

基于改进樽海鞘算法的共享单车分布密度优化

周川

江苏理工学院商学院 江苏常州 213001

摘要 针对城市共享单车分布密度优化问题,提出了一种改进樽海鞘算法。首先,将共享单车分布密度优化问题转换成函数优化问题,以等待时间、花费时间、费用及安全代价为评价指标,建立目标函数。其次,引入一维正态云模型和非线性递减控制策略来改进樽海鞘算法中引领者的搜索机制,增强对局部数据的挖掘能力;引入自适应策略来改进原算法跟随者搜索机制,避免算法陷入局部最优值。最后,通过标准测试函数以及共享单车分布密度优化仿真对所提优化算法的有效性进行了验证,结果表明:相比原樽海鞘算法、萤火虫算法及人工蜂群算法,改进的樽海鞘算法具有更好的稳定性和全局搜索能力,能够更好地实现对共享单车分布密度的优化,提升共享单车的区域利用率,对智慧交通的发展有一定的参考价值。

关键词 共享单车;分布密度优化;樽海鞘算法;自适应策略;云模型

中图分类号 TN911.1-34,TP18

Optimization of Sharing Bicycle Density Distribution Based on Improved Salp Swarm Algorithm

ZHOU Chuan

School of Business, Jiangsu University of Technology, Changzhou, Jiangsu 213001, China

Abstract In this article, an improved sea-squirt algorithm is proposed for the urban bike-sharing distribution density optimization problem. First, the sharing bicycle distribution density optimization problem is converted into a functional optimization problem, and the objective function of optimization is established with waiting time, time spent, cost and safety cost as evaluation indexes. Secondly, a one-dimensional normal cloud model and a nonlinear decreasing control strategy are introduced to improve the leader search mechanism in the Bottleneck algorithm to enhance the mining ability of local data; an adaptive strategy is introduced to improve the follower search mechanism of the original algorithm to avoid the algorithm falling into the local optimum. Finally, the effectiveness of the proposed optimization algorithm is verified by the standard test function and the simulation of shared bicycle distribution density. The results show that the improved Bottlenose sheath algorithm has better stability and global search capability than the original algorithm, firefly algorithm and artificial bee colony algorithm, and can better optimize the distribution density of shared bicycles and improve the regional utilization rate of shared bicycles, which is a reference value for the development of intelligent transportation. It has certain reference value for the development of intelligent transportation.

Keywords Public bicycle, Distribution density optimization, Salp swarm algorithm, Adaptive strategy, Cloud model

1 引言

随着信息技术、人工智能技术、现代管理技术的发展,融合了系统服务和资源共享的智慧城市模式越来越受到政府的重视。智慧城市包括智慧人文、智慧应用、智慧交通等内容,其中智慧交通与市民日常出行息息相关,一直是国内外学者们研究的热点^[1]。作为一种典型的智慧交通系统,共享单车可以为小区域内的市民出行带来便捷,有效缓解了城市交通拥堵问题,促进了社会与经济的发展^[2]。然而,共享单车也面临着不少问题,其中之一就是如何合理安排其分布密度以便最大化地满足区域用户需求以及优化公共场地的占地。

近年来,有学者将共享单车分布密度问题转换成多元函数优化问题,并引入智能优化算法来寻找最优解。例如,Zuo利用蚁群算法来优化广州越秀区15个共享单车停放点分布,仿真结果表明该算法能获得简洁、便捷的最优路径^[3]。

Duan等根据大数据统计共享单车的骑行路线,并引入贪婪算法来权衡单车用户的时间成本与费用成本^[4]。Jia等利用人工蜂群(Artificial Bee Colony Algorithm, ABC)算法来优化高峰期的共享单车分布,提高单车的利用率并降低区域安全事故发生率^[5]。但是上述智能算法极易陷入局部最优值,从而导致算法前期搜索不足、后期收敛停滞的问题。

