i^d) + \sum_{i=0}^{n_d-1} t_{\text{cost}}(p_i^d, p_{i+1}^d, p_{b_i}^d), \quad (8)$$

其中,目标 $f(\mathbf{X})$ 表示旅游期间的日平均费用,日均费用由游玩景点和交通费用两部分组成。式(4)计算的是游玩景点的日均费用, $d_{\max}$ 为旅游天数;第 $d$ 天的游玩景点费用 $\text{con}(\mathbf{r}(d))$ ( $d=1,2,\dots,d_{\max}$ )由式(5)计算,规定 $p_0$ 为酒店的编号, $\mathbf{r}(d) = [p_0^d, p_1^d, \dots, p_i^d, \dots, p_{n_d}^d]$ 是第 $d$ 天的路线, $p_0^d = p_0, p_i^d$ ( $i=1,2,\dots,n_d$ )是第 $d$ 天游玩的第 $i$ 个景点, $n_d$ 为第 $d$ 天游玩景点的个数, $P_m(p_i^d)$ 是第 $d$ 天编号为 $p_i^d$ 的景点的游玩费用, $P_h$ 为酒店每日费用;式(6)计算的是交通的日均费用,第 $p_0$ 天的交通费用 $\text{con}(\mathbf{P}_b(d))$ 由式(7)给出,其中 $\mathbf{P}_b(d) = [p_{b_0}^d, p_{b_1}^d, \dots, p_{b_i}^d, \dots, p_{b_{n_d}}^d]$ 是第 $d$ 天的交通方式, $m_{\text{cost}}(p_i^d, p_{i+1}^d, p_{b_i}^d)$ ( $i=0,1,2,\dots,n_d-1$ )表示从地点 $p_i^d$ 到地点 $p_{i+1}^d$ ,采用编号为 $p_{b_i}^d$ 的交通方式所消耗的费用, $m_{\text{cost}}(p_{n_d}^d, p_0^d, p_{b_{n_d}}^d)$ 表示从当天游玩的最后一个景点返回酒店,采用编号为 $p_{b_{n_d}}^d$ 的交通方式所消耗的费用;式(8)为第 $d$ 天的耗时,其中 $P_i(p_i^d)$ 为地点 $p_i^d$ 的游玩耗时, $t_{\text{cost}}(p_i^d, p_{i+1}^d, p_{b_i}^d)$ ( $i=0,1,2,\dots,n_d-1$ )表示从地点 $p_i^d$ 到地点 $p_{i+1}^d$ ,采用编号为 $p_{b_i}^d$ 的交通方式所消耗的时间, $t_{\text{cost}}(p_{n_d}^d, p_0^d, p_{b_{n_d}}^d)$ 表示从当天游玩的最后一个景点返回酒店,采用编号为 $p_{b_{n_d}}^d$ 的交通方式所消耗的时间。

设用户通过个性化设置选择了第 $j$ 种旅游类型,该类型对应的景点集合为 $O_j$ ,则满足约束条件:

$$\text{s.t.} \quad T(d) \leq 10, \quad 1 \leq d \leq d_{\max}, \quad (9)$$

$$\mathbf{r}(1) \cap \mathbf{r}(2) \cap \dots \cap \mathbf{r}(d) \cap \dots \cap \mathbf{r}(d_{\max}) = \{p_0\}, \quad (10)$$

$$[\mathbf{r}(1) \cup \mathbf{r}(2) \cup \dots \cup \mathbf{r}(d) \cup \dots \cup \mathbf{r}(d_{\max})] \cap O_j \neq \emptyset, \quad (11)$$

式(9)—(11)为约束条件。式(9)由假设4给出。式(10)由假设6给出,每日旅游路线的起点都是同一个酒店,且每个景点最多只能被游览一次,因此不同日期的路线交集仅包含 $p_0$ 。式(11)由假设7给出,表示为用户规划出的旅游路线中,至少包含一个景点对应的旅游类型与该用户设置的偏好旅游类型一致。

## 3 求解个性化旅游路线推荐问题的改进混合蛙跳算法

本文采用改进混合蛙跳算法对用户旅行的路径

和交通方式的选择进行组合优化.通过改进混合蛙跳算法不断迭代进化以期获得符合用户需求的最佳旅游路线与交通方式.所提算法采用调整可控精度、新增筛选准则和异常解的处理三种改进策略.

### 3.1 个体的编码方式

本模型的解由路径  $r$  与交通方式  $P_b$  组成,算法采用十进制整数编码的方式对个体进行编码.对于  $n$  个候选景点  $\{s_1, s_2, s_3, \dots, s_n\}$ ,为便于表示,依次标记为  $S = \{1, 2, 3, \dots, n\}$ ,对于 4 种交通方式 {公交, 地铁, 驾车, 步行},依次标记为  $W = \{1, 2, 3, 4\}$ ,则旅游天数为  $d_{max}$ ,第  $d$  天游玩景点数量为  $n_d$  的个体  $X$  的编码方式如图 1 所示.

