

爆炸搜索算法及其收敛性证明

曹 炬 侯学卿

(华中科技大学大学数学与统计学院 武汉 430074)

摘 要 受烟花(炸弹)爆炸的启发,结合经典优化算法提出了一种新的智能优化算法——爆炸搜索算法(Explosion Search Algorithm,ESA)。ESA 引入邻域搜索的思想,将智能优化算法与下降搜索算法进行有机结合,使得 ESA 具有强大的局部搜索能力和全局搜索能力以及好的收敛精度。对算法的收敛性进行了证明,最后通过对 benchmark 函数集进行仿真并同其他算法进行比较,验证了 ESA 的高效性。

关键词 智能优化算法,爆炸搜索算法,差商最速下降搜索,邻域搜索,收敛

中图法分类号 TP301 **文献标识码** A

Explosion Search Algorithm and its Convergence

CAO Ju HOU Xue-qing

(School of Mathematics and Statistics, Huazhong University of Science & Technology, Wuhan 430074, China)

Abstract Inspired by explosion of fireworks(bomb), a new intelligence search algorithm was proposed, which is called Explosion Search Algorithm(ESA). One theory defined as Neighborhood Search was proposed in ESA, the Steepest descent search algorithm was introduced into the ESA, which makes this new algorithm stronger ability of global search as well as local search. The convergence of the algorithm was also proved in this article. The simulation using standard benchmark functions and comparison with other algorithms proved the efficiency of the new algorithm.

Keywords Intelligence optimization algorithm, Explosion search algorithm, Difference steepest descent search, Neighborhood search, Convergence

1 引言

优化技术是一种以数学为基础,用于求解各种工程问题优化的应用技术,作为一个重要的科学分支一直受到人们的广泛重视。同时产生了许多经典的算法,来求解此类问题,例如最速下降方法^[1]、共轭梯度方法等。然而现实中有许多工程问题呈现复杂化、多极化、非线性、强约束等特点,很难用传统的算法进行求解。近些年来,人们根据自然界的一些生命现象提出了许多智能算法,如遗传算法(GA)^[2,3]、微粒群优化算法(PSO)^[4,5]等。很多 NP 问题可以通过这些优化算法得到较好的求解,但精度却无法与经典算法相比。

本文将提出一种全新的算法:爆炸搜索算法(Explosion Search Algorithm,ESA)。算法是受烟花(炸弹)爆炸现象的启发提出的。算法将智能算法与经典算法进行结合,在按照智能策略进行搜索的同时执行最速下降搜索,从而使算法具有智能算法的效率和经典算法的精度。算法具有收敛速度快、精度高、收敛性好等特点。

2 爆炸搜索算法(ESA)

炸弹爆炸时,能够对周围的一个圆形邻域产生影响。而且随着炸弹威力的不同,影响的区域大小也不相同。如果将

该区域看成空间中的一个局部区域,炸出的弹片看成区域中的点,一次爆炸就类似于对局部区域的一次探索。受此启发,如果把一次爆炸看作对炸点位置周围的一次覆盖型探索,则这种探索可看作空间中该点附近的一次局部搜索。受此启发,ESA 提出了邻域搜索的思想,在算法的实现中加入自适应因子,使得探索区域能自适应地增大,炸点可以移动较大的距离,实现了全局的搜索。同时利用差商最速下降法进行精确搜索,提高算法的精度。

2.1 爆炸搜索算法的算法流程

本文中所设计的算法主要适用的问题可以表示为 $\min f(x): \Omega \subset \mathbb{R}^n (x_i \in [l_i, r_i], i=1, 2, \dots, n)$, 即无约束的优化问题。在算法中称搜索点为炸点, $f(x)$ 为评测函数, 函数值越小表示炸点位置越好。

在爆炸搜索算法中,主要包括 4 个算子:爆炸搜索算子(Explosion Search Operator)、迁移算子(Moving Operator)、变异算子(Mutation Operator)以及历史最优点(*gbest*)下降搜索算子(Steepest descent search for *gbest*)。算法流程如下,其中 *gbest* 为历史最优点。

Initialize: 在搜索空间内初始化 M 个炸点,并计算各个炸点的破坏程度(目标函数值),并存储目标函数值最小的点到 *gbest*;

While(终止条件不满足)

 每个炸点执行爆炸搜索算子;

