

文章编号: 1674-8085(2020)01-0048-06

融合多种策略的改进粒子群算法及其在 电子商务多级物流中的应用研究

*郭海峰, 张翠玲

(沈阳理工大学自动化与电气工程学院, 辽宁, 沈阳 110159)

摘 要: 为了提高粒子群优化算法(PSO)求解复杂优化问题的能力, 本文对基于细菌趋化的粒子群优化算法(PSOBC)进行改进。PSOBC 算法是 PSO 算法的一种新思路, 可以有效地克服其易陷入局部最优、后期粒子多样性差的缺点, 故将一般反向学习策略和自适应惯性权重与 PSOBC 算法相结合, 得到一种改进的粒子群优化算法。改进的粒子群优化算法的开发能力和勘探能力都得到了很大的提高; 在求解复杂性优化问题时种群能够在搜索范围内快速收敛到局部最优处, 并且当种群密度足够小时, 及时增大种群密度即进行去全局寻优。最后将改进后算法应用到电子商务多级物流中心选址及路径规划问题上。

关键词: 粒子群优化算法; 反向学习; 自适应惯性权重; 物流中心选址; 路径规划

中图分类号: TH16;F252

文献标识码: A

DOI:10.3969/j.issn.1674-8085.2020.01.010

IMPROVED PARTICLE SWARM OPTIMIZATION ALGORITHM WITH MULTIPLE STRATEGIES AND ITS APPLICATION IN MULTI-LEVEL LOGISTICS OF E-COMMERCE

*GUO Hai-feng, ZHANG Cui-ling

(School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang, Liaoning 110159, China)

Abstract: In order to improve the ability of particle swarm optimization (PSO) to solve complex optimization problems, this paper improves the PSO algorithm based on bacterial chemotaxis (PSOBC). Aiming at the problem of slow convergence and low convergence of PSO in solving complexity problems, a general inverse learning strategy and adaptive inertia weight are combined with PSOBC algorithm to obtain an improved particle swarm optimization algorithm. The improved particle swarm optimization algorithm has greatly improved the development and exploration capabilities; it can quickly converge to the local optimum within the search capability when solving the complexity optimization problem, and increase the population density when it is small enough. The large population density is to perform global optimization. Finally, the improved algorithm is applied to the e-commerce multi-level logistics center location and path planning.

Key words: particle swarm optimization; reverse learning; adaptive inertia weight; logistics center location; path planning

粒子群优化算法^[1] (Particle Swarm Optimization, PSO)自提出以来便受到国内外学者的广泛关注。PSO 算法作为现代群体智能算法, 有

无限的发展潜力, 被广泛应用在各个领域。但传统 PSO 存在局部搜索能力差、求解精度低、易陷入局部最优和十分依赖各个参数的问题。为了解决这些

收稿日期: 2019-08-11; 修改日期: 2019-11-14

作者简介: *郭海峰(1970-), 男, 山东莱芜人, 教授, 博士, 主要从事先进控制理论与应用、物流与供应链运作技术等研究(E-mail:ghf_1970@163.com); 张翠玲(1996-), 女, 河北省邢台市人, 硕士生, 主要从事粒子群与物流优化方面的研究(E-Mail:17363190519@163.com)。

问题,很多学者从参数设置、学习策略、领域结构等方面融合其他算法对 PSO 算法进行改进,提出了许多不同的粒子群改进算法^[2-5]。在传统 PSO 中每个粒子都是一个候选解,粒子在进化过程中是不断向个体历史最优和全局最优的学习过程,导致种群易陷入局部最优。所以,要提高算法的全局搜索能力,改进学习策略是关键。文献[6]中提出了一种求解大规模问题的自学习协同粒子群算法,将自学习策略引入到协同粒子群算法中,提高了算法的全局搜索能力,使算法在求解大规模问题中表现出较大的潜力。

本文将一般反向学习策略(GOBL)^[7]和自适应惯性权重^[8]与基于细菌趋化的粒子群算法^[9]相结合,提出了新的改进算法。反向学习策略增加了种群进化的多样性,使种群进化方向由单一向个体历史最优和全局最优方向变成多方向,提高了算法的局部寻优能力^[10];自适应惯性权重改变了基本 PSO 中单一惯性,充利用种群信息,增强种群多样性和算法的全局寻优能力。

