

# 基于改进原子算法的物流配送中心选址策略

王芸博

(太原城市职业技术学院,山西 太原 030027)

**摘 要:**针对电商物流配送中心选址难以优化,导致配送成本过高、配送效率低的问题,提出一种改进原子算法的电商物流配送中心选址模型。针对基本原子算法易早熟收敛而陷入局部最优、局部收敛精度不高的问题,通过提出一种非线性控制因子对算法进行改进,提高算法的收敛精度,同时引入正弦混沌搜索策略来帮助算法跳出局部最优。最后将改进后的原子算法对电商物流配送中心选址模型进行优化。实验结果表明:优化后的选址模型有效地节约了配送成本,提高了配送效率,增加了电商企业的效益。

**关 键 词:**电商物流配送中心选址;原子算法;非线性控制因子;正弦混沌搜索

中图分类号:TP301.6

文献标识码:A

文章编号:1671-2250(2021)03-0080-08

DOI:10.13777/j.cnki.issn1671-2250.2021.03.014

## 0 引言

随着计算机物联网技术的飞速发展,电商购物以及物流产业也随之得到了迅速发展。物流配送的效率和成本直接关系到电子商家的效益。由于网络购物具有配送地址分散、配送量大以及实时性高的特点,因此,如何降低物流配送成本、缩短配送时间、提高配送效率是提高电商效益的重要途径<sup>[1]</sup>。对于物流配送而言,影响其效率的主要原因分为配送路径规划和配送中心选址两个方面。其中,配送中心选址是影响配送效率的核心问题。由于配送中心选址模型具有复杂的非线性和多约束条件<sup>[2]</sup>,因此,针对配送中心选址模型的优化问题,国内外学者进行了深入的研究。

王勇等提出一种基于 K-means 聚类方法的配送中心优化策略<sup>[3]</sup>,通过聚类的方法,确定聚类中心的位置,得到最终的配送中心地址。该策略有效地缩短了配送中心到其他配送点之间的距离,提高了配送效率;但缺点在于通过模糊集成的方法进行优化,优化速度较慢,寻优精度较群智能算法低。尚猛等提出一种改进鲸鱼算法的物流配送中心选址策略<sup>[4]</sup>,通过随机正弦策略提高了鲸鱼算法的寻优精度,使得算法寻优性能更高,提高了配送效率。生力军提出一种改进量子粒子群算法的配送中心优化策略<sup>[5]</sup>,通过量子非门对粒子群算法提供变异,提高算法的局部搜索能力,降低了配送时间,节约了配送成本。Chen 等提出一种具有 Baumer-Walvar 模型的配送中心调度策略<sup>[6]</sup>,提高了求解精度和配送效率。Chi 等提出一种基于差分的布谷鸟算法的配送中心优化策略<sup>[7]</sup>,通过结合差分算法解决了因配送中心具有复杂非线性而导致算法陷入局部最优的问题,提高了配送中心模型的求解精度,提高了配送效率。以上优化策略均通过改进智能优化算法来提高优化算法的寻优精度,并改进算法求解配送中心选址模型,但都只考虑了最优极值点对算法个体的影响,并未考虑个体之间的相互作用,因而在求解具有多约束条件问题时精度并不高。

本文提出一种改进原子算法的物流配送中心选址策略。原子算法<sup>[8]</sup>与传统的群智能算法不同,在寻优过程中充分考虑了个体间的相互作用,使得算法更易求解到全局极值点。本文针对基本原子算法局部搜

收稿日期:2020-08-14

基金项目:2019 年国家自然科学基金项目(11805091)。

作者简介:王芸博(1982-),女,山西黎城人,太原城市职业技术学院信息工程系讲师,研究方向:智能优化算法。

索能力与全局搜索能力不平衡,且易陷入局部最优的问题,通过非线性控制因子和正弦混沌搜索策略进行改进,将改进后的算法对物流配送中心选址模型进行优化,并将实验结果进行对比验证。

## 1 电商物流配送中心选址模型

设有  $N$  个需配送的供需点,即有  $N$  个买家,且  $N$  个买家的地址各不相同。如何从  $N$  个供需点中选取  $M$  个配送中心,是求解物流配送中心选址模型的关键。由于  $N$  个买家的地理位置各不相同,因此在其附近建立的配送中心成本也各不相同。这使得在建立物流配送中心选址模型时,具有较多的约束条件,且模型具有复杂非线性。物流配送中心选址模型的约束条件为:

1)每个配送中心所对应的多个配送点的货物总量应小于或等于配送中心的货物总量。此约束条件的数学表达式为:

$$\sum_{j=1}^M \psi_{i,j} \leq T_i \quad i=1,2,\dots,N \quad (1)$$

式中,  $\psi_{i,j}$  表示第  $j$  个配送中心向第  $i$  个配送点配送的货物总量。

2)每个配送点的货品(即每个买家的货品)只能由离其最近的配送中心发出,此约束条件的数学表达式为:

$$\sum_{j=1}^M Z_{i,j} \leq 1_i \quad i=1,2,\dots,N \quad (2)$$

式中,  $Z_{i,j}$  为判断因子,取值为 0 或 1,当  $Z_{i,j}=1$  时,表示配送点  $i$  的需求货物由配送中心供应。

3)当任意一个配送点无对应的配送中心时,则不对其配送点进行配送,应告知电商无法对该配送点进行配送,此约束条件的数学表达式为:

$$Z_{i,j} \leq h_i \quad (3)$$

式中,  $h_i$  为 0 或 1,当  $h_i=1$  时,表示第  $i$  个配送点有相对应的配送中心。

4)设全部配送中心的个数为  $M$ :

$$\sum_{j=1}^M h_j = M \quad i=1,2,\dots,N \quad (4)$$

5)设每个配送点都应在与其对应的配送中心可配送范围之内

$$d_{i,j} \leq d_{\max} \quad (5)$$

式中,  $d_{i,j}$  为第  $i$  个配送点和与其距离最短的第  $j$  个配送中心之间的距离;  $d_{\max}$  为最大配送距离。因此,定义物流配送中心选址模型的数学表达式为:

$$\min C = \sum_{j=1}^M h_j F_j + \sum_{i=1}^N \sum_{j=1}^M \psi_{i,j} d_{i,j} Z_{i,j} \quad (6)$$

式中,  $F_j$  为每个配送中心的建设费用。

## 2 改进的原子优化算法

### 2.1 基本原子优化算法

原子算法为 2018 年提出的一类新的智能算法,具有较强的全局搜索能力。其搜索机制源于系统中各原子之间产生的相互作用力,使得原子向最优解进行移动。设在一个系统中,存在  $NP$  个原子,维数为  $D$ ,对系统中全部原子进行位置初始化,其数学模型如下:

$$X_i^d = (X_{i,1}, X_{i,2}, \dots, X_{i,d}) \quad (7)$$

式中,  $i=1,2,\dots,NP, d=1,2,\dots,D, X_i^d$  表示第  $i$  个原子在第  $d$  维上的位置。设算法最大迭代次数为  $T_{\max}$ ,则  $X_{best}^d(t)$  表示第  $t$  次迭代过程中所产生的最优解。原子在运动过程中会产生加速度,定义加速度的数学模

型为:

$$a_i^d(t) = \frac{F_i^d(t) + G_i^d(t)}{m_i^d(t)} \quad (8)$$

式中,  $F_i^d$  表示在系统中适应度值较优的  $k$  个原子对第  $i$  个原子的作用力之和, 其数学表达式为:

$$F_i^d(t) = \sum_{j \in k_{best}} -rand_j \times \eta(t) [2(h_{ij}(t))^3 - (h_{ij}(t))] \quad (9)$$

式中,  $\alpha$  为加权系数,  $h_{ij}(t)$  为任意两原子间的距离;  $\eta(t)$  表示范围控制因子, 控制引力和斥力的作用范围;  $\eta(t)$  随迭代次数的增加而非线性递减, 有利于算法在迭代后期快速收敛, 其数学表达式为:

$$\eta(t) = \alpha \left(1 - \frac{t-1}{T_{max}}\right)^3 \times \exp\left(\frac{20t}{T_{max}}\right) \quad (10)$$