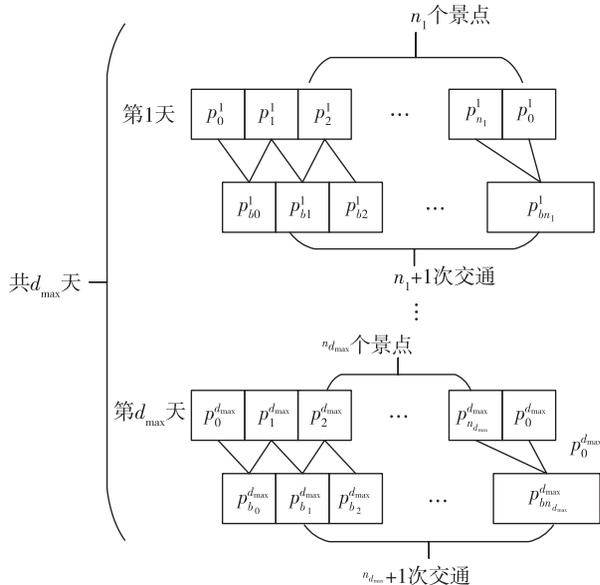


图 1 个体  $X$  编码方式示意图

Fig. 1 Schematic diagram of individual  $X$  coding mode

图 1 中:  $p_i^d (i = 1, 2, \dots, n_d)$  表示第  $d$  天游玩的第  $i$  个地点,且  $p_i^d \in S$ ;根据模型假设,约定第 0 个地点为酒店  $p_0$ ,即每日路线的起点与终点均为同一酒店,且满足  $p_0^1 = p_0^2 = \dots = p_0^d = \dots = p_0^{d_{max}} = p_0$ ,即多日游的

起点固定;  $p_{bi}^d (i = 0, 1, 2, \dots, n_d - 1)$  表示第  $d$  天从地点  $p_i^d$  到地点  $p_{i+1}^d$  采用的交通方式,  $p_{bn_d}^d$  表示第  $d$  天从游玩的最后一个景点返回酒店采用的交通方式,且  $p_{bi}^d \in W (i = 0, 1, 2, \dots, n_d)$ .

### 3.2 调整可控精度

基本蛙跳算法中,种群规模与分组数量这两个参数在进化过程中均保持不变.如果增大种群规模和分组数量,能够使得算法的搜索范围扩大、寻优能力增强,但同时也加大了运算的软硬件资源消耗,运算时间较长.因此,本文为了权衡求解精度与资源消耗及运算时间之间的互斥关系,为混合蛙跳算法设定了一个迭代次数的中间阈值  $T = f_{th} \times G$ ,其中  $G$  为算法的总迭代次数,  $f_{th}$  为控制结果精度的调节因子.当迭代次数达到  $T$  时,在原种群中加入规模与原种群规模相同的新的种群,使搜索范围扩大,该方案的示意图如图 2 所示.

由图 2 可知,在对初始种群  $A$  进行了  $T$  次的进化迭代之后得到了种群  $A^{(T)}$ ,此时在种群  $A^{(T)}$  中加入与其规模相同的种群  $B$  得到了新的种群  $(A^{(T)} + B)$ .对于新的种群进行  $(G - T)$  次的进化后算法达到了终止条件,此时种群  $(A^{(T)} + B)^{(G-T)}$  中的最优个体即为混合蛙跳算法搜索到的全局最优个体.

显然,调节因子  $f_{th}$  取值越小,种群规模扩大得就越早,搜索范围也就越大,从而使得发现最优解的概率增大.但是过早地进行种群扩大与直接增大初始种群规模是近似等效的,这样做固然可以提高结果精度,但是也会大幅度地降低收敛速度,加大软硬件资源的损耗.

在算法运行到后期时,种群缺乏多样性,且可能已陷入局部最优解,原本的搜索范围不足以搜索到潜在的更优解,于是可以选择在此时扩大搜索范围,以降低遗漏更优解的风险.在本文的仿真实验中,取调节因子  $f = 0.9$ ,即迭代次数中间阈值  $T = 0.9G$ .

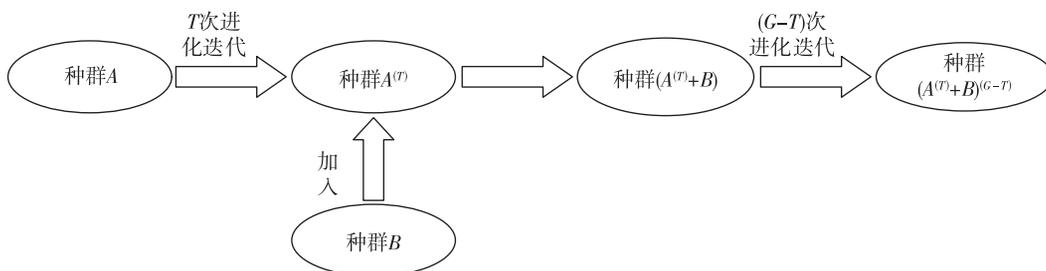


图 2 调整可控精度示意图

Fig. 2 Schematic diagram of adjusting controllable accuracy

### 3.3 新增筛选准则

在基本蛙跳算法中,对当前局部最差解  $X_w$  采用局部最优解  $X_b$  进行一次更新后,若产生的新解  $X'_{w_{new}}$  比  $X_w$  差,则会对  $X_w$  采用全局最优解  $X_g$  再进行一次更新,若第二次更新的新解  $X_{w_{new}}$  仍比  $X_w$  差,则在搜索空间内产生一个随机解来替换  $X_w$ . 新解个体的两次更新由图 3 描述.