到稿日期:2010-12-08 返修日期:2011-03-21

曹 炬(1955—),男,博士,教授,主要研究方向为运筹学和智能计算及其应用;侯学卿(1986—),男,硕士生,主要研究方向为智能算法及其应用, E-mail: hxqing_hust@gmail.com(通信作者)。

执行迁移算子;
 执行变异算子;
 gbest 最速下降更新算子;
 更新 gbest

End While

输出最好的个体。

2.2 爆炸搜索算法中各算子的实现

爆炸搜索算法的执行流程在上面已经给出,接下来给出各个算子的实现方式。其中 gbest 更新的操作为:从各个炸点以及 gbest 中选取最好的点替换 gbest。

2.2.1 爆炸搜索算子

爆炸搜索算子主要实现邻域搜索,即在一个局部区域内找到一个“最好”的点替换现有炸点位置,完成炸点位置的更新。另 $Rt=(Rt_1, Rt_2, \dots, Rt_n)$, 定义 $\|Rt\|_2$ 为搜索半径,即炸点 X 爆炸所影响的区域为 $B(X, \|Rt\|_2)$ 。本算子在搜索区域内选择一个方向,并在该方向上选择一定数量的点进行评测,选出最好的点与 X 进行比较。

令 $e_i=(0, 0, \dots, 1, \dots, 0)$ (即只有第 i 维为 1 的 n 维向量), 定义 $\bar{g}=(\bar{g}_1, \bar{g}_2, \dots, \bar{g}_n)$, 其中

$$\bar{g}_i = \frac{f(X + \Delta x_i e_i) - f(X)}{\Delta x_i}$$

称 \bar{g} 为 X 点的类梯度向量。本算子用向量 \bar{g} 代替传统算法中的梯度方向,在 \bar{g} 方向上选点评测。算子中令 $\Delta X=(\Delta x_1, \Delta x_2, \dots, \Delta x_n)=rand() \cdot Rt$, 由此计算出类梯度方向 \bar{g} 。令 Num 为评测点的个数,接下来从点集 $\{X - i \cdot \|Rt\|_2 \frac{\bar{g}}{\|\bar{g}\|_2} | i=1, 2, \dots, Num\}$ 中选择最好的一个点 X_Temp 。如果 X_Temp 比炸点 X 好,则 $X=X_Temp$ 作为新的炸点位置,否则炸点 X 不发生变化。

为了保证算法不陷入局部最小,参数 Rt 是自适应的。定义参数 $init_Rt$ (随迭代代数线性减小):

$$init_Rt = Rt_{max} - t * (Rt_{max} - Rt_{min}) / T$$

式中, t 表示当前迭代代数, T 表示最大迭代代数, Rt_{max}, Rt_{min} 表示 $init_Rt$ 的上限和下限。

如果 X_Temp 优于炸点 X , 则炸点 X 位置发生更新, $Rt=init_Rt$, 否则 $Rt=\alpha Rt$, 其中 $\alpha > 1$ 为比例因子。这样,当炸点 X 陷入局部最小时,随着 Rt 的增大,有助于 X 跳出局部最优。

2.2.2 迁移算子

令 μ 表示迁移比例,即最差的 $[\mu M]$ 个炸点发生迁移。本算子的执行流程为:选择最差的 $[\mu M]$ 个炸点,使这些炸点向历史最优值 $gbest$ 移动。更新公式如下:

$$X' = X + 2 * rand() * (gbest - X)$$

通过迁移算子可以使差的炸点向最好的位置移动,快速跳出差的位置,从而提高算法的效率。同时,该算子也能在一定程度上提高算法的收敛速度。

2.2.3 变异算子

引入变异算子主要是增加随机性、多样性,任何炸点都有可能发生变异。本算子引入参数 P_M 表示变异概率,推荐 P_M 取 $0.01 \sim 0.1$ 。炸点 $X=(x_1, x_2, \dots, x_n)$ 发生变异,按照下面公式更新位置:

$$x_i' = x_i (1 + N(0, \sigma_i))$$

式中, $N(0, \sigma_i)$ 表示一个均值为 0, 标准差为 σ_i 的正态分布随

机数,其中

$$\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n) = \gamma (X_{max} - X_{min})$$