在原子算法中, 原子个体会通过自身所收到的引力和约束力进行移动。在每次迭代过程中, 设定的最优个体越多, 原子收到的吸引力就会越大。因此, 在算法中希望原子个体在迭代前期受到较大的吸引力, 即设定较多的适应度值较优的个体对原子个体产生作用力。在迭代后期, 为避免原子个体陷入局部最优, 应使原子个体受到较小的吸引力, 即设定较少的适应度值较优的个体对原子个体产生作用力。因此, 在算法中, 设  $K$  为对第  $i$  个原子产生新引力的原子个数。其数学表达式如下:

$$K = NP - (NP - 2) \times \sqrt{\frac{t}{T_{max}}} \quad (11)$$

同时, 原子在移动过程中会受到最优原子的排斥力, 其数学表达式为:

$$G_i^d(t) = \beta \times \exp\left(\frac{20t}{T_{max}}\right) \times (X_{best}^d(t) - X_i^d(t)) \quad (12)$$

式中,  $\beta$  为乘数权重。在式(8)中,  $m_i^d(t)$  为原子质量, 其数学表达式为:

$$m_i^d(t) = \frac{\exp\left(\frac{fitness_i(t) - fitness_{best}(t)}{fitness_{worst}(t) - fitness_{best}(t)}\right)}{\sum_{j=1}^{NP} \exp\left(\frac{fitness_j(t) - fitness_{best}(t)}{fitness_{worst}(t) - fitness_{best}(t)}\right)} \quad (13)$$

式中,  $fitness_i(t)$  为第  $i$  个原子在第  $t$  次迭代的适应度函数值;  $fitness_{best}(t)$  为第  $t$  次迭代的最优原子;  $fitness_{worst}(t)$  为第  $t$  次迭代的最差原子。原子根据加速度更新其速度和位置。原子个体的速度和位置更新公式为:

$$\begin{cases} v_i^d(t+1) = rand_i^d \times v_i^d(t) + a_i^d(t) \\ X_i^d(t+1) = X_i^d(t) + v_i^d(t+1) \end{cases} \quad (14)$$

式中,  $v_i^d(t)$  为第  $i$  个原子在第  $t$  次迭代的速度;  $X_i^d(t)$  为第  $i$  个原子在第  $t$  次迭代的位置。

## 2.2 非线性控制因子

在基本原子算法中, 原子个体所受到的吸引力与相邻最优原子个体数相关。相邻最优原子的个数越多, 吸引力越大; 相邻最优原子的个数越少, 吸引力越小。在原子算法优化过程中, 系统中每一个原子个体均向最优原子不断地靠近, 并沿前进路线不断地更新自身位置。研究表明, 算法在迭代前期应进行大范围搜索, 以保证最优解包含在解空间内, 从而提高算法的全局收敛精度。算法在迭代后期, 应在最优解附近进行精确搜索, 以提高算法的局部开发能力和收敛速度, 使得算法可以快速收敛到全局极值点。在基本原子算法中, 相邻原子个数控制因子  $K$  随迭代次数的增加而线性减小, 递减速度较为平滑, 使得原子个体所受到的吸引力  $F_i^d$  线性递减, 但这并不符合优化的过程。对于原子算法而言, 在迭代前期应使原子个体受到较大的吸引力, 并在迭代过程中迅速减小, 使得原子个体可在迭代前期进行大范围搜索, 同时迅速向

极值点靠近;在迭代后期,应使原子个体受到较小的吸引力,并随迭代次数的增加而缓慢递减,防止原子个体在迭代后期迅速靠近极值点而陷入局部最优,使原子个体在极值点附近进行精确搜索,提高算法的局部搜索能力。改进后的相邻原子个数控制因子  $K$  如下式所示:

