复杂约束条件下的混合粒子群优化算法 *

丁　雷

(吉首大学 物理科学与信息工程学院, 湖南 吉首 416000)

摘　要：针对具有复杂约束条件的优化问题,提出了一种混合粒子群算法。该混合算法在将标准粒子群算法与线性搜索法有机结合的基础上,依次对粒子的每一维变量进行适当变化并同时判断其变化的效果。最后进行了数值实验,其结果表明,所提出的混合粒子群算法对于具有复杂有约束条件的优化问题有较好的优化效果。

关键词：复杂约束条件；混合粒子群算法；线性搜索；变量；综合信息

中图分类号：TP18　　**文献标志码**：A　　**文章编号**：1001-3695(2010)09-3256-03

doi:10.3969/j.issn.1001-3695.2010.09.014

Hybrid particle swarm optimization algorithm for optimization problem with complicated constraint condition

DING Lei

(*School of Physics Science & Information Engineering, Jishou University, Jishou Hunan* 416000, *China*)

Abstract: Focused on the study of the optimization problem with complicated constraint condition, this paper proposed the hybrid particle swarm optimization (PSO) algorithm. Firstly, flexibly combined the standard PSO algorithm with the line search method. Then, adaptively and respectively modified each variable of the particle and gave the searching result simultaneously. Finally, gave the digit experiment. The result shows the proposed hybrid PSO algorithm can obtain a satisfied result for the optimization problem with complicated constraint condition.

Key words: complicated constraint condition; hybrid particle swarm optimization; line search; variables; synthetical information

0　引言

目前进化算法已经被广泛应用于求解复杂的有约束条件的优化问题[1~3]。惩罚函数法是进化算法处理约束问题的常用方法,但其主要缺点是惩罚参数的选取比较困难,算法在很大程度上依赖于参数的选取。因此,为了避免因初始信息的设置不当带来的影响,将约束优化问题转换为多目标优化问题得到了极大的重视。

粒子群算法[4,5]是一种仿生算法和随机搜索算法,其根据个体的适配信息进行搜索,因此不受函数约束条件的限制,具有全局优化能力。目前粒子群算法已经成为求解约束优化问题的研究热点[6~8],但粒子群算法最大的缺点是容易出现早熟现象。粒子群算法发生早熟时有两种情况出现,一种是收敛到局部极值,另一种是过早熟,即收敛在某一位置(不是局部极值)上。对于单目标无约束条件的优化问题,其常见改善粒子群算法优化效果的方法可以归结为:通过改变粒子的速度、位置或惯性权重来刺激粒子的活性。以上方法对于复杂有约束条件的优化问题其效果并不明显,这是因为在进化过程中如果不满足约束条件,其变异后所带来的信息就起不到应有的作用。

文献[9]将线性搜索法与标准粒子群算法有机结合起来进行复杂问题的优化取得了一定的成功。但在对实际问题的优化实验发现,由于标准粒子群算法对进化后粒子的优劣判断是基于粒子所有变量来进行的综合判断[10],这种进化方式容易出现一些好的变量信息被丢失,从而使得搜索效果受到一定的影响。因此,为进一步提高搜索效果,在搜索过程中有必要对这种进化方式作适当的修正,以尽量避免一些优秀的变量被丢失。

1　有约束条件的优化问题

为了避免因初始信息的设置不当带来的影响,将约束优化问题转换为多目标优化问题,并给出了相应的并行优化方法。

1.1　问题的定义

有约束条件的优化问题可以描述如下:

$$\begin{aligned} & \min f(\boldsymbol{x}) \\ \text{s.t.}\quad & g_i(\boldsymbol{x}) \leqslant 0 \qquad i=1,\cdots,l \\ & h_i(\boldsymbol{x}) = 0 \qquad i=l+1,\cdots,m \\ & \boldsymbol{x}=(x_1,x_2,\cdots,x_n)\in \mathrm{R}^n \end{aligned} \tag{1}$$