樽海鞘算法(Salp swarm algorithm, SSA)是由Mirjalili团队于2017年提出的一种模拟樽海鞘群的协同运动与觅食行为的新兴元启发式人工智能算法^[6],被广泛用于工程优化设计^[7]、特征选择^[8]、参数辨识^[9]等优化问题中。与常见的人工智能算法相比,SSA算法具有结构参数少、鲁棒性好、收敛速度快等优点。然而,传统SSA算法也存在收敛停滞与易陷入局部最优值的不足。本文引入一维正态云模型与自适应策略来改变SSA算法固有结构,增强其优化能力,并利用改进算法解决共享单车分布密度优化问题。最后,通过仿真算例

基金项目:江苏省自然科学基金(BK20170315)

This work was supported by the Natural Science Foundation of Jiangsu Province, China(BK20170315).

通信作者:周川(nufe_zc@163.com)

验证了改进后算法的有效性。

2 问题描述

图1为共享单车市场的供求关系调研图^[10],线 ξ_i ($i=1, 2$)为共享单车的市场需求量,线 δ_i ($i=1, 2$)为共享单车的供应量。当共享单车需求量增加(由线 ξ_1 变为线 ξ_2),市场供求的平衡点从0变为1时,共享单车的供求均会增加,但费用会下降,共享单车数量也会由 G_2 增加至 G_3 。同时,随着共享单车数量的增加,土地租赁费用也会增加(由 P_2 变为 P_1),这又会导致共享单车数量从 G_3 降为 G_1 或 G_4 。由此可见,土地租赁费用是制约共享单车数量的关键因素^[11],故可通过建立数学模型权衡费用与共享单车分布数量的关系,对共享单车的分布进行优化,实现资源的合理利用。

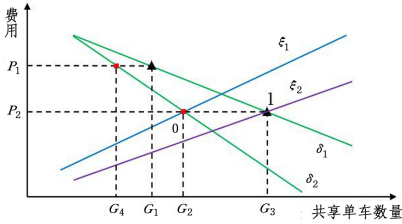


图1 共享单车市场的供求关系

Fig. 1 Supply and demand in the public bicycle market

定义A点与B点之间的行人数量为 N_{AB} ,两点之间的往返模式有步行、汽车、公交和打的4种,设每种模式选择的概率一样(均为0.25),则两点间选择共享单车的人数为:

$$N_{AB,P} = 0.25 N_{AB} \quad (1)$$

设定 D 为共享单车密度,即自行车停放区长度与人行道长度之比,即:

$$D = \frac{L_0}{L_1} = \frac{0.3(N_B - 1) + 0.2N_B}{L_1} \quad (2)$$

其中, L_0 为自行车停放区长度, L_1 为人行道长度,停车时两车间距为0.3m,智能锁的长度为0.2m, N_B 为共享单车数量。

设定共享单车购买、定位及维保的总费用为:

$$C_F = D \times L_1 (C_\rho + C_M Y + \log_e DL_1) \quad (3)$$

其中, C_ρ 为单车购买费用, C_M 为单车定位与维保费用, Y 为单车使用寿命, $\log_e DL_1$ 表征共享单车数量与总体运营成本之间的关系, e 为成本与效益之间的权重因子。

同时,共享单车的土地租赁代价 C_O 也决定着共享单车的分布,其可由行人步行到共享单车停放区域的时间代价 C_T 与行走的安全代价 C_S 来简化描述,即:

$$C_O = C_T + C_S \quad (4)$$

行人的行走速度 v 与行人密度 ρ 的关系可由下式计算:

$$v = \begin{cases} 1.27, & \rho < 0.05 \\ -0.25 \ln \rho + 0.80, & 0.05 \leq \rho < 3 \\ -0.5\rho + 2, & 3 \leq \rho < 4 \\ 0, & \rho \geq 4 \end{cases} \quad (5)$$

