求解 VRP 问题的混沌模拟退火萤火虫算法

胡云清
（山西交通职业技术学院，太原 030031）

摘要：目的 使萤火虫优化算法（GSO）能够适用于车辆路径问题（VRP）的求解，同时提高该算法的求解性能。**方法** 通过对 GSO 算法的改进，提出求解 VRP 问题的混沌模拟退火萤火虫优化算法（CSAGSO）。首先，设计改进的 GSO 算法（IGSO）使 IGSO 算法能够适应 VRP 问题的求解；其次，在 IGSO 算法中引入模拟退火机制，提出模拟退火萤火虫优化算法（SAGSO），使 IGSO 算法可有效避免陷入局部极小并最终趋于全局最优。然后，在 SAGSO 算法中引入混沌机制，提出 CSAGSO 算法，对 SAGSO 算法的荧光素浓度值进行混沌初始化和混沌扰动；最后，对标准算例集进行仿真测试。**结果** 与遗传算法、蚁群算法和粒子群算法相比，CSAGSO 算法的全局寻优能力、收敛速度及稳定性均改善了 50%以上。**结论** 对 GSO 算法的改进是合理的，且 CSAGSO 算法的全局优化能力、收敛速度和稳定性均优于遗传算法、蚁群算法和粒子群算法。

关键词：车辆路径问题；萤火虫优化算法；模拟退火
中图分类号：TP301 **文献标识码：**A **文章编号：**1001-3563(2017)07-0216-06

A Chaotic Simulated Annealing Glowworm Swarm Algorithm for Solving VRP Problem

HU Yun-qing
(Shanxi Traffic Vocational and Technical College, Taiyuan 030031, China)

ABSTRACT: The work aims to enable the glowworm swarm optimization (GSO) algorithm to be applied to the solution to the vehicle routing problem (VRP) and improve the solution performance of GSO algorithm. Based on the improvement of GSO algorithm, the chaotic simulated annealing GSO (CSAGSO) algorithm was put forward to solve the VRP. Firstly, the improved GSO (IGSO) algorithm which enabled the IGSO algorithm to adapt to the solution to VRP was designed; secondly, the simulated annealing mechanism was introduced into the IGSO algorithm, and the simulated annealing GSO (SAGSO) algorithm was proposed, which made the local optimal solution of IGSO algorithm jump out of local optimum. Then, the chaotic mechanism was introduced into the SAGSO algorithm, and the CSAGSO algorithm was proposed, which carried out the chaos initialization and chaos perturbation of the fluorescein concentration value of the SAGSO algorithm. Finally, simulation tests were carried out on a standard example set. Compared with genetic algorithm, ant colony algorithm and particle swarm optimization algorithm, the global optimization ability, convergence rate and stability of CSAGSO algorithm were improved by more than 50%. The improvement of GSO algorithm is reasonable, and the global optimization ability, convergence rate and stability of CSAGSO algorithm are better than those of the genetic algorithm, ant colony algorithm and particle swarm optimization algorithm.

KEY WORDS: vehicle routing problem; glowworm optimization algorithm; simulated annealing

物流活动包括包装、仓储、配送、装卸、流通加工等环节。其中，配送环节直接影响着物流活动的成本和服务质量，是物流活动的重要组成部分[1]，因此，国内外学者对物流配送问题进行了大量研究，其中的车辆路径问题（VRP）及求解算法更成为了理论界的研究热点之一[2—3]。近年来，学者们从仿生学机理中受到启发，提出了许多求解 VRP 问题的群智能优化算法，如遗传算法[3] (GA)、蚁群算法[4](ACO)、粒子群算法[5] (PSO)和模拟退火算法(SA)等。其中，SA 算法是由 N.Metropolis 模拟固体物质退火机制而提出的