式中, X_{max}, X_{min} 为搜索空间的上下限。 γ 推荐取 0.1。

2.2.4 gbest 下降搜索算子

该算子完成对于历史最优值附近的一维搜索。一维搜索的方向向量仍选用类负梯度方向,只是 ΔX 取极小值。令 $h=(h_1, h_2, \dots, h_n)=\Delta X=\eta(X_{max}-X_{min})$, 在本算子中 η 取非常小的值,从而保证 \bar{g} 接近梯度方向。另 λ_k 为最佳步长,在计算时 λ_k 可以按照文献[1]中的方法估计出。如果 $f(gbest - \lambda_k \bar{g}) < f(gbest)$, 则令 $gbest = gbest - \lambda_k \bar{g}$, 否则不改变 $gbest$ 。

3 收敛性证明

3.1 gbest 下降搜索算子在凸函数上的证明

假定目标函数 $f(x) (x \in R^n)$ 为凸函数且二次连续可微,并且满足下面条件:

令 $G(x) = \nabla^2 f(x)$, 存在 $0 < m < M < +\infty$, 使得

$$m \|y\|^2 \leq y^T G(x) y \leq \frac{M}{n} \|y\|^2$$

式中, n 为函数维数。

由 Taylor 展式得

$$f(x^{(k)} + h_i e_i) = f(x^{(k)}) + h_i g_k^T e_i + \frac{1}{2} h_i^2 e_i^T \nabla^2 f(x^{(k)}) e_i + \theta h_i e_i$$

其中, $0 < \theta < 1$ 。

即

$$\frac{f(x^{(k)} + h_i e_i) - f(x^{(k)})}{h_i} = g_k^T e_i + \frac{1}{2} h_i e_i^T \nabla^2 f(x^{(k)}) e_i + \theta h_i e_i$$

令 $C_k = (c_{k,1}, c_{k,2}, \dots, c_{k,n}) \in R^n$, 其中

$$c_{k,i} = h_i e_i^T \nabla^2 f(x^{(k)}) e_i + \theta h_i e_i, i=1, 2, \dots, n$$

则

$$\bar{g}_k = g_k + \frac{1}{2} C_k$$

$-g_k$ 为负梯度方向,如果能够证明 $-\bar{g}_k$ 与 $-g_k$ 的夹角小于 90° , 则 $-\bar{g}_k$ 为下降方向。

引理 1^[1] 若 $\max\{|h_1|, |h_2|, \dots, |h_n|\} \leq \frac{1}{M} \|\bar{g}_k\|$, 令 $d_k = -\bar{g}_k$, 则 d_k 为下降方向,且最佳搜索步长 λ_k 满足 $\frac{1}{2M} \leq \lambda_k \leq \frac{3}{2n}$ 。

定理 1 $d_k = -\bar{g}_k$ 与负梯度方向有一致夹角,即 $\exists \varepsilon_0 > 0, \forall k$ 有

$$\cos \langle d_k, -g_k \rangle = \frac{g_k^T \bar{g}_k}{\|g_k\| \|\bar{g}_k\|} \geq \varepsilon_0$$

证明详见文献[1]。

引理 2^[1] 令 $d_k = -\bar{g}_k, \lambda_k$ 是最佳步长因子,则对 $\forall k$ 有

$$f(x^{(k)}) - f(x^{(k)} + \lambda_k d_k) \geq \frac{1}{2M} g_k^2 \cos \langle d_k, -g_k \rangle$$

定理 2 如果 $\max\{|h_1|, |h_2|, \dots, |h_n|\} \leq \frac{1}{M} \|\bar{g}_k\|$, 则按照 gbest 最速下降搜索算子进行迭代,有 $f(gbest) \rightarrow -\infty$ 或者 $\|\bar{g}_k\|^2 \rightarrow 0$ 。

证明:由引理 2 知:

$$f(x^{(k)}) - f(x^{(k+1)}) \geq \frac{1}{2M} g_k^2 \cos \langle d_k, -g_k \rangle$$

因此有

$$f(x^{(0)}) - f(x^{(k)}) = \sum_{i=0}^{k-1} [f(x^{(i)}) - f(x^{(i+1)})]$$

$$\begin{aligned} &\geq \frac{1}{2M} \sum_{i=0}^{k-1} \|g_i\|^2 \cos(d_i, -g_i) \\ &\geq \frac{1}{2M} \sum_{i=0}^{k-1} \|g_i\|^2 \varepsilon_0 \end{aligned}$$