行人密度可由下式描述:

$$\rho = \frac{N_{AB}}{L_1 W - L_0 l} \quad (6)$$

其中, W 为人行道的宽度, l 为共享单车的长度。

联立式(2)和式(5),可计算行人步行时间 T_w 为:

$$T_w = \frac{L_0}{v} - \frac{L_0}{1.27} \quad (7)$$

进而,时间代价 C_T 的具体形式为:

$$C_T = \gamma T_w \quad (8)$$

其中, γ 为时间代价转换系数。

用户行走的安全代价也可进一步描述为^[12]:

$$C_S = N_{AB} P_S C_I \quad (9)$$

$$P_S = \theta \cdot \max \left\{ \frac{\bar{\rho} - \rho}{\rho - \underline{\rho}}, 0 \right\} \cdot \Delta t \quad (10)$$

其中, P_S 表征事故发生的概率, θ 表征事故发生的激发因子, $\bar{\rho}$ 与 $\underline{\rho}$ 为行人密度的上下限, Δt 为事故持续时间, C_I 为因事故导致人员与财产的损失。

用户选择共享单车模式的费用 C_3 可以进一步描述为:

$$C_3 = \nu_1 C_T + \nu_2 C_O \quad (11)$$

其中, ν_1 与 ν_2 为权重系数。

另外,设A点与B点之间选择共享单车的总代价为 C ,其与等待时间 C_1 、花费时间 C_2 与费用 C_3 有关,即:

$$C = \eta_1 C_1 + \eta_2 C_2 + \eta_3 C_3 \quad (12)$$

其中, η_1 、 η_2 和 η_3 为权重系数。

联立式(1)与式(12)可得共享单车分布密度优化问题的目标函数为:

$$O = CN_{AB,P} \quad (13)$$

其中,当共享单车分布密度 D 不变时,目标函数值 O 越大,说明单车布局越合理,单车的使用率越高。因此,该优化问题就是找到合适的共享单车分布密度使得行人用户数量最大。

3 算法描述

3.1 传统樽海鞘算法

在传统SSA算法结构中存在引领者与跟随者两种个体,前者指挥后者以一种链式行为趋近食物源^[13]。引领者执行全局处理,跟随者进行局部搜索,以保证整个算法能够朝着潜在最优解的方向进化。SSA算法具体的步骤如下所示:

步骤1 初始化算法条件,设置SSA的种群数目为 N ,优化问题的维度为 Ω ,截止迭代次数为 T_{max} ,樽海鞘个体的初始位置为 $x_i = [x_i^1, \dots, x_i^\Omega]$ ($i=1, \dots, N$)。

步骤2 定义第 j ($j=1, \dots, \Omega$)个樽海鞘群获取食物源的适应度值为 F_j (SSA算法的目标函数),其与目标函数的关系为:

$$F_j = \frac{1}{1 + \text{abs}(O_j)} \quad (14)$$

步骤3 种群中引领者根据下式来更新当前位置信息。

$$x_i^j = \begin{cases} F_j + c_1 \cdot [(ub_j - lb_j) \cdot c_2 + lb_j], & c_3 \geq 0.5 \\ F_j - c_1 \cdot [(ub_j - lb_j) \cdot c_2 + lb_j], & c_3 < 0.5 \end{cases} \quad (15)$$

其中, x_i^j 为引领者个体位置, ub_j 与 lb_j 为个体在 j 维位置中的上下限。 c_2 和 c_1 为(0,1)之间的随机数。 c_1 为权重系数,用来平衡全局搜索与局部搜索的能力,其表达式为:

$$c_1 = 2e^{-(4 \cdot T / T_{max})^2} \quad (16)$$

其中, T 为当前累计迭代次数。

步骤4 依据牛顿第二定律来更新种群中跟随者位置。

$$x_i^j = 0.5at_a^2 + v_0 t_a + x_i^j \quad (i \geq 2) \quad (17)$$

其中, t_a 为迭代步长, v_0 为运动初始速度, $a = (v_{final} - v_0) / t_a$ 为加速度, $v_{final} = (x_i^{j-1} - x_i^j) / t_a$ 。如果 $v_0 = 0$,式(17)则简化成:

$$x_i^j = 0.5(x_i^j + x_i^{j-1}) \quad (18)$$

根据式(15)和式(18)可模拟樽海鞘种群的觅食机制,比较当前迭代中的解和已获得的全局最优解,优胜劣汰,保留较好的食物源位置和适应度值进入到下一代。

步骤5 记录当代优化结果,然后跳转至步骤3继续迭代,直至 $T=T_{\max}$ 。

3.2 改进机制

传统 SSA 算法中,仅仅靠权重系数无法较好地平衡局部搜索与全局搜索的过程,故算法极易陷入局部最优值,从而导致进化停滞。为解决这一问题,本文引入自适应策略与一维正态云模型来改进种群中跟随者和引领者位置更新策略^[14],进而得到一种改进的算法(Improved SSA, ISSA)。具体过程如下:

借助一维正态云模型的随机性和模糊性来获取当前引领者位置 x_i^j 的领域解 nx_i^j ,提高 SSA 算法的局部数据挖掘能力,其数学描述为:

$$\begin{cases} Ex = x_i^j \\ En = ex^j \\ He = En/10 \end{cases} \quad (19)$$

其中, Ex 为期望值,表征定性概念模糊度的度量; En 为熵值,表征领域解 nx_i^j 分布广度,其值越大,分布越广; He 为超熵,表征领域解 nx_i^j 取值密度,其值越大,密度越小; ex^j 为中间变量。

例如,设定 $Ex=0.5$,选择不同的熵值与超熵,由式(19)可以计算出当前适应度的 500 个领域解,如图 2 所示。可以看出,熵值与超熵影响云模型的搜索范围与离散程度,即熵值越大,算法搜索范围越广,超熵越大,待开采的领域解越离散。

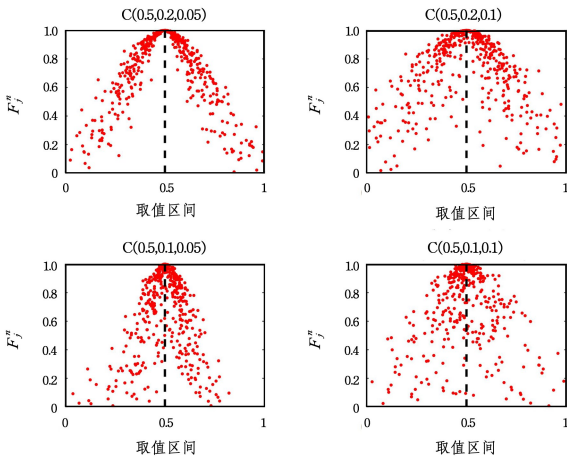


图2 产生 500 个云滴的云模型

Fig.2 Cloud model with 500 cloud droplets

当 SSA 算法迭代至一定次数后,搜索到的当前解会接近理论最优解。为控制搜索范围并提高搜索精度,引入一种非线性递减控制策略来调整 ex^j 的取值^[15]:

$$ex^j = -(ub_j - lb_j) \cdot (T/T_{\max})^2 + ub_j \quad (20)$$

同时,为帮助 SSA 算法跳出局部最优的束缚,并在前期拥有较强的全局搜索能力、后期能获得相对精确的解,在跟随者位置更新阶段引入自适应策略^[16],其数学表达式为:

$$\begin{cases} x_i^j = w(T) \cdot (x_i^j + x_{i-1}^j) \\ w(T) = w_{\max} \cdot rand - \frac{T}{T_{\max}}(w_{\max} - w_{\min}) \end{cases} \quad (21)$$

其中, w_{\max} 和 w_{\min} 分别为权重系数 w 的上、下限, $rand \in (0, 1)$ 。若 w 的值过大,则算法搜索效率低,局部数据挖掘能力弱,不易得到最优解;反之,若 w 的值过小,算法收敛精度高,但全局搜索能力偏弱,容易陷入局部最优。通过自适应策略可有效处理上述问题,使得 SSA 算法具有较好的性能。