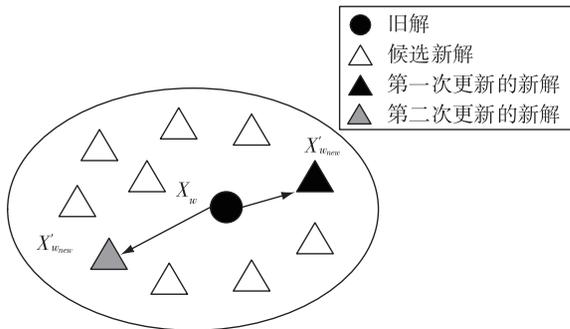


图 3 基本蛙跳算法两次个体更新示意图

Fig. 3 Schematic diagram of two individual updates for basic frog leaping algorithm

图 3 中黑色圆点表示子群体的局部最差解  $X_w$ , 黑色三角表示第一次更新产生的新解  $X_{w_{new}}$ , 灰色三角代表第二次更新产生的新解  $X'_{w_{new}}$ , 白色三角表示尚未搜索到的候选新解. 根据混合蛙跳算法的筛选规则,若  $X_{w_{new}}$  和  $X'_{w_{new}}$  都比  $X_w$  差,则会直接舍去  $X_w$ , 用搜索空间内的一个随机解替代. 这也就意味着,两次更新不仅决定了产生的新解的取舍,也决定了当前解  $X_w$  的取舍. 在  $X_{w_{new}}$  和  $X'_{w_{new}}$  都比  $X_w$  差的情况下,当舍去  $X_w$  后,当前状态下的搜索区域会被产生的随机解的搜索区域替换. 与经历过多次搜索更新操作的  $X_w$  相比,随机解具有更大的不确定性,以随机解为更新对象的搜索区域在很大概率上会劣于  $X_w$ , 因此,原混合蛙跳算法仅通过判断适应度大小来决定当前解  $X_w$  的取舍,可能会造成具有较大搜索潜力的  $X_w$  的丢失.

针对上述问题,本文在基本蛙跳算法的筛选规则中新增了一条:如果在第二次更新后的新解  $X'_{w_{new}}$  仍劣于  $X_w$ ,则计算  $X'_{w_{new}}$  与  $X_w$  之间的欧拉距离(即跳跃步长的幅值)来衡量变化程度,若变化程度大于阈值  $\varepsilon$ ,则说明当前  $X_w$  的搜索范围较大,仍采用  $X'_{w_{new}}$  替代  $X_w$ ;反之舍去,随机产生新解代替当前  $X_w$ . 特别地,在本模型中取

$$\varepsilon = \frac{\max\{\|D_{1\max}\|_2, \|D_{2\max}\|_2\}}{2}, \quad (12)$$

其中,  $D_{1\max} = X_b - X_w$ ,  $D_{2\max} = X_g - X_w$ , 它们分别表示第一次和第二次更新中,跳跃步长的最大可能值;  $\|\cdot\|_2$  表示向量的 2 范数. 所提新增筛选规则的特点是:通过在个体的更新过程中判断实际步长的幅值大小,既可以降低丢失具有较大搜索潜力的  $X_w$  的风险,又能够防止算法陷入局部最优. 改进混合蛙跳算法筛选准则的伪代码如算法 2 所示.

#### 算法 2 改进混合蛙跳算法的筛选准则

输入:当前子群局部最差解  $X_w$ , 第二次更新后得到新解的  $X'_w$ .

输出:更新后的解  $X_w$ .

- a) if  $F(X'_{w_{new}}) \geq F(X_w)$  then
- b) if  $\|X'_{w_{new}} - X_w\|_2 < \varepsilon$  then
- c) 在搜索空间里产生随机解  $X_{w_{new}}$  代替  $X_w$ ;
- d) else
- e)  $X_w = X_{w_{new}}$
- f) end if
- g) end if

### 3.4 异常解的处理

在生成解的过程中,经常会出现重复、交叉等异常解的情况,若是采用强制性的循环来避免异常解的产生会大大增加算法运算时长甚至陷入死循环. 为了解决这一系列问题,本文在生成每一个随机个体之后,立即判断其是否为异常解,若为异常解,则用一个数值足够大的惩罚因子  $\Delta C$  代替其目标值,即

$$f(X_h) = \Delta C. \quad (13)$$

显然,通过引入惩罚因子后异常解的目标值被赋予了一个较大的惩罚值,使其在后续的子群更新中被淘汰,简化了模型,提高了运算速度.