1 融合多种策略的改进粒子群优化算法

1.1 PSO 存在的问题及改进方法

传统 PSO 存在局部搜索能力差、收敛速度快,易陷入局部最优且搜索精度低;而对于复杂优化问题,往往存在很多极值点,PSO 在求解过程中收敛速度慢、收敛度低且很难得到最优解。

针对 PSO 易陷入局部最优的问题,采用 PSOBC^[7]算法中以种群密度来度量种群当前的状态,当种群密度过高时,进行“吸引”操作,加快种群的收敛速度,从而得到较优解;当种群密度过低时,判断种群目前已达到某个局部最优,此时进行“排斥”操作,来增加种群的多样性,避免种群陷入这个局部最优,以寻找下一个局部最优;最终从多个局部最优选出全局最优。针对种群在“吸引”阶段中收敛过快的问题,采用自适应惯性权重^[8](如公式(1)所示),粒子根据自身所处状态来调节惯性权重,而非以统一的惯性权重不加区别地对自身速度予以保持。而“排斥”操作,通过粒子跟

随历史最差位置和全局最差位置来实现,速度更新如公式(2)所示。

$$\begin{cases} (w_{start} - w_{end}) * (fitness_i - fitness_{min}) / (mean(fitness) - fitness_{min}) + w_{end}, & others \\ w_{start}, & fitness_i > mean(fitness) \end{cases} \quad (1)$$

其中, $fitness_i$ 为当前粒子的适应值; $fitness_{min}$ 为当前种群中适应值的最小值; $mean(fitness)$ 为当代粒子的平均适应值。

$$v_{id}^{t+1} = w v_{id}^t + c_1 r_1 (x_{id}^t - w_{id}^t) + c_2 r_2 (x_{id}^t - w_{gd}^t) \quad (2)$$

其中, w_{id}^t 、 w_{gd}^t 分别表示第 i 个粒子的历史最差位置和全局最差位置在 d 维上的分量。

针对 PSO 在求解复杂优化问题时收敛度慢和收敛度低的问题,将一般反向学习策略(GOBL)引入到算法^[8]中。因为当给定一个随机候选解 x 时,其对立点 x' 可能更靠优化问题的最优解,所以在引入反向学习策略后,不仅可以加快收敛速度,一定程度上提高了算法的寻优能力。

GOBL 定义: 设 $\tilde{x}_i$ 是粒子 x_i 的反向解, 则

$$\tilde{x}_{ij} = k(a_{ij} + b_{ij}) - x_{ij} \quad (3)$$

其中, k 为服从均匀分布的(0,1)之间的随机数; $[a_{ij}, b_{ij}]$ 是粒子 i 在 j 维度的搜索范围; 若 $\tilde{x}_i \notin [a_{ij}, b_{ij}]$, 则将 $\tilde{x}_i$ 取 $[a_{ij}, b_{ij}]$ 的随机值; 即 $\tilde{x}_i$ 始终是 $[a_{ij}, b_{ij}]$ 之间的位置。

1.2 改进后粒子群优化算法

改进后粒子群优化算法伪代码:

- 1) initialize P, X, V
- 2) 更新搜索范围;
- 3) for $i=1:PopSize$
生成反向种群 G
- 4) end
- 5) fitness(X); fitness(G) // 计算适应度
- 6) for $i=PopSize+1:2*PopSize$
选取前 Popsiz 个粒子
- 7) end
- 8) Pbest, Gbest // 选取种群历史最优和个体历史最优
- 9) while iter<Maxiter
- 10) iter=iter+1;

```

11)   div=div/(sqrt(UB^2+LB^2)*PopSize) //
判断种群多样性
12)   根据种群密度取 Mode 值
13)   if Mode==1
14)   w_now=((w_start-w_end)*(Maxiter-iter)/M
axiter)+w_end // 更新惯性权重
15)   V=w_now*V+c1*R1*(Pbest-X)+c2*R2*(G
best-X) // 更新速度
16)   X=X+V;
17)   fitness(X)
18)   更新 Pbest, Gbest
19)   else
20)   反向学习生成种群 G, 并计算种群适
应度, 选取前 PopSize 个粒子
21)   end
22) end
23) 输出最优值