ISSA 算法的流程图如图 3 所示。

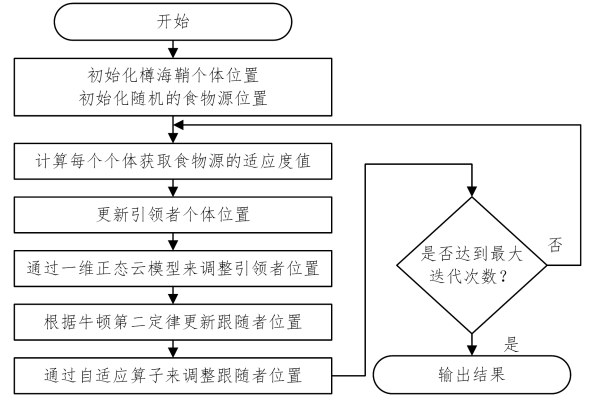


图3 ISSA 算法的流程图

Fig.3 Flow chart of ISSA

4 仿真分析

4.1 标准测试函数

为了测试本文所提 ISSA 算法的有效性,采用 3 个标准测试函数进行性能测试^[17]。同时,引入萤火虫算法(Glowworm Swarm Optimization, GSO)算法、ABC 算法、SSA 算法与 ISSA 算法进行比较。3 个标准测试函数的维数均为 10,其形式如下所示:

$$\begin{cases} f_1(x) = \sum_{i=1}^n x_i^2, & x \in [-100, 100] \\ f_2(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2, & x \in [-100, 100] \\ f_3(x) = \sum_{i=1}^n ix_i^2, & x \in [-10, 10] \end{cases} \quad (22)$$

4 种算法各运行 500 次,重复 3 次,并取平均值作为最终结果。4 种算法的结构参数如表 1 所列。这里需说明的是, GSO 算法和 ABC 算法解结构参数的解释可见参考文献^[18]和文献^[19]。

表1 4 种算法的结构参数

Table 1 Structural parameters of the four algorithms

算法	结构参数
GSO	$M=20, \gamma=0.6, \rho=0.4, s=0.03, \beta=0.1, n_r=8$
ABC	$SN=20, limit=50$
SSA	$N=20$
ISSA	$N=20$

标准测试函数的测试结果如表 2 所列。若不加改进算子, SSA 算法的寻优效果要弱于 GSO 算法与 ABC 算法,但增加一维正态云模型算子和自适应策略后, ISSA 在数据开采与数据挖掘方面的能力明显提升,在标准测试函数优化上,优化的最大值(Max)、最小值(Min)、平均值(Mean)和标准差(Std) 4 个评价指标是最小的,较 GSO 算法、ABC 算法和 SSA 算法有显著的量级提升。这表明, ISSA 算法具有较好的稳定性和全局搜索能力,能够深度挖掘数据。

表2 标准测试函数的测试结果

Table 2 Results of the benchmark functions

函数	算法	Max	Min	Mean	Std
f_1	GSO	4.31×10^{-10}	1.01×10^{-12}	1.59×10^{-10}	8.72×10^{-11}
	ABC	6.19×10^{-12}	2.15×10^{-14}	3.24×10^{-12}	1.60×10^{-8}
	SSA	1.82×10^{-9}	3.31×10^{-10}	9.57×10^{-10}	3.18×10^{-10}
	ISSA	5.81×10^{-23}	8.45×10^{-32}	4.81×10^{-28}	3.36×10^{-30}
f_2	GSO	2.53×10^{-5}	4.02×10^{-9}	5.51×10^{-7}	1.21×10^{-6}
	ABC	6.51×10^{-10}	4.11×10^{-15}	5.09×10^{-13}	1.21×10^{-10}
	SSA	7.13×10^{-2}	2.52×10^{-9}	5.48×10^{-7}	1.53×10^{-6}
	ISSA	9.60×10^{-46}	7.04×10^{-50}	1.18×10^{-48}	7.35×10^{-47}
f_3	GSO	9.99×10^{-10}	1.17×10^{-11}	1.06×10^{-10}	3.63×10^{-11}
	ABC	1.16×10^{-15}	6.36×10^{-18}	3.87×10^{-16}	4.22×10^{-16}
	SSA	4.42×10^{-10}	1.15×10^{-11}	1.07×10^{-10}	9.08×10^{-11}
	ISSA	5.18×10^{-30}	1.39×10^{-33}	8.96×10^{-31}	1.11×10^{-28}