### 3.5 改进算法的实现

基于上述 3 种改进策略,用于求解个性化旅游路线推荐问题的改进混合蛙跳算法 ISFLA (Improved Shuffled Frog Leaping Algorithm) 的伪代码如算法 3 所示.

#### 算法 3 改进蛙跳算法伪代码 ISFLA

输入:子群个数  $M$ , 子群内个体数量  $I$ , 种群个体总数量  $I \times M$ , 子群更新次数  $N$ , 混合迭代次数  $G$ , 当前迭代次数  $g$ .

输出:全局最优解.

- a) 初始化种群,计算个体目标函数值;
- b) for  $g = 1$  to  $G$  do
- c) if  $g = T$  then
- d) 根据 3.2 节扩大种群规模;

- e) end if
- f) 对所有个体按目标函数值大小排序并分组,根据排序与分组结果记录当前全局最优解  $X_g$ 、各组局部最优解  $X_b$ 、局部最差解  $X_w$ ;
- g) for  $i = 1$  to  $N$  do
- h)  $X_{w_{new}} = X_w + \lambda(X_b - X_w)$  //  $\lambda$  为  $0 \sim 1$  的随机数;
- i) if  $F(X_{w_{new}}) < F(X_w)$  then //  $F$  表示个体的目标值;
- j)  $X_w = X_{w_{new}}$
- k) else
- $X'_{w_{new}} = X_w + \lambda(X_g - X_w)$  //  $\lambda$  为  $0 \sim 1$  的随机数;
- l) end if
- m) if  $F(X'_{w_{new}}) < F(X_w)$  then
- n)  $X_w = X'_{w_{new}}$
- o) else
- 执行算法 2;
- p) end if
- q) 对每组个体按目标函数值大小排序,重新确定局部最优解  $X_b$ 、局部最差解  $X_w$ ;
- r) end for
- s) 将所有个体混洗,记录全局最优个体  $X_g$ ;
- t) end for

## 4 仿真实验

### 4.1 数据集的建立

本文实验以南京游为例建立数据集.搜集南京周边的 19 个热门景点数据,假设旅游时长为 3 d,每天游玩 3 个景点.通过百度地图搜集目标地点的数

据,每组数据包括坐标、名称、景区的开放时间、景区的门票价格、不同出行方式所需的时间和费用以及对应的食宿价格.表 2—4 分别列出了各景点经纬度、门票费用及游玩时间.由于版面限制,费用和消耗时间的数据不在此列出.以历史人文、自然景观、美食购物作为 3 种个性化旅游类别.对各景点的分类如下:历史人文 = {南京长江大桥,颐和路,鸡鸣寺,中山陵,明孝陵,夫子庙,总统府,侵华日军南京大屠杀遇难同胞纪念馆,中华门,大报恩寺,雨花台风景区,牛首山,南京博物院};自然景观 = {南京长江大桥,栖霞山,玄武湖,中山陵,南京眼步行桥,中华门,雨花台风景区,牛首山};美食购物 = {新街口,夫子庙,湖南路}.

### 4.2 实验结果与分析

在 CPU 为 Intel(R) Core(TM) i7-7700HQ 2.8 GHz、内存 8 GB、WINDOWS 10 系统上使用 Matlab 2016a 进行仿真实验.

实验中的参数取值如下:全局混合迭代次数、初始子群数量、子群内个体数量、种群总规模、子群更新次数、结果精确度的调节因子.为验证所提策略的有效性,分别使用基本蛙跳算法 SFLA(算法 1)、改进蛙跳算法 I(SFLA+调整可控精度+异常解的处理)、改进蛙跳算法 II(SFLA+新增筛选准则+异常解的处理)以及所提改进蛙跳算法 ISFLA(SFLA+调整可控精度+新增筛选准则+异常解的处理,即算法 3)求解个性化旅游路线推荐问题的数学模型,4 种算法分别独立地运行 30 次,求取它们在 30 次运行中求得的最少费用的最佳值、均值和方差,比较结果如表 5 所示.

