

基于包装配送问题的混沌蛙跳布谷鸟算法研究

张异

(重庆工商职业学院, 重庆 401520)

摘要: **目的** 设计一种求解包装配送问题的混沌蛙跳布谷鸟算法 (Chaotic Frog Leaping Cuckoo Search Algorithm, CFLCSA)。**方法** 对鸟巢个体进行实数编码, 引入混沌机制和随机蛙跳算法, 增强算法种群多样性和局部搜索能力, 并利用 E-n33-k4 和 E-n76-k8 算例来验证算法的求解性能。**结果** CFLCSA 算法能够求得 E-n33-k4 已知最优解, 求得 E-n76-k8 的最短配送距离与已知最优解的误差仅为 5.03%, 且算法求解结果及平均运行时间均优于混沌蚁群算法 (Chaotic Ant Colony Algorithm, CACA)、改进遗传算法 (Improved Genetic Algorithm, IGA) 和禁忌搜索算法 (Tabu Search, TS)。**结论** CFLCSA 算法求解性能优于 CACA 算法、IGA 算法和 TS 算法, 是一种较好的包装配送问题求解方法。

关键词: 包装配送; 布谷鸟算法; 混沌机制; 蛙跳算法

中图分类号: TP301 文献标识码: A 文章编号: 1001-3563(2019)05-0174-06

DOI: 10.19554/j.cnki.1001-3563.2019.05.024

Chaos Frog Leaping Cuckoo Search Algorithm in Packaging Distribution Problem

ZHANG Yi

(Chongqing Technology and Business Institute, Chongqing 401520, China)

ABSTRACT: The work aims to design a chaotic frog leaping cuckoo search algorithm (CFLCSA) to solve packaging distribution problem. The nests were coded by real number and the chaotic mechanism and random leaping frog algorithm were introduced to enhance the population diversity and local search capability of CFLCSA. E-n33-k4 and E-n76-k8 were also used to verify the performance of CFLCSA. The known optimal solution of E-n33-k4 was obtained by CFLCSA algorithm, the error between the shortest distance of E-n76-k8 and the known optimal solution was only 5.03%, and solution results and mean running time of CFLCSA were better than Chaotic Ant Colony Algorithm (CACA), Improved Genetic Algorithm (IGA) and Tabu Search (TS) algorithm. CFLCSA algorithm is superior to CACA, IGA and TS algorithm in solving performance and is a better solution for packaging distribution problem.

KEY WORDS: packaging distribution; cuckoo search algorithm; chaotic mechanism; leaping frog algorithm

包装配送是现代物流的一个重要领域, 其中的车辆路径问题 (Vehicle Routing Problem, VRP) 对于提高企业服务质量、降低物流成本具有重大意义^[1-2]。针对该问题, 学者们先后提出了大量启发式算法, 如 Clarke 等设计的 C-W 算法、Gillett 等提出的 Sweep 算法和 Bramel 等提出的基于选址问题转化的 LBH 算法。这些算法虽然极大程度地丰富了 VRP 问题求解

手段, 但仍然存在一些缺陷, 如 C-W 算法边缘点难以组合, Sweep 算法非渐进优化导致求解精度不高, LBH 算法转化过程复杂且运行效率较低^[3]。近年来, 现代启发式算法逐渐成为了求解 VRP 问题的主要方法。例如, REN C Y 等^[4]构造了具有邻里结构的初始个体, 并引入 1-move 和 2-opt 算子, 提高了禁忌搜索算法 (TS) 的求解性能。周生伟等^[5]设计了基于随机

收稿日期: 2018-11-18

基金项目: 重庆市教委人文社科项目 (16SKGH209); 重庆市社科规划项目 (2018YBGL071)

作者简介: 张异 (1973—), 女, 硕士, 重庆工商职业学院副教授, 主要研究方向为供应链管理、物流管理。

贪婪自适应搜索机制的改进遗传算法 (IGA)。谈晓勇^[6]则利用混沌机制对标准蚁群算法信息素浓度进行扰动,提出了求解 VRP 问题的混沌蚁群算法 (CACA)。