其中:$\boldsymbol{x}\in\Omega\subseteq S$ 为决策向量;Ω 为可行域;S 为决策空间;$f(\boldsymbol{x})$,$g_i(\boldsymbol{x})$,$h_i(\boldsymbol{x})$($i=1,\cdots,m$)均为 R^n上的 n 元函数;$f(\boldsymbol{x})$ 为原目标函数;$g_i(\boldsymbol{x})$ 和 $h_i(\boldsymbol{x})$ 为约束条件。将式(1)变换为两个目标函数,有

$$\begin{aligned} & J=\min f(f_0(\boldsymbol{x}),f_1(\boldsymbol{x})) \\ & \boldsymbol{x}=(x_1,x_2,\cdots,x_n)\in \mathrm{R}^n \end{aligned} \tag{2}$$

其中:$f_0(\boldsymbol{x})$ 为原目标函数,$f_1(\boldsymbol{x})$ 为约束违反程度函数。$f_1(\boldsymbol{x})$

收稿日期：2010-01-31；**修回日期**：2010-03-19　**基金项目**：国家自然科学基金资助项目(60702076)；湖南省自然科学基金资助项目(07JJ6109)

作者简介：丁雷(1972-),男,湖南临湘人,副教授,博士,主要研究方向为计算机网络、人工智能(dinglei_39@yahoo.com.cn).

的形式如下:

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{l}\max(0, g_i(\boldsymbol{x})) + \sum_{i=l+1}^{m}\text{abs}(h_i(\boldsymbol{x})) \tag{3}$$

1.2　并行优化方法

Deb[11]提出了基于锦标赛选择算子的并行优化方法。为了发挥不可行解的作用,Takahama 等人[12]在 Deb 算法的基础上提出了一种 a 约束水平比较原则。本文在以上研究的基础上采用如下的比较原则,给定成对的个体 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$,令 u_1 和 u_2 分别表示个体 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 的约束违反程度,则有:a)如果 $u_1 > u_2$ 且 $u_1 > u_0$,则个体 $\boldsymbol{x}_2$ 优先;b)如果 $u_1 = u_2$,则具有更小目标函数的个体优先;c)如果 $u_1 \leqslant u_0$ 且 $u_2 \leqslant u_0$,则具有更小目标函数的个体优先。

为了方便以后的描述,给出如下几个定义:

定义 1　假设给定的目标函数为 $f_i(\boldsymbol{x}), i=0,\cdots,k$,如果 $\forall i \in \{1,\cdots,k\}$ 满足 $f_i(\boldsymbol{x}_1) \leqslant f_i(\boldsymbol{x}_2)$,且至少存在一个 $f_j(\boldsymbol{x}_1) < f_j(\boldsymbol{x}_2), j \in \{1,\cdots,k\}$,则记为 $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$,即 $\boldsymbol{x}_1$ Pareto 优超 $\boldsymbol{x}_2$。

定义 2　Pareto 最优解。对于可行解 $\boldsymbol{x}^* \in S$,当且仅当不存在另一个可行解 $\boldsymbol{x} \in S$,使所有不等式 $f_j(\boldsymbol{x}) \leqslant f_j(\boldsymbol{x}^*)$ 成立,其中 $j=1,\cdots,q$,且至少存在一个 $j \in \{1,\cdots,q\}$,使严格不等式成立 $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$,则称 $\boldsymbol{x}^*$ 为一个 Pareto 最优解。

2　混合粒子群算法

当算法的收敛速度变慢或停滞时,根据两个不同的全局最优解获得线性搜索的方向。如果线性搜索不到一个最优的解,则依次对粒子的每一维变量进行适当变化(此时其他变量保持不变),并同时给出相应的搜索结果。