表 2 各景点经纬度

Table 2 List of longitude and latitude of each tourist attraction

景点	经度/(°E)	纬度/(°N)	景点	经度/(°E)	纬度/(°N)
南京长江大桥	118.737 383	32.132 964	夫子庙	118.795 270	32.026 938
栖霞山	118.976 373	32.163 867	南京眼步行桥	118.708 978	32.001 869
颐和路	118.776 349	32.068 715	侵华日军南京大屠杀遇难同胞纪念馆	118.749 884	32.041 149
玄武湖	118.803 993	32.076 375	中华门	118.788 219	32.018 965
鸡鸣寺	118.801 861	32.067 125	大报恩寺	118.789 441	32.015 439
中山陵	118.861 453	32.059 739	雨花台风景区	118.788 054	32.004 004
明孝陵	118.847 373	32.056 812	牛首山	118.750 198	31.904 998
南京大学	118.786 007	32.061 430	南京博物院	118.831 650	32.046 772
总统府	118.803 804	32.050 571	湖南路	118.789 795	32.075 605
新街口	118.790 227	32.047 056			

表3 各景点门票费用

Table 3 List of admission fee for each tourist attraction

元			
景点	门票费用	景点	门票费用
南京长江大桥	31.1	夫子庙	79.2
栖霞山	84.3	南京眼步行桥	79.4
颐和路	100.6	侵华日军南京大屠杀 遇难同胞纪念馆	73.2
玄武湖	135.2	中华门	84.2
鸡鸣寺	129.0	大报恩寺	207.2
中山陵	59.6	雨花台风景区	66.1
明孝陵	183.6	牛首山	155.0
南京大学	59.0	南京博物院	72.6
总统府	135.6	湖南路	80.0
新街口	79.2		

通过观察表5中各算法的运行结果可知,在引入了改进策略后,算法的性能得到了一定程度的提高.与基本蛙跳算法 SFLA 相比,改进蛙跳算法 I 求得的最少费用均值有所减少,方差也有所降低.这是由于改进蛙跳算法 I 引入了控制结果精度的调节因

子,当迭代次数达到设定的中间阈值时,在原种群中加入规模与原种群规模相同的新种群,从而使得搜索范围扩大,群体的多样性增强,有利于增强算法的全局搜索能力.改进蛙跳算法 II 求得的最少费用均值也优于 SFLA.原因是在基本混合蛙跳算法的筛选规则中,如果两个新解都比旧解差,则会直接舍去旧解,这就导致在舍去了旧解后,当前状态下的候选新解也会被舍去很大一部分甚至是全部.而在本文所提筛选规则中,个体在更新时,跳动的实际步长为最大步长乘以一个 $[0,1]$ 内的随机数,通过判断实际步长的幅值大小可以很好地降低丢失具有较大潜力旧解的风险,能够防止陷入局部最优.从表5的结果可以看出,本文所提算法 ISFLA 在同时引入调整可控精度、新增筛选准则及异常解的处理策略后,结合了3个改进措施的优点,使得求得的最少费用的最佳值、均值和方差都得到改善,搜索结果也较 SFLA 和仅引入部分改进策略的改进算法 I 和 II 更加精确,且结果分布的离散程度更小,说明所提算法 ISFLA 搜索的稳定性更强.

表4 各景点游玩时间

Table 4 List of recommended visiting time for each tourist attraction

						h		
景点	开放时段	建议游玩时间	景点	开放时段	建议游玩时间	景点	开放时段	建议游玩时间
夫子庙	全天	2.0	总统府	08:30—17:00(周一闭馆)	3.0	南京长江大桥	07:30—18:00	1.5
中山陵	08:30—17:00	2.0	侵华日军南京大屠杀 遇难同胞纪念馆	08:30—16:30(周一闭馆)	1.5	雨花台	08:00—17:00	3.0
新街口	全天	1.5	大报恩寺	09:00—17:30	2.5	颐和路	全天	1.5
牛首山	09:00—17:00	2.0	南京博物院	09:00—17:00	3.0	中华门	08:00—20:30	1.5
栖霞山	08:30—16:30	2.0	南京大学	全天	2.0	玄武湖	全天	2.0
鸡鸣寺	07:30—17:00	1.5	南京眼	全天	0.5	明孝陵	07:00—18:30	3.0
湖南路	全天	1.5						

表5 算法改进前后结果对比

Table 5 Comparison of least spending before and after the algorithm improvement

元			
算法	最少费用的最佳值	最少费用的均值	最少费用的方差
SFLA	256.7	269.583 3	29.182 8
改进蛙跳算法 I	256.7	265.436 7	28.637 1
改进蛙跳算法 II	256.7	268.086 7	29.479 6
ISFLA	254.7	264.893 3	16.312 4

利用显著度水平为 0.05 的秩和检验法将上述改进蛙跳算法 I、改进蛙跳算法 II、ISFLA 与基本蛙跳算法 SFLA 在 30 次运行中的结果进行了统计测试,结果如表 6 所示.其中“+”表示相应算法的搜索

结果显著优于基本蛙跳算法,“=”表示两者无显著差异,“-”表示显著劣于基本蛙跳算法.