布谷鸟算法 (Cuckoo Search Algorithm, CSA) 是 Yang 和 Deb 于 2009 年提出的一种新型智能算法^[7],主要利用 lévy 飞行机制全局搜索寄生鸟巢,并通过偏好随机游走的局部搜索方式更新被发现鸟巢的位置。分析文献可知,CSA 算法主要应用于神经网络优化、智能图像识别、多目标优化等连续域问题,在离散型问题上研究较少,集中在 0—1 背包问题、TSP 问题等方面,鲜有文献对 VRP 问题进行研究。在此背景下,文中对 CSA 算法编码方式进行改进,同时引进混沌机制和蛙跳算法,提出一种求解包装配送 VRP 问题的 CFLCSA 算法。

1 包装配送 VRP 问题的数学模型

包装配送 VRP 问题可以描述为:现有 n 个客户,第 i 个客户的包装需求量为 $g_i (i=1,2,\dots,n)$,配送中心派出车辆进行配送,车辆载质量为 q ,且 $q > g_i$,求满足各客户需求的最短配送路线。

为了合理安排路线,需要预估配送所需车辆数。实际生活中,车辆数取决于货物装车(或卸车)的复杂程度,装卸越简单,约束条件越少,车辆实际载货量就越多,所需车辆数就越少^[8-9]。通常情况下,可根据式(1)确定配送车辆数。

$$m = \lceil \sum g_i / \alpha q \rceil + 1 \tag{1}$$

式中: m 为配送车辆数; $\lceil \cdot \rceil$ 为不大于括号内数值的最大整数; α 为装卸复杂程度,取值为(0,1),且一般设置为0.95。

用 d_{ij} 表示客户 i 到 j 的距离,0 表示配送中心,定义决策变量为:

$$x_{ijk} = \begin{cases} 1 & \text{车辆 } k \text{ 从客户 } i \text{ 行驶到 } j \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{客户 } i \text{ 由车辆 } k \text{ 配送} \\ 0 & \text{otherwise} \end{cases}$$

则包装配送 VRP 问题的数学模型为:

$$\min f = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m d_{ij} x_{ijk} \tag{2}$$

约束条件如下:

$$\sum_{i=1}^n g_i y_{ik} \leq q \quad k=1,2,\dots,m \tag{3}$$

$$\sum_{i=1}^m y_{ik} = \begin{cases} 1 & i=1,2,\dots,n \\ m & i=0 \end{cases} \tag{4}$$

$$\sum_{i=0}^n x_{ijk} = y_{jk} \quad j=1,2,\dots,n; k=1,2,\dots,m \tag{5}$$

$$\sum_{j=0}^n x_{ijk} = y_{jk} \quad i=1,2,\dots,n; k=1,2,\dots,m \tag{6}$$

$$x_{ijk} = 0/1 \quad i,j=0,1,\dots,n; k=1,2,\dots,m \tag{7}$$

$$y_{ik} = 0/1 \quad i=1,2,\dots,n; k=1,2,\dots,m \tag{8}$$

其中,式(2)表示路径最短的目标函数;式(3)表示所有车辆不超载;式(4)表示每个客户只由一辆车配送,所有配送任务由 m 辆车协同完成;式(5)和式(6)表示达到和离开任一客户的车辆只有 1 辆;式(7)和式(8)表示 0~1 变量约束。

2 基本布谷鸟算法

CSA 算法主要通过模拟布谷鸟选巢产蛋的行为寻找问题最优解^[10],主要求解步骤如下所述。

1) 随机生成 N 个鸟巢个体 $X = \{X_1, X_2, \dots, X_i, \dots, X_N\}$, 计算鸟巢适应度值并找出最优个体。

2) 判断是否达到最大迭代次数。达到,输出最优鸟巢;否则,进入步骤 3)。

3) 根据式(9)所示 lévy 飞行机制更新鸟巢^[11]。

$$X_i^{t+1} = X_i^t + \alpha \oplus Levy(\lambda) \oplus (X_{best}^t - X_i^t) \tag{9}$$