收稿日期：2016-05-17
作者简介：胡云清(1981—)，女，硕士，山西交通职业技术学院讲师，主要研究方向为物流管理。

一种智能算法，其演算机制与组合优化问题十分相似，主要被学者们应用于 TSP、VRP 等组合优化问题的求解[1]。与其他智能算法相比，SA 算法最大的优点在于可有效避免陷入局部极小并最终趋于全局最优，有较强的全局优化能力。萤火虫优化算法(GSO)是 Krishnand 和 Ghose 于 2005 年提出的一种新型仿生群智能优化算法[6]。该算法思想来源于自然界中萤火虫通过发光来寻偶或者觅食，且萤火虫发出的光越亮越绚丽，其吸引力越大，最终大部分萤火虫聚集在几个位置的现象。经过多年的发展，GSO 算法已经在多模态函数优化、预测方法改进、多源信号追踪及定位等领域中得到了成功的应用。例如吴伟民等[7]提出了求解多模态函数优化的改进 GSO 算法。该算法在 GSO 算法基础上，引入了尝试性移动策略，并以邻域平均距离为参考，对个体步长进行了调整，弥补了 GSO 算法峰值发现率低、收敛速度慢和求解精度不高的缺点；吕明[8]利用 GSO 算法对 BP 神经网络的初始权值和阀值进行了优化，提高了 BP 神经网络方法的预测精度；欧阳桢等[9]则设计了一种基于 GSO 算法的匹配跟踪算法，改善了噪声干扰情况下的声音辨识的结果。随着学者们对 GSO 算法的研究不断深入，不管是实践方面的应用成果，还是理论方面的探讨文献都越来越多，研究范围也越来越广泛。分析现有研究可知，截止目前为止，GSO 算法多用于连续型论域中的目标优化问题，缺少对离散型问题，尤其是 VRP 问题的研究。同时，GSO 算法存在容易陷入局部最优解、收敛速度较慢等缺点，这也必然会影响 GSO 算法求解 VRP 问题的性能。

1　VRP 问题数学模型

用 V 表示配送网络中的节点集，$V=\{0,\ 1\cdots n\}$，其中 0 表示配送中心，其余节点表示客户；K 表示配送车辆集，$K=\{1,2\cdots m\}$，m 表示完成配送任务所需车辆数；Q 表示车辆的最大承载能力；d_{ij} 表示客户 i 到 j 的距离；q_i 为客户 i 的货物需求量。

设定决策变量：

$$x_{ijk}=\begin{cases}1,\text{车辆}k\text{配送完客户}i\text{后前往客户}\ j\ \text{配送}\\0,\text{否则}\end{cases}$$

$$y_{ik}=\begin{cases}1,\ \text{车辆}k\text{配送客户}i\\0,\ \text{否则}\end{cases}$$

则 VRP 问题的数学模型为[10—11]：

$$\min\ f=\sum_{k\in K}\sum_{i\in V}\sum_{j\in V}d_{ij}x_{ijk} \tag{1}$$

$$\text{ST:}\ \sum_{i\in V\setminus\{0\}}q_i\sum_{j\in V}x_{ijk}\le Q,\forall k\in K \tag{2}$$

$$\sum_{k\in K}\sum_{i\in V}x_{ijk}=1,\forall j\in V\setminus\{0\} \tag{3}$$

$$\sum_{k\in K}\sum_{j\in V}x_{ijk}=1,\forall i\in V\setminus\{0\} \tag{4}$$

$$\sum_{i\in V\setminus\{0\}}x_{i0k}=\sum_{j\in V\setminus\{0\}}x_{0jk}=1,\forall k\in K \tag{5}$$

$$\sum_{k\in K}\sum_{i\in V\setminus\{0\}}x_{i0k}=\sum_{k\in K}\sum_{j\in V\setminus\{0\}}x_{0jk}=m \tag{6}$$

$$\sum_{k\in K}\sum_{i\in S}\sum_{j\in\bar{S}}x_{ijk}\ge 1,\forall S\subseteq V\setminus\{0\},|S|\ge 2 \tag{7}$$

$$\sum_{j\in V}x_{ijk}=\sum_{j\in V}x_{jik},\forall k\in K,\forall i\in V\setminus\{0\} \tag{8}$$

$$x_{ijk}(x_{ijk}-1)=0,\forall i,j\in V,\forall k\in K \tag{9}$$

$$y_{ik}(y_{ik}-1)=0,\forall i\in V,\forall k\in K \tag{10}$$

其中，式(1)为行驶距离最短的目标函数；式(2)为车辆载重约束；式(3)和式(4)为客户配送唯一性约束；式(5)和式(6)为配送中心约束；式(7)为子回路约束；式(8)为客户车辆流守恒约束；式(9)和式(10)为 0~1 变量约束。