4.2 共享单车分布密度优化仿真

本仿真算例借鉴文献[20]的例子,以某大学校园为例来研究其附近的共享单车的优化分布问题。从校园东门(A点)到地铁站口(B点),有500人次,东门与地铁站附近的人行道长度为100m,另外每辆共享单车的购买费用为0.0580万元、维保费用为0.0160万元/年、使用寿命为10年。仿真参数如表3所列。

表3 仿真参数

Table 3 Simulation parameters

参数	参数含义	数值
$\bar{\rho}$	行人密度上限/(人/m ²)	2.8
$\underline{\rho}$	行人密度下限/(人/m ²)	0.02
ρ	当前行人密度/(人/m ²)	0.50
Δt	事故持续时间/min	10
C_I	因事故导致人员与财产的损失/万元	0.5
θ	事故发生的激发因子	0.1
v	行人的行走速度/(m/s)	1.27
ϵ	成本与效益之间的权重因子	0.8
v_1	权重系数	0.2
v_2	权重系数	0.8
W	人行道的宽度/m	2
l	共享单车的长度/m	0.86
γ	时间代价转化系数	0.3
η_1	权重系数	0.02
η_2	权重系数	0.05
η_3	权重系数	0.03

仿真中,共享单车数量的搜索范围为20~100辆,引入ABC算法和GSO算法来与ISSA算法作比较,迭代总次数为50,每种算法运行3次,保留其中最好的结果。3种算法的迭代曲线如图4所示,可以看出ISSA算法的收敛速度要明显优于其他两种算法,同时ISSA算法、ABC算法和GSO算法获得的最优目标函数值分别为0.0026、0.0034与0.0038。ISSA算法比其他两种算法的优化能力分别高出了30.77%和46.15%,此时最优的共享单车量为20辆。

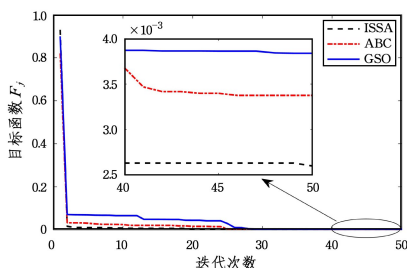


图4 3种算法的迭代曲线

Fig. 4 Iteration curves of the three algorithms

进一步地,分析行人密度对共享单车最优分布密度的影响。选取行人密度从0.05人每平方米步进到2.5人每平方米,步进值为0.1,共25个点,利用ISSA算法来优化共享单车分布密度,如图5所示。可以看出,随着行人密度的增加,共享单车最优分布密度上升。当行人密度为1.25人每平方米至1.75人每平方米时,共享单车数量会陡增,这可能是由于行人数量过多时引起安全事故的概率会增加,故急需大量增加单车的数量;当行人密度超过1.85人每平方米时,共享单车分布密度基本稳定,此时需要考虑增加共享停放区域来解决共享单车饱和问题。

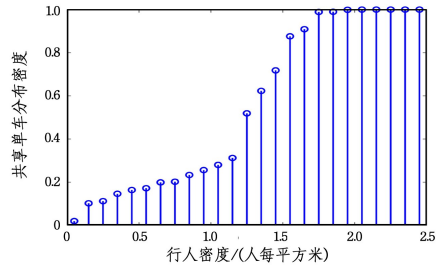


图5 行人密度与共享单车分布密度的关系

Fig. 5 Relation between the pedestrian density and bike-sharing distribution density