表6 秩和检验结果

Table 6 Results of rank sum test

算法	统计测试结果
改进蛙跳算法 I	+
改进蛙跳算法 II	=
ISFLA	+

通过观察表6的统计测试结果可知,在引入新增筛选准则后,改进蛙跳算法 II 与基本蛙跳算法得到的结果无显著差异,而加入调整可控精度策略的

改进蛙跳算法 I 以及所提改进蛙跳算法 ISFLA 的求解结果均显著优于基本蛙跳算法,进一步说明本文所提改进策略是可行而有效的。

图 4 给出了 4 种对比算法的收敛曲线.由图 4 可见,当总的迭代次数相同时,对于仅引入调整可控精度策略的改进算法 I,虽然它搜索到的结果优于基本蛙跳算法 SFLA,但是其收敛速度相较于 SFLA 并没有明显的提升;对于仅引入新增筛选准则的改进蛙跳算法 II,虽然它对最优解精度的改进并不明显,但是它能够以较快的速度收敛;对于同时引入调整可控精度策略和新增筛选准则的所提算法 ISFLA,它将改进策略的优点有机地结合,既具有较快的收敛速度,也能够跳出局部极值,获得了更高的求解精度,算法的全局搜索性能得到了大幅度地提升。

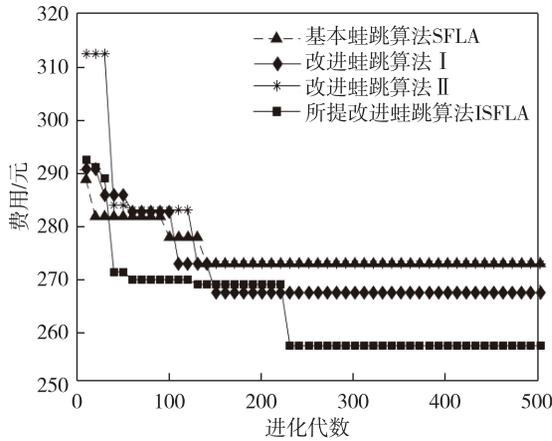


图 4 算法收敛曲线对比

Fig. 4 Comparison of convergence curves

由图 4 还可以看出,当引入改进策略后,所提改进算法 ISFLA 的收敛速度较 SFLA 有小幅度降低,其主要原因是在 ISFLA 的求解过程中,为了避免陷入局部最优,在进化到一定程度时扩大了种群的搜索范围.然而,与 SFLA 相比,ISFLA 能够收敛到精度更优的解,显著提高了算法的求解效率。

进一步,将模型与相应的个性化旅游设置相结合,以花费最少为目标进行优化时得出自然景观、历史人文、美食购物的路线规划如图 5—10 所示。

从图 5—10 可以看出,当旅游者选择游览 3 d 且每天游览 3 个景点,以费用最低为优化目标时,自然景观、历史人文、美食购物 3 种个性化旅游方式下各需总花费 1 031.2、1 012.1 和 1 057.2 元,ISFLA 推荐出的路线能够较好地满足用户的个性需求。