2　混沌模拟退火萤火虫优化算法

2.1　萤火虫优化算法

GSO 算法的基本步骤如下所述[12—14]。

1）根据式(11)更新萤火虫 a 在 t 时刻的荧光素浓度值 $l_a(t)$：

$$l_a(t)=(1-\rho)l_a(t-1)+\gamma F(x_a(t)) \tag{11}$$

式中：$\rho\in[0,1]$为荧光素挥发系数；$\gamma\in[0,1]$为荧光素增强系数；$x_a(t)$为萤火虫 a 在 t 时刻的位置；$F(x_a(t))$为对应的目标函数值。

2）每只萤火虫选择动态决策半径 $r_{dy}^a(t)$ 内荧光素浓度值大于自身的个体组成邻域集 $N_a(t)$：

$$N_a(t)=\left\{b:\|x_b(t)-x_a(t)\|<r_{dy}^a(t);l_a(t)<l_b(t)\right\} \tag{12}$$

3）根据式(13)计算个体 a 移向邻域 $N_a(t)$中个体 c 的概率。

$$p_{ac}(t)=\frac{l_c(t)-l_a(t)}{\sum_{b\in N_a(t)}l_b(t)-l_a(t)} \tag{13}$$

4）采用轮盘赌方式选择移向个体，进行移动，并通过式(14)更新个体位置(s 为移动步长)。

$$x_a(t+1)=x_a(t)+s\cdot\left(\frac{x_c(t)-x_a(t)}{\|x_c(t)-x_a(t)\|}\right) \tag{14}$$

5）更新个体动态决策半径公式见式(15)。

$$r_{dy}^a(t+1)=\min\{r_s,\max\{0,r_{dy}^a(t)+\beta(num_t-|N_a(t)|)\}\} \tag{15}$$

式中：r_s 为萤火虫的最大感知半径；β 为动态决策半径更新系数；num_t 为个体邻域集内可以包含的萤火虫数目的阀值。

2.2　改进萤火虫优化算法

2.2.1　荧光素更新策略的设定

根据式(11)，荧光素浓度值更新与目标函数值有

关。文中以路径最短为优化目标，目标函数值越小，优化效果越好，而在 GSO 算法中，荧光素浓度值越大个体越优，因此，文中以式(1)的倒数对各路径的荧光素浓度值进行更新，见式(16)和式(17)。

$$l_{ij}(t)=(1-\rho)l_{ij}(t-1)+\gamma/f_{\mathrm{a}}(t) \tag{16}$$

$$l_{ji}(t)=l_{ij}(t)=(l_{ij}(t)+l_{ji}(t))/2 \tag{17}$$

式中：$l_{ij}(t)$为第 t 次迭代中客户 i 到 j 路径上的荧光素浓度值；$f_{\mathrm{a}}(t)$为个体 a 对应的目标函数值。为了降低萤火虫选择较劣路径的概率，加快算法收敛，对于萤火虫没有经过的路径，文中只进行挥发操作，即令式(16)中 γ=0。

2.2.2 萤火虫转移概率的设定

文中结合 VRP 问题的特点对萤火虫转移概率进行改进，见式(18)。

$$p_{i_{\mathrm{a}}j}(t)=\frac{l_{i_{\mathrm{a}}j}(t)}{\sum_{u\in I_{\mathrm{a}}(t)}\mathrm{dis}(i_{\mathrm{a}},u)+\mathrm{dis}(i_{\mathrm{a}},j)^2} \tag{18}$$

式中：i_{a}为个体 a 所在客户点；$p_{i_{\mathrm{a}}j}(t)$为第 t 次迭代中 a 从客户点 i_{a}移向 j 的概率；$\mathrm{dis}(i_{\mathrm{a}},j)$为客户 i_{a}到 j 的距离；$I_{\mathrm{a}}(t)$为 a 的邻域集。

