

文章编号:1009-3087(2014)06-0038-06

# 一种新的混合差分粒子群优化算法及其应用

沈济南,梁芳\*,郑明辉

(湖北民族学院 计算机科学与技术系,湖北 恩施 445000)

**摘要:**针对 B2C 电子商务物流配送优化精度不高的问题,提出基于一种新的混合差分粒子群启发式优化算法的 B2C 电子商务物流配送优化方案。首先,将粒子种群作为辅助变异算子与差分进化算法种群进行交叉操作,产生的新子代继承了父代和母代的优势特性,从而避免了单一算法的早熟收敛和收敛速度过慢的问题。通过与已有的改进算法仿真对比,该算法能够有效地跳出局部极值,防止算法早熟且收敛速度很快。其次,借鉴已有文献方法对混合算法在 B2C 路径优化问题中的工程应用进行了实验研究,通过仿真显示所设计配送方案具有更快的计算速度和更优的目标收敛值。

**关键词:**混合差分粒子群;定向变异;物流路径优化

**中图分类号:**TP18

**文献标志码:**A

## New Hybrid Differential Evolution and Particle Swarm Optimization Algorithm and Its Application

SHEN Jinan, LIANG Fang\*, ZHENG Minghui

(Dept. of Computer Sci. & Technol., Hubei Minzu Univ., Enshi 445000, China)

**Abstract:** In order to solve the problem that the B2C electronic commerce logistics distribution optimization accuracy is not high, a new hybrid differential evolution and particle swarm optimization heuristic optimization algorithm based B2C e-commerce logistics distribution optimization was proposed. Firstly, the particle swarm population was used as auxiliary mutation operator to do the crossover operation with differential evolution algorithm population, resulting new generation that inherits the advantages of characteristics of parents, thus avoiding the single algorithm premature convergence and low convergence rate. Through simulation comparison with the other existing improved algorithm, the algorithm here can effectively escape from local minima, prevent premature convergence, and the convergence is fastest. The simulation showed the designed distribution scheme has the faster calculation speed and better convergence of the target value.

**Key words:** hybrid differential particle swarm; directional variation; logistics path optimization

近年来以遗传算法、粒子群算法和差分进化算法为代表的群智能优化算法得到了迅速的发展和广泛应用,与传统优化方法相比,这些智能优化算法的优势在于能够保证收敛的前提下,对所求优化问题动力学信息不苛求,具有全局优化能力<sup>[1]</sup>。

差分进化算法(DE)是众多群智能优化算法中性能相对优秀的一种仿生计算策略,如果说标准遗传算法的基因编码是一种二进制的编码方式,那么

差分进化算法类似于一种实数编码的遗传算法。影响差分进化算法性能的操作主要有3个:一是差分变异操作,二是差分交叉操作,三是差分选择操作。可能算法设计者在设计之初考虑的重点是针对低维算法快速收敛,所以选择操作采用的是一对一贪婪机制,这种机制有利于收敛但是个体趋同性太强,非常容易被局部干扰极值点吸引而早熟<sup>[2-3]</sup>。

粒子群(或微粒群)算法(PSO)也是一种典型

收稿日期:2014-06-27

基金项目:国家自然科学基金资助项目(61173175; 61262078)

作者简介:沈济南(1980—),男,讲师,硕士。研究方向:计算机软件与理论;信息安全。E-mail:shenjinan@163.com

\*通信联系人 E-mail:maple\_liang@163.com

网络出版时间:2014-9-1 18:13:41 网络出版地址: <http://www.cnki.net/kcms/detail/51.1596.T.20140903.1435.004.html>

<http://jsuese.scu.edu.cn>

的群智能算法,算法的最大特点是机理简单并且易于程序实现,在维持种群个体的多样性方面具有独到的特点,相对于遗传算法等具有相对较高的算法运行效率。在 PSO 算法中,群体中所有的粒子具有相同的搜索行为,并表现出相同的全局或局部搜索能力<sup>[4]</sup>。

PSO 和 DE 算法都是一种基于群体进化的智能优化算法,他们自身的特点和优势,同样也都存在缺陷和不足。首先,DE 是一种基于随机种群搜索的算法,它通过特有的变异、交叉和选择操作使种群能够快速进化到最优值附近,算法简单控制参数少易于实现<sup>[5]</sup>。DE 算法在具有简易和快速收敛性的同时,其在种群多样性方面优势不明显,算法易陷入局部极小。其次,PSO 算法也比较简单便于实现,并且在种群多样性保持上具有一定优势,收敛速度相对较快。

