第 10 卷 第 8 期 2010 年 3 月
1671-1815(2010)08-1870-04
科 学 技 术 与 工 程
Science Technology and Engineering
Vol. 10 No. 8 Mar. 2010

计算机技术

一个求解约束工程设计问题的混合局部收缩微粒群算法

于龙文 刘国志
(辽宁石油化工大学理学院信息与计算科学系,抚顺 113001)

摘 要 提出一个求解约束工程设计问题的新的混合算法——与可行基规则相结合的局部收缩微粒群算法。与惩罚函数法相比,可行基规则不需要额外的参数,且指引粒子迅速飞向可行域。利用 3 个工程设计问题进行仿真计算比较,仿真结果表明了新算法是求解约束工程设计问题的一个高效的算法。

关键词 可行基规则 微粒群算法 约束最优化
中图法分类号 TP18; **文献标志码** A

非线性规划是运筹学的一个重要分支,广泛应用于军事、经济、工程技术和管理科学。约束优化问题可以描述为

$$\begin{cases} \min f(x) \\ \text{s. t. } g_i(x) \leqslant 0, \quad i = 1,2,\cdots,n \end{cases} \tag{1}$$

关于求解约束优化问题有许多方法,传统的数学规划法要求目标函数和约束条件是可微的,且获得的最优解通常是局部最优解。近年来,进化算法由于其不要求函数是可导的和连续的,且能够求出全局最优解,已被应用求解各种优化问题。众所周知,惩罚函数法由于其简单且易实施一直是最流行的约束处理技术,其原理是通过引入罚函数,将约束优化问题转换为无约束优化问题。由于采用非稳态的罚函数方法,罚值和罚因子的选择与具体问题有关,给该方法应用带来了一定困难[1]。所以,Coello[2] 把遗传算法应用于决定解和罚因子的进化,提出了协同进化的遗传算法;He 和 Wang[3] 把 PSO 算法应用于决定解和罚因子的进化,提出了协同进化的微粒群算法,取得了较好的结果,但同样存在系数的选择问题。为了避免罚因子的选取,He 和 Wang[1] 又提出了一个与可行基规则相结合微粒群算法求解约束优化问题。本文在此基础上,提出一个与可行基规则相结合局部收缩的微粒群混合算法求解约束工程优化问题,并通过实例验证了算法的有效性。

1 局部收缩的微粒群算法和可行基规则

1.1 局部收缩的微粒群算法

令 n 表示搜索空间的维数,$x_i = (x_{i1}, x_{i2}, \cdots, x_{in})$ 表示微粒 i 当前的位置,$p_i = (p_{i1}, p_{i2}, \cdots, p_{in})$ 表示微粒 i 曾经达到的最好位置。种群中最优微粒的序号用 g 表示,微粒 i 的速度用 $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$ 表示。每个微粒根据式(2)来更新自己的速度和位置:

$$\begin{cases} v_i^{k+1} = \omega_i^k (v_i^k + c_1 \text{rand}(\cdot)(p_i^k - x_i^k)) + \\ \qquad c_2 \text{Rand}(\cdot)(p_g - x_i^k) \\ x_i^{k+1} = x_i^k + v_i^{k+1} \end{cases} \tag{2}$$

式(2)中:k 表示迭代次数,c_1,c_2 为学习因子,rand(·),Rand(·)是[0,1]区间的随机数,ω_i^k 为惯性权重,关于惯性权重的调整分为线性和非线性两类,按式(3)动态调整。

设第 i 个微粒在前 $k-1$,k 次迭代的目标函数的最小值分别为 f_{bi}^{k-1},f_{bi}^{k},第 k 次迭代的目标函数值

2009 年 12 月 8 日收到 国家自然科学基金(50771052)资助
第一作者简介:于龙文(1956—),男,吉林长春人,学士/副教授,研究方向:最优化理论与算法、微粒群算法。