两边同时取极限,则 $g_{best} \rightarrow -\infty$ 或者 $\|\bar{g}_k\|^2 \rightarrow 0$ 。

3.2 爆炸搜索算法收敛性证明

引理 3 对于 n 维连续函数 $f(x)$ ($x_i \in [l_i, r_i], l_i < r_i, i=1, 2, \dots, n$), 如果存在最小值点的集合 $X^* = \{x^* | f(x^*) = \min\{f(x)\}\}$ 以及极小值点的集合 Y^* , 如果 $Y^* - X^* \neq \emptyset$, 那么必存在集合 Opt , 使得 $X^* \subseteq Opt, f(x^{opt}) < f(x), \forall x^{opt} \in Opt, x \in Y^* - X^*$, 令 Opt^* 为所有满足上式所有集合的并集 (最大集合); 如果 $Y^* - X^* = \emptyset, Opt^*$ 为 $f(x)$ 的整个定义域, 则 $V(Opt^*) > 0$, 其中 $V(*)$ 为体积测度。

证明: 如果 $Y^* - X^* = \emptyset$, 则结论成立。如果 $Y^* - X^* \neq \emptyset$, 令 $y^* = \min\{f(x) | x \in Y^* - X^*\}$, 令集合 $Q = \{x | \min\{f(x)\} \leq f(x) < y^*\}$, 则 $Q \subseteq Opt^*$ 。由条件可知 Opt^* 中点的函数值小于 y^* , 因此 Opt^* 中的点属于 Q , 即 $Opt^* \subseteq Q$ 。所以 $Opt^* = Q$ 。因为 $f(x)$ 为连续函数且 $\min\{f(x)\} < y^*$, 所以 $V(Opt^*) > 0$ 。

定理 3(爆炸搜索算法收敛性) 对于 n 维连续目标函数 $f(x)$ ($x_i \in [l_i, r_i], i=1, 2, \dots, n$), 如果在 Opt^* 内连续二次可微且满足式(1), 则

1) 对于凸函数, g_{best} 收敛于最优点。

2) 在爆炸搜索算法中, 如果 $g_{best} \in Opt^*$, 则 g_{best} 收敛于最优点。

3) 爆炸搜索算法以概率 1 收敛。

证明: 1) 定理 2 给出了 g_{best} 最速下降搜索算子在凸函数上的收敛性证明。算法中其他算子会加速 g_{best} 的下降速度, 因此对于凸函数算法是收敛的。

2) 如果 $g_{best} \in Opt^*$, 3.1 节中引理 2 仍然成立, 则 $\|\bar{g}_k\|^2 \rightarrow 0$, 即最终 g_{best} 收敛于极小值, 由于 Opt^* 中的极小值为最小值, 因此 g_{best} 收敛于最优点。

3) 变异算子使得炸点位置可以在搜索空间内任意移动, 由于 $V(Opt^*) > 0$, 因此任意炸点位置更新进入集合 Opt^* 的概率是大于 0 的。如果炸点进入 Opt^* 集合, 则 g_{best} 进入, 并以 0 概率更新出 Opt^* 。根据随机过程的知识可知, 最终 g_{best} 以概率 1 进入集合 Opt^* , 由 2) 可知 g_{best} 以概率 1 收敛。

注: 在大量的实际问题中, 迁移算子和爆炸搜索算子都能够加速 g_{best} 进入集合 Opt^* 。

4 仿真实验及分析

本节将对著名的 benchmark 函数进行仿真。限于篇幅, 选取部分数据罗列。下面给出这 4 个 benchmark 函数, 其中 f_3 的最优值为 -12569.5, 其他 3 个均为 0。

$$f_1(x) = \sum_{i=1}^{30} x_i^2, |x_i| \leq 100$$

$$f_2(x) = \sum_{i=1}^{30} [x_i \sin(\sqrt{|x_i|})], |x_i| \leq 500$$

$$f_3(x) = \sum_{i=1}^{30} [x_i^2 - 10 \cos(2\pi x_i) + 10], |x_i| \leq 5.12$$

$$f_4(x) = -20 \exp(-0.2 \sqrt{\sum_{i=1}^{30} x_i^2 / 30}) - \exp(\sum_{i=1}^{30} \cos(2\pi x_i) / 30) + 20 + e, |x_i| \leq 32$$