式中: X_i^t 为鸟巢 i 在时刻 t 的位置; X_{best}^t 为时刻 t 的最优鸟巢; α 为服从正态分布的步长控制因子,通常设置为 $0.01R$, R 是服从 $N(0, 1)$ 分布的随机数; \oplus 为点乘运算; $Levy(\lambda)$ 为服从 lévy 分布的随机数,由式(10)计算。

$$Levy(\lambda) = \frac{\varphi \mu}{|\nu|^{\frac{1}{\beta}}} \tag{10}$$

式中: β 通常设置为 $3/2$; μ, ν 为服从 $N(0,1)$ 分布的随机数; φ 按照式(11)取值,其中 $\Gamma(\cdot)$ 为 gamma 函数。

$$\varphi = \left\{ \frac{\Gamma(1+\beta) \sin(\pi\beta/2)}{2^{(\beta-1)/2} \Gamma[(1+\beta)/2] \beta} \right\}^{1/\beta} \tag{11}$$

4) 生成[0,1]间的随机数 r , 设更新后鸟巢被宿主发现的概率为 p_i 。若 $r < p_i$, 保留该鸟巢。否则,根据式(12)随机产生新鸟巢。算法转入步骤 2), 进行下一次迭代。

$$X_i^{t+1} = X_i^t + r(X_a^t - X_c^t) \tag{12}$$

式中: X_a^t, X_c^t 为第 t 次迭代中除 X_i^t 外的任意 2 个鸟巢位置。

3 混沌蛙跳布谷鸟算法在包装配送 VRP 问题中的应用

3.1 构造解向量

文中采用实数编码构造鸟巢个体,即对有 m 辆

车、 n 个客户的包装配送 VRP 问题，用 $m+n-1$ 维向量 $X = \{x_1, x_2 \dots x_{m+n-1}\}$ 表示一个个体，具体解码过程如下所述。

已知某包装配送网络有 8 个客户，完成任务所需车辆数为 3。用 $\bar{X} = \{1, 2, 3, 4, 5, 6, 7, 8, 0, 0\}$ 表示客户向量，其中 $\{0\}$ 表示配送中心，数量为车辆数减 1。随机产生一个鸟巢个体 $X = \{3.1, 2.5, 1.2, 5.4, 6.9, 4.3, 4.4, 8.5, 4.6, 5.5\}$ ，对其进行升幂排序和整数化处理，并与 \bar{X} 相对应。

$$\bar{X}: \{1, 2, 3, 4, 5, 6, 7, 8, 0, 0\}$$

$$X: \{3, 2, 1, 7, 9, 4, 5, 10, 6, 8\}$$

将 X 视为 \bar{X} 的配送顺序，则可以得到配送方案为 3—2—1—6—7—0—4—0—5—8。其中，车辆 1 方案为 0—3—2—1—6—7—0，车辆 2 方案为 0—4—0，车辆 3 为 0—5—8—0。

3.2 构造适应度函数

对于包装配送 VRP 问题，目标函数即为适应度函数。为方便算法编程，如式 (13) 所示，文中将车辆载重约束整合到目标函数中，得到新的适应度函数。

$$\min f = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m d_{ij} x_{ijk} + B \sum_{k=1}^m \max(\sum_{i=1}^n g_i y_{ik} - q, 0) \tag{13}$$