2.2.3 改进萤火虫优化算法

在 IGSO 算法中，需要将荧光素分配到配送网络中各路径上，且用萤火虫个体的访问次序表示一条可行的路径编码，算法的求解步骤为：

1）将所有萤火虫放置在配送中心，车辆载重量设置为 0。

2）以萤火虫当前位置为起点、车辆最大载重为约束，选择未访问客户点组成邻域集。

3）萤火虫邻域集是否为空？是，则萤火虫返回配送中心，车辆载重清零，进入 2），否则，根据式(18)计算萤火虫的移动概率，进行移动，更新车辆载重。

4）所有客户是否全部被访问？是，萤火虫个体返回配送中心，算法进入 5）；否则，算法进入 2）。

5）选出此次迭代中的最优个体，根据式(16)和(17)更新对应路径上的荧光素浓度值。最优个体没有经过的路径，只进行挥发操作。

6）算法是否达到了最大迭代次数？是，算法执行完毕，输出当前的最优个体，即为问题的最终解；否则，返回 1），进入下一次迭代。

2.3 混沌模拟退火萤火虫优化算法

2.3.1 模拟退火机制的引入

在所有萤火虫个体均遍历了全部客户后，选出此次循环中的最优个体 i[15]。记最优个体 i 对应的目标函数值为 f_i，最优路径为 B_i。将 B_i 设为模拟退火算法的初始集，且设定一个随时间变化的温度 T，T 的变化范围为[$T_{\min}$, $T_{\max}$]。根据模拟退火原理从 B_i 中产生新的可行解 Δf，计算目标函数的变化值 Δf：

$$\Delta f=f_j-f_i \tag{19}$$

若 $\Delta f<0$，则接受新的可行解 f_j，将其设为最优解；否则，考虑热运动的影响。

$$p=\exp(-\Delta f/T) \tag{20}$$

若 $p>\theta$(θ 为(0,1)之间的随机数)，则接受 f_j 为最优解；否则，拒绝 f_j，最优解仍然为 f_j。完成一轮循环后，根据式(21)进行降温。

$$T(t+1)=T(t)\alpha \tag{21}$$

式中：$\alpha\in(0,1)$为降温系数。

2.3.2 混沌机制的引入

混沌机制对模拟退火萤火虫优化算法（SAGSO）的改进包括对荧光素浓度值的混沌初始化和混沌扰动 2 种方式。利用混沌系统的随机性保证初始荧光素浓度值的多样性，使萤火虫能够快速选择较优路径，进而提高算法收敛速度。首先，产生一组混沌变量；然后，将混沌变量赋予到配送网络中 16 各路径上作为混沌初始荧光素浓度值。文中通过 Logistics 映射产生混沌变量，见式(22)。

$$y(t+1)=\mu y(t)[(1-y(t)] \tag{22}$$

式中：$y(t)$为混沌变量；$\mu\in[0,4]$为控制参数。当 μ=4，$0\leqslant y(t)\leqslant 1$ 时，Logistics 映射处于完全混沌状态。

为了进一步提高 SAGSO 算法的全局寻优能力，文中引入了混沌扰动机制，对陷入局部最优解的萤火虫个体进行混沌扰动。当萤火虫个体陷入局部最优解时，对应路径的荧光素浓度值更新公式见式(23)。

$$l_{ij}(t)=(1-\rho)l_{ij}(t-1)+\gamma/f_{\mathrm{a}}(t)+\eta y_{ij}(t) \tag{23}$$

式中：η 为调节系数，是一个常数；$y_{ij}(t)$为第 t 次迭代中客户 i 到客户 j 的路径对应的混沌变量。

2.3.3 混沌模拟退火萤火虫优化算法

1）设置算法初始参数，并根据式(22)对各路径的荧光素浓度值进行混沌初始化。

2）将所有萤火虫放置在配送中心，车辆载重设置为 0。

3）以萤火虫当前位置作为起点、车辆最大载重作为约束，选择未访问的客户组成邻域集。

4）萤火虫邻域集是否为空？是，则萤火虫返回配送中心，车辆载重清零，进入 3）；否则，根据式(18)计算萤火虫的移动概率，进行移动，更新车辆载重。

5）所有客户是否全部访问完毕？是，萤火虫个体返回配送中心，算法进入 6）；否则，算法进入 3）。

6）选出此次迭代中的最优个体；将最优个体作为模拟退火算法的初始集，根据模拟退火原理产生新的可行解，按式(19)和(20)判断新的可行解是否为更优解？是，判断新的可行解对应的路径是否满足车辆载重约束，满足则接受新的可行解为最优解，不满足则重新生成可行解；否则，原来的最优个体仍为最优解。