4.1 参数设定

在仿真中, 迭代代数 $T=500$, 炸点规模 $M=50$, 根据经验选定 $P_M=0.05, \gamma=0.1, \mu=0.2, \alpha=1.5, Num=10$,

$$R_{t_{max}} = 0.01(X_{max} - X_{min})$$

$$R_{t_{min}} = 10^{-4}(X_{max} - X_{min})$$

PSO 算法采用自适应惯性权策略 (CPSO), GA 交叉变异概率分别为 0.8、0.01, 采用高斯扰动进行变异操作^[6]。

4.2 仿真结果

表 1 给出了 ESA 的仿真结果, 同时与 CPSO 算法和 GA 进行了比较。通过仿真结果可以看出, ESA 取得了很好的结果, 在 500 次迭代内收敛到优秀的点。图 1—图 4 给出了这几个函数在仿真时最优值的变化曲线, 可以看出 ESA 算法的收敛速度也是比较快的。从仿真结果可以得到, ESA 具备搜索能力强、收敛快、精度高、稳定性高等优点。

表 1 仿真结果

函数		F1	F2	F3	F4
ESA	平均	6.3E-07	-11559.7	2.0E-10	9.6E-12
	标准差	7.0E-07	446.81	4.2E-10	2.4E-12
CPSO	平均	449.17	-5865.8	87.38	10.782
	标准差	293.30	716.59	17.79	1.4929
GA	平均	0.0239	-9249.8	9.13	0.8742
	标准差	0.0287	717.49	6.23	0.6086

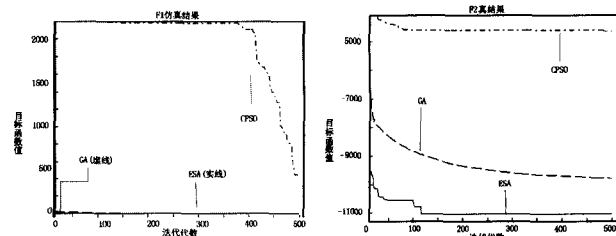


图 1 F1 仿真最优值的变化情况 图 2 F2 仿真最优值的变化情况

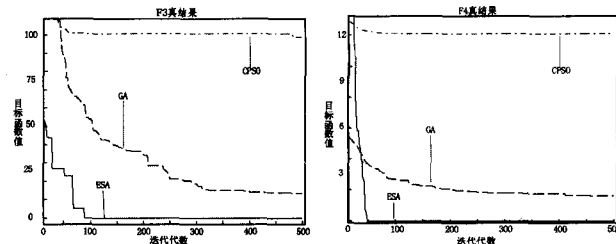


图 3 F3 仿真最优值的变化情况 图 4 F4 仿真最优值的变化情况

结束语 受烟花(炸弹)爆炸的启发, 本文提出了一种新型智能优化算法——爆炸搜索算法 (ESA)。ESA 将智能算法的思想和经典算法进行有机结合, 具有强大的搜索能力, 且具备经典算法的收敛精度; 同时给出了算法的收敛性证明。从仿真的结果可以看出, 新算法具有搜索能力强、收敛快、精度高、效果好等优点, 经典算法的引入弥补了传统智能算法精度的不足。在以后的工作里, 可以对 ESA 中算子做进一步的优化, 例如通过智能地引导爆炸来提高搜索效率, 增强自适应策略, 同时可以考虑用 ESA 解决一些典型的实际问题。

参考文献

- [1] 赵小平. 差商最速下降法及其收敛性[J]. 华东化工学院学报, 1992, 18(6): 807-811

(下转第 251 页)

表明,本文在使用增量聚类算法后比未使用增量聚类的检索在时间效率上得到很大的提高(见表3)。

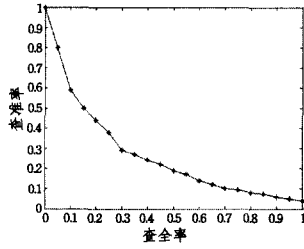


图5 查全率-查准率曲线

表3 本文算法检索时间

模型	顶点数	面片数	检索出模型数量	原始检索时间(s)	增量聚类检索时间(s)
Human	10,999	21,994	15	0.094	0.0014
Cup	15,198	30,396	10	0.109	0.0015
Glasses	7,016	14,028	13	0.089	0.0012
Airplane	5,400	10,796	15	0.086	0.0013
Ant	6,370	12,736	10	0.083	0.0011
Teddy	13,826	27,648	12	0.107	0.0016
Horse	14,680	29,356	12	0.111	0.0011

