具有启发式探测及自学习特征的降维对称微粒群算法

邵增珍 王洪国 刘 弘

(山东师范大学信息科学与工程学院 济南 250014)

摘 要 提出对称微粒群算法SymPSO_HD,用以提高PSO算法的搜索能力。引入种群分布熵以保证种群的分布性;引入具有探测特征的启发式粒子,用以影响普通粒子的位置;提出邻域内的克隆变异选择策略及全局范围内的降维对称粒子策略,用以增强粒子的局部及全局学习能力。仿真实验及分析结果表明,SymPSO_HD算法搜索能力稳定,适应性强,能以较大概率收敛到全局最优。

关键词 降维对称微粒群算法,种群分布熵,启发式探测,克隆变异

中图法分类号 TP18 **文献标识码** A

Dimensionality Reduction Symmetrical PSO Algorithm Characterized by Heuristic Detection and Self-learning

SHAO Zeng-zhen WANG Hong-guo LIU Hong

(Institute of Information Science and Engineering, Shandong Normal University, Jinan 250014, China)

Abstract A novel Symmetrical PSO algorithm (SymPSO-HD) was proposed to improve the search ability of PSO algorithm. To initialize the cluster effectively, population scatter entropy strategy was introduced. In order to improve the particle's position vector, a special kind of particle characterized by detection was proposed too. And to enhance the particle's learning ability both in local and global domain, we put forward the clone-mutation-selection strategy in neighborhood and the dimensionality reduction symmetrical strategy in global area. Simulation and analysis show that SymPSO-HD algorithm has stable search ability and strong adaptability, and can converge to the global optimum with large probability.

Keywords Dimensionality reduction symmetrical PSO, Population scatter entropy, Heuristic detection, Clone and mutation

标准PSO算法于1995年由Eberhart[1,2]等人提出,它是一种基于种群搜索的进化计算技术,可解决大量非线性、不可微及多维多峰问题。目前PSO及改进算法已广泛应用于函数优化[3]、工业控制[4]、神经网络参数训练、模式识别、多目标优化[5]等领域,并取得了较好的效果。PSO算法的突出优点是原理简单,实现方便,适应性强,但也存在着容易陷于局部最优等问题。因此,国内外学者从多方面对基本PSO进行改进以提高其性能。有的学者通过调整PSO公式中的参数来实现粒子自我学习及向外界学习的平衡[6],借以提高全局搜索能力;有的学者重点考虑粒子的分布性[7]对算法性能的影响,从粒子分布的拓扑结构上给予细致分析,并得出了一些结果;有的学者将PSO算法同其它算法混合,取长补短,形成混合粒子群算法[8];有的学者将多种群思想引入PSO算法[9],利用种群之间的协同以提高种群进化的多样性;有的学者采用反弹机制、变异机制等力图使得粒子跳出局部最优,进而提高全局搜索能力。

针对PSO算法容易陷于局部最优的不足,本文在文献[10]改进策略HPSO_TVAC的基础上,提出具有启发式探测及自学习特征的对称微粒群算法SymPSO_HD,并增加以下策略:将单维搜索空间分区,计算种群分布熵以保证种群的分布性;在种群进化过程中,引入具有随机探测特征的启发式静态粒子,用以影响普通粒子的位置向量;引入粒子的局部范围内的邻域克隆变异选择策略,以增强粒子局部学习能力;引入粒子的降维对称粒子策略,以增强粒子的全局学习能力。仿真实验与分析结果表明,该算法能显著提高搜索效率,对搜索空间维数不敏感,显示出较强的实用性。

