

6-14-2018

## Monkey Algorithm for Solving Logistics Center Location Selection Problem

Xiaoping Xu

*School of Sciences, Xi'an University of Technology, Xi'an 710054, China;*

Dongjie Zhang

*School of Sciences, Xi'an University of Technology, Xi'an 710054, China;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

---

This Original Article is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

# Monkey Algorithm for Solving Logistics Center Location Selection Problem

## Abstract

**Abstract:** The location selection of logistics center is the most basic problem that affects the whole development of logistics center; an improved monkey algorithm is used to solve the problem. *The basic idea is that the chaos variables are used to generate the initial feasible solution of the monkey algorithm, the decreasing factor is taken as the climbing step in the climbing process and a chaotic search method is introduced in the looking process to improve the performance of the algorithm.* Through simulation experiment, the solution results of typical function and logistics center location problem are given to illustrate the feasibility of the proposed method.

## Keywords

logistics center location, monkey algorithm, chaotic variables, descending factor, chaotic search method

## Recommended Citation

Xu Xiaoping, Zhang Dongjie. Monkey Algorithm for Solving Logistics Center Location Selection Problem[J]. Journal of System Simulation, 2018, 30(6): 2272-2278.

# 求解物流中心选址问题的猴群算法

徐小平, 张东洁

(西安理工大学 理学院, 陕西 西安 710054)

**摘要:** 物流中心选址问题是影响物流中心整体建设发展的最基础问题, 加强物流中心选址问题研究具有十分重要的意义。然而, 它是一个被人们广泛关注的 NP 问题。因此, 本文利用一种改进的猴群算法来求解物流中心选址问题。给出一种改进方法, 其基本思想为采用混沌变量产生猴群算法的初始可行解, 在爬过程中采用递减的因子作为爬步长以及在望过程中引用混沌搜索法来提高算法性能。利用改进算法求解物流中心选址问题。在仿真实验中, 分别利用对典型函数和物流中心选址问题的求解结果说明了所提方法是可行的。

**关键词:** 物流中心选址; 猴群算法; 混沌变量; 递减的因子; 混沌搜索法

中图分类号: TP273+.4A 文献标识码: A 文章编号: 1004-731X (2018) 06-2272-07

DOI: 10.16182/j.issn1004731x.joss.201806034

## Monkey Algorithm for Solving Logistics Center Location Selection Problem

Xu Xiaoping, Zhang Dongjie

(School of Sciences, Xi'an University of Technology, Xi'an 710054, China)

**Abstract:** The location selection of logistics center is the most basic problem that affects the whole development of logistics center; an improved monkey algorithm is used to solve the problem. *The basic idea is that the chaos variables are used to generate the initial feasible solution of the monkey algorithm, the decreasing factor is taken as the climbing step in the climbing process and a chaotic search method is introduced in the looking process to improve the performance of the algorithm.* Through simulation experiment, the solution results of typical function and logistics center location problem are given to illustrate the feasibility of the proposed method.

**Keywords:** logistics center location; monkey algorithm; chaotic variables; descending factor; chaotic search method

## 引言

物流中心选址问题在物流系统规划中起着重要的决策作用, 它是物流调动中的核心问题。它是含有多种变量及各种相关复杂约束条件的模型, 是

著名的 NP 问题。随着科学技术的突飞猛进和客户需求的多样化, 使得物流业迅速地发展。其中, 物流中心作为物流系统中的节点起着越来越重要的角色, 深入地研究物流中心的选址问题将直接关系到总物流配送成本的高低<sup>[1-2]</sup>。物流中心就是连接厂家和客户的纽扣, 通过厂家为不同客户提供产品, 因而, 适当地为物流中心选址可以有效改善产品流节约成本<sup>[3]</sup>。此问题目前已引起众多学者的研讨, 而且给出了各种算法的进行求解, 如差分进化算法<sup>[4]</sup>、粒子群算法<sup>[5]</sup>和遗传算法等<sup>[6]</sup>。但是, 由