为f_i^k,全体微粒前 k 次迭代的目标函数的最小值为f_g^k。

$$\omega_i^{k+1}=\begin{cases}\omega_{\max},f_{bi}^{k-1}-f_{bi}^k>\dfrac{3}{4}(f_i^k-f_g^k)\ 且\ 2\omega_i^k\geqslant\omega_{\max}\\ 2\omega_i^k,f_{bi}^{k-1}-f_{bi}^k>\dfrac{3}{4}(f_i^k-f_g^k)\ 且\ 2\omega_i^k<\omega_{\max}\\ \omega_{\min},f_{bi}^{k-1}-f_{bi}^k<\dfrac{1}{10}(f_i^k-f_g^k)\ 且\ \dfrac{1}{2}\omega_i^k\leqslant\omega_{\min}\\ \dfrac{1}{2}\omega_i^k,f_{bi}^{k-1}-f_{bi}^k<\dfrac{1}{10}(f_i^k-f_g^k)\ 且\ \dfrac{1}{2}\omega_i^k>\omega_{\min}\\ \omega_i^k,\dfrac{1}{10}(f_i^k-f_g^k)\leqslant f_{bi}^{k-1}-f_{bi}^k\leqslant\dfrac{3}{4}(f_i^k-f_g^k)\end{cases}\tag{3}$$

1.2　可行基规则

利用文[1]提出的可行基规则处理约束条件,不可行解约束偏差值按式(4)计算。

$$\mathrm{viol}(x)=\sum_{j=1}^{N}[\max(g_j(x),0)]\tag{4}$$

假设 $P_i(k)$代表第 i 粒子在第 k 代的历史最好位置 p_{best},$X_i(k+1)$代表第 i 粒子在 $k+1$ 代的新产生的位置。在标准的 PSO,$P_i(k+1)=X_i(k+1)$,当且仅当$f(X_i(k+1))<f(P_i(k))$。现按如下规则对P_{best}进行更新:

(1) $P_i(k)$是不可行的,但 $X_i(k+1)$是可行的;

(2) $P_i(k)$和 $X_i(k+1)$都是可行的,但$f(X_i(k+1))<f(P_i(k))$;

(3) $P_i(k)$和 $X_i(k+1)$都是不可行的,但$\mathrm{viol}(X_i(k+1))<\mathrm{viol}(P_i(k))$。

类似地,关于所有粒子的历史最好位置 g_{best},在每一代按着上述规则进行更新。

2　仿真与分析

在实验中,为了评价比较 HLPSO 算法与其它算法的性能,采用文献[1]中的 3 个典型的工程设计问题(例 1:a tension/compression string design problem;例 2:a weldel beam design problem;例 3:a pressure vessel design problem)。HLPSO 的参数设置:种群大小为 20,学习因子 $c_1=c_2=2.05$,$\omega_{\min}=0.73$,$\omega_{\max}=0.85$(在例 1 中,$\omega_{\max}=0.9$),惯性权重 ω 按式(3)非线性动态调整。每个例子独立运行 20 次。

表 1 和表 2 列出了例 1 的不同方法最优解比较和统计结果。从表 1 可以看出,由 HLPSO 算法获得的最优解优于其他的进化算法。表 2 表明了HLPSO 算法的搜索质量好于其他的进化算法,由 HLPSO 算法找到的最差解好于其他的进化算法的最好解且标准差也是最小的。

表 1　由不同方法得到的例 1 的最优解比较

Methods	$x_1(d)$	$x_2(D)$	$x_3(P)$	$f(x)$
Belegundu[4]	0.050 00	0.315 900	14.250 000	0.012 833 4
Arora[5]	0.053 396	0.399 180	9.185 400	0.012 730 3
Coello[2]	0.051 480	0.351 661	11.632 201	0.012 704 8
Coello and Montes[6]	0.051 989	0.363 965	10.890 522	0.012 681 0
Coello and Becerra[7]	0.050 000	0.317 395	14.031 795	0.012 721 0
CPSO[3]	0.051 728	0.357 644	11.244 543	0.012 674 7
HPSO[1]	0.051 706	0.357 126	11.265 083	0.012 665 2
HLPSO	0.500 000 00	0.317 425 40	4.452 770 66	0.005 120 68