式中： B 为惩罚函数，取值为非常大的正实数。由式 (13) 可知，当车辆超载时，适应度值取非常大的正值，在算法迭代中自然会被淘汰。

3.3 混沌初始化

混沌是一种广泛存在于社会和自然界中的非周期性运动，能够在一定范围内不重复地遍历所有状态，具有显著的遍历性、随机性和规律性^[12]。如式 (14) 和 (15) 所示，文中主要利用混沌 Logistic 映射随机生成初始鸟巢个体，进而提高初始种群多样性，增强算法求解效率。

$$X_{n+1} = \mu X_n (1 - X_n) \tag{14}$$

$$n = 0, 1, 2, \dots; 0 \leq \mu \leq 4 \tag{15}$$

3.4 蛙跳算法

蛙跳算法最初由 Eusuff 和 Lansley 于 2000 年提出，是一种模拟青蛙觅食过程的群体智能亚启发式算法^[13-15]。蛙跳算法的核心思想在于，将青蛙种群按照一定规则划分为若干子群，子群个体按照式 (16—18) 进行局部搜索，直到满足子群进化条件结束。然后，所有子群重新融合、划分、进化，直到满足条件为止。

$$S = r(X_{bl} - X_{wl}) \tag{16}$$

$$X_{wl_new} = X_{wl} + S \tag{17}$$

$$-S_{max} < S < S_{max} \tag{18}$$

式中： r 为 $[0, 1]$ 间的随机数； X_{bl} 、 X_{wl} 分别为第 l 个子群中的最优个体和最差个体； S 为青蛙的局部搜索半径，由 X_{bl} 、 X_{wl} 根据式 (16) 计算； S_{max} 为青蛙的最大跳动步长；如果更新后的 X_{wl_new} 优于原来最差个体 X_{wl} ，则用 X_{wl_new} 代替 X_{wl} ，反之则在子群 l 中随机产生一个个体代替 X_{wl} ，直到满足进化条件结束。

3.5 CFLCSA 算法在包装配送 VRP 问题中的实现步骤

综上所述，CFLCSA 算法在包装配送 VRP 问题中的实现步骤如下所述。

1) 设置鸟巢种群规模 N 、最大迭代次数 T 、鸟巢被发现概率 p ；蛙跳局部搜索次数 T_1 、子群数量 L 、最大跳动步长 S_{max} 。

2) 随机生成一个 $0 \sim 1$ 之间的 $m+n-1$ 维鸟巢个体 X_1 ，根据式 (14) 和 (15) 得到其他个体 X_2, X_3, \dots, X_N 。

3) 根据式 (13) 计算每个鸟巢 i 的适应度值 $f(i)$ ，确定当前种群最优个体 X^* 及对应适应度值 f^* ， $i=1, 2, \dots, N$ 。

4) 算法迭代次数是否达到 T 。达到，输出最优个体 X^* 及适应度值 f^* ，算法停止。反之，进入步骤 5)。

5) 将 N 个鸟巢随机分成 L 个子群。确定每个子群 l 中最好个体 X_{bl} 和最差个体 X_{wl} ，根据式 (16) 至 (18) 得到个体 X_{wl_new} ，如果 $X_{wl_new} < X_{wl}$ ，则用 X_{wl_new} 代替 X_{wl} 作为子群最差个体，即 $X_{wl} = X_{wl_new}$ 。反之，随机选取子群中任一个个体代替原来的 X_{wl} 。重新确定子群最优个体和最差个体，进行蛙跳局部搜索，直到搜索次数 T_1 为止。将所有子群融合。

6) 根据式 (9—11) 更新鸟巢个体 i ，得到新一代鸟巢种群。

7) 对鸟巢个体 i ，随机生成 $[0, 1]$ 之间的数值 r ，若 $r < p$ ，保留该鸟巢。否则，根据式 (12) 随机产生新鸟巢。转入步骤 3)，进入下一次迭代。

4 仿真研究

为验证 CFLCSA 算法的有效性，文中选取 VRP 问题的标准算例 E-n33-k4 和 E-n76-k8 进行仿真测试。如表 1 和表 2 所示，E-n33-k4 为 33 个客户的小规模算例，车辆最大载质量为 8000，根据式 (1) 可得配送车辆数为 4。E-n76-k8 为 76 个客户的大规模算例，车辆最大载质量为 180，配送车辆数为 8。