国内外进化算法研究者早就注意到 2 种算法所存在的问题,提出了很多有针对性的解决方案。文献[6]设计了一种类似于专家周期调整的搜索区域自动调整方案,根据个体进化情况和迭代次数对搜索空间进行有目的的收放,从而使全局极值点尽可能的处在相对合理的位置,通过这种新颖的方式提高算法性能。文献[7]提出了 2 种新的改进算法:DERL 和 DELB,并进行了大量对比仿真,验证了改进算法的有效性。文献[8]将改进的目标定位在算法的粒子更新公式上,增加并设计了自适应惯性权重因子  $\omega$ 。文献[9]提出一种采用繁殖概率取代适应值的杂交粒子群算法框架,这种交叉操作类似于差分进化算法的交叉操作,实现了提高算法性能的目的。

另一方面,为突破单一算法的局限,继承和发扬算法各自的优点,国内外学者对混合种群智能优化算法进行了很多研究和探讨。文献[1]设计基于遗传粒子群的多群多层混合算法,下层的遗传算法主要负责算法的全局开发,上层的由底层精英组成的种群采用粒子群算法负责局部的深度探索。文献[10]设计了差分和粒子群双种群算法,并提出一种协同信息交换机制。文献[11]提出一种差分进化与粒子群双种群协同进化算法(DEPSO),仿真结果显示了有效性。

作者深入研究差分和粒子群 2 种算法的不同特点,提出了一种新的混合启发式优化算法(NHDEPSO),并对 NHDEPSO 算法在物流路径优化问题中的应用进行了实验研究。

## 1 NHDEPSO 算法

### 1.1 基本 DE 算法

算法首先要取得 1 组在搜索空间上随机初始化的种群:

$$\mathbf{X}^0 = [x_1^0, x_2^0, \dots, x_{N_p}^0] \quad (1)$$

$N_p$  为种群规模,经过一系列规定的操作,第  $t$  代个体进化为:

$$\mathbf{x}_i^t = [x_{i,1}^t, x_{i,2}^t, \dots, x_{i,D}^t] \quad (2)$$

式中,  $D$  为所优化问题的维数。

DE/rand/1、DE/best/2 变异形式:

$$x_m = x_{s3}^t + F \times (x_{s1}^t - x_{s2}^t) \quad (3)$$

$$x_m = x_{\text{gbest}}^t + F \times [(x_{s1}^t - x_{s2}^t) + (x_{s3}^t - x_{s4}^t)] \quad (4)$$

式(3)和(4)中,  $x_{\text{gbest}}^t$  为本代进化种群的最佳个体位置;  $x_{s1}^t, x_{s2}^t, x_{s3}^t, x_{s4}^t$  为 4 个各异的随机粒子;  $F \in [0, 2]$  为缩放因子。

交叉操作:

$$x_{Tj} = \begin{cases} x_{mj}, & \text{rand} \leq CR; \\ x_{ij}, & \text{rand} > CR \end{cases} \quad (5)$$

式中,  $j = 1, 2, \dots, D$ 。

选择操作:

$$x_i^{t+1} = \begin{cases} x_T, & f(x_T) < f(x_i^t); \\ x_i^t, & f(x_T) \geq f(x_i^t) \end{cases} \quad (6)$$

### 1.2 算法改进

标准 PSO 算法位置变换公式:

$$\underbrace{p_i^j(t+1)}_{\text{下一刻位置}} = \underbrace{p_i^j(t)}_{\text{当前位置}} + \underbrace{\omega \cdot v_i^j(t)}_{\text{当前速度参考}} + \underbrace{c_1 \cdot r_1 \cdot (pb_i^j(t) - p_i^j(t))}_{\text{个体历史最优参考 } Ob_j} + \underbrace{c_2 \cdot r_2 \cdot (pg_i^j(t) - p_i^j(t))}_{\text{群体最优参考 } G_{gj}} \quad (7)$$