表 2　例 1 不同方法的统计结果

Methods	Best	Mean	Worst	Std./10^{-5}
Belegundu[4]	0.012 833 4	N/A	N/A	N/A
Arora[5]	0.012 730 3	N/A	N/A	N/A
Coello[2]	0.012 704 8	0.012 769 0	0.012 822	3.939 0
Coello and Montes[6]	0.012 681 0	0.012 742 0	0.012 973	5.900 0
Coello and Becerra[7]	0.012 721 0	0.013 681	0.015 116	804 1520
CPSO[3]	0.012 674 7	0.012 730 0	0.012 924	51 985
HPSO[1]	0.012 665 2	0.012 707 2	0.012 719 1	1.582 4
HLPSO	0.005 1206 8	0.005 120 69	0.005 120 71	0.001

表 3 和表 4 列出了例 2 的不同方法最优解比较和统计结果。从表 3 可以看出,由 HLPSO 算法获得的最优解优于其他的进化算法。表 4 表明了HLPSO 算法的搜索质量好于其他的进化算法,由 HLPSO 算法找到的最差解好于其他进化算法的最好解且标准差也是最小的。

表 3 由不同方法得到的例 2 的最优解比较

Methods	$x_1(h)$	$x_2(l)$	$x_3(t)$	$x_4(b)$	$f(x)$
Coello[2]	0.208 800	3.420 500	8.997 500	0.210 000	1.748 309
Coello and Montes[6]	0.205 986	3.471 328	9.020 224	0.206 480	1.728 226
Coello and Becerra[7]	0.205 700	3.470 500	9.036 600	0.205 700	1.724 852
CPSO[3]	0.202 369	3.544 214	9.048 210	0.205 723	1.728 024
HPSO[1]	0.205 730	3.470 489	9.036 624	0.205 730	1.724 852
HLPSO	0.182 862 69	3.804 280 24	9.585 003 30	0.182 862 93	1.641 866 03

表 4 例 2 不同方法的统计结果

Methods	Best	Mean	Worst	Std.
Coello[2]	1.748 309	1.771 973	1.785 835	0.011 220
Coello and Montes[6]	1.728 226	1.792 654	1.993 408	0.074 713
Coello and Becerra[7]	1.724 852	1.971 809	3.179 709	0.443 131
CPSO[3]	1.728 024	1.748 831	1.782 143	0.012 926
HPSO[1]	1.724 852	1.749 040	1.814 295	0.040 049
HLPSO	1.641 866 03	1.641 896 91	1.641 966 21	0.000 025

表 5 和表 6 列出了例 3 的不同方法最优解比较和统计结果。从表 5 可以看出,由 HLPSO 算法获得的最优解优于其他的进化算法。表 6 表明了 HLPSO 算法的搜索质量好于其他的进化算法,由 HLPSO 算法找到的最差解好于其他的进化算法的最差解且标准差也是最小的。

表 5 由不同方法得到的例 3 的最优解比较

Methods	$x_1(T_s)$	$x_2(T_h)$	$x_3(R)$	$x_4(L)$	$f(x)$
Sandgren[8]	1.125 0	0.625 0	47.700 0	117.701 0	8129.800 0
Kannan[9]	1.125 0	0.625 0	58.291 0	43.690 0	7198.042 8
Deb[10]	0.937 5	0.500 0	48.329 0	112.679 0	6 410.381 1
Coello[2]	0.812 5	0.437 5	40.323 9	200.000 0	6 288.744 5
Coello and Montes[6]	0.812 5	0.437 5	42.097 4	176.654 0	6 059.946 3
CPSO[3]	0.812 5	0.437 5	42.091 3	176.746 5	6 061.077 7
HPSO[1]	0.812 5	0.437 5	42.098 4	176.636 6	6 059.714 3
HLPSO	0.783 345 16	0.389 297 36	40.580 191 36	196.417 225 23	5 901.661 278

表 6 例 3 不同方法的统计结果

Methods	Best	Mean	Worst	Std.
Sandgren[8]	8 129.800 0	N/A	N/A	N/A
Kannan[9]	7 198.042 8	N/A	N/A	N/A
Deb[10]	6 410.381 1	N/A	N/A	N/A
Coello[2]	6 288.744 5	6 293.843 2	6 308.149 7	7.413 3
Coello and Montes[6]	6 059.946 3	6 177.253 3	6 469.322 0	130.929 7
CPSO[3]	6 061.077 7	6 147.133 2	6 363.804 1	86.454 5
HPSO[1]	6 059.714 3	6 099.932 3	6 288.677 0	86.202 2
HLPSO	5 901.661 278	5 989.671 264	6 189.858 664	0.000 01

