

# 改进的果蝇优化算法在城市物流配送中心选址中的应用

于博

郑州轻工业学院经济与管理学院, 河南 郑州 450002

**摘要:** 为实现物流配送中心选址的合理配置, 本文提出一种基于 Logistic 混沌系统的果蝇优化算法, 通过我国 31 个城市物流配送中心坐标及其需求量的要求, 建立物流配送中心选址的数学模型, 并运用其算法进行优化求解, 实现配送路径的优化配置, 节约成本。该改进的果蝇优化算法既充分发挥 Logistic 混沌系统和果蝇优化算法的优点, 又克服了果蝇优化算法的局部最优问题。仿真结果表明该算法具有收敛速度快、精度高的优点。

**关键词:** 果蝇优化算法; 物流配送中心; 选址

**中图分类号:** F22

**文献标识码:** A

**文章编号:** 1000-2324(2015)04-0632-04

## Application of the Improved Fruit Flying Optimization Algorithm in Urban Logistics Distribution Center Location

YU Bo

School of Economics and Management/Zhengzhou University of Light Industry, Zhengzhou 450002, China

**Abstract:** To realize the reasonable allocation in logistics distribution center location, this paper provided a fruit flying optimization algorithm based on Logistic chaotic system to optimize the logistics distribution center location through the mathematics model set up on 31 urban logistics distribution center coordinates and the material demand quantity requirements in China and to achieve the optimization allocation in delivery routes and save a cost. This improved algorithm fully takes advantages of Logistic chaotic system and fruit flying distribution algorithm and overcomes the defects of them so as to show the advantages of fast convergence and high accuracy in simulation examples.

**Keywords:** Fruit flying optimization algorithm; logistics distribution center; location

随着电子商务的快速增长, 物流业作为一个新兴行业正在迅速发展。物流配送中心作为连接客户和供应商的中间纽带, 其选址方式决定了物流配送方式和配送距离, 影响物流系统的工作效率和经济效益, 研究物流配送中心的选址具有重要的现实意义和理论价值。

### 1 改进的果蝇优化算法

#### 1.1 果蝇优化算法

果蝇优化算法(Fruit Fly Optimization Algorithm, FOA)<sup>[1]</sup>具有控制参数少、收敛速度快和收敛精度高的优点, 但其存在局部最优问题, 容易陷入局部极小值的缺点。本文结合 Logistic 混沌系统具有遍历性、规律性和随机性的优点, 提出一种基于 Logistic 混沌系统的果蝇优化算法, 克服果蝇优化算法的局部最优问题。

果蝇优化算法具体流程如下<sup>[2]</sup>:

(1) 设置果蝇优化算法的果蝇群体大小  $Popsiz$  和最大迭代次数  $Iteration$ , 随机初始化果蝇群体位置, 初始化结果分别用  $X\_begin$  和  $Y\_begin$  表示;

(2) 根据公式 (1) 和公式 (2) 计算果蝇个体进行寻优的随机方向和距离;

$$x_i = X\_begin + Value \times rand() \quad (1)$$

$$y_i = Y\_begin + Value \times rand() \quad (2)$$

公式 (1) 和公式 (2) 中,  $Value$  表示果蝇的搜索距离;  $x_i$  和  $y_i$  分别表示果蝇个体的下一时刻的位置。

(3) 根据公式 (3) 估计果蝇个体和原点之间的距离  $a_i$ , 之后运用公式 (4) 计算果蝇个体的味道浓度  $s_i$ ;

**收稿日期:** 2014-06-13

**修回日期:** 2014-07-22

**基金项目:** 河南省科技公关项目“农产品冷链物流体系建设及关键冷链技术研究”(112102310055)

**作者简介:** 于博(1980-),男,河南汝州人,硕士,讲师,研究方向:电子商务、物流。E-mail:boyu13@gmail.com

数字优先出版: 2015-07-30 <http://www.cnki.net>

$$d_i = \sqrt{x_i^2 + y_i^2} \quad (3)$$

$$s_i = 1/d_i \quad (4)$$

(4) 味道浓度  $s_i$  代入公式 (5) 味道浓度判定函数, 计算出该果蝇个体当前位置的味道浓:

$$Smell_i = Function(s_i) \quad (5)$$

(5) 找到果蝇群体中最佳味道浓度值和最佳位置, 最佳味道浓度由  $Smell_b$  表示, 最佳位置由  $x_b$  和  $y_b$  表示;

(6) 保留并记录果蝇最佳位置和最佳味道浓度, 最佳味道浓度  $Smell_{best} = Smell_b$ , 果蝇初始位置  $X_{begin} = x_b$ ,  $Y_{begin} = y_b$ , 同时果蝇群体朝着该最佳位置搜寻过去;