2.1　标准粒子群算法

在标准粒子群算法中,PSO 算法在每一次的迭代中,粒子的每一维变量如第 d 维变量通过跟踪两个极值($p_{i,d}^{\text{best}}, g_d^{\text{best}}$)来更新自己,其中 $p_{i,d}^{\text{best}}$ 表示第 i 个个体极值的第 d 维变量,g_d^{best} 表示当代全局极值的第 d 维变量。粒子变量的速度和位置的更新方式为

$$\begin{cases} v_{i,d}^{(k+1)} = w^{(k+1)} \times v_{i,d}^{(k)} + c_1 \times r_1 \times (p_{i,d}^{\text{best}} - s_{i,d}^{(k)}) + \\ c_2 \times r_2 \times (g_d^{\text{best}} - s_{i,d}^{(k)}) \\ s_{i,d}^{(k+1)} = s_{i,d}^{(k)} + v_{i,d}^{(k+1)} \end{cases} \tag{4}$$

其中:$v_{i,d}^{(k)}$ 是第 i 个粒子第 d 维变量在第 k 次迭代中的速度;r_1、r_2 是介于(0,1)的随机数;$s_{i,d}^{(k)}$ 是第 i 个粒子第 d 维变量在第 k 次迭代中的位置;c_1 和 c_2 是学习因子;$w^{(k)}$ 是惯性权重,且 $w^{(k+1)} = w_0 - w_1 \times k/l$($w_0$、$w_1$ 为设定的初始权重值,l 为设定的总迭代次数,k 为迭代次数)。

2.2　混合粒子群算法的实现

文献[9]在文献[13]研究的基础上,在粒子群搜索过程中,将其经历过的全局最优解记录下来,如

$$\boldsymbol{g}^{\text{best}} = (\boldsymbol{g}_1^{\text{best}}, \boldsymbol{g}_2^{\text{best}}, \cdots, \boldsymbol{g}_p^{\text{best}}) \tag{5}$$

其中:$\boldsymbol{g}_i^{\text{best}}(i=1,\cdots,p)$ 表示粒子群算法在其进化过程中所经历过的全局最优解,p 表示总共获得了 p 个不同的全局最优解。可从当前全局最优解和前一个全局最优解获得线性搜索的方向为

$$\boldsymbol{d} = (\boldsymbol{g}_p^{\text{best}} - \boldsymbol{g}_{p-1}^{\text{best}}) / |\boldsymbol{g}_p^{\text{best}} - \boldsymbol{g}_{p-1}^{\text{best}}| \tag{6}$$

粒子根据式(7)寻找一个同时 Pareto 优超 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 的个体 $\boldsymbol{x}_3$,其中 n 表示搜索的次数,m 表示搜索的步长,有

$$\boldsymbol{x}_3 = \boldsymbol{x}_1 + n \times m \times \boldsymbol{d} \tag{7}$$

从式(6)和(7)可以看出,线性搜索过程是将个体的每个变量同时进行相应的变化。

考虑到标准粒子群算法是基于粒子所有变量的综合信息来判断其优劣,因此依次对粒子的每一维变量单独进行适当的变化(此时其他的变量保持不变)来搜索,有

$$\boldsymbol{x}_{3,j} = \boldsymbol{x}_{1,j} + n' \times m' \times d_0 \tag{8}$$

其中:$\boldsymbol{x}_{3,j}$ 表示个体 $\boldsymbol{x}_3$ 的第 j 个变量;$\boldsymbol{x}_{1,j}$ 表示个体 $\boldsymbol{x}_1$ 的第 j 个变量;n' 表示搜索的次数;m' 表示搜索的步长;d_0 为给定的一个初始值。当搜索到 $\boldsymbol{x}_3 \prec \boldsymbol{x}_1$ 时,则停止下来。