式中:上标  $i$  对应于第  $i$  个个体 ( $i = 1, \dots, N, N$  为种群规模),下标  $j$  对应于粒子的第  $j$  维 ( $j = 1, \dots, n$ ),  $t$  表示当前迭代代数;  $\omega$  惯性因子;  $\mathbf{P}^i = [p_1^i, \dots, p_n^i]^T$  和  $\mathbf{v}^i = [v_1^i, \dots, v_n^i]^T$  各自代表第  $i$  个个体的位置、速度,  $|v_j^i| \leq v_{\max}$ ;  $c_1, c_2$  分别是个体历史最优和全体最优的加速因子;  $r_1, r_2$  为  $[0, 1]$  内随机数。

对于差分进化算法和粒子群算法的改进,已有很多学者进行了研究,这里采用两种简单的改进方式,来提高算法的性能。文献[12]从矢量分析层面进行算法变异的理论性研究,设计了四向量交叉变异形式主要目的是平衡个体多样与算法运行效率。个体变异操作的方程为:

$$x_i^{t+1} = x_i^t + F((x_{i1}^t - x_i^t) + (x_{i2}^t - x_{i3}^t)) \quad (8)$$

文献[13]提出平常遇到的 PSO 算法基本是混

合粒子“位置”及“速度”的交叉更新方式,并且其他已有的改进算法也大多采用围绕这两个量进行操作改进(增加、删除或修改),使得算法描述越来越复杂。对此作者经过研究分析相关文献资料,提出一种简化的不含速度项的粒子群位置变换公式:

$$p_j^i(t+1) = \omega p_j^i(t) + c_1 r_1 (pb_j^i(t) - p_j^i(t)) + c_2 r_2 (pg_j^i(t) - p_j^i(t)) \quad (9)$$

式中:等号右侧的首项常被称为算法“历史”项,代表历史对当前的影响;次项被称为“认知”项,代表个体对本身的思考;末项被称为“社会”项,代表个体与邻居个体的信息互享。

### 1.3 NHDEPSO 算法步骤

Step1:设置 NHDEPSO 算法的相关参数,如  $N_p$ 、 $D$ 、 $F$ 、 $CR$ 、 $x_{\min}$ 、 $x_{\max}$ 、 $c_1$ 、 $c_2$  等参数,迭代次数  $t = 0$ , 设置变空间次数最大值  $T$ 。

Step2:在给定的搜索区域  $[x_{\min}, x_{\max}]$  中采用随机的方式生成  $N_p$  个个体  $P(0)$ 。

Step3:计算群体  $P(t)$  中每一个体的评价函数值  $f(i)$ ,并根据评价值对  $pb_j^i(t)$  和  $pg_j^i(t)$  进行赋值。

Step4:根据式(8)方式进行变异生成下一代种群  $P_1(t)$ ,按照式(9)进行变异操作产生群体  $P_2(t)$ 。

Step5: $P_1(t)$  和  $P_2(t)$  按照式(5)进行交叉操作产生下一代种群  $P'(t)$ 。

Step6:按照式(6)对种群  $P_1(t)$  和  $P'(t)$  进行选择操作产生子代种群  $P(t+1)$ ,并对种群  $P(t+1)$  个体进行评价。

Step7:判断当前 NHDEPSO 算法是否符合终止要求,是则终止 NHDEPSO 算法并输出,否则跳转步骤3继续进行参数优化。

## 2 算法性能分析

设定 PS-GDE 算法的实验仿真参数:国际上通用的进化计算仿真参数选取标准中,算法维数一般取  $D = 30$ ,而种群数量选取与之相关一般取所设定维度的 5 ~ 10 倍<sup>[4]</sup>,种群数量的选取要平衡 2 个方面:种群数量大有利于多样性维持但是增加算法计算负担,二是种群数量小与之相反,需要根据经验选取相对合理的种群数量。取  $NP = 200$ ,迭代最大次数 8 000,文献[4]给出的差分缩放因子的取值范围是  $F \in [0.4, 0.8]$ ,取  $F = 0.6$ ,差分交叉概率因子的变化范围  $CR \in [0.3, 0.9]$ ,取  $CR = 0.6$ 。设置 PSO 参数  $\omega = 0.729$ ,  $c_1 = c_2 = 1.494 45$ <sup>[14]</sup>。