接着,分析行人数量对共享单车最优分布密度的影响。选取行人数量从20人步进到200人,步进值为20,共10个点,利用ISSA算法来优化共享单车分布密度,如图6所示。可以看出,随着行人数量的增加,共享单车的最优分布密度也基本呈线性增加趋势,不过要指出的是过分地增加共享单车的数量,总体运营成本也会增加。

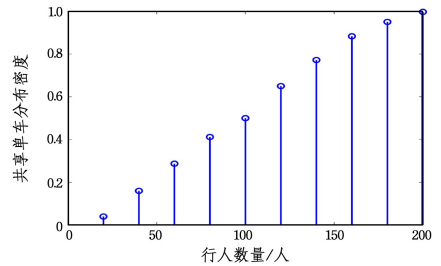


图6 行人数量与共享单车分布密度的关系

Fig. 6 Relation between the pedestrian quantity and bike-sharing distribution density

最后,对共享单车分布密度优化模型进行了分析,即探讨权重系数 v_1, v_2 和事故发生的激发因子 θ 对共享单车最优分布密度 D_s 、单车数量 N_B 及行人密度的影响。设定A点到B点往返人次为1500,人行道长度为1000m,有2条人行道,共享单车数量的搜索范围为100~1000辆,利用ISSA算法计算的结果如表4和表5所列。从表4可以看出,随着 v_1 的增加,时间代价 C_T 上升,需要增加共享单车的数量,故最优分布密度 D_s 也随之上升,此时能够满足更大密度的用户,不过也会带来更多的安全事故。从表5可以看出,随着共享单车最优分布密度的增加,事故发生的概率上升,当 θ 值超过0.7时,应该增加新的共享单车停放区,实现用户的分流,以降低安全事故的发生率。

表4 权重系数敏感度分析

Table 4 Sensitivity analysis for weight factors

指标	(v_1, v_2)				
	(0.1, 0.9)	(0.2, 0.8)	(0.3, 0.7)	(0.4, 0.6)	(0.5, 0.5)
D_s	0.113	0.264	0.377	0.420	0.482
N_B	323	377	438	581	666
ρ	0.344	0.356	0.378	0.392	0.401

表5 事故发生的激发因子敏感度分析

Table 5 Sensitivity analysis for the accident occur rate

指标	θ				
	0.1	0.3	0.5	0.7	0.9
D_s	0.121	0.255	0.483	0.499	—
N_B	202	487	729	883	—
ρ	0.218	0.375	0.399	0.465	—

结束语 本文提出了一种改进的樽海鞘优化算法来解决城市共享单车最优分布密度问题,主要结论如下:

(1)标准测试函数结果表明,ISSA算法在Max, Min, Mean和Std 4个评价指标上均比GSO算法、ABC算法和SSA算法优越,其具有较好的稳定性和全局数据开采能力。

(2)共享单车分布密度优化仿真结果表明,ISSA算法比ABC算法和GSO算法能更好、更快地找到从校园东门(A点)到地铁站口(B点)的最优共享单车优化密度。

(3)根据行人密度和行人数量可以合理控制共享单车分布密度。需要综合权衡共享单车密度、运营成本以及安全代价之间的关系,从而更合理地布局区域内共享单车的数量。

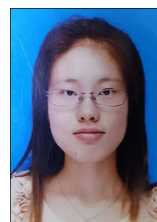
在今后的研究中,还会考虑更多的影响因素,进一步完善共享单车分布模型,同时研究其他智能优化算法或神经网络来解决共享单车最优化分布问题。