$$K = -23.4 \ln\left(\frac{t}{T_{\max}}\right) - 0.249 \quad (15)$$

改进后的相邻原子个数控制因子  $K$  的更新曲线如图1所示。

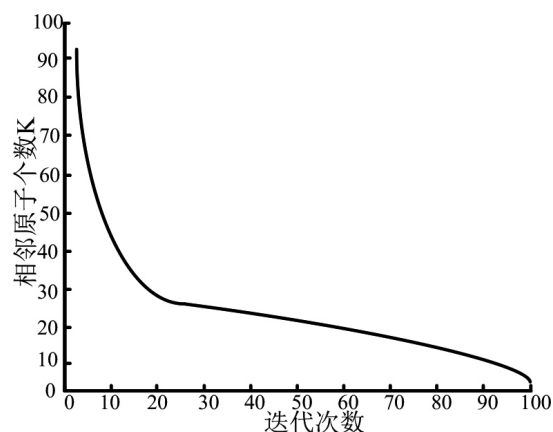


图1 改进后的相邻原子个数控制因子  $K$  的更新曲线

### 2.3 正弦混沌搜索

在基本原子算法中,系统中的原子个体因受全局极值点的吸引而不断地向极值点靠近。由于全局极值点所处位置不发生变化,最终导致系统中全部个体均围绕在当前全局极值点附近,造成算法早熟收敛陷入局部最优。针对此问题,本文引入正弦混沌搜索策略,对每次迭代所产生的最优解  $X_{best}^d(t)$  进行混沌搜索,并将当前最优解  $X_{best}^d(t)$  与正弦混沌搜索所产生的最优解  $X_{sin\_best}^d(t)$  进行贪婪选择。若  $X_{sin\_best}^d(t)$  优于  $X_{best}^d(t)$ ,则将  $X_{best}^d(t)$  的位置进行替换,以避免原子个体早熟收敛陷入局部最优。正弦混沌搜索模型为:

$$\begin{aligned} Z^{t+1} &= \sin(0.5 * Z^t, t=0, 1, \dots, N) \\ \text{s.t. } -1 &\leq Z^t \leq 1, Z^t \neq 0 \end{aligned} \quad (16)$$

式中,  $Z^0 = (z_1^0, z_2^0, \dots, z_D^0)$ , 表示在第  $i$  次迭代过程中随机生成的维数为  $D$  的不为 0 的向量;  $Z^t$  和  $Z^{t+1}$  表示第  $t$  次迭代和第  $t+1$  产生的混沌变量。基于正弦混沌搜索的原子算法优化过程如下:

**Step1:** 对原子算法迭代过程中产生的最优解  $X_{best}^d(t)$  进行混沌处理,过程如式(17)所示。式中,  $V_i^{t+1}$  为通过混沌搜索后的新个体,  $\alpha$  为调节因子,  $V_{\min}$  为搜索空间最小值,  $V_{\max}$  为搜索空间最大值,  $\delta$  为  $[0, 1]$  之间的随机数。

$$\begin{aligned} V_i^{t+1} &= X_{best}^d(t) + \alpha Z^{t+1}, V_i^{t+1} \in [V_{\min}, V_{\max}] \\ \alpha &= \begin{cases} 1 & \delta \geq 0.5 \\ -1 & \delta < 0.5 \end{cases} \end{aligned} \quad (17)$$

**Step2:** 对  $X_{best}^d(t)$  进行归一化处理,其数学表达式为:

$$\tilde{x}_{i,j} = \frac{(x_{i,j} - X_{j\min})}{(X_{j\max} - X_{j\min})}, i=1, 2, \dots, N; j=1, 2, \dots, D \quad (18)$$

式中,  $X_{j\min}$  和  $X_{j\max}$  分别为搜索可行域中的最小值和最大值;  $X_{best}^d(t) = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ 。

**Step3:** 生成混沌序列。随机生成矩阵  $Z^0$ , 并根据式(16)迭代生成  $S$  个混沌序列。

**Step4:** 将所生成的  $S$  个混沌序列通过式(17)生成  $S$  个混沌变量序列  $V_i^k = (v_{i,1}^k, v_{i,2}^k, \dots, v_{i,j}^k, \dots, v_{i,D}^k)$ , 其中  $k=(1, 2, \dots, S)$ 。

**Step5:** 通过式(19)对混沌变量序列  $V_i^k$  进行反归一化处理,产生新的解集  $U_i$ 。解集  $U_i$  中的元素可表示为:

$$u_{i,j} = x_{i,j} + \frac{X_{j\max} - X_{j\min}}{2} \times (2v_{i,j} - 1) \quad (19)$$