测试函数如表 1 所示。

表 1 测试函数

Tab. 1 Test function

函数名	表达式	取值范围
Dejong	$f_1 = \sum_{i=1}^n x_i^2$	$ x_i  \leq 100$
Griewank	$f_2 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$ x_i  \leq 100$
Rosenbrock	$f_3 = \sum_{i=1}^n (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$ x_i  \leq 100$

对比算法选取标准 DE、PSO 和 DEPSO 算法,DE 和 PSO 参数的设置同上,DEPSO 参数设置参见文献[11]中的设置,仿真精度  $VTR = 10^{-6}$ 。

仿真结果如图 1~3 所示,为清晰显示出对比曲线位置,采用对数方式对收敛曲线进行放大。

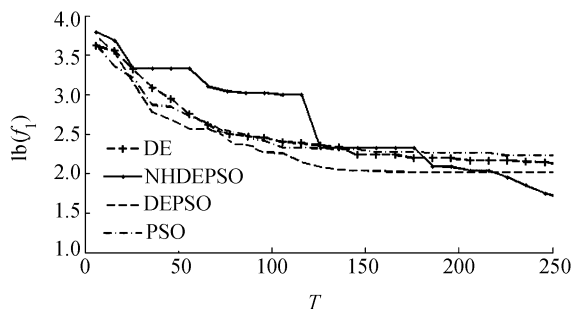


图 1 目标函数  $f_1$  的收敛曲线

Fig. 1 Convergence curves of  $f_1$  objective function

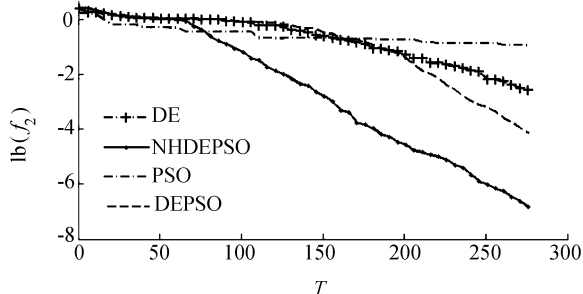


图 2 目标函数  $f_2$  的收敛曲线

Fig. 2 Convergence curves of  $f_2$  objective function

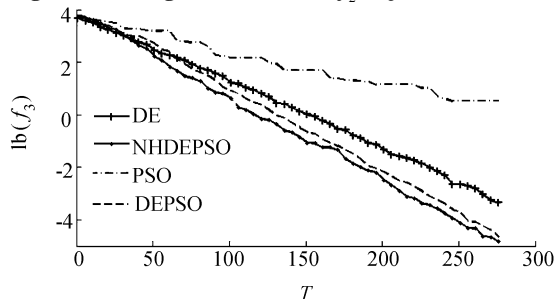


图 3 目标函数  $f_3$  的收敛曲线

Fig. 3 Convergence curves of  $f_3$  objective function

对文献[15]测试函数  $f_1 \sim f_4$ 、文献[16]测试函数  $f_7$  和  $f_8$  进行仿真分析,参数设置同上。

为便于比较 4 种方法计算 6 个测试函数的收敛情况,参考文献[15],若求得的结果与理想最优解误差小于 $10^{-4}$ ,就视其为达到收敛,最大迭代代数为 4 000。算法第一次达到收敛的代数、时间分别称为

收敛代数、收敛时间。采用 MATLAB 2013a 编程工具,计算机的配置为:AMD Athlon X4 641 2.8 GHz,4 GHz 内存,Win7。每个算法独立执行 50 次。平均试验的结果如表 2 所示。

表 2 基准函数的优化结果比较

Tab.2 Comparison of results of the benchmark function optimization

算法	性能	$f_1$	$f_2$	$f_3$	$f_4$	$f_7$	$f_8$
PSO	收敛率	0.27	0.31	0.21	0.19	0.35	0.30
	平均收敛代数	2 358	3 520	3 200	3 321	1 920	1 832
	平均收敛时间/ms	45.32	86.22	75.26	82.35	37.25	35.26
DE	收敛率	0.34	0.24	0.33	0.23	0.40	0.38
	平均收敛代数	2 198	3 652	2 635	3 457	1 735	1 507
	平均收敛时间/ms	41.25	61.23	49.67	58.35	32.48	27.89
DEPSO	收敛率	0.60	0.58	0.73	0.67	0.80	0.90
	平均收敛代数	1 423	2 347	1 378	1 736	536	796
	平均收敛时间/ms	37.25	46.35	35.26	39.57	15.6	23.12
NHDEPSO	收敛率	0.82	0.79	0.92	0.89	0.93	0.97
	平均收敛代数	807	1 423	652	1 576	825	756
	平均收敛时间/ms	25.12	37.35	20.14	37.58	22.25	20.35