整个混合粒子群多目标优化算法步骤如下:

a)首先将有约束条件的优化问题转换为有两个目标的优化函数,并初始化学习因子、惯性权重、每个粒子的初始位置和初始速度、最大迭代次数,以及 u_0 的初始值(在以后的迭代过程中,每迭代到一定的次数,u_0 作相应的变换)。

b)利用约束水平比较法来计算当代全局最优解、个体历史最优解。如果当代全局最优解和上次迭代过程中的全局最优解不一样,则将上一个全局最优解记录下来。

c)根据个体历史最优解和当代全局历史最优解调整粒子速度和位置。在规定的次数范围内判断粒子群算法是否停滞,如果没有则转到 b),否则转 d)。

d)根据当代全局最优解 $\boldsymbol{g}_p^{\text{best}}$ 和所记录的前一个全局最优解 $\boldsymbol{g}_{p-1}^{\text{best}}$ 获得线性搜索方向,按照式(7)进行线性搜索,如果搜索到一个更优的解则将它作为粒子群算法的当代全局最优解,并将 $\boldsymbol{g}_p^{\text{best}}$ 记录下来。为防止可能出现搜索不到可行解的情况,设置一个最大的搜索次数。如果找不到一个更优的解,则按照式(8)依次分别对粒子的每个变量进行线性搜索,同样为防止可能出现搜索不到可行解的情况,设置一个最大的搜索次数。

e)判断迭代次数达到最大次数没有,如果达到则结束;否则转 b)。

3　数值实验

选取三个经典的最小化问题的测试函数来检验新算法,并将获得的结果与目前已知的一些文献利用其分算法所获得的最优结果进行对比。

选取的第一个函数如式(9)所示:

$$\begin{aligned} &f_1(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + \\ &\quad 37.293239x_1 - 40792.141 \\ &\text{s.t.}\quad g_1 = -85.334407 - 0.0056858x_2x_5 - \\ &\quad 0.0006262x_1x_4 + 0.0022053x_3x_5 \leqslant 0 \\ &g_2 = 85.334407 + 0.0056858x_2x_5 + \\ &\quad 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leqslant 0 \\ &g_3 = -80.51249 - 0.0071317x_2x_5 - \\ &\quad 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leqslant 0 \\ &g_4 = 80.51249 + 0.0071317x_2x_5 + \\ &\quad 0.0029955x_1x_5 + 0.0021813x_3^2 - 110 \leqslant 0 \\ &g_5 = -9.300961 - 0.0047026x_3x_5 - \\ &\quad 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leqslant 0 \\ &g_6 = 9.300961 + 0.00472026x_3x_5 + \\ &\quad 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leqslant 0 \\ &78 \leqslant x_1 \leqslant 102, 33 \leqslant x_2 \leqslant 45, 27 \leqslant x_i \leqslant 45 (i=3,4,5) \end{aligned} \tag{9}$$

文献[11, 14]中非线性规划方法获得的最优解为(78.0, 33.0, 29.995, 45.0, 36.776)。文献[11, 15]中遗传算法获得的最优解为(80.49, 35.07, 32.05, 40.33, 33.34)。混合粒子群算法获得的最优解为(78.36248616423723, 33.23250309833431, 30.14009295690651, 44.48992472357583, 36.52205628689010)。将以上优化结果的原目标函数值与约束违反程度进行对比,如表1所示。

表1 函数1的不同优化算法优化结果对比

优化算法	原目标函数值	约束违反程度
非线性规划法	−30 665.5	2.055e-004
遗传算法	−30 005.7	0
混合粒子群算法	−30 610.8	0

从表1可以看出,本文提出的混合粒子群算法在保证是可行解的条件下,计算的原目标值比遗传算法计算的值有很大程度上的改善,而且接近非线性规划得到的值。

选取的第二个函数如式(10)所示:

$$\begin{aligned}
&f_2(\boldsymbol{x})=x_1+x_2+x_3\\
&\text{s.t. } g_1(\boldsymbol{x})=-1+0.0025(x_4+x_6)\le 0\\
&g_2(\boldsymbol{x})=-1+0.0025(x_5+x_7-x_4)\le 0\\
&g_3(\boldsymbol{x})=-1+0.01(x_8-x_5)\le 0\\
&g_4(\boldsymbol{x})=-x_1x_6+833.33252x_4+100x_1-\\
&\quad 83333.333\le 0\\
&g_5(\boldsymbol{x})=-x_2x_7+1250x_5+x_2x_4-\\
&\quad 1250x_4\le 0\\
&g_6(\boldsymbol{x})=-x_3x_8+1250000+x_3x_5-\\
&\quad 2500x_5\le 0\\
&100\le x_1\le 10000\\
&1000\le x_i\le 10000 \quad (i=2,3)\\
&10\le x_i\le 1000 \quad (i=4,\cdots,8)
\end{aligned}\tag{10}$$