```

2 问题描述与数学模型

2.1 问题描述

多级物流配送中心选址及路径规划问题需要从一级配送中心 M 和二级配送中心 H 中选择部分物流中心进行建设, 即闭环供应链网络关键节点的构建, 然后规划供货商 T 、一级物流中心 M 、二级物流中心 H 及城市区 C 之间的服务与被服务关系, 即完成整个闭环供应链中的网络搭建。闭环供应链运营包括配送、运输、退货处理、横向调度及纵向调度等操作 (如图 1 所示)。

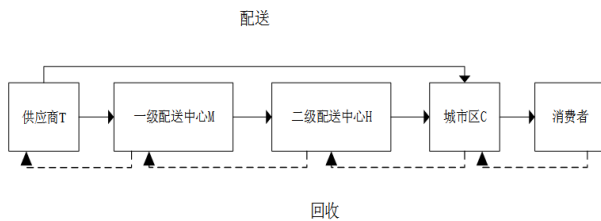


图 1 多级物流中的产品流动
Fig.1 Product flow in multi-level logistics

当某城市客户区有产品需求时, 首先由服务于该客户区的二级物流配送中心提供, 若该二级配送中心缺货, 则优先从其他二级配送中心调度即纵向调度, 当所有二级配送中心都无法满足时, 由一级物流配送中心提供紧急配送服务即横向调度, 且每

级配送中都有回收退回产品的能力。

为了方便模拟解决实际问题, 先对问题做了如下假设: 1) 一级配送中心 M 不存在缺货和容量不足的问题; 2) 只有供货商有鉴定和修复退回产品的能力; 3) 客户的产品需求单一; 4) 暂不考虑运输时货车的承载能力。

2.2 数学模型

2.2.1 符号定义

符号含义	符号	符号含义	符号
供应商集	T	设施点的建造成本	F
一级备选配送中心集	M	设施点的可利用时长	V
二级备选配送中心集	H	城市区的需求	D
城市区集	R	配送处理成本	S
退货收集成本	t	设施点 i 运到 j 的产品数	q_{ij}
二级配送中心的库存量	θ	设施点 i 到 j 的距离	d_{ij}
二级配送中心横向调度单位运输成本	f	城市区 r 的退货率	kr
二级配送中心单位新产品存储容量系数	γ	二级配送中心单位退回产品存储容量系数	β
节点间的单位产品运输成本	$G_{tm}, G_{mh}, G_{hr}, G_{rh}, G_{hm}, G_{mr}, G_{hh}$	二级配送中心的库存处理最大容量	Q
一级配送中心是否建设	$k_m \in (0,1)$	二级配送中心是否建设	$b_k \in (0,1)$
一级配送中心 m 是否服务于二级配送中心 h	$c_{mh} \in (0,1)$	二级配送中心 h 是否服务于城市区 r	$c_{hr} \in (0,1)$

2.2.2 数学模型

建立数学模型, 即在保证满足各个城市区产品需求(如公式(6)所示)、二级物流配送中心库存处理最大容量(如公式(7)所示)的条件下, 使整个闭环供应链的成本最小, 即公式(4)所表示含义。

$$\begin{aligned}
\min = & \sum_{m \in M} F_m k_m / V_m + \sum_{h \in H} F_h b_h / V_h + \\
& \sum_{t \in T} \sum_{m \in M} G_{tm} q_{tm} d_{tm} + \sum_{m \in M} \sum_{h \in H} (G_{mh} d_{mh} + S_m) q_{mh} c_{mh} + \\
& \sum_{h' \in H} \sum_{h \in H} (f d_{h'h} + S_{h'}) q_{h'h} + \\
& \sum_{m \in M} \sum_{r \in R} (G_{mr} d_{mr} + S_m) q_{mr} + \\
& \sum_{r \in R} \sum_{h \in H} (G_{rh} d_{rh} + t_h) q_{hr} k_r c_{hr} +
\end{aligned}$$

$$\sum_{h \in H} \sum_{m \in M} \left\{ (G_{hm} d_{hm} + t_m) \sum_{r \in R} q_{hr} k_r c_{hr} \right\} + \sum_{m \in M} \sum_{t \in T} \left\{ G_{mt} d_{mt} \sum_{h \in H} \sum_{r \in R} q_{hr} k_r c_{hr} c_{mh} \right\} \quad (4)$$