7）最优解是否小于给定值？是，则认定算法陷入局部最优，启动混沌扰动机制，根据式(23)和(17)对最优个体所对应路径的荧光素浓度值进行更新；否则，则根据式(16)和(17)进行荧光素浓度值的更新；在 2 种荧光素浓度值更新方式中，最优个体没有经过的路径均只进行挥发操作。

8）算法是否达到了最大迭代次数？是，算法执行完毕，输出当前的最优个体，即为问题的最终解；否则，返回 2），进入下一次迭代。

3　仿真分析

3.1　仿真 1

仿真实验分别采用 IGSO 算法、SAGSO 算法和 CSAGSO 算法对仿真算例集进行求解。对每个算例运行 20 次，统计各算例求得的最好解、最好解与已知最优解的误差、平均值、平均值与已知最优解的误差和平均收敛代数等指标。由于文章篇幅限制，文中只选择了最好解与已知最优解的误差（误差 1）、平均值与已知最优解的误差（误差 2）和平均收敛代数进行展示，结果见表 1。误差 1=（最好解−已知最优解）/已知最优解×100%，误差 2=（平均值−已知最优解）/已知最优解×100%，收敛代数表示算法 20 次求解的平均收敛代数。

根据表 1，在全局寻优能力上，IGSO 算法不能求得仿真算例集中任一算例的最优解（对应“误差 1”的值不为 0），SAGSO 算法和 CSAGSO 算法能够分别求出 4 个和 7 个标准算例的最优解（对应“误差 1”的值为 0），且 3 个算法的“误差 1”的平均值依次为 8.19, 3.41 和 1.07，说明 IGSO 算法的全局寻优能力最差，最容易陷入局部最优解，而 SAGSO 算法和 CSAGSO 算法的全局优化能力依次增强。这是由于 IGSO 算法仅对 GSO 算法进行了初步改进，使其能够适应 VRP 问题的求解，并没有解决 GSO 算法中存在的“早熟”问题，因此算法全局优化能力较差。SAGSO 算法在 IGSO 算法的基础上，利用模拟退火算法的概率突跳性使 IGSO 算法的可有效避免陷入局部极小并最终趋于全局最优，进而提高了算法的全局寻优能力。CSAGSO 算法在 SAGSO 算法中引入混沌机制，通过对 SAGSO 算法中陷入局部最优解的个体进行浑沌扰动，进一步最高了算法的全局寻优能力。由此可知，文中对 GSO 算法的一系列改进能够提高算法的全局寻优能力。

在收敛速度上，IGSO 算法和 SAGSO 算法的收敛代数相差不大，而 CSAGSO 算法的收敛代数得以大幅度缩短，说明 IGSO 算法和 SAGSO 算法的收敛速度较慢，CSAGSO 算法的收敛速度最快。这是由于 GSO 算法存在收敛速度较慢的缺点，而 IGSO 算法并没有针对 GSO 算法的收敛速度进行改进，所以 IGSO 算法的收敛速度仍然较慢。同理，SAGSO 算法引入