文献[11, 16~18]中用遗传算法计算的最优解为(579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979)。文献[19]利用Pareto强度方法获得的最优解为(579.30643727677, 135.97061959552, 5109.97096365681, 182.01767893950, 295.60116145374, 217.98232106050, 286.41651748576, 395.601 16145374)。利用混合粒子群算法得到的最优解为(579.31584752485, 1359.94587512697, 5110.06147138780, 182.01743194760, 295.59875949996, 217.98011315705, 286.41621755227, 395.59821396766)。将以上优化结果的原目标函数值和约束违反程度进行对比,如表2所示。

表2 函数2的不同优化算法优化结果对比

优化算法	原目标函数值	约束违反程度
Pareto强度算法	7 049.2480	2.1828e-9
遗传算法	7 049.3307	0
混合粒子群算法	7 049.3232	0

从表2可以看出,采用Pareto强度演化计算的最优解,严格说来为非可行解,而混合粒子群算法在保证可行解的基础上比遗传算法所计算得到的原目标函数的值更小。

选取的第三个函数如式(11)所示:

$$\begin{aligned}
&f_3(\boldsymbol{x})=x_1^2+x_2^2+x_1x_2-14x_1-16x_2+(x_3-10)^2+\\
&\quad 4(x_4-5)^2+(x_5-3)^2+2(x_6-1)^2+5x_7^2+\\
&\quad 7(x_8-11)^2+2(x_9-10)^2+(x_{10}-7)^2+45\\
&\text{s.t. } -105+4x_1+5x_2-3x_7+9x_8\le 0\\
&10x_1-8x_2-17x_7+2x_8\le 0\\
&-8x_1+2x_2+5x_9-2x_{10}-12\le 0\\
&3(x_1-2)^2+4(x_2-3)^2+2x_3^2-7x_4-120\le 0\\
&5x_1^2+8x_2+(x_3-6)^2-2x_4-40\le 0\\
&x_1^2+2(x_2-2)^2-2x_1x_2+14x_5-6x_6\le 0\\
&0.5(x_1-8)^2+2(x_2-4)^2+3x_5^2-x_6-30\le 0\\
&-3x_1+6x_2+12(x_9-8)^2-7x_{10}\le 0\\
&-10\le x_i\le 10(i=1,\cdots,10)
\end{aligned}\tag{11}$$

文献[11, 16~18]中遗传算法的最优解为(2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)。文献[19]采用Pareto强度方法来计算该优化方程,但没有提供具体的解。利用混合粒子群优化算法得到的最优解为(2.15965915281396, 2.39596367138456, 8.78078919830212, 5.11056810908457, 0.99443333706598, 1.42513984799026, 1.29675530256629, 9.80797899328002, 8.25494124510128, 8.39468017288231)。将以上优化结果的原目标函数值与约束违反程度进行对比,如表3所示。

表3 函数3的不同优化算法结果对比

优化算法	原目标函数值	约束违反程度
Pareto强度算法	24.306209068	/
遗传算法	24.3062091	1.7507e-5
混合粒子群算法	24.3176122	0

从表3可以看出,混合粒子群算法得到的最优解在保证是可行解的前提下,计算的原目标函数值与遗传算法计算的值非常接近。

4 结束语

通过对以上三个经典有复杂约束条件优化问题所进行的数值实验表明,提出的混合粒子群算法能在一定程度上克服粒子群算法发生早熟时所带来的影响,获得较好的优化效果。因此,本文所提出的混合粒子群算法为今后有复杂条件问题的优化提供了一种新的思路。

参考文献:

[1] IRUTHAYARJAN M W, BASKAR S. Evolutionary algorithms based design of multivariable PID controller[J]. Expert Systems with Applications, 2009, 36(5): 9159-9167.