收稿日期: 2016-08-05 修回日期: 2017-05-10;  
基金项目: 国家自然科学基金(61773016), 西安理工大学科技创新计划(2016CX013);  
作者简介: 徐小平(1973-), 男, 陕西蓝田, 博士, 教授, 研究方向为进化算法, 系统建模理论等; 张东洁(1990-), 女, 陕西西安, 硕士生, 研究方向为系统优化、控制与辨识。

<http://www.china-simulation.com>

• 2272 •

于他们对该问题的求解依然存在精度不高的缺点, 因而, 如何设计和选择配送中心仍为当前一个研究热点问题, 具有重要的实际意义。

猴群算法是 Zhao 提出的一个群体智能优化算法<sup>[7]</sup>。该算法的突出优点是在求解高维的优化问题时, 无需考虑函数是否可导或可微, 只需要计算当前位置的伪梯度, 就可以确定爬过程中的搜索的方向。而且该算法需调整的参数少、实现过程简单和容易理解。因此, 猴群算法得到了人们广泛地关注, 在各个领域得到快速地发展, 例如在加气站项目进度问题<sup>[8]</sup>、入侵检测问题<sup>[9]</sup>、混合动力系统优化<sup>[10]</sup>和云资源分配<sup>[11]</sup>等领域。但是在实际应用中, 猴群算法自身存在容易陷入局部最优解和求解的精度不高的缺点, 因此, 对该算法进行改进是很必要的。

本文尝试利用猴群算法求解物流中心选址问题。首先, 为了提高猴群算法的优化性能, 给出了改进算法。其次, 建立猴群算法与物流中心选址问题的对应关系。接着, 对物流中心选址问题进行求解。最后, 通过仿真实验结果说明所提出算法是合理的。

## 1 物流中心选址问题简介

物流中心选址问题是指: 在一定数量的位置中, 选取一些地点, 并且让为所有位置供应需求, 选取的这些地点称为物流中心<sup>[12]</sup>。这里作如下假设: (1) 物流中心刚好满足所有位置的需求; (2) 每个位置有且仅由一个物流中心供应; (3) 不考虑其它的费用。从而, 就可以建立如下的数学模型。

$$\min F = \sum_{i \in N} \sum_{j \in M} w_i d_{ij} x_{ij} \quad (1)$$

$$\text{s.t.} \quad \sum_{j \in M_i} x_{ij} = 1, i \in N \quad (2)$$

$$\sum_{j \in M_j} q_j = p \quad (3)$$

式中: 目标函数  $\min F$  是使各位置的需求量到与之最近的物流中心的距离乘积的和最小, 其中,  $N = \{1, 2, \dots, n\}$  是所有位置的编号集合; 其  $M_i$  为物流

中心  $i$  可达到的范围,  $w_i$  是编号为  $i$  的位置的需求量;  $d_{ij}$  是编号为  $i$  的位置到物流中心  $j$  的距离, 且  $d_{ij} \leq s$ ,  $s$  表示位置在物流中心配送距离的上限;  $x_{ij}$  为 0-1 变量, 表示编号为  $i$  的位置的需求量由物流中心  $j$  供应, 当其为 1 时, 表示编号为  $j$  的位置被选为物流中心, 否则为 0;  $q_j$  表示物流中心的服务需求分配关系, 且  $q_j$  为 0-1 变量; 式(2)保证每个位置只能由一个物流中心供应, 其中  $x_{ij} \leq q_j, i \leq N, j \leq M_j$ ; 式(3)表示物流中心的数量为  $p$ 。

## 2 猴群算法

### 2.1 基本猴群算法简介与分析

基本猴群算法是一种启发式的群智能优化算法。该算法主要是由初始化、爬过程、望过程和跳过程组成。初始化是随机给出一组初始解, 爬过程是在自己的局域内寻找最高点, 望过程是寻找邻近区域的最优解, 跳过程是寻找其它区域是否存在更优的解<sup>[7]</sup>。它具有算法结构简单, 参数少、对问题的维数不敏感等优点。