式中,  $u_{i,j}$  为解集  $U_i$  中的元素。

**Step6:** 计算元素  $u_{i,j}$  的适应度函数值,并与  $X_{best}^d(t)$  的适应度函数进行对比,进行贪婪选择。

## 2.4 改进原子算法的优化过程

改进后的原子算法的优化过程如下:

Step1:初始化原子算法的种群规模  $NP$ 、维数  $D$  和最大迭代次数  $T_{\max}$ ,并对原子种群进行初始化;

Step2:计算初始化后的原子个体的适应度函数值,并保存最优值  $X_{best}^d(t)$ ;

Step3:通过式(8)计算原子个体的加速度;

Step4:通过式(14)更新原子个体的位置和速度,并计算原子个体的适应度函数值,更新最优个体  $X_{best}^d(t)$ ;

Step5:通过正弦混沌策略对最优个体  $X_{best}^d(t)$  进行混沌搜索,并将搜索所得结果与  $X_{best}^d(t)$  进行对比,保存适应度值较优的个体为当前迭代的最优解;

Step6:判断是否达到最大迭代次数,若是则输出最优解,否则返回 step3。

## 2.5 数值仿真实验

为了验证本文所提的改进原子优化算法(Improved atom search optimization, IASO)的算法性能,选择 8 个基准测试函数,对 IASO 算法进行数值仿真实验,对算法的收敛精度和稳定性进行验证。8 个测试函数如表 1 所示。其中, F1~F5 为单峰测试函数,用于验证 IASO 算法的局部开发能力; F6~F8 为多峰测试函数,用于验证 IASO 算法的全

局搜索能力。为了说明实验结果的有效性,将实验结果与改进鲸鱼算法<sup>[4]</sup>和量子粒子群算法<sup>[5]</sup>的实验结果进行对比验证。为保证对比结果的公平性,三种算法的种群规模均为 100, 迭代次数均为 200。三种算法独立运行 50 次, 取得实验结果的最小值、平均值和标准差。结果如表 2 所示,最优值用加粗字体表示。

表 1 测试函数

函数	名称	搜索精度	搜索范围	最优值
F1	Sphere	$1 \times 10^{-8}$	$[-100, 100]$	0
F2	Schwefel2.22	$1 \times 10^{-8}$	$[-10, 10]$	0
F3	Schwefel1.2	$1 \times 10^{-8}$	$[-100, 100]$	0
F4	Schwefel2.21	$1 \times 10^{-8}$	$[-100, 100]$	0
F5	Rosenbrock	$1 \times 10^{-8}$	$[-5, 10]$	0
F6	Rastrigin	$1 \times 10^{-8}$	$[-5.12, 5.12]$	0
F7	Ackley	$1 \times 10^{-8}$	$[-32, 32]$	0
F8	Griewank	$1 \times 10^{-8}$	$[-600, 600]$	0

表 2 十二个测试函数的测试结果

函数	取值类型	改进原子优化算法	改进鲸鱼优化算法	量子粒子群算法	函数	取值类型	改进原子优化算法	改进鲸鱼优化算法	量子粒子群算法
F1	min	0	0	0	F5	min	$2.04 \times 10^{-04}$	$4.77 \times 10^{-02}$	$7.54 \times 10^{-02}$
	std	0	0	0		std	$4.30 \times 10^{-01}$	$6.37 \times 10^{-01}$	$6.37 \times 10^{-01}$
	ave	0	0	0		ave	$2.55 \times 10^{-01}$	$5.62 \times 10^{-01}$	$5.49 \times 10^{-01}$
F2	min	$4.79 \times 10^{-07}$	$7.88 \times 10^{-06}$	$5.30 \times 10^{-05}$	F6	min	$4.77 \times 10^{-25}$	$4.77 \times 10^{-16}$	$3.39 \times 10^{-22}$
	std	$5.06 \times 10^{-04}$	$3.25 \times 10^{-03}$	$4.66 \times 10^{-01}$		std	$5.90 \times 10^{-21}$	$3.59 \times 10^{-14}$	$5.37 \times 10^{-20}$
	ave	$4.21 \times 10^{-04}$	$2.17 \times 10^{-03}$	$3.48 \times 10^{-01}$		ave	$5.21 \times 10^{-21}$	$2.40 \times 10^{-14}$	$4.28 \times 10^{-20}$
F3	min	$2.37 \times 10^{-15}$	$6.37 \times 10^{-09}$	$8.88 \times 10^{-08}$	F7	min	0	$3.20 \times 10^{-27}$	$1.79 \times 10^{-25}$
	std	$6.33 \times 10^{-12}$	$7.54 \times 10^{-06}$	$7.49 \times 10^{-06}$		std	0	$3.18 \times 10^{-24}$	$4.30 \times 10^{-22}$
	ave	$5.27 \times 10^{-12}$	$6.24 \times 10^{-06}$	$5.26 \times 10^{-06}$		ave	0	$1.55 \times 10^{-24}$	$2.50 \times 10^{-22}$
F4	min	$5.73 \times 10^{-16}$	$4.58 \times 10^{-09}$	$9.67 \times 10^{-11}$	F8	min	0	0	0
	std	$8.60 \times 10^{-14}$	$6.31 \times 10^{-07}$	$7.50 \times 10^{-09}$		std	0	0	0
	ave	$4.59 \times 10^{-14}$	$5.22 \times 10^{-07}$	$6.51 \times 10^{-09}$		ave	0	0	0