[2] YILDIZ A R. A novel hybrid immune algorithm for global optimization in design and manufacturing[J]. Robotics and Computer-Integrated Manufacturing, 2009, 25(2): 261-270.

[3] HO P Y, SHIMIZU K. Evolutionary constrained optimization using an addition of ranking method and a percentage-based tolerance value adjustment scheme[J]. Information Sciences, 2007, 177(14): 2985-3004.

[4] KENNEDY J, EBERHART R C. Particle swarm optimization[C]//Proc of IEEE International Conference on Neural Networks. Perth: IEEE Press, 1995: 1942-1948.

[5] EBERHART R C, KENNEDY J. A new optimizer using particles swarm theory[C]//Proc of the 6th International Symposium on Micro Machine and Human Science. Nagoya: IEEE Press, 1995: 39-43.

[6] HE Q, WANG Ling. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization[J]. Applied Mathematics and Computation, 2007, 186(2): 1407-1422.

[7] LUO Yi-qing, YUAN Xi-gang, LIU Yong-jiang. An improved PSO algorithm for solving nonconvex NLP/MINLP problems with equality constraints[J]. Computers & Chemical Engineering, 2007, 31(3): 153-162.

(下转第3267页)

与原始心电信号的相关系数(所有相关系数均已取模)。从表中可以看出,采用本文算法恢复出的心电信号与原始信号的相关系数均达到 0.999 9 以上。可见,本文算法对心电信号也具有很好的工频干扰消除效果,恢复精度较高。

表 2　恢复信号与原始心电信号的相关系数

序号	相关系数	序号	相关系数
1	0.99999859328890	11	0.99999858422010
2	0.99999858019034	12	0.99999858774065
3	0.99999857671491	13	0.99999857021049
4	0.99999859986169	14	0.99999856939976
5	0.99999859705296	15	0.99999858451002
6	0.99999857248732	16	0.99999857030801
7	0.99999856861435	17	0.99999857273840
8	0.99999858295986	18	0.99999858749690
9	0.99999859970136	19	0.99999860025378
10	0.99999858592779	20	0.99999860034837

5　结束语

本文针对微弱信号采集过程中的工频干扰消除问题,利用人工构造观测信号的方法,使系统模型符合盲源分离的数学模型要求,将工频干扰消除问题转换为盲信号分离问题。利用粒子群优化算法寻找使盲分离问题判据最大化的分离矩阵,进而实现了消除微弱信号中工频干扰的目的。在粒子群优化算法的求解过程中,利用对分离矩阵的直接辨识转换成对一系列 Givens 旋转矩阵的辨识方法,减少了对未知元素辨识的数量,有效降低了算法的计算量。仿真结果验证了算法的有效性,说明了本算法具有优秀的工频干扰消除性能以及良好的发展前景。

参考文献:

[1] 沈凤麟. 生物医学随机信号处理[M]. 合肥:中国科技大学出版社,1999: 441-442.

[2] KENNEDY J, EBERHART R C. Particle swarm optimization[C]//Proc of IEEE International Conference on Neutral Networks. Perth: [s. n.], 1995: 1942-1948.

[3] SHI Yu-hui, EBERHART R C. A modified particle swarm optimizer[C]//Proc of IEEE World Congress on Computational Intelligence. Anchorage, Alaska:[s. n.], 1998:69-73.

[4] TICHAVSKY P, KOLDOVSKY Z, YEREDOR A, *et al*. A hybrid technique for blind separation of non-Gaussian and time-correlated sources using a multicomponent approach[J]. IEEE Trans on Neural Networks, 2008, 19(3): 421-430.

[5] SUN T Y, LIU Chan-cheng, HSIEH S T, *et al*. Blind separation with unknown number of sources based on auto-trimmed neural network[J]. Neurocomputing, 2008, 71(10): 2271-2280.

[6] LI Shu-jun, LI Cheng-qing, LO K T, *et al*. Cryptanalyzing an encryption scheme based on blind source separation[J]. IEEE Trans on Circuits and Systems I: Regular Papers, 2008, 55(4): 1055-1063.