在实际应用中, 基本猴群算法自身存在一些不足。其一, 初始化对算法的结果会产生一定的影响。由随机初始化首先给出一组解可能与最优值相差甚远, 也可能接近最优值, 这一随机初始化不容易找到最优解。而初始化位置越均匀, 对算法的优化结果的影响就越小<sup>[13]</sup>。其二, 爬过程在一次循环中需要执行两次, 而这个过程的作用是求解局部最优值, 它可以直接影响算法的运行时间和求解精度。尤其爬步长大小对精度的影响极大, 爬步长越小, 算法的求解精度越高, 相应计算的次数增加, 花费的时间增多。其三, 望过程中加入混沌搜索法可以提高搜索效率。因此, 对算法的优化性能的提高, 还有很大的提升空间。

### 2.2 改进的猴群算法

基于上述分析, 为了提高算法的优化性能, 这里在算法的解初始化、爬过程和跳过程中给出了相

应改进思想，并且给出具体的方法。

(1) 解的表示和初始化

设  $X_i$  为目标函数的一个可行解，表示第  $i$  只猴子当前的位置，用向量  $X_i=(x_{i1},x_{i2},\dots,x_{in})$  来表示，分量  $x_{ij}$  由下式生成。猴群位置的初始化位置会对最优结果产生一定的影响。由于混沌变量的遍历性和无重复性的特点<sup>[14]</sup>，因此，这里采用混沌算法给出猴群的初始位置，具体用如下经典的 Logistic 函数来产生混沌变量：

$$h_j=4 \cdot h_{j-1} \cdot (1-h_{j-1}), h_j \in [0,1] \quad (4)$$

$$x_{ij} = x_{\min} + h_j \cdot (x_{\max} - x_{\min}) \quad (5)$$

当  $h_0 \notin \{0,0.25,0.5,0.75\}$  时，混沌变量的可以取到值域领域内的所有值。这样就可以避免在开始时陷入局部最优解。其中， $i=1,2,\dots,M$ ， $M$  表示猴群的规模， $x_{\min}$  和  $x_{\max}$  分别为变量  $x_{ij}$  的上限和下限，分量  $x_{ij}$  为第  $i$  只猴子在第  $j$  维的实际位置， $j=1,2,\dots,n$ ， $n$  为优化问题的维数。

(2) 爬过程

爬过程是每只猴子在自己的小范围内通过逐步迭代寻找优化问题的目标函数值的过程，具体过程如下：

1) 随机生成向量  $\Delta X_i=(\Delta x_{i1},\Delta x_{i2},\dots,\Delta x_{in})$ ， $i=1,2,\dots,M$ ，分量  $\Delta x_{ij}$  以均等的概率取  $a$  或者  $-a$ ，其中， $j=1,2,\dots,n$ ， $a=(a\Delta > 0)$  为猴群每次爬的步长。

2) 计算，其中  $j=1,2,\dots,n$ ，向量  $f'_i(X_i)=(f'_{i1}(X_i),f'_{i2}(X_i),\dots,f'_{in}(X_i))$  为目标函数所在位置的伪梯度。

3) 令  $Y_i = X_i + a \cdot \text{sign}(f'_i(X_i))$ ，其中， $j=1,2,\dots,n$ ， $\text{sign}$  为符号函数。

4) 如果向量  $Y_i=(y_{i1},y_{i2},\dots,y_{in})$  在变量的范围内，并且  $f(Y_i) < f(X_i)$ ，更新  $X_i$  为  $Y_i$ ；否则， $X_i$  不变。

5) 重复 1)~4)，直至相邻迭代次数间的目标函数没有变化或者达到给定的执行次数  $N_c$ 。

上述是基本猴群算法的爬过程，爬步长决定算法搜索的精度和需要的 CPU 时间，爬步长的值在一定范围内越小，算法的搜索精度越高，相应的需

要 CPU 时间越长，反之亦然。因此为了权衡精度与 CPU 时间的一个平衡，引入递减因子：

$$a_k = e^{1 - \frac{100}{100-k+1}} \quad (6)$$

来代替固定的爬步长，并且设置爬步长的精度。其中， $k$  为当前的迭代次数， $a_k$  是迭代  $k$  次的爬步长。此时，爬步长的范围可以在一定的区间内呈现递减，从而爬过程的搜索效率就提高了，达到期望的效果。