从图 1 ~ 3 及表 2 对比数据可以看出,NHDEPSO 算法在 $f_1 \sim f_3, f_1 \sim f_4, f_7 \sim f_8$  测试函数上的寻优性能普遍好于其他几种对比算法。例如如图 1 中,在 $f_1$  函数上 DE、PSO 和 DEPSO 算法均趋向于早熟收敛,但是 NHDEPSO 却能够在保持种群多样性,防止早熟收敛以及收敛速度上具有明显优势。

### 3 NHDEPSO 物流路径优化算法

#### 3.1 B2C 电子商务物流配送数学模型

文献[17]设计了 B2C 电子商务物流路径优化目标:

$$\text{Min} \sum_{k \in K} \sum_{\substack{i \in D \\ (i,p) \in E}} \sum_{\substack{j \in D \\ (i,j) \in E}} C_{ij} d_{ij} x_{ijk} + \sum_{k \in K} B_k \text{Max}_{j \in D} (z_{jk}) \quad (10)$$

满足如下条件:

$$\left\{ \begin{aligned} &\sum_{\substack{i \in D \\ (i,p) \in E}} x_{ipk} - \sum_{\substack{j \in D \\ (p,j) \in E}} x_{pj k} = 0, \forall p \in S, k \in K_p; \\ &\sum_{\substack{i \in H \\ (i,j) \in E}} x_{ijk} \geq z_{jk}, \forall j \in D, k \in K; \\ &\sum_{k \in K} z_{jk} = 1, \forall j \in D; \\ &\sum_{k \in K_p} \text{Max}_{j \in D} (z_{jk}) \leq A_p, \forall p \in S \end{aligned} \right. \quad (11)$$

$$\left\{ \begin{aligned} &\sum_{j \in D} \left( \sum_{l \in L} \omega_l q_{jl} \right) z_{jk} \leq Q, \forall k \in K; \\ &\sum_{j \in D} q_{jl} y_{ij} - v_{il} \leq 0, \forall i \in S, l \in L; \\ &x_{ijk} = (0,1), \forall i \in H, j \in H, k \in K; \\ &y_{ij} = (0,1), \forall i \in S, j \in D; \\ &z_{jk} = (0,1), \forall j \in D, k \in K \end{aligned} \right. \quad (12)$$

模型中, $y_{ij}$  表示顾客  $j$  是否由配送中心  $i$  配送, $x_{ijk}$  表示车辆  $k$  是否从节点  $i$  开往节点  $j, z_{jk}$  表示顾客  $j$  是否由车辆  $k$  配送, $G$  为配送站, $L$  为物流商品的类别集, $K$  为物流配送车辆集, $B_k$  为车辆  $k$  的启动费用, $A_p$  表示站点  $p$  的可用车数量, $q_{jl}$  表示顾客  $j$  的商品需求量, $C_{ij}$  表示车辆  $k$  在路线  $(i, j)$  间的单位运输费用, $d_{ij}$  表示节点  $i, j$  间的距离, $\omega_l$  为重量系数, $Q$  为单车装载量, $v_{il}$  表示配送中心  $i$  商品供应量。

#### 3.2 基于 NHDEPSO 算法的配送路线优化仿真

3 个配送中心的实际配送网络图如图 4 所示,基于 NHDEPSO 算法 B2C 物流配送最优配送路径如图 5 所示。

3 个配送中心可用车辆数均为 3 辆,车辆启动费用为 400 元,最大载重量均为 10 t,单位距离费用 5 元,3 类商品的重量系数分别为  $\omega_1 = 0.2 \text{ t/件}, \omega_2 = 0.3 \text{ t/件}, \omega_3 = 0.5 \text{ t/件}$ ,该模型的最优配送值是 2 085 元,其他

相关参数见文献[18]。为了验证基于 NHDEPSO 算法的物流分配方案的有效性,将基于 NHDEPSO 算法的优化方案分别与基于 DE 和 PSO 算法进行对比,各随机计算 10 次求平均值,如表 3 所示。