(7) 进入迭代寻优, 重复迭代步骤 (2) ~ (5), 同时判断味道浓度是否好于前一迭代味道浓度; 若成立, 则执行步骤 (6)。

## 1.2 Logistic 混沌系统

已知 Logistic 混沌系统的表达式如公式 (6) 所示<sup>[3]</sup>:

$$x(n+1) = ux(n)(1-x(n)) \quad x(n) \in [0, 1] \quad (6)$$

公式 (6) 中,  $n$  表示迭代次数;  $u$  表示混沌控制参数, 当  $u=4$  时, Logistic 系统处于混沌状态。混沌变量  $Cx_i$  的一种变换计算公式如 (7) 所示:

$$Cx(n+1)_i = 4Cx(n)_i(1-Cx(n)_i) \quad i = 1, 2, \dots, N \quad (7)$$

公式 (7) 中,  $Cx(n)_i$  表示混沌映射的第  $i$  个混沌变量  $Cx_i$  在第  $n$  步混沌变量之后的值, 当  $Cx_i \in [0, 1]$ , 并且  $Cx_i \notin \{0.25, 0.50, 0.75\}$  时, 系统处于混沌状态。公式 (7) 的优化参数  $x_i \in [a_i, b_i]$ , 可通过公式 (8) 和公式 (9) 同混沌变量  $Cx_i \in [0, 1]$  进行相互映射变换。

$$Cx_i = (x_i - a_i) / (b_i - a_i) \quad (8)$$

$$x'_i = a_i + Cx_i(b_i - a_i) \quad (9)$$

公式 (9) 中,  $x'_i$  表示混沌映射之后的第  $i$  个混沌变量  $Cx_i$  转化成常规变量的值。

## 1.3 改进的果蝇优化算法

改进的果蝇优化算法的主要流程如下:

(1) 设置果蝇优化算法的果蝇群体大小  $Popsiz$  和最大迭代次数  $Iteration$ , 随机初始化果蝇群体位置, 处于  $[0, 1]$  之间, 并用行向量  $Z_i$  表示;

(2) 根据公式 (8) 将  $Z_i$  分量映射成混沌变量  $Cz(n)_i$ ,  $Cz(n)_i \in [0, 1]$ ;

(3) 运用公式 (7) 对混沌变量  $Cz(n)_i$  进行混沌映射;

(4) 利用公式 (9) 对每个分量进行映射变换, 映射为处于  $[a_i, b_i]$  之间的常规变量  $z'_i$ , 并计算适应度  $f(z'_i)$ , 选取种群中最小的  $f(z'_i)$ , 记录选取的最小  $f(z'_i)$ ,  $fit(gen) = f(z'_i)$ ;

(5) 进行寻优迭代处理, 重复流程 (2) ~ (4);

(6) 若停止条件满足, 则选择最小  $fit$ , 使得  $Smell_{best} = \min(fit(gen))$ , 此时记录最小  $f(z'_i)$  所对应的味道浓度  $S_g$ ;

(7) 为了保证要获得的初值在初次迭代的最优参数的小领域内, 如果参数大于 0, 则  $B \in [0, 1]$ , 通过反复验证, 本文选取  $B=0.25$ , 在选取的味道浓度  $S_g$  附近运用  $S_i = S_g + 2B \times rand() - B$  生成微小扰动种群, 同时运用  $S_i$  评估  $f(S_i)$ , 选取种群中适应度最小的  $f(S_i)$ , 并令  $Bestsmell = \min(f(S_i))$ , 如果  $Bestsmell < Smell_{best}$ , 那么则  $Smell_{best} = Bestsmell$ , 令  $S_g = S_i$ ;

(8) 进行二次迭代寻优, 更新味道浓度  $S_i$ , 重复步骤 (7);

(9) 若停止条件满足, 输出  $Smell_{best}$ ,  $P = S_g$ 。

## 2 物流配送中心选址数学模型

为了实现城市物流配送中心的最优选择, 进行如下假设<sup>[4]</sup>:

(1) 物流配送中心的物资容量总能够满足各需求点的物资需求, 并且配送中心的物资总量由其配送需求点的物资需求量决定;

(2) 一个配送中心仅供应一个需求点;

(3) 不考虑物流配送中心到工厂的运输费用。

基于上述假设, 可以建立物流配送中心选址的数学模型, 其配送示意图如图 1 所示。该数学模型是一个选址和分配模型, 在满足距离上限的条件下, 需要从  $n$  个需求点中找出配送中心并向各需求点配送物品<sup>[5]</sup>。目标函数是各配送中心到需求点的需求量和距离的乘积之和最小, 目标函数为:

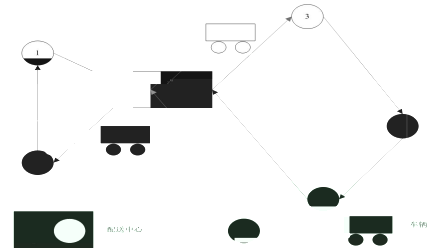


图 1 配送示意图

Fig.1 Schematic diagram for distribution

$$\min F = \sum_{i \in N} \sum_{j \in M_i} w_i d_{ij} Z_{ij} \tag{10}$$

约束条件:

$$\sum_{j \in M_i} Z_{ij} = 1, i \in N \tag{11}$$

$$Z_{ij} \leq h_j, i \in N, j \in M_i \tag{12}$$

$$\sum_{j \in M_i} h_j = p \tag{13}$$

$$Z_{ij}, h_j \in \{0, 1\}, i \in N, j \in M_i \tag{14}$$

$$d_{ij} \leq s \tag{15}$$

其中,  $N = \{1, 2, \dots, n\}$  是所有需求点的序号集合;  $M_i$  表示到需求点  $i$  的距离小于  $S$  的备选配送中心集合,  $i \in N, M_i \subseteq N$ ;  $w_i$  表示需求点的需求量;  $d_{ij}$  表示从需求点  $i$  到离它最近的配送中心  $j$  的距离;  $Z_{ij}$  表示 0~1 变量, 表示用户和物流中心的服务需求分配关系, 当  $Z_{ij} = 1$  时, 表示需求点  $j$  的需求量由配送中心  $j$  供应, 否则  $Z_{ij} = 0$ ;  $h_j$  表示 0~1 变量, 当  $h_j = 1$  时, 表示需求点  $j$  被选为配送中心;  $S$  表示新建配送中心由它服务的需求点的距离上限。

公式 (11) 保证每个需求点只能由一个配送中心服务; 公式 (12) 确保需求点的需求量只能被选为配送中心的点供应, 即没有配送中心的地点不会有客户; 公式 (13) 规定了被选为配送中心的数量为  $p$ ; 公式 (15) 保证了需求点在配送中心可配送到的范围内。

### 3 IFOA 算法优化物流配送中心选址

IFOA 算法优化物流配送中心选址的步骤如下, 流程图如图 2 所示:

Step 1: 设定 IFOA 算法的种群大小  $Popsiz$ , 最大迭代次数  $Iteration$ ;

Step 2: 根据适应度函数公式 (10) 计算果蝇个体的适应度函数值, 寻找果蝇个体和全局最优个体的位置和最优值;

Step 3: 果蝇种群速度和位置的更新;

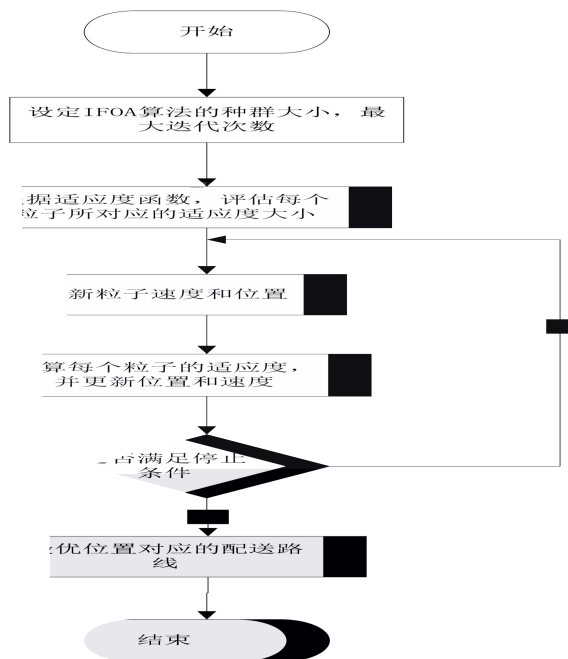


图 2 IFOA 算法优化物流配送中心选址的流程图

Fig.2 Flow chart for optimizing logistics distribution center location based on IFOA

Step 4: 计算适应度同时更新位置和速度;

Step 5: 若  $gen > Iteration$ , 保存最优解; 反之  $gen = gen + 1$ , 转到 Step3;