1 基于分布熵的种群初始化调整策略

种群在初始化过程中,需要保证个体分布的不确定性(即随机性)。显然,除某些特殊情况外,应避免所有个体集中在较小的局部区域。假设将搜索空间分为D个子空间,每个子空间含有n_i个个体,$i=1,2,\cdots,D$,指定N为个体总数,则

$$\overline{S}=-\sum_{i=1}^{D}\frac{n_i}{N}\log\frac{n_i}{N}$$

称为种群在搜索空间中的分布熵。$\overline{S}$越大,则种群越分散,否则就越集中。问题的关键在于子空间的划分方式。文献[11]以二维搜索空间为例,提出将搜索空间Ω扩展为一个圆形区

到稿日期:2009-06-24 返修日期:2009-08-30 本文受国家自然科学基金(60970004),山东省科技攻关项目(2009GG10001008),济南市高校院所自主创新项目(200906001)资助。

邵增珍(1976-),男,博士生,讲师,CCF会员,主要研究方向为计算智能、人工社会等,E-mail:Shaozengzhen@163.com;王洪国(1966-),男,教授,博士生导师,主要研究方向为电子政务等;刘 弘(1955-),女,教授,博士生导师,CCF高级会员,主要研究方向为人工智能、创新设计等。

域 Ω'(称为 Ω 的陪域)的方法。该方法首先计算陪域 Ω' 的中心点 $\overline{X}=\frac{\sum_{i=1}^{N}x_i}{N}$,然后以该中心点为圆心,画若干半径逐渐倍增的同心圆,并将相邻同心圆之间的区域(环)定义为一个子空间。该方法操作简单,具有一定作用,但分析后发现,这种划分子空间的方法仅适合某些特定场合,当各个维的长度相差较大时将失去作用。

本文提出的对搜索空间进行有效划分的"n 维等分空间法",实际上就是对搜索空间的每一维都进行划分,种群的分布熵取各维空间分布熵的加权和。设讨论均在由 $[d_l^i,d_u^i]$,$i=1,2,\cdots,n$ 确定的 n 维欧氏空间 Ω 中进行,K 为每一维被分割的段数。

定义 1 设每个子段个体分布数目为 n_{ij},$j=1,2,\cdots,K$,则称

$$S_i=-\sum_{j=1}^{K}\frac{n_{ij}}{N}\log\frac{n_{ij}}{N} \tag{1}$$

为种群在第 i 维的种群分布熵。

定义 2 假设 S_i 是欧氏空间 Ω 第 i 维的种群分布熵,则

$$\overline{S}=(\alpha_1 S_1+\alpha_2 S_2+\cdots+\alpha_n S_n)=\sum_{i=1}^{n}\alpha_i S_i \tag{2}$$

称为欧氏空间 Ω 的种群分布熵。

式中,$\alpha_i\in[0,1]$,$\alpha_i\in R^*$,且 $\sum\alpha_i=1$。特别地,可取 $\alpha_i=\alpha_j=\frac{1}{n}$,$i,j=1,2,\cdots,n$,则式(2)简化为

$$\overline{S}=\frac{1}{n}\sum_{i=1}^{n}S_i \tag{3}$$

假设已基于随机方法生成初始种群,记 $S_0<1$ 为熵阈值。如果种群分布熵 $S<S_0$,说明种群的随机性不够,需要对其进行一定的扩散操作,以确保 $S\geqslant S_0$。我们采取的方法是利用扩散函数 $disperse(\min\{S_i\})$ 实现该功能,其作用是分散最小分布熵子空间 $\min\{S_i\}$ 中的个体,并保证其分布熵取得某个较大的值。具体实现时,可采用混沌方程 $pr(t+1)=\lambda\cdot pr(t)\cdot(1-pr(t))$ 生成随机扩散步长。其中 $pr\in(0,1)\wedge pr\notin\{0.25,0.5,0.75\}$,$\lambda=4$。$\lambda=4$ 时,生成的序列具有很好的混沌效果,且 pr 的所有取值均可保证在(0,1)之间。之所以 pr 的初始值不能取 0.25,0.5 和 0.75,是因为这些取值将导致 pr 的后期序列全部变为 0,无法实现混沌效果。图 1 是 pr 初值为 0.3 时的取值变化情况。