将 CFLCSA 算法参数设置为：鸟巢种群规模 $N=60$ 、最大迭代次数 $T=100$ 、鸟巢被发现概率 $p=0.2$ ；蛙跳局部搜索次数 $T_1=20$ 、子群数量 $L=4$ 、最大跳动步长 $S_{max}=1$ 。在“Intel i5-8250U, 8 GB RAM, Win XP,

表 1 E-n33-k4 的实验数据
Tab.1 Experimental data of E-n33-k4

序号	坐标	需求量	序号	坐标	需求量	序号	坐标	需求量
1	(292,495)	0	12	(314,435)	1500	23	(314,394)	1300
2	(298,427)	700	13	(311,442)	150	24	(313,378)	700
3	(309,445)	400	14	(304,427)	250	25	(304,382)	750
4	(307,464)	400	15	(293,421)	1600	26	(295,402)	1400
5	(336,475)	1200	16	(296,418)	450	27	(283,406)	4000
6	(320,439)	40	17	(261,384)	700	28	(279,399)	600
7	(321,437)	80	18	(297,410)	550	29	(271,401)	1000
8	(322,437)	2000	19	(315,407)	650	30	(264,414)	500
9	(323,433)	900	20	(314,406)	200	31	(277,439)	2500
10	(324,433)	600	21	(321,391)	400	32	(290,434)	1700
11	(323,429)	750	22	(321,398)	300	33	(319,433)	1100

表 2 E-n76-k8 的实验数据
Tab.2 Experimental data of E-n76-k8

序号	坐标	需求量	序号	坐标	需求量	序号	坐标	需求量
1	(40,40)	0	27	(41,46)	18	53	(54,38)	19
2	(22,22)	18	28	(55,34)	17	54	(55,57)	22
3	(36,26)	26	29	(35,16)	29	55	(67,41)	16
4	(21,45)	11	30	(52,26)	13	56	(10,70)	7
5	(45,35)	30	31	(43,26)	22	57	(6,25)	26
6	(55,20)	21	32	(31,76)	25	58	(65,27)	14
7	(33,34)	19	33	(22,53)	28	59	(40,60)	21
8	(50,50)	15	34	(26,29)	27	60	(70,64)	24
9	(55,45)	16	35	(50,40)	19	61	(64,4)	13
...
26	(17,64)	14	52	(29,39)	12	76	(40,37)	20

Matlab 2012”环境下，分别对 E-n33-k4 和 E-n76-k8 算例求解 30 次，求得最短配送距离为 835 和 772，最优配送线路见图 1 和图 2。