Step 6: 根据最优位置对应的物流配送中心的最佳位置和最优配送路径。

### 4 实验仿真

为了证明改进的果蝇优化算法进行物流配送中心选址优化选择的可行性和有效性, 收集中国 31 个城市的坐标, 每个用户的位置和其物资需求表如表 1 所示, 本文数据是经过规范化处理之后的数据, 从中选择 6 个作为物流配送中心<sup>[6]</sup>。设定果蝇种群规模  $Sizepop=20$ , 最大迭代次数  $Iteration=100$ , 物流配送中心的优化结果如图 4~7 所示:

表 1 用户位置和其物资需求量  
Table1 Users' location and their material demand

$j$	$(U_j, V_j)$	$b_j$	$j$	$(U_j, V_j)$	$b_j$
1	(1304, 2312)	20	17	(3918, 2179)	90
2	(3639, 1315)	90	18	(4061, 2370)	70
3	(4177, 2244)	90	19	(3780, 2212)	100
4	(3712, 1399)	60	20	(3676, 2578)	50
5	(3488, 1535)	70	21	(4029, 2838)	50
6	(3326, 1556)	70	22	(4263, 2931)	50
7	(3288, 1229)	40	23	(3429, 1908)	80
8	(4196, 1044)	90	24	(3507, 2376)	70
9	(4312, 790)	90	25	(3394, 2643)	80
10	(4386, 570)	70	26	(3439, 3201)	40
11	(3007, 1970)	60	27	(2935, 3240)	40
12	(2562, 1756)	40	28	(3140, 3550)	60
13	(2788, 1491)	40	29	(2545, 2357)	70
14	(2381, 1676)	40	30	(2778, 2826)	50
15	(1332, 695)	20	31	(2370, 2975)	30
16	(3715, 1678)	80			

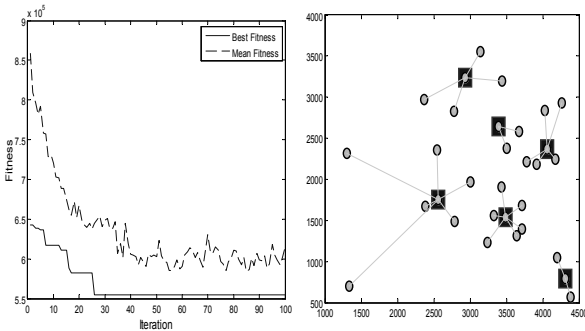


图 4 配送中心为 4 时, 配送路线和收敛图  
Fig.4 The delivery route and convergence when distribution center is 4

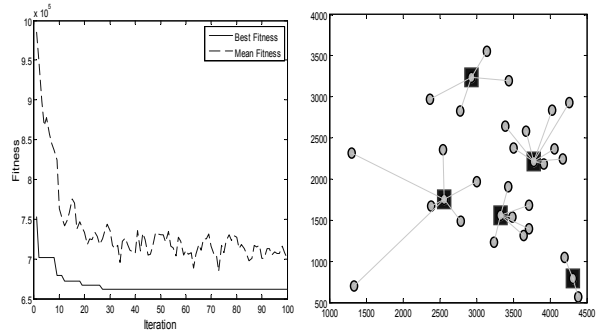


图 5 配送中心为 5 时, 配送路线和收敛图  
Fig.5 The delivery route and convergence when distribution center is 5

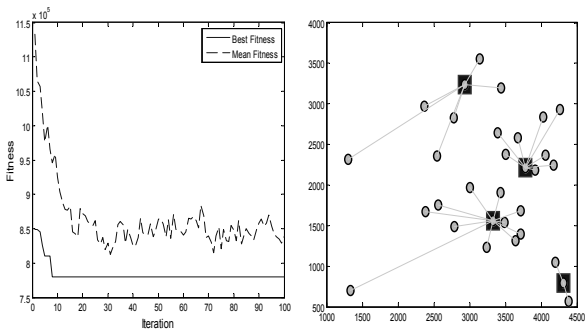


图 6 配送中心为 6 时, 配送路线和收敛图  
Fig.6 The delivery route and convergence when distribution center is 6

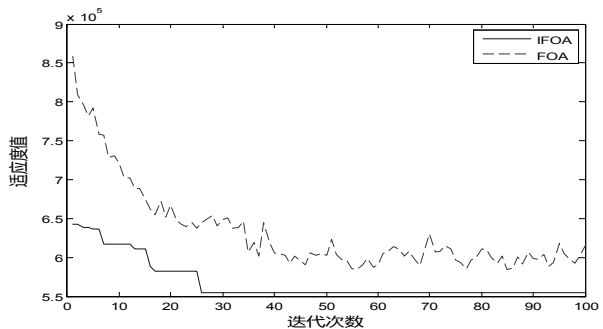


图 7 IFOA、FOA 收敛性对比图  
Fig.7 Comparison of convergences between IFOA and FOA

(下转第 639 页)