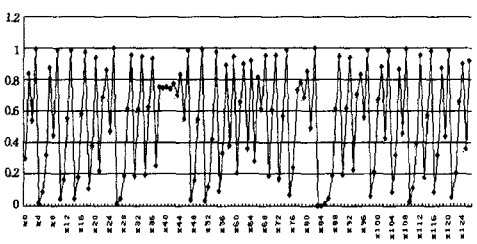

图 1 $pr=0.3$ 时的取值变化情况

2 启发式吸引粒子与粒子学习策略

2.1 启发式吸引粒子

对粒子 r,文献[10]中有关粒子速度的计算公式为:

$$\vec{v}_r(t+1)=\omega\times\vec{v}_r(t)+c_1\times rand1()\times(\vec{p}_r-\vec{x}_r)+c_2\times rand2()\times(\vec{p}_g-\vec{x}_r) \tag{4}$$

式中,$\omega=0.5+rand3()/2$,$c_1=(c_{1f}-c_{1r})t/MAXT+c_{1r}$,$c_2=(c_{2f}-c_{2r})t/MAXT+c_{2r}$;位置计算公式同文献[1]。

上式中的 $rand1()$,$rand2()$ 和 $rand3()$ 都是独立的均匀分布随机函数,其范围均为[0,1]。ω 的平均取值为 0.729,而 c_1 的取值从 2.5 降至 0.5,c_2 的取值从 0.5 增至 2.5[10]。

为改进粒子的位置,在位置计算公式中增加了一个趋向向量。第 t 次迭代时,系统随机选择 n 维空间的第 i 维,并按均匀分布原则在每个子段内随机生成一个静态粒子(该粒子不属于种群,且速度为 0)。计算该子段内所有静态粒子的平均适应度作为该子段的适应度,记为 $f_{ik(t)}$,$k=1,2,\cdots,K$。假设 $f_{im(t)}=\max\{f_{ik(t)}\mid k=1,2,\cdots,K\}$ 为具有最高适应度的子段(称为吸引子空间),从统计意义上看,该子段所定义的子空间中包含全局最优点的概率也最大,故有必要增强所有粒子飞向该区域的概率。称吸引子空间中适应值最高的静态粒子为吸引粒子,其位置向量记为 $\vec{x}_a$。对粒子 r,构建其指向吸引粒子的标准方向向量 $\vec{D}_r(t+1)=std(\vec{x}_a-\vec{x}_r(t))$,$std(\ )$ 的作用是将向量单位化,使得 $|\vec{D}_r(t+1)|=1$。称向量 $\vec{D}_r(t+1)$ 为粒子 r 在第 $t+1$ 次迭代时的趋向向量。基于此,对粒子 r 的位置变化公式修改为

$$\vec{x}_r(t+1)=\vec{x}_r(t)+\vec{v}_r(t+1)+\gamma|\vec{v}_r(t+1)|\cdot\vec{D}_r(t+1) \tag{5}$$

式(5)增加了 $\gamma|\vec{v}_r(t+1)|\cdot\vec{D}_r(t+1)$ 作为启发项。γ 称为趋向向量影响因子,同迭代次数 t 成正相关性,用于调整启发项的影响能力,一般可设 $\gamma\in[1,3]$。启发项具有较强的导向性,是增大粒子选择正确飞行方向概率的重要因素。实验结果表明,趋向向量在很小计算代价的基础上提高了粒子搜索的目的性,能有效加快收敛速度。吸引粒子的启发作用如图 2 所示。