(3) 望过程

在通过爬过程之后，每只猴子都到达了各自范围领域所在的最高峰，即目标函数在局部范围内找到最优解。然后，向四周张望，在视野范围内的邻近领域，观察是否存在比当前位置所在的山峰更高的山峰。如果存在，就跳到更高的位置。紧接着重复爬过程(这里，爬步长为递减因子的爬过程)，具体过程如下：

1) 在视野范围内  $(x_{ij}-b,x_{ij}+b)$ ，随机产生实数  $y_{ij}$ ，令  $Y_i=(y_{i1},y_{i2},\dots,y_{in})$ ，那么， $Y=(Y_1,Y_2,\dots,Y_M)$ 。其中， $b$  为视野长度，它是猴子在当前位置可以张望到的范围。

2) 如果向量  $Y_i=(y_{i1},y_{i2},\dots,y_{in})$  在变量的范围内，并且  $f(Y_i) < f(X_i)$ ，更新  $X_i$  为  $Y_i$ ；否则，重复 1)，直至找到可行的  $Y_i$ 。

3) 以  $Y$  作为初始位置，执行爬过程(爬步长为递减因子的爬过程)。

(4) 跳过程

跳过程的主要目的是由当前的搜索领域转移到新的领域搜索。求出所有猴子的重心位置，作为跳的支点，使每只猴子从当前位置朝向支点的方向，跳到新的领域，进行搜索，具体过程如下：

1) 在跳区间  $[c,d]$  内随机生成一个实数  $h_j$ ，这里  $h_j$  是由公式(4)生成。采用混沌搜索方法可以更好地遍历整个局部空间，这样可以使其不错过局部最优解。

$$2) \quad y_{ij} = x_{ij} + h_j(p_j - x_{ij}), j=1,2,\dots,n \quad (7)$$

$$p_j = \frac{1}{M} \sum_{i=1}^M x_{ij}, \quad p=(p_1,p_2,\dots,p_n) \text{ 称为支点。}$$

3) 如果向量  $Y_i=(y_{i1}, y_{i2}, \dots, y_{in})$  在变量的范围内, 并且  $f(Y_i) < f(X_i)$ , 更新  $X_i$  为  $Y_i$ ; 否则, 重复 1) 和 2), 直至找到可行的  $Y_i$ 。

#### (5) 终止条件

重复执行(2)~(4), 直至目标函数在迭代过程中没有变化, 或者到达预先设定的迭代次数  $N$ , 算法终止。输出最优解和最优位置。

### 3 利用改进的猴群算法求解物流中心选址问题

求解物流中心的选址问题就是求满足条件的公式(1)的最小值。将物流选址与猴群算法建立如下对应关系如表 1。

表 1 对应关系

Tab. 1 Corresponding relationship

猴群算法	物流中心的选址问题
猴子的位置	公式(1)的一组可行解
种群规模	公式(1)的可行解的数量
维数	备选的城市个数
适应度	公式(1)的目标函数值
编码方式	二进制编码

以(1)式为目标函数, 采用二进制编码表示猴群的位置, 这里的二进制编码将城市的编码随机设为 0 或者 1, 当城市选为物流中心时为 1, 否则为 0。例如,  $X=(1,0,1,0,0,1,0,0,1,0)$  备选的城市个数为 10, 物流中心选址是编号 1,3,6,9 的城市。具体求解步骤如下:

Step 1: 设定猴群算法的参数值。

Step 2: 利用(5)式给出种群中每只猴子的初始位置。计算出猴群中个体当前位置的目标函数值, 找出当前最优的位置及对应的函数值并记录。

Step 3: 执行爬过程, 将爬过程的爬步长由固定值改为可变的步长因子(7)式。计算爬过程后, 猴群中个体当前位置的目标函数值, 如果有更好的目标函数值, 则更新猴群位置到较好目标函数值相对应的位置。

Step 4: 执行改进的望过程, 在区间  $(x_{ij}-b, x_{ij}+b)$

内随机产生 0-1 变量  $y_{ij}$ ,  $j=1,2,\dots,n$ , 搜索到新的猴群个体的位置  $Y_i=(y_{i1} \cdot y_{i2}, \dots, y_{in})$ , 如果优于原来的位置, 更新到新的位置, 否则, 执行 Step 3。