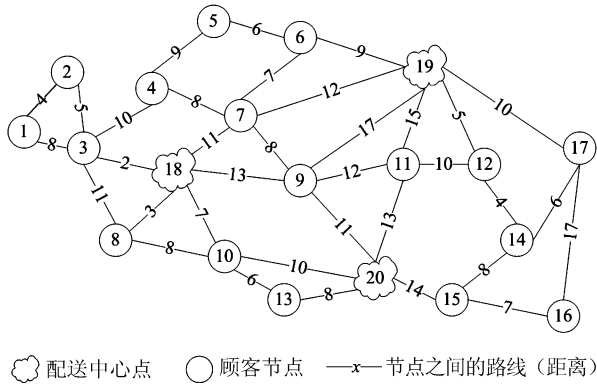


图 4 实际配送网络图

Fig. 4 Actual distribution network diagram

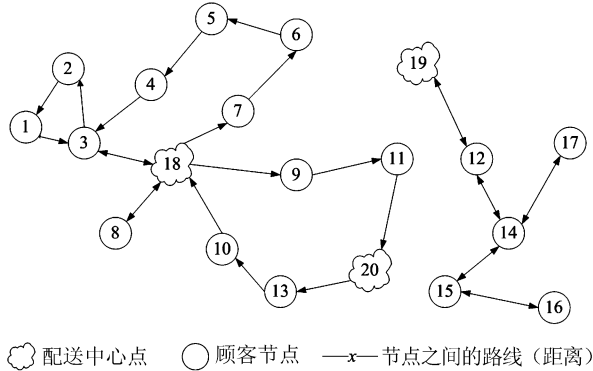


图 5 基于 NHDEPSO 算法的车辆最优配送路

Fig. 5 Vehicle optimum distribution path based on NHDEPSO algorithm

表 3 仿真结果对比

Tab. 3 Comparison of the simulation results

计算次数	目标值/元			计算时间/s		
	PSO	DE	NHDEPSO	PSO	DE	NHDEPSO
1	2 352	2 148	2 085	0.952	0.756	0.658
2	2 248	2 096	2 085	1.253	0.563	0.547
3	2 153	2 085	2 085	1.123	0.624	0.589
4	2 458	2 085	2 085	1.258	0.524	0.668
5	2 247	2 124	2 085	1.145	0.468	0.758
6	2 189	2 085	2 085	0.975	0.587	0.987
7	2 315	2 178	2 085	1.365	0.652	0.752
8	2 478	2 085	2 097	1.287	0.498	0.675
9	2 185	2 179	2 085	0.976	0.513	0.694
10	2 368	2 085	2 085	0.953	0.487	0.587
平均值	2 299.3	2 115.0	2 086.2	1.129	0.567	0.692
标准差	113.95	39.58	3.79	0.133	0.070	0.085

从表 3 对比数据可以看出,虽然在程序运算时间上基于 NHDEPSO 的物流配送优化算法略长于基于 DE 算法的优化算法,但是在目标函数的优化值上基于 NHDEPSO 算法的优化值要具有明显优势,10 次优化计算的最优值的平均值和标准值均小于基于其他两种基本算法的物流配送方案。因此,基于 NHDEPSO 算法的 B2C 物流配送路线方案是可行和高效的。

## 4 结论

从遗传思想角度出发根据差分进化和粒子群算法的不同性能特点,设计了一种新的混合差分进化粒子群混合算法 NHDEPSO,该算法能够较为全面的继承差分进化和粒子群 2 种算法各自特点,取长补短的提高混合算法的性能。通过仿真对比显示该算法在保持种群多样性和收敛性能上要优于所对比算法。最后结合该 NHDEPSO 算法对 B2C 电子商务物流配送路径优化问题提出一种可行的解决方案,仿真结果显示 NHDEPSO 算法的路径选择方案要优于对比的基于差分进化和粒子群算法的物流配送选取方案。