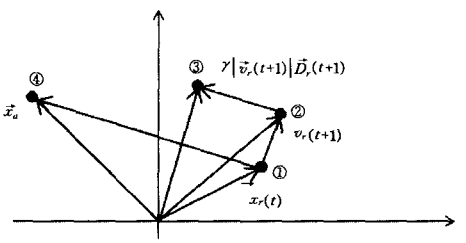

图 2 吸引粒子作用示意图

图 2 中,假设圆点①为粒子 r 在第 t 次迭代时所处位置,圆点④为吸引粒子位置。标准 PSO 算法中,经过第 $t+1$ 次迭代时速度向量的修正,粒子 r 的位置将偏移到圆点②。在吸引粒子的吸引作用下,系统产生趋向向量 $\vec{D}_r(t+1)$,该向量的正相关向量 $\gamma|\vec{V}_r(t+1)|\cdot\vec{D}_r(t+1)$ 对位置②又增加了一个飞向位置④的拉动作用,使得粒子 r 在第 $t+1$ 次迭代时位置偏移到圆点③位置。一般来说,位置③比位置②更接近吸引粒子。影响因子 γ 越大,粒子飞向吸引粒子的趋势越明显,因此系统运行后期增大影响因子是有必要的。

2.2 短程学习——克隆变异选择策略

粒子 r 在进化过程中,不仅要遵从个体惯性,还要根据个体认知 $\vec{p}_r$ 和群体认知 $\vec{p}_g$ 不断调整其飞行方向。如果 r 本身拥有系统当前最佳位置,除非有更为强势的粒子出现,其飞行方向将保持不变[12];如果 r 仅是当前最优而不是全局最优粒子,它将对其他粒子产生误导而可能使整个群体陷入局部最优;如果 r 不是当前最优粒子,则除了向自身最佳位置和全局最佳位置指引的方向飞行之外,本身没有更强的学习能力。

基于此，引入了粒子的短程学习机制——克隆变异选择策略，用于更新 $\vec{p}_r$。

以二维空间为例说明 $\vec{p}_r$ 的更新策略。为方便起见，以粒子 r 的当前位置为中心，以两个维长度的 $1/K$ 为边长构造一个矩形空间(或椭圆空间)，称该空间为 r 的邻域 σ_r。克隆 r 的 m 个陪体，并按照柯西规则在 σ_r 内对陪体进行微变异，然后计算它们的适应值并选择最优陪体代替粒子 r。通过克隆、变异和选择过程，粒子 r 可对邻域范围进行有限探测，学习其中的优秀位置，可实现粒子在局部区域的优化，提高算法的收敛能力。另外，粒子 r 在向其邻域内陪体学习的同时，还可向当前邻域内的其他粒子以相同的方式学习，这可进一步提高粒子的性能。

2.3 长程学习——降维对称粒子策略

粒子的短程学习涉及范围较小，有时难以跳出局部最优。为进一步提高搜索范围，并尽量减少由此带来的计算量，提出了降维对称粒子的长程学习策略。对 n 维空间来说，如果保持其中的 $n-2$ 维的坐标不变，则 n 维空间的可投影为二维空间。在投影时，选择哪两维是随机的，而且随着迭代次数不断变化，这可保证粒子学习的覆盖性。对二维空间来说，每个粒子基于搜索空间的中心点、搜索空间的两条对角线、两条中心轴，都各有一个对称点，称这些对称点为标准对称点，也就是基本粒子的对称粒子。在进化过程中，每个粒子在完成短程学习后，还要生成几个对称粒子，实现长程学习，如图 3 所示。

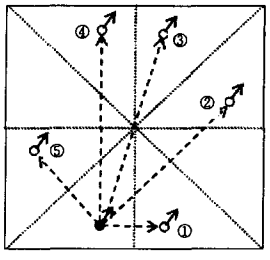

图 3 基本粒子的对称粒子

图 3 中，黑色实心圆点代表基本粒子，空心圆点代表对称粒子，其编号从①到⑤，分别是基本粒子同垂直中心轴、主对角线、中心点、水平中心轴以及次对角线对称产生的。定义基本粒子的位置坐标是 $B(x_0,y_0)$，影粒子的坐标用 sB_1-sB_5 表示。假设搜索空间是以 (a,b)，$(a+w,b+h)$ 为左下角和右上角顶点的矩形区域，则有