从检索时间可以看出,本文的增量聚类检索方法对于顶点数和面片数都高达 10^4 的大规模模型,也可以得到接近实时的检索速度,比之未使用增量聚类的检索时间得到很大的提高。

结束语 本文将 C2ICM 增量式聚类的思想应用到三维模型的检索工作上,针对模型面片数多、模型种类多样的大规模模型库,该方法能有效地提取模型的特征,并且高效地执行模型的特征匹配工作。另外,增量聚类的方法也可在 GPU 上实现,可以大大提高模型检索的速度,且具备良好的可扩展性,下一步将对本文算法的可并行性进行研究,以寻求具有更高检索速度的方法。

参考文献

[1] 郑伯川,彭维,张引,等. 3D模型检索技术综述[J]. 计算机辅助设计与图形学学报,2004,16(7):873-881

[2] Mihael A, Gabi K, Peter K H, et al. 3D shape histograms for similarity search and classification in spatial databases[C]//Proceedings of the 6th International Symposium on Large Spatial Databases(SSD' 99). Hong Kong, 1999:207-226

[3] Motofumi T S, Toshikasu K, Nobuyuki O. A similarity retrieval

of 3D polygonal models using rotation invariant shape descriptors [C]//Proceedings IEEE International Conference on Systems, Man and Cybernetics (SMC2000). Nashville, Tennessee, 2000:2946-2952

[4] Tangelder J W, Veltkamp R C. Polyhedral model retrieval using weighted point sets [J]. International Journal of Image and Graphics, 2003, 3(1):209-229

[5] Robert O, Thomas F, Bernard C, et al. Shape distributions [J]. ACM Transactions on Graphics, 2002, 21(4):807-832

[6] Saupé D, Vranic D V. 3D Model Retrieval with Spherical Harmonics and Moments[C]//Proceedings of DAGM-Symposium. 2001:392-397

[7] Kazhdan M M, Chazelle B, Dobkin D P, et al. A Reflective Symmetry Descriptor[C]//Proceedings of European Conference on Computer Vision (ECCV). Copenhagen, Denmark, 2002:642-656

[8] Vranic D V. 3D Model Retrieval[C]//Proceedings of the Spring Conference on Computer Graphics and its Applications (SCCG2000). Budmerice, Slovakia, 2000:89-93

[9] Ohbuchi R, Minamitani T, Takei T. Shape-similarity search of 3D models by using enhanced shape functions[J]. International Journal of Computer Applications in Technology (IJCAT), 2005, 23(2):70-85

[10] Paquet E, Rioux M. A query by content system for three-dimensional model and image databases management [J]. Image and Vision Computing, 1999, 17(2):157-166

[11] Novomi M, Klein R. A geometric approach to 3D objects comparison[C]//Proceedings of International Conference on Shape Modeling and Applications (SMI 2001). Genova, Italy, 2001:167-175

[12] Berker B, Cebe M, I-smet Z Y. Very-large scale incremental clustering [R]. Bilkent; Bilkent University, 2007

[13] 任化敏,张勇东,林守勋. GPU加速的基于增量式聚类的视频拷贝检测方法[J]. 计算机辅助设计与图形学学报, 2010, 22(3):449-456

[14] Chen D, Tian X, Shen Y, et al. On visual similarity based 3D model retrieval[J]. EUROGRAPHICS, 2003, 22(3):223-232

(上接第233页)

[2] Tsoulos I G. Modifications of real code genetic algorithm for global optimization[J]. Applied Mathematics and Computation, 2008, 203:598-607

[3] Aryanezhad M B, Hemati M. A new genetic algorithm for solving nonconvex nonlinear programming problems[J]. Applied Mathematics and Computation, 2008, 199:186-194

[4] 朱红求,阳春华,桂卫华,等. 一种带混沌变异的粒子群优化算法[J]. 计算机科学, 2010, 37(3):215-217

[5] Kennedy J, Eberhart R. Particle swarm optimization[C]//IEEE International Conference on Neural Networks (ICNN' 95). IEEE, 1995:1942-1947

[6] 王凌,刘波. 微粒群优化与调度算法[M]. 北京:清华大学出版社, 2001