Step 5: 执行跳过程, 计算  $x'_{ij} = x_{ij} + \text{round}(\theta |p_j - x_{ij}|)$ , 其中  $j=1,2,\dots,n$ ,  $\text{round}$  为取整函数, 计算猴群中个体当前位置的目标函数值, 如果有更好的目标函数值, 则更新猴群位置到较好目标函数值相对应的位置。

Step 6: 重复 Step 3~Step 5, 直至找到猴群中个体所在最高的山峰或者达到预定的次数  $n_{\text{Max}}$ , 算法结束, 输出最优位置和最优函数值, 即得到物流中心的城市位置序号。

### 4 数值仿真

为了说明本文所给的改进猴群算法 (Improved Monkey Algorithm, IMA) 的有效性, 这里选取如下典型测试函数和物流中心选址问题进行求解。

#### 4.1 典型测试函数

(1) Sphere 函数:

$$f_1(x) = \sum_{i=1}^n x_i^2, |x_i| \leq 100$$

(2) Ackley 函数:

$$f_2(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sqrt{\sum_{i=1}^n \cos(2\pi x_i)}) + 10 + \exp(1), |x_i| \leq 32$$

(3) Rastrigin 函数

$$f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10), |x_i| \leq 5.12$$

以上 3 个测试函数的理论最小值都为 0。在仿真实验中, 借鉴文献[7]的函数参数设置, 并且经过多次实验, 将 IMA 的参数设置为: 种群规模  $M=10$ , 总的迭代次数  $n_{\text{Max}}=1000$ , 爬步长  $a=0.1$ , 爬步长的精度  $1 \times 10^{-6}$ , 爬的迭代次数  $N_c=20$ , 视野长度  $b=1$ , 望的迭代次数  $N_w=10$ , 跳区间  $[c,d]=[-1,1]$ 。在 10 维的情况下, 对 3 个测试函数分别求解一次, 其寻优过程曲线分别为图 1~3 所示。