$$sB_1(x_1,y_1):x_1=2a+w-x_0,\ y_1=y_0$$

$$sB_2(x_2,y_2):x_2=x_0+hT_2,\ y_2=y_0+wT_2$$

式中，$T_2=\dfrac{-2(hx_0+wy_0-bw-hw-ah)}{w^2+h^2}$；

$$sB_3(x_3,y_3):x_3=2a+w-x_0,\ y_3=2b+h-y_0;$$

$$sB_4(x_4,y_4):x_4=x_0,\ y_4=2b+h-y_0;$$

$$sB_5(x_5,y_5):x_5=x_0-hT_5,\ y_5=y_0+wT_5,$$

式中，$T_5=\dfrac{-2(-hx_0+wy_0-bw+ah)}{w^2+h^2}$ (6)

计算这些对称粒子的适应度，最优粒子将替代基本粒子进入基本种群，其速度大小及方向可在原来的基础上增加一个扰动。

分析发现，对称粒子的计算量很小，但基本粒子的学习空间跨度却非常大，这对群体离开局部最优极为有利，能大大提高系统的寻优能力。另外，该策略对搜索空间的维数不敏感，不会出现维数灾问题。需要说明的是，构造对称粒子过程中，需要对邻近搜索空间中心点位置的粒子进行特别处理。因为邻近中心位置的粒子的对称粒子位置也邻近中心位置，这使得粒子的长程学习退化为短程学习，不利于全局收敛性的提高。解决的办法是根据搜索空间的跨度特点，以中心点为中心，定义一个泛中心搜索区域，凡是落在此区域的基本粒子生成对称粒子时仍然落在泛中心搜索区域，则根据由基本粒子位置和对称粒子位置确定的学习方向增加一个额外的同向扰动，以达到扩散对称粒子的效果。

3 算法复杂性分析

假设群落中粒子总数为 PopSize，搜索空间为 $\Omega\subset R^n$，其每一维被分割成的子段数目为 K。下面对算法的复杂性进行讨论。

3.1 计算复杂性讨论

SymPSO-HD 算法在 HPSO_TVAC 算法的基础上，增加了种群分布熵的判断、种群扩散操作、吸引粒子的构建、短程学习及长程学习等操作。种群分布熵的计算复杂度为 $O((K\times n\times PopSize)+O(K\times n)=O((K\times n\times(PopSize+1))\approx O(K\times n\times PopSize)$；如果种群需要扩散操作，则其计算复杂度为 $O(K\times PopSize)$；吸引粒子的计算复杂度较为复杂，达到了 $O(t\times K\times n\times PopSize)$，其中 t 为迭代次数，随着 t 的增加，其计算量逐渐增加；短程学习的计算复杂度为 $O(n\times PopSize\times m)$；长程学习的计算机复杂度为 $O(5\times PopSize)$。分析表明，SymPSO-HD 算法为多项式算法，且适用于高维问题。

3.2 空间复杂性讨论

由于要计算吸引粒子，故需记忆所有粒子信息。第 t 次迭代时，系统需分类保存约 $PopSize\times t\times K\times n$ 条记录。另外，短程学习及长程学习过程中，记忆粒子的邻域陪体也需要约 $PopSize\times m$ 个粒子的空间。这些都是为提高系统寻优能力而付出的代价。

4 实验验证与结果分析

对 HPSO-TVAC 算法[10]和本文 SymPSO-HD 算法进行实验对比分析。为分析 SymPSO-HD 中各种策略的作用，将仅具有短程学习能力的算法命名为 SymPSO-HD* 算法，并也将其参与比较。实验硬件环境均为 Intel PIV2.8 双核，内存 2G，软件环境 VS2005 及 Matlab7.0。