从表 2 单峰测试函数的测试结果可以看出, 相较改进鲸鱼算法和量子粒子群算法的实验结果而言, 本文所提的改进原子算法的最小值和平均值最小。这说明改进原子算法的局部搜索能力要强于其他两种算法, 因为在每次迭代过程中均对当前迭代所得最优解进行正弦混沌搜索, 丰富了种群的多样性, 也防止了算法早熟收敛而陷入局部最优, 可以更加精确地寻找到全局最优解。同时, 改进原子算法求解的标准差最小, 说明改进原子算法的寻优稳定性和鲁棒性优于其他两种算法。这进一步表明在迭代过程中改进原子算法未出现早熟收敛的现象, 正弦混沌搜索策略有效提高了算法的局部寻优精度。

从表 2 多峰测试函数的测试结果可以看出, 改进原子算法求解的最小值和平均值同样为最小。对于测试函数 F7 和 F8 而言, 可以求解到理论最小值, 这说明通过对相邻原子个数控制因子计算方式的改进, 可以有效地平衡算法的全局搜索能力和局部搜索能力, 使原子个体可以在迭代前期进行有效快速的大范围搜索, 将全局最优解包含在搜索范围中, 又在迭代后期极速减小相邻最优原子对原子个体的新引力, 使得原子个体可以在最优解附近进行精确搜索, 不会陷入局部最优。此外, 改进原子算法求解的标准差同样最小, 这进一步说明了改进原子算法具有较强的寻优鲁棒性, 算法性能较优, 收敛精度得到了有效的提高。

总之, 通过正弦混沌搜索和非线性控制因子对算法进行改进, 算法性能得到有效的提高, 可用于对电商物流配送中心选址模型进行优化。

3 基于改进原子算法的物流配送中心选址优化实验

将改进的原子算法对电商物流配送中心选址模型进行优化, 其中每一个配送点均可视为一个原子个体, 因此原子个体可表示为  $X=[x_1,x_2,x_3\cdots,x_N]$ 。其中  $N$  表示配送点总个数, 改进原子算法从  $N$  个配送点中选取  $M$  个配送点作为配送中心, 使得配送的路劲最短、成本最低。设最终选取的最优粒子为  $X=[1,0,1,0,1,0]$ , 表示在 6 个配送点中, 选取第 1、3、5 个配送点作为配送中心。

为了验证改进原子算法对配送中心选址模型的优化效果, 选取 30 个配送点位置和供需量为基准进行实验, 同时将优化结果与基于改进鲸鱼算法的配送中心优化结果和量子粒子群算法的配送中心优化结果进行对比, 验证本文所提优化策略的高效性。其中 30 个配送点的坐标位置和供需量如表 3 所示, 实验结果如表 4 和图 2~图 4