[7] ZHANG Hong-juan, SHI Zhen-wei, GUO Chong-hui. Blind source extraction based on generalized autocorrelations and complexity pursuit[J]. Neurocomputing, 2009, 72(10-12): 2556-2562.

[8] BELL A J, SEJNOWSKI T J. An information-maximization approach to blind separation and blind deconvolution[J]. Neuralcomputation, 1995, 7(6): 1129-1159.

[9] HYVARINEN A. Fast and robust fixed-point algorithms for independent component analysis[J]. IEEE Trans on Neural Networks, 1999, 10(3): 626-634.

[10] AMARI S, CICHOCKI A, YANG H H. A new learning algorithm for blind signal separation[C]//Proc of Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 1996: 757-763.

[11] 凌云,高军,张汝杰,等. 随时间推移地震勘探处理方法研究[J]. 石油地球物理勘探,2001,36(2):173-179.

[12] 刘洋. 强工频干扰波的提取与消除方法[J]. 石油物探, 2003, 42(2):154-159.

[13] 周静. 心电信号中工频干扰的消除[J]. 生物医学工程研究, 2003, 22(4): 61-64.

[14] 吴小培, 詹长安, 周荷琴, 等. 采用独立分量分析方法消除信号中的工频干扰[J]. 中国科技大学学报, 2000, 30(6):671-676.

[15] 刘俊豪. 基于粒子群算法和鱼群算法的盲源分离的研究[D]. 太原: 太原理工大学, 2006.

[16] 覃和仁, 谢胜利. 基于 QR 分解与罚函数方法的盲分离算法[J]. 计算机工程, 2003, 29(17):55-57.

(上接第 3258 页)

[8] AKJIRATIKARL C, YENRADEE P, DRAKE P R. PSO-based algorithm for home care worker scheduling in the UK[J]. Computers & Industrial Engineering, 2007, 53(4): 559-583.

[9] 丁雷, 吴敏, 曹卫华, 等. 基于混合粒子群算法的铅锌烧结过程产量质量优化[J]. 中国有色金属学报, 2008, 18(6): 1152-1158.

[10] BERGH F van den. An analysis of particle swarm optimizers[D]. South Africa: University of Pretoria, 2002.

[11] DEB K. An efficient constraint handling method for genetic algorithms[J]. Computer Methods in Applied Mechanics and Engineering, 2000, 186(2-4): 311-338.

[12] TAKAHAMA T, SAKAI S. Constrained optimization by applying the α constrained method to the nonlinear simplex method with mutations[J]. IEEE Trans on Evolutionary Computation, 2005, 9(5): 437-451.

[13] CAMPONOGARA E, TALUKDAR S N. A genetic algorithm for constrained and multi-objective optimization[C]//Proc of the 3rd Nordic Workshop on Genetic Algorithm and Their Application. Finland: University of Vaasa Press, 1997: 49-62.

[14] HIMMELBLAU D M. Applied nonlinear programming[M]. New York: McGraw-Hill, 1972.

[15] HOMAIFAR A, LAI S H V, QI X. Constrained optimization via genetic algorithms[J]. Simulation, 1994, 62(4): 242-254.

[16] RUNARSSON T P, YAO Xin. Search biases in constrained evolutionary optimization[J]. IEEE Trans on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2005, 35(2): 233-243.

[17] MING Yu-chi, KIM J H, JO J. A population ecology inspired parent selection strategy for numerical constrained optimization problems[J]. Applied Mathematics and Computation, 2007, 190(1): 292-304.

[18] MICHALEWICZ Z. Genetic algorithms, numerical optimization, and constraints[C]//Proc of the 6th International Conference on Genetic Algorithms. San Mateo: Morgan Kaufmann Publishers, 1995: 151-158.

[19] 周育人, 李元香, 王勇, 等. Pareto 强度值演化算法求解约束优化问题[J]. 软件学报, 2003, 14(7): 1243-1249.