4.1 Benchmark 函数选择

选择实值函数对算法的性能进行测试。表 1 列出了用于实验的 5 个测试函数。同文献[10]相同，以上函数在运行时 V_{max} 取值分别是 100，100，10，600，100。除函数 f_6 外，其余函数 f_1-f_4 均为多维函数，具体实验时 n 取 20，30。所有函数均要求取其全局最小值作为最优值，已知所有函数的最优值均为 0。我们设定当函数值小于等于 0.01 时，就认为函数已经收敛到最优点。

表 1 Benchmark 函数

函数名称	函数形式	搜索区域
Sphere function	$f_1(x)=\sum_{i=1}^{n}x_i^2$	$(-100,100)^n$

Rosenbrock function	$f_2(x)=\sum_{i=1}^{n}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	$(-100,100)^n$
Rastrigrin function	$f_3(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	$(-10,10)^n$
Griewank function	$f_4(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	$(-600,600)^n$
Schaffer's f6	$f_6(x)=0.5-\frac{\sin(\sqrt{x^2+y^2})^2-0.5}{(1.0+0.001(x^2+y^2))^2}$	$(-100,100)^2$

4.2 算法性能对比实验

设种群个体数目 $PopSize=100$，最大迭代次数 GMAX 根据函数难度适当调整，精度控制在 10^{-4}，每个算法执行 50 次。执行函数 f_4 的寻优过程时，$K=10$，其余函数均设置 $K=5$。试验数据见表 2。表中黑体部分代表最优结果。从表 2 左侧可以看出，算法 SymPSO-HD 在平均函数值方面的性能明显优于 HPSO-TVAC 的性能，但是稳定性方面相差不大。算法 SymPSO-HD* 由于没有增加对称粒子策略，其性能相对于 SymPSO-HD 来说稍低。从各算法收敛到最优解的次数及 50 次实验中收敛到最优解的平均代数比较分析（表 2 右侧）可知，SymPSO-HD 算法能在较少的迭代次数就收敛到最优解，相比 HPSO-TVAC 算法来说具有一定的优势。HPSO-TVAC 算法在函数 f_1，f_3 和 f_4 上表现较为优秀，而 SymPSO-HD 算法求解 f_4 时效果更好一点，这可用函数 f_4 的搜索区域较大，而长程学习所起的作用也较大来解释。

表 2 算法性能对比

函数	维数	GMAX	平均函数值 /（标准差）			收敛到最优值的次数 /（平均迭代次数）		
			HPSO-TVAC	SymPSO-HD*	SymPSO-HD	HPSO-TVAC	SymPSO-HD*	SymPSO-HD
f_1	20	2000	0.0129/(0.107)	**0.0104/(0.0614)**	0.0104/(0.0301)	48/(549)	49/(421)	**49/(401)**
	30	3000	0.0131/(0.1948)	**0.0124/(0.0107)**	0.0129/(0.0519)	46/(860)	49/(899)	**49/(801)**
f_2	20	4000	14.9172/(10.6821)	17.8817/(12.9547)	**13.0007/(9.1114)**	21/(2836)	24/(2203)	**25/(2190)**
	30	5000	14.2214/(10.0119)	19.5547/(15.9510)	**13.4237/(8.1209)**	16/(4550)	21/(3241)	**29/(3007)**
f_3	20	4000	0.0417/(2.4812)	0.0351/(1.7145)	**0.0327/(2.1413)**	**39/(2527)**	39/(2550)	42/(2421)
	30	5000	**0.0614/(0.0907)**	0.0771/(1.9245)	0.1027/(3.1410)	**45/(3021)**	40/(3218)	41/(3114)
f_4	20	4000	0.0171/(0.0304)	0.0250/(1.0148)	**0.0129/(0.0091)**	27/(1618)	37/(1600)	**44/(1409)**
	30	5000	0.0232/(0.0493)	0.0346/(2.1024)	**0.0174/(0.0645)**	30/(2334)	36/(2280)	**46/(2204)**
f_6	2	1000	**0.0132/(0.0071)**	0.0139/(0.6484)	0.0133/(0.0124)	27/(705)	45/(493)	**48/(407)**