### 参考文献:

- [1] Jin Min, Lu Huaxiang. A multi-subgroup hierarchical hybrid of genetic algorithm and particle swarm optimization [J]. Control Theory & Applications, 2013, 30(10): 1231 - 1237. [金敏,鲁华祥.一种遗传算法与粒子群优化的多子群分层混合算法[J].控制理论与应用,2013,30(10):1231 - 1237.]
- [2] Storm R. On the usage of differential evolution for function optimization[C]//Proceedings of Biennial Conference of the North American Fuzzy Information Processing Society. Berkley: NAFIPS, 1996: 519 - 526.
- [3] Zhang Jie, Pei Fang. Multi-strategy cooperative evolutionary particle swarm optimization [J]. Application Research of Computers, 2013, 30(10): 2965 - 2967. [张洁,裴芳.多策略协同进行粒子群优化算法[J].计算机应用研究,2013,30(10):2965 - 2967.]
- [4] Korm F, Muth S. Influence sets based on reverse nearest neighbor queries [C]//Proceedings of ACM SIGMOD International Conference on Management of Date. New York: ACM Press, 2000: 201 - 212.

- [5] Cheema M A, Zhang W J, Lin X M. Continuous reverse K nearest neighbors queries in Euclidean space and in spatial networks[J]. The VLDB Journal, 2012, 21(1): 69 - 95.
- [6] Yao Feng, Yang Weidong, Zhang Ming, et al. Improved space-adaptive-based differential evolution algorithm[J]. Control Theory & Applications, 2010, 27(1): 32 - 38. [姚峰, 杨卫东, 张明, 等. 改进自适应空间差分进化算法[J]. 控制理论与应用, 2010, 27(1): 32 - 38.]
- [7] Kaelo P, Ali M M. A numerical study of some modified differential evolution algorithms[J]. European Journal of Operational Research, 2006, 169: 1176 - 1184.
- [8] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]//Proceedings of IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 1998: 69 - 73.
- [9] Lovbjerg M, Rasmussen T K, Krink T. Hybrid particle swarm optimization with breeding and subpopulations [C]//Proceedings of the IEEE International Conference on Evolutionary Computation. San Francisco: IEEE, 2000: 1217 - 1222.
- [10] Yang Chunhua, Qian Xiaoshan, Gui Weihua. Hybrid algorithm of chaotic differential evolution and particle swarm optimization [J]. Application Research of Computers, 2011, 28(2): 32 - 35. [阳春华, 钱晓山, 桂卫华. 一种混沌差分进化和粒子群优化混合算法[J]. 计算机应用研究, 2011, 28(2): 32 - 35.]
- [11] Sayah S, Hamouda A. A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems[J]. Applied Soft Computing, 2013, 13(4): 1608 - 1619.
- [12] Yang Weidong, Yao Feng, Zhang Ming. Differential evolution algorithm based on adaptive crossover probability factor and its application[J]. Information and Control, 2010, 39(2): 187 - 193. [杨卫东, 姚峰, 张明. 基于自适应交叉概率因子的差分进化算法及其应用[J]. 信息与控制, 2010, 39(2): 187 - 193.]
- [13] Hu Wang, Li Zhishu. A simpler and more effective particle swarm optimization algorithm[J]. Journal of Software, 2007, 18(4): 861 - 868. [胡旺, 李志蜀. 一种更简化而高效的粒子群优化算法[J]. 软件学报, 2007, 18(4): 861 - 868.]
- [14] Eberhart R C, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization [C]//Proceedings of the IEEE Conference on Evolutionary Computation. New York: IEEE, 2000: 84 - 88.
- [15] Wang L F, Zeng J C. A cooperative evolutionary algorithm based on particle swarm optimization and simulated annealing algorithm [J]. Acta Automatica Sinica, 2006, 32(4): 630 - 635.
- [16] Du H F, Gong M G, Liu R C. Adaptive chaos clonal evolutionary programming [J]. Science in China, Series E: Information Sciences, 2005, 35(8): 817 - 829.
- [17] Jiang Zhongzhong, Wang Dingwei. Model and algorithm for logistics distribution routing of B2C ecommerce [J]. Information and Control, 2005, 34(1): 481 - 485. [蒋忠中, 汪定伟. B2C 电子商务中物流配送路径优化的模型与算法[J]. 信息与控制, 2005, 34(1): 481 - 485.]
- [18] 汪定伟, 王俊伟, 王洪峰, 等. 智能优化算法 [M]. 北京: 高等教育出版社, 2007.

(编辑 杨 蓓)