由图 1 和图 2 可知，CFLCSA 算法可以求得 E-n33-k4 的已知最优解，求得 E-n76-k8 的最短配送

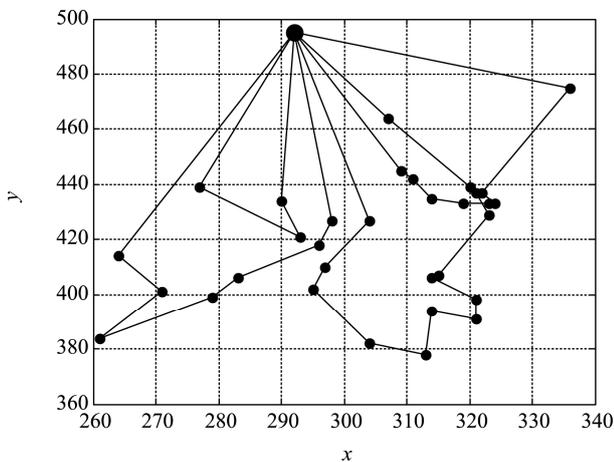


图 1 E-n33-k4 最优配送方案
Fig.1 Optimal delivery scheme of E-n33-k4

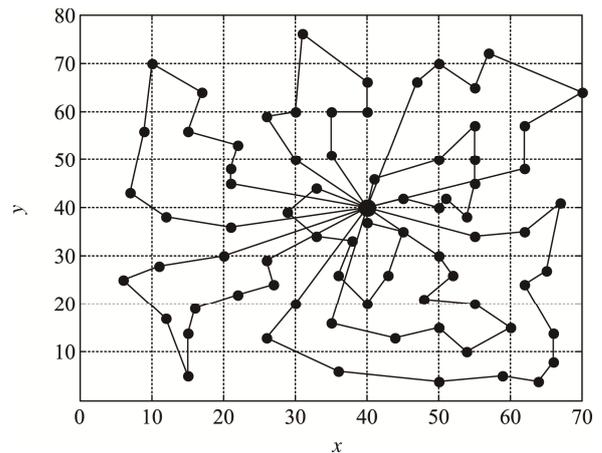


图 2 E-n76-k8 最优配送方案
Fig.2 Optimal delivery scheme of E-n76-k8

距离与已知最优解 (735) 误差仅为 5.03%，算法具有较好求解性能。在相同实验环境下，分别采用 TS 算法^[4]、IGA 算法^[5]和 CACA 算法^[6]对上述算例进行求解，算法参数按照对应参考文献设置。对各算例分别求解 30 次，求得最短配送距离及对应运行时间、

平均运行时间见图3—5。

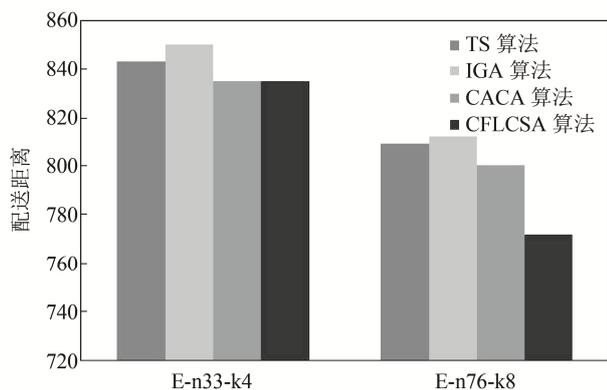


图3 各算法最短配送距离

Fig.3 Shortest distribution distance of each algorithm

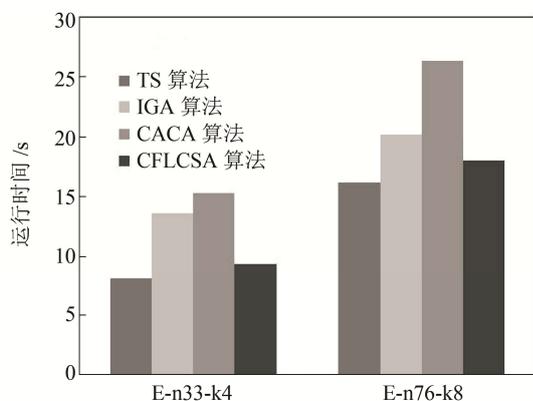


图4 各算法最优方案运行时间

Fig.4 Optimal scheme run time of each algorithm

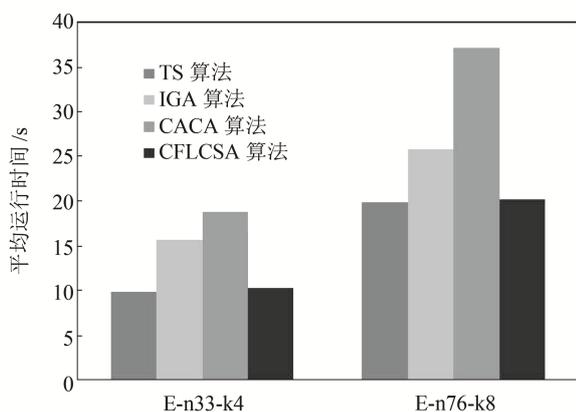


图5 各算法平均运行时间

Fig.5 Average run time of each algorithm

从图3、图4和图5对比结果可知，CFLCSA算法和CACA算法均可求得E-n33-k4的已知最优解，CFLCSA算法虽不能求得E-n76-k8的已知最优解，但其求解结果优于CACA算法、IGA算法和TS算法。同时，CFLCSA算法较强的寻优能力并没有对搜索速度造成负担，算法求得最优调度方案的运行时间、平均运行时间最短。综上分析，CFLCSA算法求解性能优于CACA算法、IGA算法和TS算法，是一种较好