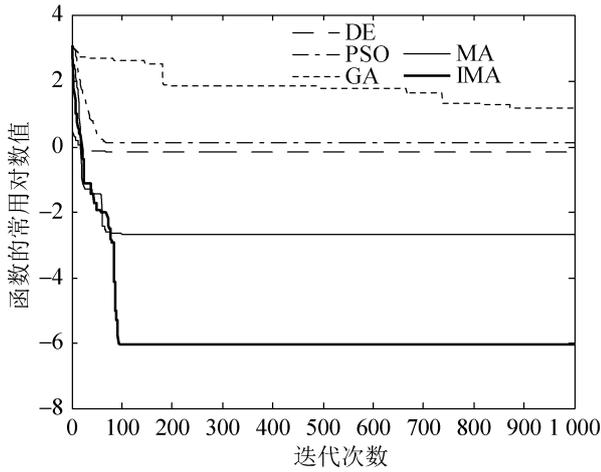


图1 对 Sphere 函数的优化曲线

Fig. 1 Optimization curves for the Sphere function

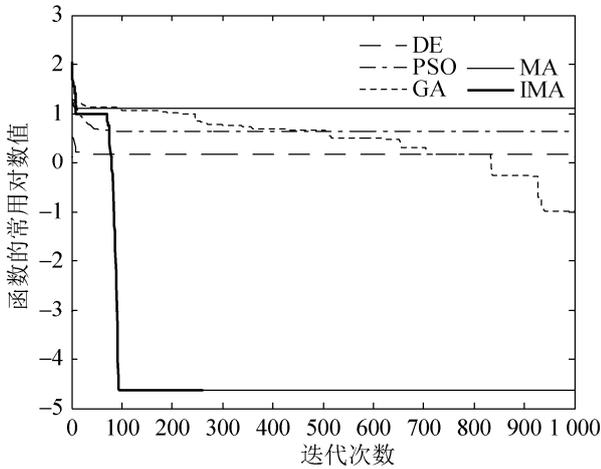


图2 对 Ackley 函数的优化曲线

Fig. 2 Optimization curves for the Ackley function

其次，选用了差分进化算法 (Differential Evolution Algorithm, DE)<sup>[4]</sup>、粒子群算法 (particle Swarm Optimization, PSO)<sup>[5]</sup>、遗传算法 (Genetic Algorithm, GA)<sup>[6]</sup>、基本猴群算法 (Monkey Algorithm, MA)<sup>[7]</sup> 对分别求解上述 3 个测试函数，将它们与所给改进方法比较。在 10 维的情况下，进行 1 次求解它们寻优过程曲线分别为图 1~3 所示。对以上 3 种算法，分别利用参考文献[4-6]中 DE, PSO, GA 的参数设置，种群规模  $M=20$ ，总的迭代次数  $nMax=1000$ 。DE 其余参数为：交叉概率  $p_{cross}=0.6$ ，变异概率  $p_{mutation}=0.01$ ，杂交参数  $CR=0.3$ ；PSO 其它参数设置为：学习因子  $c_1=2.05$ ，学习因子  $c_2=2.05$ ，惯性权重  $w=1$ ，设置精度

$eps=10^{-6}$ ，起始惯性权重值  $w_{start}=0.95$ ，终止惯性权重值  $w_{end}=0.4$ ，粒子间最大距离  $MaxDist=0.28$ ；GA 的其它参数设置为：交叉概率  $p_{cross}=0.6$ ，变异概率  $p_{mutation}=0.01$ 。其中，MA 的参数设置和 IMA 相同。

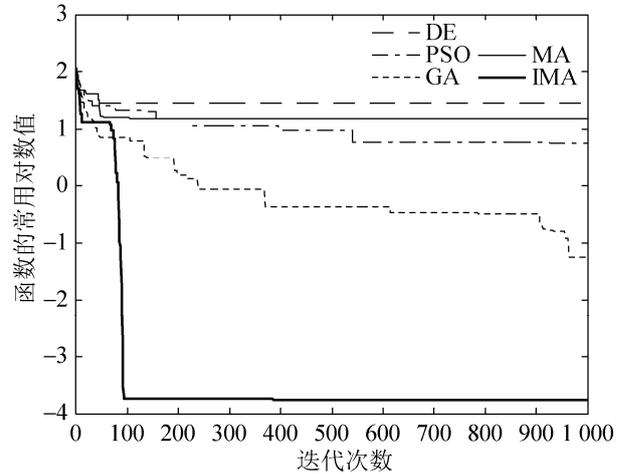


图3 对 Rastrigin 函数的优化曲线

Fig. 3 Optimization curves for the Rastrigin function

接着，进一步利用 DE、PSO、GA、MA 和 IMA 对上 3 个测试函数分别进行 50 次求解，它们的最优值、平均值和方差均罗列在表 2 中。

## 4.2 物流中心选址问题

这里以中国 31 个城市为例，表 3 为城市的坐标和所需配送的货物量。

在仿真中，涉及到的五种算法的参数取值均同于 4.1 部分的参数设置，需选取的物流中心的个数设置为  $p=6$ 。利用 cplex 软件 (记为 cplex)、DE、PSO、GA、MA 和 IMA 分别对该物流中心选址问题求解一次，寻优过程曲线如图 4 所示，最优结果分别为：762 770, 649 988, 634 177, 631 850, 563 036 和 549 650，相应选址的城市编号分别为：2-11-14-18-25-29, 4-8-13-18-26-29, 1-12-18-19-20-30, 4-8-17-20-27-29, 5-9-12-17-24-27, 5-9-12-17-20-27。