4 结论

提出了一个新的与可行基规则相结合的混合局部收缩的微粒群算法。它克服了惩罚函数法的缺点,是求解约束工程设计问题的一个高效的算法。仿真结果和比较表明了 HLPSO 混合算法在解的搜索质量、效率和关于初始点的鲁棒性都远优于其他的进化算法。此外,由 HLPSO 混合算法搜索到最优解好于已发现的最优解。

参 考 文 献

1 He Qie, Wang Ling. A hybrid particle swarm optimizatuion with a feasibility-based rule for constrained optimization. Applied Mathematics and Computation,2007;186(2):1407—1422

2 Coello C A C. Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind,2000;41:113—127

3 He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell,2007;20(1):89—99

4 刘国志,苗 晨. Powell 搜索法和惯性权重非线性调整局部收缩微粒群算法的混合算法. 吉林大学学报,2008;6:1149—1154

5 Coello C A C, Montes E M. Contraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inform,2002;16:193—203

6 Coello C A C, Becerra R C. Efficient evolutionary optimization through the use of a cultural algorithm. Eng Optimiz,2004:36(2):219—236

7 Belegundu A D. A sdudy of mathematical programming methods for structural optimization. Department of Civil and Environmental Engineering, University of Iowa, IA,1982

8 Arora J S. Introduction to Optimum Design. New York: McGraw-Hill, 1989

9 Kannan B K, Kramer S N. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, J. Mech. Des. 1994;116:318—320

10 Deb K, Gene AS: a robust optimal design technigue for mechanical component design. In:(Eds.) Dasgupta D, Michalewicz Z. Evolutionary algorithms in engineering applications, Berlin: Springger-Verlag, 1997:497—514

11 Sandgren E. Nonlinear integer and discrete programming in mechanical engineering systems. J Mech Des,1990;112(1):223—229

12 Koziel S, Michalewicz Z. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evol Comput, 1999;7(1):19—44

13 Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization, IEEE Trans Evol Compt,2000;4(3):284—294

A Hybrid Local Constriction Approach Particle Swarm Optimization for Constrained Engineering Design Problems

YU Long-wen, LIU Guo-zhi

(Department of Information & Computational Science, College of Science, Liaoning University of Petroleum & Chemical Technology, Fushun 113001, P. R. China)

[**Abstract**] The hybrid local constriction particle swarm optimization (HLPSO) with a feasibility-based rule is proposed to solve constrained engineering design problems. In contrast to the penalty function method, the rule requires no additional parameters and can guide the swarm to the feasible region quickly. Simulation and comparisons based on well-known constrained engineering design problems demonstrate the effectiveness, efficiency and robustness of the proposed HLPSO.

[**Key words**] feasibility-based rule　particle swarm optimization　unconstrained optimization

(上接第 1869 页)

8 Kennedy J, Eberhart R. A discrete binary version of the particle swarm optimization algorithm. Systems Man and Cybernetics, IEEE ICSMC,1997;(5):4104—4108

9 Shi Y, Eberhart R. Parameter selection in particle swarm optimization. In: Proc Int Conf Evol Program,1998:591—600

10 Kennedy J. The Particle Swarm: Social Adaptation of knowledge. In: Proc IEEE Int Conf. Evolutionary Computation,1997:303—308

11 Choi S, Cichocki A, and Amari S. Flexible Independent Component Analysis. Journal of VLSI Signal Processing, 2000; 26 (1/2): 25—38

Research on Independent Component Analysis Based on Particle Swarm Optimization Algorithms

ZHANG Wen-xi, ZHENG Mao[1*]

(Department of Electronic and Communication Engineering, Changsha University, Changsha 410003, P. R. China; School of Electronic Science and Engineering, National University of Defense Technology[1], Changsha 410073, P. R. China)

[**Abstract**] On the basis of analyzing the independent component analysis algorithms, a novel method based on particle swarm optimization was proposed to minimize the mutual information, which through improving position vector and velocity vector to get the global optimization solution and then separate the mixed signals. The simulation results showed that the independent component analysis based on particle swarm optimization was a more efficient algorithm.

[**Key words**] independent component analysis　mutual information　particle swarm optimization　fitness function