表 3 三十个配送点地理位置及供需量

城市 序号	$(X_i,Y_i)$	供需 量	城市 序号	$(X_i,Y_i)$	供需 量
1	(1304,2312)	30	16	(3715,1678)	90
2	(3639,1315)	80	17	(3918,2179)	100
3	(4177,2244)	80	18	(4061,2370)	60
4	(3712,1399)	50	19	(3780,2212)	90
5	(3757,1187)	80	20	(3676,2578)	60
6	(3326,1556)	80	21	(4029,2838)	60
7	(3238,1229)	50	22	(4263,2931)	60
8	(4196,1044)	90	23	(3429,1908)	60
9	(4312,790)	90	24	(3507,2376)	80
10	(4386,570)	60	25	(3394,2643)	70
11	(3007,1970)	70	26	(3439,3201)	50
12	(2562,1756)	50	27	(2935,3240)	50
13	(2788,1491)	50	28	(3140,3550)	70
14	(2381,1676)	50	29	(2545,2357)	60
15	(1332,695)	40	30	(2778,2826)	40

表 4 三种算法的选址对比结果

算法	选址方案	平均配送费用	迭代次数	运行时间/s
改进原子优化算法	(27,12,20,17,5,22)	1032	29	11.4
改进鲸鱼优化算法	(12,27,20,17,7,22)	1754	56	35.2
量子粒子群算法	(27,12,25,5,18,22)	1593	41	20.8

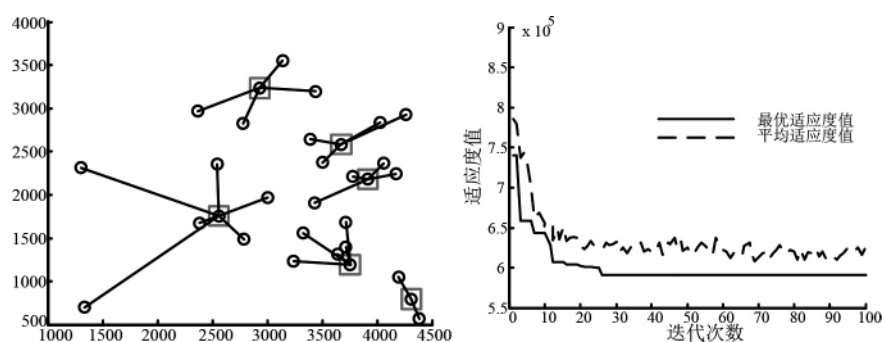


图2 基于改进原子算法的配送中心寻优结果

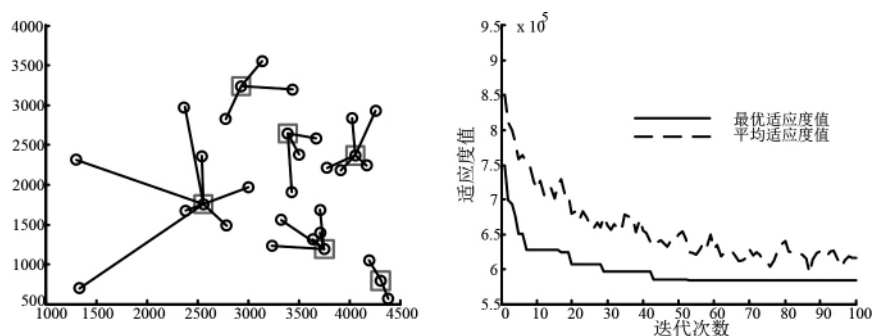


图3 基于量子粒子群算法的配送中心寻优结果

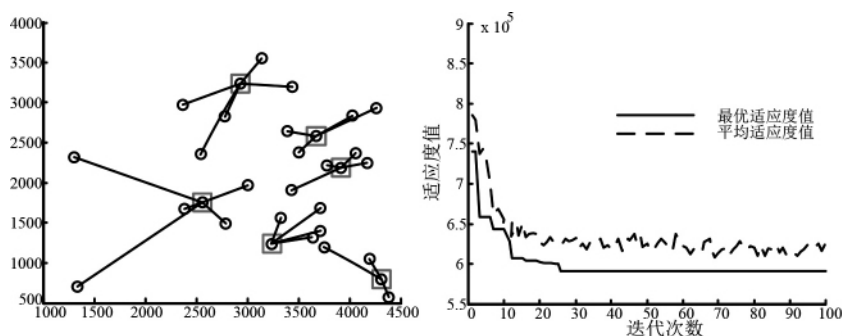


图4 基于改进鲸鱼算法的配送中心寻优结果

所示。

对比三种算法的优化结果,从表4可以看出,改进原子算法优化后的配送中心选址模型配送货品所花费的配送费用最小,配送距离最短。这说明本文所提原子优化算法的寻优精度很高,可以找到更加合理的配送中心位置。同时,原子算法在优化过程中取得最优结果所需迭代次数较少,计算所需的时间最短,精度最优,这说明本文所提改进原子算法的算法性能要优于其他三种算法,收敛速度更快。

