结合增广 Lagrange 罚函数的约束优化差分进化算法*

龙 文[1a,1b]，徐松金[2]

(1. 贵州财经学院 a. 贵州省经济系统仿真重点实验室；b. 数学与统计学院，贵阳 550004；2. 铜仁学院 数学与计算机科学系，贵州 铜仁 554300)

摘 要：利用增广 Lagrange 罚函数处理问题的约束条件，提出了一种新的约束优化差分进化算法。基于增广 Lagrange 惩罚函数，将原约束优化问题转换为界约束优化问题。在进化过程中，根据个体的适应度值将种群分为精英种群和普通种群，分别采用不同的变异策略，以平衡算法的全局和局部搜索能力。用 10 个经典 Benchmark 问题进行了测试，实验结果表明，该算法能有效地处理不同的约束优化问题。

关键词：约束优化问题；差分进化算法；增广 Lagrange 罚函数；变异策略

中图分类号：TP301.6　　**文献标志码**：A　　**文章编号**：1001-3695(2012)05-1673-03

doi:10.3969/j.issn.1001-3695.2012.05.019

Constrained optimization differential evolution algorithm using augmented Lagrange penalty function

LONG Wen[1a,1b], XU Song-jin[2]

(1. *a. Guizhou Key Laboratory of Economics System Simulation, b. School of Mathematics & Statistics, Guizhou University of Finance & Economics, Guiyang* 550004, *China*; 2. *Dept. of Mathematics & Computer, Tongren University, Tongren Guizhou* 554300, *China*)

Abstract: Using augmented Lagrange penalty function to deal with the constrained conditions, this paper proposed a modified constrained optimization differential evolution algorithm. It converted the general constrained optimization problem into a bound constrained optimization problem. In the process of evolution, divided the initial population into two subpopulations, i.e. elite and general subpopulations, which used different mutation strategies to balance the ability of global and local search respectively. It tested ten classic Benchmarks problems, the experiment results show that the proposed algorithm is an effective way for constrained optimization problems.

Key words: constrained optimization problem; differential evolution algorithm; augmented Lagrange penalty function; mutation strategy

科学研究和工程应用中大多数优化问题都含有等式约束和不等式约束，而且这些约束条件通常都是非线性的，这使得约束优化问题的求解要比无约束优化问题复杂得多。不失一般性，非线性约束优化(最小化)问题可描述为

$$\begin{aligned} &\min \quad f(\boldsymbol{x}) \\ &\text{s.t.} \quad g_j(\boldsymbol{x}) \leqslant 0 \quad j=1,2,\cdots,p \\ &h_j(\boldsymbol{x}) = 0 \quad j=p+1,p+2,\cdots,m \\ &l_i \leqslant x_i \leqslant u_i \quad i=1,2,\cdots,n \end{aligned} \tag{1}$$

其中：$f(\boldsymbol{x})$为目标函数；$g_j(\boldsymbol{x})$和 $h_j(\boldsymbol{x})$分别为不等式约束条件和等式约束条件；$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)\in \mathrm{R}^n$ 称为 n 维决策向量；l_i 和 u_i 分别为变量 x_i 的上、下界。

与遗传算法的原理相似，差分进化算法也是通过个体在变异、交叉以及选择算子的作用下向更高的适应度进化以达到寻求问题最优解的目标[1]。差分进化算法已被证明在收敛速度和稳定性方面都超过了其他几种知名的随机算法，对于大多数的数值 Benchmark 问题，差分进化算法要优于粒子群和其他进化算法[2,3]。与传统的优化方法相比，差分进化算法不需要借助问题的梯度信息，更适合于求解约束优化问题，但它需要结合一种合适的约束处理技术。利用差分进化算法求解约束优化问题已成为进化计算领域的研究热点之一，并且提出了大量的约束优化差分进化算法[4,5]。

本文采用增广 Lagrange 罚函数法处理约束条件，将非线性约束优化问题转换为一系列带界约束的无约束优化问题，然后利用差分进化算法进行优化。为保证种群个体均匀分布在搜索空间中，利用佳点集方法初始化种群。在进化过程中，将种群分为精英子种群和普通子种群，分别采用不同的变异策略，同时根据不同的搜索阶段自适应地调整各子种群的个体数目。

1 增广 Lagrange 惩罚函数法

在求解约束优化问题中，罚函数法是最常用的约束处理技术之一，其基本思想是在目标函数中加上一个能反映是否满足约束的惩罚项，从而构成一个无约束的广义目标函数，然后利用优化算法对该广义目标函数进行寻优，使得算法在惩罚项的作用下找到问题的最优解[6]。

对于如式(1)所示的约束优化问题，如果没有界约束，则可用增广 Lagrange 函数法来求解上述问题。对给定的 Lagrange 乘子向量$\boldsymbol{\lambda}^k$和罚参数向量$\boldsymbol{\sigma}^k$，增广 Lagrange 函数法第 k

收稿日期：2011-10-30；**修回日期**：2011-11-30　　**基金项目**：国家自然科学基金资助项目(61074069)

作者简介：龙文(1977-)，男，湖南隆回人，讲师，博士，主要研究方向为进化计算、约束优化及应用(lw770457@163.com)；徐松金(1972-)，湖南隆回人，讲师，硕士，主要研究方向为优化理论与应用.