图 5 选择自然景观游且以费用最少为优化目标的推荐路线

Fig. 5 Recommended routes for natural landscape tour with least spending

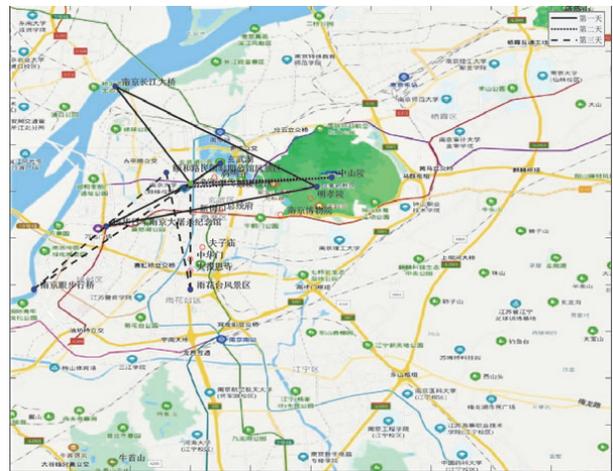


图 6 选择自然景观游且以费用最少为优化目标的轨迹地图

Fig. 6 Trajectory map for natural landscape tour with least spending

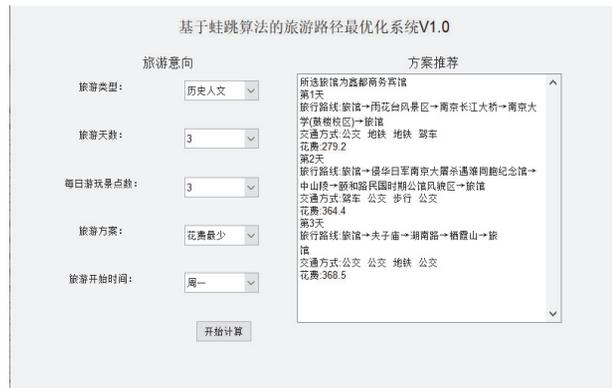


图 7 选择历史人文游且以费用最少为优化目标的推荐路线

Fig. 7 Recommended routes for historical & cultural tour with least spending

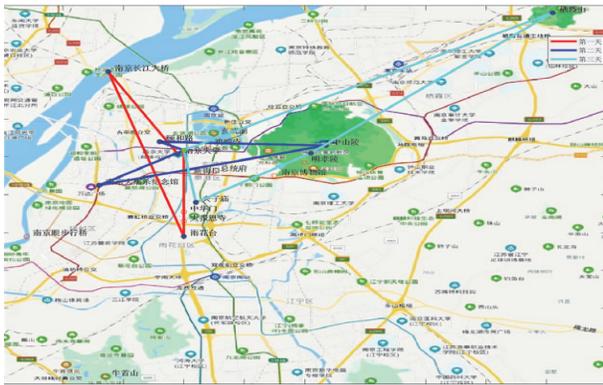


图8 选择历史人文游且以费用最少为优化目标的轨迹地图

Fig. 8 Trajectory map for historical & cultural tour with least spending

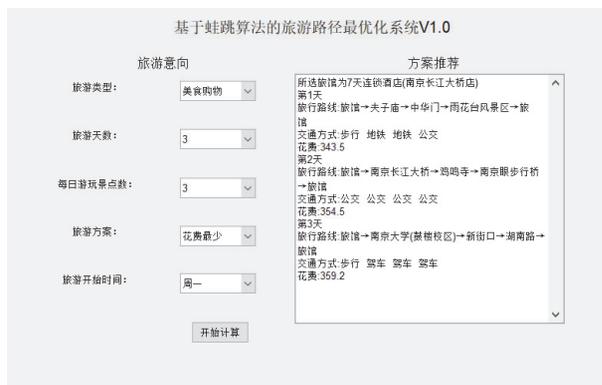


图9 选择美食购物游且以费用最少为优化目标的推荐路线

Fig. 9 Recommended routes for food & shopping tour with least spending

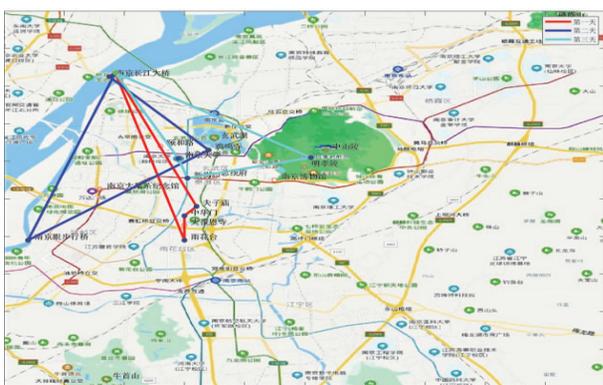


图10 选择美食购物游且以费用最少为优化目标的轨迹地图

Fig. 10 Trajectory map for food & shopping tour with least spending

## 5 总结

本文针对旅游路线个性化推荐问题,设计了一

种改进的混合蛙跳算法.通过调整可控精度,新增筛选准则和即时处理异常解来增加群体多样性、降低遗漏最优解风险,强化了局部搜索能力,提高了算法的求解精度.以南京游作为实例,采用改进混合蛙跳算法进行求解,并根据用户的个性选择推荐出了具体的出行方案,在一定程度上为旅游者的出行提供了参考.以多次运行中的最佳值、均值和方差作为测度,验证了不同改进策略的有效性.实验结果表明,在求解精度方面,所提改进算法相较于基本的蛙跳算法有了显著的提升.然而,本文算法也存在着不足,例如本文采用的是单目标优化,而现实生活中旅游者可能有多个旅游需求需要同时优化,如何高效求解个性化旅游路线推荐的多目标优化问题,将是下一步的研究方向.

## 参考文献

### References

- [ 1 ] 杨云鹏,袁光辉,金阳,等.全国5A级景区旅游路线规划问题研究[J].数学的实践与认识,2016,46(15):74-80  
YANG Yunpeng, YUAN Guanghui, JIN Yang, et al. Research on the national 5A level scenic tourist route planning[J]. Mathematics in Practice and Theory, 2016, 46(15): 74-80
- [ 2 ] Zhu D X, Wu Y J, Wang X D. A dynamic programming algorithm for a generalized LCS problem with multiple subsequence inclusion constraints [ J ]. Internet of Vehicles-Safe and Intelligent Mobility, 2015:439-446
- [ 3 ] Wu C C, Bai D Y, Azzouz A, et al. A branch-and-bound algorithm and four metaheuristics for minimizing total completion time for a two-stage assembly flow-shop scheduling problem with learning consideration[J]. Engineering Optimization, 2020, 52(6): 1009-1036
- [ 4 ] Luo Q, Wang H B, Zheng Y, et al. Research on path planning of mobile robot based on improved ant colony algorithm[J]. Neural Computing and Applications, 2020, 32(6): 1555-1566
- [ 5 ] 杨萍.区域旅游者行为模式及影响研究[J].经济问题探索, 2003(6): 114-117  
YANG Ping. Research on the pattern and influence of district traveler's behavior[J]. Inquiry into Economic Problems, 2003(6): 114-117
- [ 6 ] 明勇,王华军.基于改进混合蛙跳和GIS的城市垃圾车回收路径优化设计[J].计算机测量与控制, 2014, 22(12): 4054-4057  
MING Yong, WANG Huajun. Design for optimization of city rubbish collection route planning based on improved compound frog hop and GIS[J]. Computer Measurement & Control, 2014, 22(12): 4054-4057
- [ 7 ] 黄于欣,蒋洪杰.基于改进蚁群算法的旅游景区路径规划[J].河南科学, 2018, 36(6): 823-829  
HUANG Yuxin, JIANG Hongjie. An improved ant colony