约束条件为:

$$\sum_{t \in T} \sum_{m \in M} q_{tm} = \sum_{m \in M} \sum_{t \in T} q_{mh} = \sum_{h \in H} \sum_{r \in R} q_{hr} + \sum_{m \in M} \sum_{r \in R} q_{mr} \quad (5)$$

$$q_{mh} \gamma + \sum_{r \in R} q_{hr} k_r \beta \leq Q_h, h \in H \quad (6)$$

$$\sum_{r \in R} q_{hr} (\gamma + \beta_{kr}) \leq Q_h \quad (7)$$

3 算法设计

在求解该问题时, 需要对每个粒子进行编码。

编码方式为: 粒子表示为 $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$, 其中 d 为粒子的维度, 与一级备选物流配送中心个数、二级备选物流中心个数和城市区个数之和相等, 且 $x_i \in (0,1)$ 。根据算法的思想, 每个粒子都应该对应一个该问题的候选解, 因此应将每个粒子都映射成为一个问题的候选解。映射方法主要包括从备选集中确定保留的设施点和路径规划 (即对服务于本身的上一级被保留的设施点的选择) 两方面。因为路

径规划依赖于保留设施点的选择, 所以先选择保留的设施点, 再进行路径规划, 此过程称为解码。解码过程如图 2 所示, $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 对应 t 个供应商, 解码后的 0/1 表示 0 或 1, 0 表示该供应商未被一级配送中心选择, 反之, 解码为 1 表示被选择; $x_{i(t+1)}, x_{i(t+2)}, \dots, x_{i(t+m)}$ 对应 m 个一级备选配送中心, 解码后为 $[1, k]$ 之间的一个整数 1 或 0 (k 为被选的供应商个数), 0 表示该一级配送中心未被选择, 1 表示该一级中心被选择且由第 1 个被选择的供货商为其提供服务; 二级配送中心的选择与一级配送中心解码策略相同; 而对于城市区的不同之处在与对应值不为 0, 因为一定有二级配送中心为其提供服务。