由图2~图4可知,本文所提改进原子算法求解的适应度函数值比其他两种算法求解的适应度函数值都小,这进一步说明改进原子算法的寻优精度最高,且不易陷入局部最优。值得注意的是,改进原子算法求解的适应度函数曲线在迭代前期迅速下降,这说明算法在迭代前期的寻优范围更广,可以更加快速地收敛到全局最优解。此外,基于改进原子算法的配送中心选址策略可以求解到更小的适应度函数值,说明基于改进原子算法的配送中心优化选址策略具有更高的求解精度,在很大程度上提高了配送效率,节约了配送成本。

#### 4 结语

为提高电商效益,本文针对物流配送中心选址模型难以优化、配送成本高、配送效率低的问题,提出

一种改进原子算法的电商物流配送中心选址优化策略。针对基本原子算法易早熟收敛和算法整体寻优性能不平衡的问题,通过正弦混沌搜索策略和非线性控制因子策略进行改进,平衡了算法的全局搜索能力和局部搜索能力,避免了算法易早熟收敛陷入局部最优的缺陷,提高了算法的局部开发能力。将改进后的原子算法对电商物流配送中心选址模型进行优化,实验结果表明:优化后的模型较大幅度地提高了配送效率,降低了配送成本。

#### 参考文献:

- [1]徐小平,师喜婷. 关于物流配送中心供需优化选址仿真[J]. 计算机仿真,2018(10):345-349+423.
- [2]徐小平,杨转,刘龙. 求解物流配送中心选址问题的蜘蛛猴算法[J]. 计算机工程与应用,2020(1):150-157.
- [3]王勇,黄思奇,刘永,许茂增. 基于 K-means 聚类方法的物流多配送中心选址优化研究[J]. 公路交通科技,2020(1):141-148.
- [4]尚猛,康建英,曹峻玮,万志鹏. 基于改进鲸鱼优化算法的物流配送中心选址策略[J]. 计算机应用与软件,2019(6):254-259.
- [5]生力军. 基于量子粒子群算法的物流配送中心选址[J]. 科学技术与工程,2019(11):183-187.
- [6]Chen H. Location Problem of Distribution Center Based on Baumer Walvar Model: Taking Jiaji Logistics as an Example [J]. Open Journal of Business and Management, 2019(2):1042-1052.
- [7]Chi R, Su Y, Qu Z et al. A Hybridization of Cuckoo Search and Differential Evolution for the Logistics Distribution Center Location Problem[J]. Mathematical Problems in Engineering, 2019(65):221-232.
- [8]Weiguo Zhao, Liying Wang, Zhenxing Zhang. A novel atom search optimization for dispersion coefficient estimation in groundwater[J]. Future Generation Computer Systems, 2018(91):601-610.

## Location Strategy of Logistics Distribution Center Based on Improved Atomic Algorithm

WANG Yun-bo

(Taiyuan City Vocational College, Taiyuan Shanxi 030027)

**Abstract:** Concerning the location of e-commerce logistics distribution center is difficult to optimize, resulting in high distribution cost and low distribution efficiency, this paper proposed an improved atomic algorithm for e-commerce logistics distribution center location model. In order to solve the problem of premature convergence, falling into local optimum and low accuracy of local convergence of basic atomic algorithm, a nonlinear control factor was proposed to improve the convergence accuracy of the algorithm. At the same time, sine chaos search strategy was introduced to help the algorithm jump out of local optimum. Finally, the efficiency of the distribution center was improved, and the results showed that the efficiency of the distribution center was improved.

**Key Words:** logistics distribution center model; atom search optimization; nonlinear control factor; sine chaos search

(责任编辑:顾亿天)