- algorithm for path planning in scenic zones [J]. *Henan Science*, 2018, 36(6): 823-829
- [ 8 ] 杨晓敏.基于蚁群算法的黄河金三角旅游路线规划研究[J]. *计算机时代*, 2018(12): 61-63  
YANG Xiaomin. Research on tourist route planning of the Yellow River golden triangle based on ant colony algorithm [J]. *Computer Era*, 2018(12): 61-63
- [ 9 ] 陈春朝,李恒宇,罗均.基于混合蛙跳算法优化人工势场的路径规划方法研究[J]. *河南理工大学学报(自然科学版)*, 2018, 37(5): 105-110  
CHEN Chunchao, LI Hengyu, LUO Jun. Study on path planning algorithm based on artificial potential field optimized by shuffled frog leaping algorithm [J]. *Journal of Henan Polytechnic University (Natural Science)*, 2018, 37(5): 105-110
- [ 10 ] Shiripour S, Mahdavi-Amiri N, Mahdavi I. Planning a capacitated road network with flexible travel times: a genetic algorithm [J]. *Journal of Mathematical Modelling and Algorithms in Operations Research*, 2015, 14(4): 425-451
- [ 11 ] Reyes-Rubiano L, Calvet L, Juan A A, et al. A biased-randomized variable neighborhood search for sustainable multi-depot vehicle routing problems [J]. *Journal of Heuristics*, 2020, 26(3): 401-422
- [ 12 ] Sánchez-Oro J, López-Sánchez A D, Colmenar J M. A general variable neighborhood search for solving the multi-objective open vehicle routing problem [J]. *Journal of Heuristics*, 2020, 26(3): 423-452
- [ 13 ] 姜慧楠,张向锋.一种人工鱼群-蛙跳混合优化算法[J]. *上海电机学院学报*, 2019, 22(6): 330-336, 344  
JIANG Huinan, ZHANG Xiangfeng. A hybrid artificial fish swarm and shuffled frog leaping algorithm [J]. *Journal of Shanghai Dianji University*, 2019, 22(6): 330-336, 344
- [ 14 ] 万慧云,蒋艳.基于蚁群算法的5A景点旅游路线规划问题研究[J]. *软件导刊*, 2019, 18(4): 141-144  
WAN Huiyun, JIANG Yan. Research on China's 5A scenic spot tourism route planning based on ant colony algorithm [J]. *Software Guide*, 2019, 18(4): 141-144
- [ 15 ] 杨静.旅游路线的优化设计研究:以中国201个5A级景区为例[J]. *新经济*, 2016(11): 24  
YANG Jing. Study on the optimal design of tourist routes: a case study of 201 5A scenic spots in China [J]. *New Economy*, 2016(11): 24

## Personalized travel route recommendation based on an improved shuffled frog leaping algorithm

SHEN Xiaoning<sup>1</sup> WANG Senlin<sup>1</sup> WU Junchao<sup>1</sup> QIU Youhui<sup>1</sup> ZHANG Lei<sup>1</sup> LI Changfeng<sup>1</sup> WANG Yufang<sup>1</sup>

<sup>1</sup> School of Automation/Collaborative Innovation Center of Atmospheric Environment and Equipment Technology/Jiangsu Key Laboratory of Big Data Analysis Technology, Nanjing University of Information Science & Technology, Nanjing 210044

**Abstract** Tourists expect to get travel experiences like low-cost, convenient itinerary, high comfort and so on. Meanwhile, they have different tourist interests such as history and culture, natural landscape, food and shopping, etc. Therefore, a personalized travel route recommendation method based on an improved shuffled frog leaping algorithm is proposed in this paper. A model is established to optimize the personalized travel route recommendation problem, and an improved shuffled frog leaping algorithm is designed based on the characteristics of the model. By adjusting the controllable accuracy, incorporating new selection criteria and handling abnormal solutions in time, the population diversity is increased and the risk of missing the optimal solution is reduced, which enhance the local search ability and searching accuracy of the algorithm. The personalized travel route recommendation for three-day tour in Nanjing is taken as an instance to verify the proposed method. Relevant data are collected, including admission fees, opening hours, and accommodation expenses for well-known tourist attractions in Nanjing. The results show that compared with basic shuffled frog leaping algorithm, the proposed method can recommend touring routes with higher accuracy and better meet the users' individual interests.

**Key words** personalized travel; route recommendation; shuffled frog leaping algorithm; screening criteria; controllable accuracy