表 1　IGSO 算法、SAGSO 算法和 CSAGSO 算法仿真结果对比

Tab.1 The simulation results comparison of IGSO、SAGSO and CSAGSO

算例	客户数	已知最优解	IGSO			SAGSO			CSAGSO		
			误差 1	误差 2	收敛代数	误差 1	误差 2	收敛代数	误差 1	误差 2	收敛代数
A-n32-k5	32	784	2.68	12.12	71.26	0.00	5.36	68.33	0.00	1.79	20.85
A-n33-k5	33	661	3.03	18.46	70.67	0.61	7.72	70.15	0.30	3.48	22.48
A-n33-k6	33	742	3.5	12.13	72.11	0.81	7.55	71.54	0.00	4.72	21.81
A-n37-k5	37	669	3.14	19.43	74.19	0.00	6.13	70.92	0.00	3.74	23.69
A-n38-k5	38	730	3.15	22.88	74.44	0.27	9.73	69.32	0.00	6.16	22.63
A-n39-k5	39	822	3.53	14.96	76.82	0.00	11.19	72.60	0.00	5.35	25.57
A-n39-k6	39	831	3.85	15.28	75.93	0.00	10.47	72.27	0.12	5.42	24.23
A-n44-k6	44	937	3.20	19.85	79.78	0.32	16.22	75.69	0.00	4.91	28.31
A-n48-k7	46	1073	6.71	20.97	81.26	2.70	10.62	70.81	0.00	8.48	27.57
A-n53-k7	53	1010	11.19	21.98	84.19	1.49	16.44	75.98	0.50	9.11	30.63
A-n55-k9	55	1073	9.23	19.01	82.85	3.63	12.21	78.51	2.33	6.71	30.93
A-n60-k9	60	1354	10.56	25.78	86.18	4.65	19.87	80.86	2.29	9.75	32.27
A-n63-k9	63	1616	10.21	22.96	85.61	5.38	15.53	81.58	1.67	11.26	31.98
A-n63-k10	63	1314	13.24	26.03	87.95	6.70	18.95	79.83	3.04	9.51	30.94
A-n65-k9	65	1174	11.07	27.51	87.40	10.48	18.99	82.99	3.41	10.48	35.07
A-n69-k9	69	1159	16.82	25.63	90.69	9.92	18.46	83.40	1.73	11.99	35.14
A-n80-k10	80	1763	24.05	30.4	96.11	11.06	24.67	89.02	2.78	16.79	39.09
平均值			8.19	20.9	81.03	3.41	13.54	76.10	1.07	7.63	28.42

的模拟退火算法只是针对 IGSO 算法的全局寻优能力进行了改进，所以其收敛速度相对于 IGSO 算法并没有明显提高。CSAGSO 算法通过混沌机制对荧光素浓度值进行了混沌初始化，加快了算法寻找优秀子路径的速度，进而提高了算法收敛速度。由此可知，文中对 GSO 算法的改进能够加快算法收敛速度。

在算法稳定性上，IGSO 算法、SAGSO 算法和 CSAGSO 算法的“误差 2”的值逐渐减少，说明 3 个算法的稳定性依次增强。由此可知，文中对 GSO 算法的改进可以增强算法的稳定性。

3.2 仿真 2

为了进一步验证 CSAGSO 算法的求解性能，仿真实验分别采用 GA 算法[3]、ACO 算法[4]和 PSO 算法[5]求解仿真实验 1 中的算例集。对每个算例运行 20 次，统计相应指标，并将其与 CSAGSO 算法的运行结果进行对比，见表 2。

根据表 2，在全局寻优能力上，CSAGSO 算法能够求得最优解的算例个数最多（7 个），且“误差 1”的均值最小（1.07），所以 CSAGSO 算法的全局寻优能力最强；GA 算法能够求得最优解的算例个数（4 个）和“误差 1”的均值（2.06）次之，故 GA 算法的全局寻优能力次之；ACO 算法和 PSO 算法能够求得最优解的算例个数最少（均为 2 个），所以 ACO 算法和 PSO 算法的全局寻优能力最弱。但由于 ACO 算法的“误差 1”的均值（2.97）比 PSO 算法的均值（5.12）小，说明 ACO 算法的全局优化能力比 PSO 算法强；在收敛速度上，各算法收敛代数的均值从小到大依次为 CSAGSO 算法（28.42）、ACO 算法（48.41）、PSO 算法（54.36）和 GA 算法（69.48），说明 CSAGSO 算法的收敛速度最快，ACO 算法次之，而 PSO 算法和 GA 算法的收敛速度最慢；在算法稳定性上，CSAGSO 算法对应的“误差 2”的均值为 7.63，GA 算法、ACO 算法和 PSO 算法的分别为 13.39, 10.17 和 14.74，说明算法稳定性由强到弱依次为 CSAGSO 算法、ACO 算法、GA 算法和 PSO 算法。