将上述的求解过程分别进行 50 次，将它们所求结果的最优值、平均值和方差罗列在表 4 中。

从仿真结果可以看出，本文所给算法能够较快收敛到最优值，且精度较高。

表 2 对 3 个测试函数的寻优结果  
Tab. 2 Optimization results of three functions

测试函数	算法	最优数值解	平均数值解	方差
Sphere	DE	0.143 7	7.509 4	86.932 6
	PSO	0.376 2	0.568 2	0.028 3
	GA	$1.125 2 \times 10^{-4}$	$8.522 6 \times 10^{-4}$	$2.678 2 \times 10^{-7}$
	MA	$0.343 1 \times 10^{-4}$	$1.109 2 \times 10^{-3}$	$4.085 3 \times 10^{-7}$
	IMA	$3.565 6 \times 10^{-10}$	$9.750 0 \times 10^{-9}$	$3.727 9 \times 10^{-16}$
Ackley	DE	8.130 0	9.424 1	2.016 7
	PSO	8.354 8	8.859 9	0.137 8
	GA	7.622 2	7.631 4	$1.016 3 \times 10^{-4}$
	MA	7.086 8	8.861 5	2.345 9
	IMA	$9.031 5 \times 10^{-5}$	$2.686 9 \times 10^{-4}$	$6.080 8 \times 10^{-8}$
Rastrigin	DE	0.043 9	0.504 8	0.3134
	PSO	0.278 7	23.001 8	109.203 3
	GA	$2.697 2 \times 10^{-5}$	$0.179 9 \times 10^{-2}$	$2.392 5 \times 10^{-5}$
	MA	4.016 3	6.473 8	13.202 0
	IMA	$3.742 9 \times 10^{-6}$	$6.273 8 \times 10^{-6}$	$2.447 4 \times 10^{-12}$

表 3 中国 31 个城市坐标及所需货物量

Tab. 3 Coordinates of the 31 cities in China

城市编号	城市 x 坐标	城市 y 坐标	货物量
1	1 304	2 312	20
2	3 639	1 315	90
3	4 177	2 244	90
4	3 712	1 399	60
5	3 488	1 535	70
6	3 326	1 556	70
7	3 238	1 229	40
8	4 196	1 044	90
9	4 312	790	90
10	4 386	570	70
11	3 007	1 970	60
12	2 562	1 756	40
13	2 788	1 491	40
14	2 381	1 676	40
15	1 332	695	20
16	3 715	1 678	80
17	3 918	2 179	90
18	4 061	2 370	70
19	3 780	2 212	100
20	3 676	2 578	50
21	4 029	2 838	50
22	4 263	2 931	50
23	3 429	1 908	80
24	3 507	2 376	70
25	3 394	2 643	80
26	3 439	3 201	40
27	2 935	3 240	40
28	3 140	3 550	60
29	2 545	2 357	70
30	2 778	2 826	50
31	2 370	2 975	30

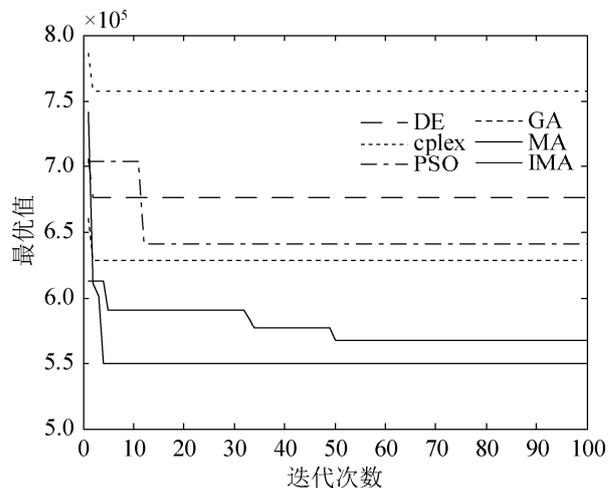


图 4 物流中心选址寻优曲线图

Fig. 4 Optimization curves of logistics center location

表 4 对中国 31 个城市的求解结果

Tab. 4 Results of solving the 31 cities in China

算法	最优值	平均值	方差
cplex	762 770	762 770	0
DE	615 230	639 866	$2.580 5 \times 10^8$
PSO	604 640	620 850	$1.222 7 \times 10^8$
GA	602 810	625 209	$4.223 7 \times 10^8$
MA	556 847	579 853	$2.042 8 \times 10^8$
IMA	549 648	550 218	$3.264 \times 10^5$