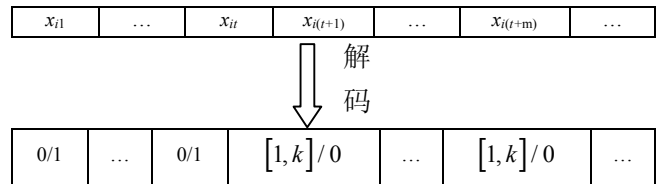


图 2 粒子解码过程

Fig.2 Particle decoding process

本文以问题的目标函数 (即公式 (4)) 作为适值函数, 算法流程如图 3 所示。

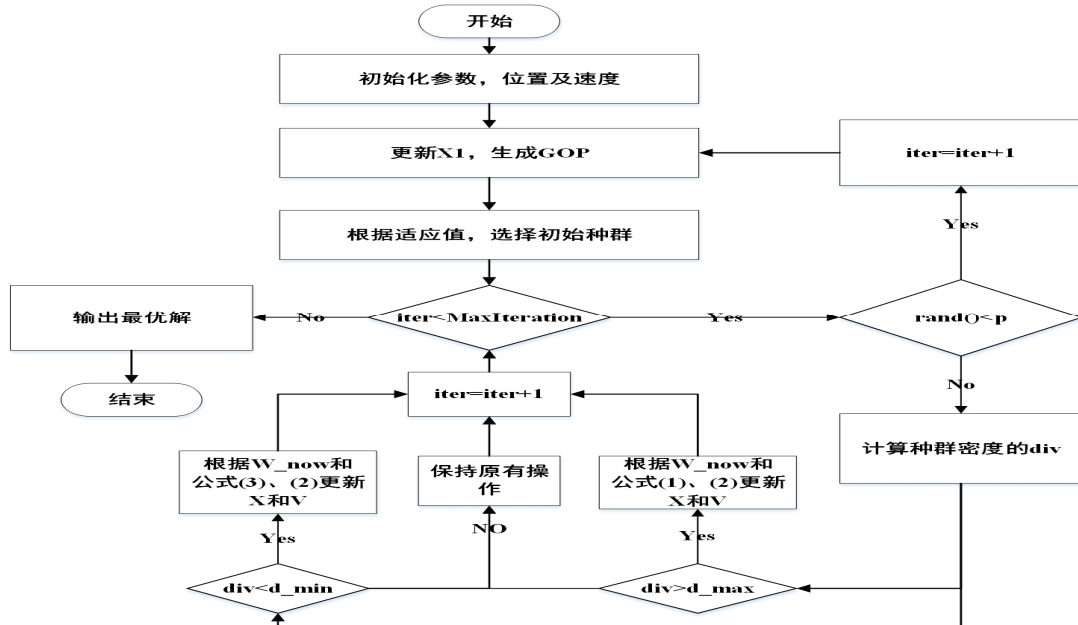


图 3 算法流程图

Fig.3 Algorithm flow chart

4 算例分析

使用改进算法求解多级物流中心选址及其路径规划问题，种群大小 $N=30$ ，最大迭代次数 $\text{MaxIteration}=200$ ，惯性权重 $w \in [0.9, 0.4]$ ，种群密度 $d_{\max} = 0.25, d_{\min} = 0.0005$ ，反向学习概率 $p=0.3$ ，学习因子 $c_1 = c_2 = 1.49618$ ，独立运行 30 次。

实验所用数据为参考文献[11]中的实验数据。该组数据的供应商、一级备选物流中心、二级备选物流配送中心和城市区的个数分别为 2,2,6,14；它们的位置分布如图 4 所示。

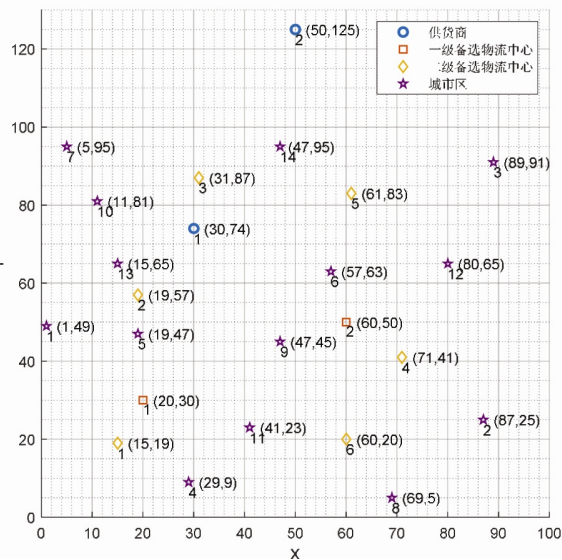


图 4 设施点位置分布
Fig.4 Location distribution of facilities

一级和二级备选物流中心的配送处理成本 $S=0.8$ ；一级备选物流中心的退回处理成本为 0.5，可利用月份均为 120，建造成本依次为 50 万、60 万；二级备选物流中心的退货收集成本 $t=1$ ，产品的储存容量系数 $\gamma=1, \beta=2$ ；处理容量 $Q=1600$ ，库存量 $\theta=1000$ ，可利用月份均为 60，建造成本依次为 8 万、10 万、13 万、10 万、11 万、9 万。城市区的需求量如表 1 所示。供应链运作的基本数据如表 2 所示。

表 1 城市区的需求量
Table 1 Demand in urban areas

城市区	1	2	3	4	5	6	7
月需求量	300	300	158	135	110	220	140
城市区	8	9	10	11	12	13	14
月需求量	105	100	105	220	215	330	280

表 2 供应链运作的基本数据
Table 2 Basic data of supply chain operation

G_{lm}	G_{mh}	G_{hy}	G_{yh}
0.02	0.045	0.05	0.08
G_{hm}	G_{hh}	G_{mr}	k_r
0.06	0.4	0.8	0.3