的包装配送VRP问题求解方法。

5 结语

文中主要对包装配送VRP问题的CFLCSA算法进行了研究。该算法采用实数对个体编码，利用混沌机制生成初始种群，并通过蛙跳算法进行局部搜索，能够适应于大规模包装配送VRP问题的求解。下一步，将对CFLCSA算法在其他领域的应用进行研究。

参考文献：

- [1] 樊贵香. 混合模拟植物生长算法在包装件配送中的应用[J]. 包装工程, 2016, 37(13): 43—49.
FAN Gui-xiang. Application of Hybrid Plant Growth Simulation Algorithm in Packaged Products Distribution[J]. Packaging Engineering, 2016, 37(13): 43—49.
- [2] UCHOA E, PECIN D, PESSOA A, et al. New Benchmark Instances for the Capacitated Vehicle Routing Problem[J]. European Journal of Operational Research, 2017, 257(3): 845—858.
- [3] ZHOU Y, WANG J. A Local Search-Based Multi-objective Optimization Algorithm for Multi-objective Vehicle Routing Problem With Time Windows[J]. IEEE Systems Journal, 2017, 9(3): 1100—1113.
- [4] REN C Y. Tabu Search Algorithm for Capacitated Vehicle Routing Problem[J]. Advanced Materials Research, 2013, 753—755: 3060—3063.
- [5] 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.
- [6] 谈晓勇, 林鹰. 基于混沌蚁群算法的应急救援车辆调度优化[J]. 计算机应用研究, 2014, 31(9): 2640—2643.
TAN Xiao-yong, LIN Ying. Vehicle Scheduling Optimization Based on Chaos Ant Colony System Algorithm in Emergency Rescue[J]. Application Research of Computers, 2014, 31(9): 2640—2643.
- [7] PANDEY A C, RAJPOOT D S, SARASWAT M. Twitter Sentiment Analysis Using Hybrid Cuckoo Search Method[J]. Information Processing & Management, 2017, 53(4): 764—779.
- [8] TEYMOURIAN E, KAYVANFAR V, KOMAKI G 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.
- [9] KIM G, ONG Y S, CHEONG T, et al. Solving the Dynamic Vehicle Routing Problem Under Traffic Congestion[J]. IEEE Transactions on Intelligent Transporta-

- tion Systems, 2016, 17(8): 2367—2380.
- [10] ZAMANI A A, TAVAKOLI S, ETEDAI S. Fractional Order PID Control Design for Semi-active Control of Smart Base-isolated Structures: A Multi-objective Cuckoo Search Approach[J]. *Isa Transactions*, 2017, 67: 222.
- [11] REHMAN S, ALI S S, KHAN S A. Wind Farm Layout Design Using Cuckoo Search Algorithms[J]. *Applied Artificial Intelligence*, 2017, 30(10): 899—922.
- [12] COSTA D D A, SAVI M A. Chaos Control of an SMA-pendulum System Using Thermal Actuation with Extended Time-delayed Feedback Approach[J]. *Nonlinear Dynamics*, 2018, 93(2): 571—583.
- [13] MARIOTTI C. A New Leapfrog Scheme for Rotational Motion in 3D[J]. *International Journal for Numerical Methods in Engineering*, 2016, 107(4): 273—289.
- [14] JIA D, ZOU S, LI M, et al. Adaptive Multi-path Routing Based on an Improved Leapfrog Algorithm[J]. *Information Sciences*, 2016, 367/368: 615—629.
- [15] MINESAKI Y. Equilibrium Solutions of the Logarithmic Hamiltonian Leapfrog for the N-body Problem[J]. *Astrophysical Journal*, 2018, 32(11): 857—854.