表 2 CSAGSO 算法、GA 算法、ACO 算法和 PSO 算法仿真结果对比

Tab.2 The simulation results comparison of CSAGSO、GA、ACO and PSO

算例	客户数	CSAGSO			GA			ACO			PSO		
		误差 1	误差 2	收敛代数	误差 1	误差 2	收敛代数	误差 1	误差 2	收敛代数	误差 1	误差 2	收敛代数
A-n32-k5	32	0.00	1.79	20.85	0.00	3.03	58.43	0.00	2.78	36.88	0.00	3.58	45.76
A-n33-k5	33	0.30	3.48	22.48	0.00	3.11	60.54	0.00	2.83	39.99	0.98	4.33	43.14
A-n33-k6	33	0.00	4.72	21.81	0.87	3.80	59.17	1.58	3.56	38.81	0.00	4.69	45.85
A-n37-k5	37	0.00	3.74	23.69	0.00	4.56	61.12	2.89	3.01	40.52	3.75	4.78	48.51
A-n38-k5	38	0.00	6.16	22.63	0.00	7.81	62.23	1.93	7.83	41.06	3.88	7.99	48.22
A-n39-k5	39	0.00	5.35	25.57	1.18	8.13	60.28	2.35	7.95	42.55	2.97	7.68	47.43
A-n39-k6	39	0.12	5.42	24.23	1.24	9.80	62.85	2.09	8.11	42.53	3.03	10.26	50.82
A-n44-k6	44	0.00	4.91	28.31	1.78	10.23	65.70	2.18	8.72	46.47	4.25	12.78	52.24
A-n48-k7	46	0.00	8.48	27.57	1.65	11.99	69.06	1.99	9.08	43.98	4.78	13.91	51.21
A-n53-k7	53	0.50	9.11	30.63	2.01	16.64	71.41	2.01	12.33	49.36	5.35	18.27	55.61
A-n55-k9	55	2.33	6.71	30.93	1.97	17.09	74.71	2.56	13.56	51.79	5.38	17.66	57.72
A-n60-k9	60	2.29	9.75	32.27	3.31	19.71	70.87	3.68	15.49	55.45	6.98	20.54	60.11
A-n63-k9	63	1.67	11.26	31.98	3.56	18.85	76.80	3.97	16.97	53.04	7.03	19.97	59.17
A-n63-k10	63	3.04	9.51	30.94	4.01	20.38	76.68	4.53	16.02	55.65	7.45	21.37	61.88
A-n65-k9	65	3.41	10.48	35.07	4.21	20.09	80.74	5.67	12.32	58.73	8.06	24.60	63.22
A-n69-k9	69	1.73	11.99	35.14	4.13	23.94	82.99	4.99	11.74	59.36	10.88	26.95	63.86
A-n80-k10	80	2.78	16.79	39.09	5.06	28.55	87.65	7.99	20.53	66.82	12.35	31.28	69.36
平均值		1.07	7.63	28.42	2.06	13.39	69.48	2.97	10.17	48.41	5.12	14.74	54.36

4 结语

对求解 VRP 问题的 CSAGSO 算法进行了研究。首先，对 GSO 算法进行了改进，提出了 IGSO 算法，并 IGSO 算法的基础上引入模拟退火机制，提出了 SAGSO 算法；然后，在 SAGSO 算法中引入混沌机制，设计了 CSAGSO 算法。仿真结果表明，对于 GSO

算法的一系列改进是合理的，改善了算法的求解性能，且提出的 CSAGSO 算法在全局寻优能力、收敛速度和稳定性上优于传统智能算法。

参考文献

[1] 裴小兵, 贾定芳. 基于模拟退火算法的城市物流多目标配送车辆路径优化研究[J]. 数学的实践与认识, 2016(2): 105—113.
PEI Xiao-bing, JIA Ding-fang. Optimizing Multi-objective Vehicle Routing Problem in City Logistics Based on Simulated Annealing Algorithm[J]. Mathematics in Practice and Theory, 2016(2): 105—113.

[2] 胡云清. 改进智能水滴算法在车辆调度问题中的应用[J]. 包装工程, 2016, 37(9): 63—67.
HU Yun-qing. Application of Improved Intelligent Water Drop Algorithm in Vehicle Scheduling Problem[J]. Packaging Engineering, 2016, 37(9): 63—67.