步所求得子问题为

$$\min P(\boldsymbol{x},\lambda^k,\sigma^k) \tag{2}$$

其中:$P(\boldsymbol{x},\lambda,\sigma)$是增广 Lagrange 罚函数[7]。

$$P(\boldsymbol{x},\lambda,\sigma)=f(\boldsymbol{x})-\sum_{i=1}^{p}[\lambda_i g_i(\boldsymbol{x})-\frac{1}{2}\sigma_i(g_i(\boldsymbol{x}))^2]-\sum_{i=p+1}^{m}\bar{P}_i(\boldsymbol{x},\lambda,\sigma) \tag{3}$$

其中:

$$\bar{P}_i=\begin{cases}\lambda_i g_i(\boldsymbol{x})-\frac{1}{2}\sigma_i(g_i(\boldsymbol{x}))^2 & \lambda_i-\sigma_i g_i(\boldsymbol{x})>0\\ \frac{1}{2}\lambda_i^2/\sigma_i & \text{否则}\end{cases}$$

如果上述问题存在界约束,上面的增广 Lagrange 函数法需要进行修改。在第 k 步,求解如下界约束子问题:

$$\begin{cases}\min\ P(\boldsymbol{x},\lambda^k,\sigma^k)\\ \text{s.t.}\quad l\leqslant\boldsymbol{x}\leqslant u\end{cases} \tag{4}$$

其中:$P(\boldsymbol{x},\lambda,\sigma)$是式(3)所定义的罚函数。

在利用增广 Lagrange 罚函数法求解约束优化问题时,乘子向量λ 和罚参数向量σ 一般很难确定。因此,在这里采用文献[7]中的方法对乘子向量λ 和罚参数向量σ 进行初始化和修正。Lagrange 乘子向量的初始值λ_0一般设定为用户预先对每个约束乘子的初始估计。关于 Lagrange 乘子向量的修正,本文假设在第 k 步,λ^k和σ^k已知,$\bar{x}_k$ 是界约束问题式(4)的解,Lagrange 乘子应作如下修正[7]:

$$\begin{cases}\lambda_j^{k+1}=\lambda_j^k-\sigma_j^k g_j(\bar{x}_k) & j=1,\cdots,p\\ \lambda_j^{k+1}=\max\{\lambda_j^k-\sigma_j^k h_j(\bar{x}_k),0\} & j=p+1,\cdots,m\end{cases} \tag{5}$$

罚参数向量初始值σ_0通常设定为$\sigma_0=\sigma\cdot(1,\cdots,1)^{\mathrm{T}}$,其中$\sigma=10$ 或$\sigma=100$[7]。在第 k 步,假设σ^k已知,则

$$\sigma^{k+1}=\gamma\cdot\sigma^k \tag{6}$$

其中:γ 通常取常数 10[7]。

2 改进的差分进化算法

对于界约束优化子问题式(4)的求解,可以采用拟牛顿法等传统的数值计算方法,但需要计算问题的梯度,且找到的解大多为局部最优解。如前所述,差分进化算法是一种模拟自然进化的全局优化方法,具有收敛速度快、全局搜索能力强等特点。因此,本文提出一种改进的差分进化算法来求解。

2.1 种群初始化

由差分进化算法的机理可知,变异、交叉和选择操作在一定程度上依赖于种群,初始种群的优劣对于进化算法的收敛性以及搜索效率会产生较大的影响。另外,在求解具体问题前,对其全局最优解处在什么位置还一无所知,如果随机初始化种群个体,不利于搜索到全局最优解,可能导致要增加迭代次数或种群大小,这势必会增加算法的计算量。因此,在初始化种群个体时应尽可能地均匀分布在整个搜索空间中[8]。

佳点集方法是一种有效的、可以减少实验次数的实验方法。在相同取点数个数的条件下,佳点序列要比其他方法选取的点序列更均匀。设种群规模为 N, 个体 $A_N^i=\{a_1^i,a_2^i,\cdots,a_s^i\}$, $i=1,2,\cdots,N$,首先在 s 维空间 H 中取含有 N 个点的佳点集。s 维空间 H 中取佳点的方法[9]如下:$P_n(i)=\{(\{r_1*k\},\{r_2*k\},\cdots,\{r_s*k\}),k=1,2,\cdots,N\}$, 其中取

$$r_i=\left\{2\cos\frac{2\pi i}{t}\right\}\quad 1\leqslant i\leqslant s \tag{7}$$

t 是满足 $t\geqslant 2s+3$ 的最小素数;或者取

$$r_i=\mathrm{e}^i\quad 1\leqslant i\leqslant s \tag{8}$$

其中:$\{r_i*k\}$表示 r_i*k 的小数部分。

当 $a_k^i(k=1,2,\cdots,s;i=1,2,\cdots,N)$为实数编码时, 设 a_k^i 的取值为 $\alpha_k\leqslant a_k\leqslant\beta_k$, 取 $a_k^i=\alpha_k+\{r_k*i\}*(\beta_k-\alpha_k)$即可。