4.3 参数 K 对算法性能的影响

K 的不同取值，对算法的性能具有不同的影响。本文通过实验测试了 K 对算法性能的影响。按照 K 取 5，8，10，20 以及 5～20 之间随机取值共 5 种取值策略进行实验。对每个函数来说，K 的每次取值算法都执行 50 次。对所有实验结果进行统计后发现，K 的取值变化对函数 f_1，f_3 和 f_6 的影响较小，对函数 f_2，f_4 的影响较大。对函数 f_6 来说，虽然其形式较为复杂，但因其维数较低（$n=2$），所以参数 K 的影响有限。分析发现，f_2 和 f_4 属于多维函数，而且搜索区域都相对很大，这是 K 取值较大导致算法性能也较好的原因之一。当 K 在 5～20 之间取值时，各函数的效果较好，这可作为一条基本规律。

结束语 SymPSO-HD 算法针对 PSO 算法的缺陷，借鉴 HPSO-TVAC 算法，提出了一系列改进策略。对比实验表明，这些改进策略对提高算法的全局寻优能力及算法的稳定性具有优良性能。本算法的缺点是存储空间有所增加，迭代速度也有待进一步提高。下一步将针对吸引粒子的有效工作时间、基本粒子的速度变化规律方面展开研究。另外，多种群协同的粒子群落算法也是后期研究的重点。

参 考 文 献

[1] Kennedy J, Eberhart R C. Particle Swarm optimization [C]// Proceedings of IEEE International Conference on Neural Networks. Perth: IEEE Inc., 1995: 1942-1948

[2] Parsopoulos K E, Vrahatis M N. Recent approaches to global optimization problems through Particle Swarm Optimization [J]. Natural Computing, 2002(1): 235-306

[3] Lhotská L, Macaš M, Burša M. PSO and ACO in Optimization Problems[C]// E. Corchado, et al., eds. IDEAL 2006, LNCS 4224. 2006: 1390-1398

[4] Yang Qingyun, Sun Jigui, Zhang Juyang, et al. A Hybrid Discrete Particle Swarm Algorithm for Open-Shop Problems[C]// T. -D. Wang, et al., eds. SEAL 2006, LNCS 4247. 2006: 158-165

[5] Chang C S, Kwan C M. Evaluation of Evolutionary Algorithms for Multi-objective Train Schedule Optimization [C] // G. I. Webb, Xinghuo Yu, eds. AI 2004, LNAI 3339. 2004: 803-815

[6] Clerc M, Kennedy J. The particle swarm-explosion, stability and convergence in a multi-dimensional complex space [J]. IEEE Trans. on Evolutionary Computation, 2002, 6(1): 58-73

[7] Parsopoulos K E, Vrahatis M N. UPSO: A unified particle swarm optimization scheme[C]// Proc. of the ICCMSE 2004. Attica: VSP Int'l Science Publishers, 2004: 868-873

[8] Zheng X, Liu H. A hybrid vertical mutation and self-adaptation based MOPSO[J]. Computers and Mathematics with Applications, 2008. doi: 10.1016/j.camwa.2008.09.023

[9] 马慧民，叶春明. 基于文化进化的并行粒子群算法[J]. 计算机工程，2008，34(2)：193-195

[10] Ratnaweera A, Halgamuge S K, Watson H C. Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 240-255

[11] 郭京蕾. 基于等级熵的自适应粒子群优化算法[J]. 华中科技大学学报，2009，37(4)：65-68

[12] 吕艳萍，李绍滋，等. 自适应扩散混合变异机制微粒群算法[J]. 软件学报，2007，18(11)：2740-2751

[13] 孙亮，代存杰，张克云. 新型混合粒子群优化算法[J]. 重庆工学院学报：自然科学版，2008，22(2)：146-149