## 5 结论

鉴于猴群算法易陷入局部最优解, 收敛精度不高的缺点, 文中提出了一种改进方法, 并将其应用到物流配送中心选址问题的求解。最后, 对典型测

试函数进行寻优,结果说明了算法的性能有明显的提高。并且通过对物流配送中心选址问题的求解,进一步显示了所给改进方法是合理的。

### 参考文献:

- [1] Alfredo G, Díaz H. Cross Entropy for Multiobjective Optimization Problems with Linear Relaxations[J]. European Journal of Operational Research (S0377-2217), 2015, 243(2): 362-368.
- [2] Francesco C, Giulio G, Shaban A. Multi-criteria Logistics Distribution Network Design for mass Customization[J]. International Journal of Applied Decision Sciences (S1755-8085), 2014, 7(2):151-167.
- [3] Jayaram J, Avittathur B. Green Supply Chains: a Perspective from an Emerging Economy[J]. International Journal of Production Economics (S0925-5273), 2015, 1(164): 234-244.
- [4] 贺媛媛, 赵秦, 栗惠英. 基于GIS与差分进化优化的应急物流配送中心选址[J]. 物流技术, 2013, 32(15): 139-140.  
He Y Y, Zhao Q, Li H Y. Location of Emergency Logistics Distribution Centers Based on GIS and Differential Evolutionary Optimization[J]. Logistics SciTech, 2013, 32(15): 139-140.
- [5] Hua X, Hu X, Yuan W. Research Optimization on Logistics Distribution Center Location based on Adaptive Particle Swarm Algorithm[J]. International Journal for Light and Electron Optics (S0030-4026), 2016, 127(20): 8443-8450.
- [6] 胡大伟, 陈诚. 遗传算法(GA)和禁忌搜索算法(TS)在配送中心选址和路线问题中的应用[J]. 系统工程理论与实践, 2007, 27(9): 171-176.  
Hu D W, Chen C. Application of GA and TS in Logistics Distribution Center Location and Routing Problem[J]. Systems Engineering Theory, 2007, 27(9): 171-176.
- [7] Zhao R, Tang W. Monkey Algorithm for Global Numerical Optimization[J]. Journal of Uncertain Systems (S1752-8917), 2008, 2(3): 164-175.
- [8] 赵涛, 夏雨, 宗玛利. 基于猴群算法的加气站项目进度研究[J]. 价值工程, 2010, 29(8): 90-92.  
Zhao T, Xia Y, Zong M L. Optimization of Gas Filling Station Project Scheduling Problem Based on Monkey Algorithm[J]. Value Engineering, 2010, 29(8): 90-92.
- [9] 张佳佳, 张亚平, 孙济洲. 基于猴群算法的入侵检测技术[J]. 计算机工程, 2011, 37(14): 131-133.  
Zhang J J, Zhang Y P, Sun J Z. Intrusion Detection Technology Based on Monkey Algorithm[J]. Computer Engineering, 2011, 37(14): 131-133.
- [10] Ituarte-Villarreal M, Lopez N, Espiritu F. Using the Monkey Algorithm for Hybrid Power Systems Optimization[J]. Procedia Computer Science (S1877-0509), 2012, 12: 344-349.
- [11] 史振华, 陈暄. 云计算中一种改进的猴群算法在资源分配中研究[J]. 计算机测量与控制, 2015, 23(9): 3188-3191.  
Shi Z H, Chen X. Research of an Improved Monkey Algorithm of Cloud Computing in Resource Allocation[J]. Computer Measurement and Control, 2015, 23(9): 3188-3191.
- [12] 张婷婷. 有竞争的物流配送中心选址模型研究[J]. 物流科技, 2014, 37(11): 77-80.  
Zhang T T. Study on Competitive Location Model of Logistics Distribution Center[J]. Logistics SciTech, 2014, 37(11): 77-80.
- [13] 周刘喜, 张兴华, 李纬. 一种改进的多目标粒子群优化算法[J]. 计算机工程与应用, 2009, 45(33): 38-41.  
Zhou L X, Zhang X H, Li W. Improved Multi-objective Particle Swarm Optimization Algorithm[J]. Computer Engineering and Applications, 2009, 45(33): 38-41.
- [14] Yuan X, Dai X, Wu L. A Mutative-scale Pseudo-parallel Chaos Optimization Algorithm[J]. Soft Computing (S1432-7643), 2015, 19(5): 1215-1227.