[3] 周生伟, 蒋同海, 张荣辉. 改进遗传算法求解 VRP 问题[J]. 计算机仿真, 2013, 30(12): 140—143.
ZHOU Sheng-wei, JIANG Tong-hai, ZHANG Rong-hui. Improved Genetic Algorithm for VRP[J]. Computer Simulation, 2013, 30(12): 140—143.

[4] 陈迎欣. 基于改进蚁群算法的车辆路径优化问题研究[J]. 计算机应用研究, 2012, 29(6): 2031—2034.
CHEN Ying-xin. Study on VRP Based on Improved Ant Colony Optimization[J]. Application Research of Computers, 2012, 29(6): 2031—2034.

[5] 施彦, 韩力群, 陈秀新. 基于集成协同 PSO 算法的车辆路径优化仿真[J]. 计算机仿真, 2012, 29(6): 339—342.
SHI Yan, HAN Li-qun, CHEN Xiu-xin. Ensemble Collaborative PSO Algorithm for Vehicle Routing Problem Optimization and Simulation[J]. Computer Simulation, 2012, 29(6): 339—342.

[6] KRISHNAND K N, GHOSE D. Detection of Multiple Source Locations Using a Glowworm Metaphor with Applications to Collective Robotics[C]// Proceeding of IEEE Swarm Intelligence Symposium Piscataway, IEEE Press, 2005: 84—91.

[7] 吴伟民, 亢少将, 林志毅, 等. 基于改进萤火虫算法的多模函数优化[J]. 计算机应用与软件, 2014, 31(1): 283—285.
WU Wei-min, KANG Shao-jiang, LIN Zhi-yi, et al. Multimodal Function Optimization Based on Improved Glowworm Swarm Optimization[J]. Computer Application and Software, 2014, 31(1): 283—285.

[8] 吕明. 基于萤火虫优化 BP 神经网络的数控机床故障诊断[J]. 机械设计与研究, 2014, 30(6): 77—80.
LYU Ming. Research on CNC Machine Fault Diagnosis Based on the GSO-BP Neural Network[J]. Machine Design and Research, 2014, 30(6): 77—80.

[9] 欧阳桢, 李应. 基于萤火虫算法的匹配追踪用于生态声音辨识[J]. 计算机工程与应用, 2015, 51(2): 198—204.
OUYANG Zhen, LI Ying. Glowworm Swarm Optimization and Matching Pursuit Sparse Decomposition for Ecological Environmental Sounds Identification[J]. Computer Engineering and Applications, 2015, 51(2): 198—204.

[10] TEYMOURIAN E, KAYVANFAR V, KOMAKI G H M, et al. Enhanced Intelligent Water Drops and Cuckoo Search Algorithms for Solving the Capacitated Vehicle Routing Problem[J]. Information Sciences, 2016(334/335): 354—378.

[11] 曹平方, 李灵, 李诗珍. 基于分枝界定的 VRP 模型精确算法研究及应用[J]. 包装工程, 2014, 35(17): 97— 101.
CAO Ping-fang, LI Ling, LI Shi-zhen. Research and Application of the Accurate Algorithm of VRP Model Based on Branch and Bound Method[J]. Packaging Engineering, 2014, 35(17): 97—101.

[12] KRISHNAND K N, GHOSE D. Glowworm Swarm Based Optimization Algorithm for Multimodal Functions with Collective Robotics Applications[J]. Multiagent and Grid System, 2006, 2(3): 209—222.

[13] MARINAKI M, MARINAKIS Y. A Glowworm Swarm Optimization Algorithm for the Vehicle Routing Problem with Stochastic Demands[J]. Expert Systems with Applications, 2016(46): 145—163.

[14] WU B, QIAN C, NI W H. The Improvement of Glowworm Swarm Optimization for Continuous Optimization Problems[J]. Expert Systems with Applications, 2012, 39(7): 6335—6342.

[15] CHEN S M, CHIEN C Y. Solving the Traveling Salesman Problem Based on the Genetic Simulated Annealing Ant Colony System with Particle Swarm Optimization Techniques[J]. Expert Systems with Applications, 2011, 38(12): 14439—14450.