参考文献

- [1] IRIMTAT A, KREJCAR O, KERTESZ A, et al. Future trends and current state of smart city concepts: A survey[J]. IEEE Access, 2020, 8: 86448-86467.
- [2] LI X P, LIU L, WANG C. Research on scheduling optimization problem of shared bikes[J]. Mathematics in Practice and Theory, 2021, 51(6): 30-40.
- [3] ZUO N N. Study on urban sharing bike optimization distribution based on swarm intelligent optimization algorithm[J]. Modern Electronics Technique, 2021, 44(1): 115-119.
- [4] DUAN Y, WU J, ZHENG H. A greedy approach for vehicle routing when rebalancing bike sharing systems[C]// 2018 IEEE Global Communications Conference (GLOBECOM). IEEE, 2018: 1-7.
- [5] JIA H, MIAO H, TIAN G, et al. Multi-objective bike repositioning in bike-sharing systems via a modified artificial bee colony algorithm[J]. IEEE Transactions on Automation Science and Engineering, 2019, 17(2): 909-920.
- [6] DING L, GAO Z Q, YU Q. Optimal attitude control for quadrotor aircraft based on improved salp swarm algorithm[J]. Transactions of the Chinese Society for Agricultural Machinery, 2019, 50(10): 243-250.
- [7] MIRJALILI S, GANDOMI A H, MIRJALILI S Z, et al. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems[J]. Advances in Engineering Software, 2017, 114: 163-191.
- [8] HEGAZY A E, MAKHLOUF M A, EL-TAWEL G S. Im-

proved salp swarm algorithm for feature selection[J]. Journal of King Saud University-Computer and Information Sciences, 2020, 32(3): 335-344.

- [9] ZHANG J, WANG Z, LUO X. Parameter estimation for soil water retention curve using the salp swarm algorithm[J]. Water, 2018, 10(6): 815-825.
- [10] CHEN J, WANG W, CHEN X, et al. Research on the layout of bike rental stations around a railway station[J]. Journal of Wuhan University of Technology (Transportation Science & Engineering), 2013, 37(6): 1206-1210.
- [11] CAI S, LONG X, LI L, et al. Determinants of intention and behavior of low carbon commuting through bicycle-sharing in China[J]. Journal of Cleaner Production, 2019, 212: 602-609.
- [12] HOOGENDOORN S P, DAAMEN W. Free speed distributions for pedestrian traffic[C]// Proc. 85th Annual Meeting of Transportation Research Board. Washington DC, 2006: 22-26.
- [13] FARIS H, MIRJALILI S, ALJARAH I, et al. Salp swarm algorithm: theory, literature review, and application in extreme learning machines[J]. Nature-inspired Optimizers, 2020: 185-199.
- [14] YAN F, XU K. Methodology and case study of quantitative preliminary hazard analysis based on cloud model[J]. Journal of Loss Prevention in the Process Industries, 2019, 60: 116-124.
- [15] LI R X, DING L. Path planning for unmanned air vehicles using artificial bee colony algorithm based on cloud model[J]. Computer Science, 2015, 42(S2): 89-92.
- [16] WU J, NAN R, CHEN L. Improved salp swarm algorithm based on weight factor and adaptive mutation[J]. Journal of Experimental & Theoretical Artificial Intelligence, 2019, 31(3): 493-515.
- [17] ÖZYÖN S, YAŞAR C, TEMURTAŞ H. Incremental gravitational search algorithm for high-dimensional benchmark functions[J]. Neural Computing and Applications, 2019, 31(8): 3779-3803.
- [18] SAMPATHKUMAR A, MULERIKKAL J, SIVARAM M. Glowworm swarm optimization for effectual load balancing and routing strategies in wireless sensor networks[J]. Wireless Networks, 2020, 26(6): 4227-4238.
- [19] CHEN J, YU W, TIAN J, et al. Image contrast enhancement using an artificial bee colony algorithm[J]. Swarm and Evolutionary Computation, 2018, 38: 287-294.
- [20] YANG Z, SUN Y, LI J, et al. Optimization of Public Bi-cycle Distribution Density Considering the Price Curve of Public Space Occupancy[J]. Journal of Urban Planning and Development, 2020, 146(3): 1-9.



ZHOU Chuan, born in 1990, master, associate researcher. Her main research interests include intelligent algorithm and data mining.