采用 Matlab R2018b 在 Windows10 系统中，运用改进的粒子群优化算法对上述问题进行求解，最终结果为 26435。此时选择建设第一个一级物流中心，且选择第一个供应商对其进行供货；选择建设第 2、第 5、第 6 三个二级配送中心，其中二级配送中心与城市区之间的服务关系如图 5 所示。虽然此时第 6 个城市区距离第 5 个二级物流配送中心比较近，但是此时却选择了第 2 个二级物流配送中心；与第 6 个城市区情况类似的有第 8、第 13 城市区。但是此时这种选择即没有横向调度也没有纵向调度，相比之下成本反而会降低。

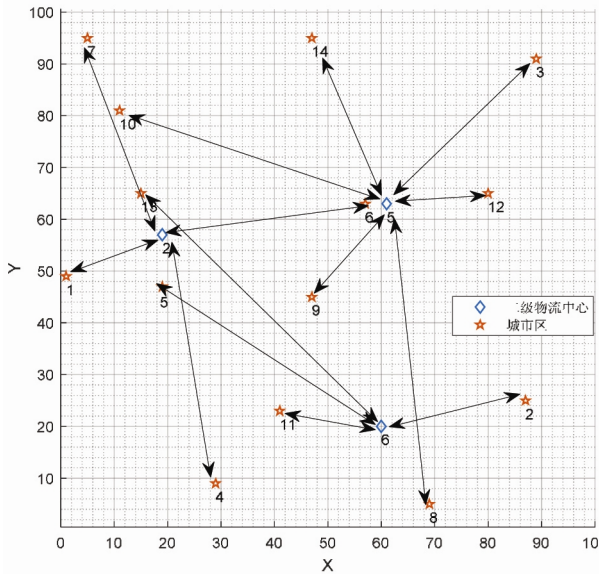


图 5 二级物流中心与城市区的服务关系
Fig.5 Service relationship between secondary logistics center and urban area

5 结束语

本文对基于细菌趋化的粒子群算法进行改进，以及针对多级物流中心选址和路径规划问题设计合适的编码，能够有效求解该问题。虽然改进算法是有效的，但是无法破解群智能算法求解不唯一的难题，每次都只能找到一个较优解，且无法复现。

在这方面仍需要继续改进,继续提高算法的搜索能力。

参考文献:

- [1] Kennedyj, eberhartc. Particleswarmoptim- ization[C]. IEEEInternationalConferenceonNeuralNetworks. Perth, Australia, 1995:1942-1948.
- [2] 刘浩然,崔静闯,卢泽丹,等. 一种改进的雁群扩展粒子群算法研究[J]. 计量学报, 2019, 40(03): 498-504.
- [3] 张兰勇,孟坤,刘胜,等. 基于改进双粒子群算法的舰船电力系统网络故障重构[J]. 电力系统保护与控制, 2019, 47(09): 90-96.
- [4] 丁芳,宋小静. 共享适应度粒子群在双机ETV中的应用[J]. 计算机测量与控制, 2018, 26(11): 228-232+247.
- [5] 谷磊. 基于改进粒子群算法优化的车辆运动侧倾控制研究[J]. 井冈山大学学报: 自然科学版, 2018, 39(06): 74-78.
- [6] 肖根福,刘欢,李东洋,等. 一种求解大规模问题的自学习协同粒子群算法[J]. 井冈山大学学报: 自然科学版, 2018(3): 32-37.
- [7] Angh, wuzj, rahnamay ANS, et al. Enhancing particleswarmoptimizationusing generalizedop- position- based learning[J]. InformationSciences, 2011, 181: 4699-4714.
- [8] 康岚兰,董文永,宋婉娟,等. 无惯性自适应精英变异反向粒子群优化算法[J]. 通信学报, 2017, 38(08): 66-78
- [9] 周涛. 基于细菌趋化的改进粒子群算法在电力系统无功优化中的应用[J]. 上海电力学院学报, 2014, 30(4): 315-323.
- [10] 曹生让,丁晓群,王庆燕,等. 基于反向云自适应粒子群算法的多目标无功优化[J]. 中国电力, 2018, 7(51): 21-27.
- [11] 李帅. 基于电子商务的物流与供应链网络优化问题研究[D]. 沈阳: 沈阳理工大学. 2014.