因此,本文采用佳点集方法生成初始种群,从而保证了初始种群的多样性。图 1 是采用佳点集方法产生的规模为 80 的二维初始种群分布。从图 1 可以看出,利用佳点集方法产生的初始种群个体分布均匀,具有较好的多样性。

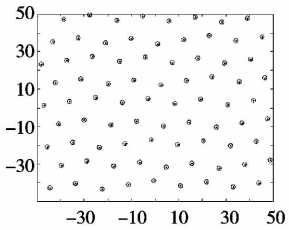

图1 佳点集方法产生的初始种群分布

2.2 变异操作

变异操作是差分进化算法的核心操作。根据变异个体的生产方法不同,形成了多种不同的变异策略,如将 x_i^t 作为 t 代的一个目标向量 $\boldsymbol{x}_i^t=\{x_{i1}^t,x_{i2}^t,\cdots,x_{in}^t\}$,常用的差分进化变异策略如下:

a) DE/rand/1 变异策略

$$v_i^{t+1}=x_{r1}^t+F(x_{r2}^t-x_{r3}^t) \tag{9}$$

b) DE/best/1 变异策略

$$v_i^{t+1}=x_{\text{best}}^t+F(x_{r2}^t-x_{r3}^t) \tag{10}$$

其中:$r_1,r_2,r_3\in\{1,2,\cdots,N\}$是互不相同且与 i 不同的随机个体;x_{best}^t为当前种群中适应度函数值最好的个体;$F\in[0,2]$为缩放因子,主要用来调节向量差异的步长幅度,以避免进化过程的停滞。

对于 DE/rand/1 变异策略,由式(9)可知,变异个体 v_i^{t+1} 由三个互不相同的随机个体组成,无须任何适应度函数值信息,有利于保持种群的多样性,因而其全局搜索能力强,但存在收敛速度慢、搜索效率不高等缺点;对于 DE/best/1 变异策略,由式(10)可知,变异个体 v_i^{t+1} 由 x_{best}^t作引导,因而局部搜索能力强、搜索精度高、收敛速度快,但会增加算法陷入局部最优的概率。根据个体的适应度值,将群体分为两个子种群,即精英子种群和普通子种群,其个体数分别为 N_1 和 N_2,满足 $N_1+N_2=N$,N 为种群规模。结合两种变异策略的特点,发挥各自优点,在进化过程中,精英种群采用 DE/best/1 变异策略,承担局部搜索任务,在其附近进行搜索,加快算法收敛;普通种群采用 DE/rand/1 变异策略,承担算法的全局搜索任务,后期用来跳出局部最优,避免算法早熟收敛。在迭代过程中,算法在搜索初始阶段应偏重于全局搜索,因而此阶段的 N_2 值应取得较大,以增强算法的全局搜索能力;在搜索后期,主要考虑局部精确搜索,这时 N_1 的取值应大些,以提高算法的局部搜索能力。因此,每次迭代过程中每个子种群的个体数目是自适应动态调整的。

2.3 交叉操作

对于群体中第 i 个个体 $\boldsymbol{x}_i^t$,将与 x_m 进行交叉操作,产生实

验个体 x_T。为保证个体 x_i^t 的进化,首先通过随机选择,使得 x_T 的 D 维变量中至少有一维由 x_m 贡献,而对于其他维,可利用一个交叉概率因子 C_r 决定 x_T 中哪位由 x_i^t 贡献,哪位由 x_m 贡献。交叉操作的具体形式为

$$x_{Tj}=\begin{cases}x_{mj} & \text{rand}()\leqslant C_r\\ x_{ij}^t & \text{rand}()>C_r\end{cases}\qquad j=1,2,\cdots,D \tag{11}$$

式中:rand()为[0,1]之间均匀分布的随机数,$C_r\in[0,1]$。由式(11)可知,若 C_r 越大,则 x_m 对 x_T 的贡献越多,当 $C_r=1$ 时,$x_T=x_m$,有利于局部搜索和加速收敛速度;如果 C_r 越小,则 x_i^t 对 x_T 的贡献越多,当 $C_r=0$ 时,$x_T=x_i^t$,有利于保持种群的多样性和全局搜索。因此,本文采用自适应动态调整交叉概率 C_r 的大小策略,即

$$C_r=C_{r\min}+(C_{r\max}-C_{r\min})\times\frac{t}{t_{\max}} \tag{12}$$

式中:$C_{r\min}$ 和 $C_{r\max}$ 分别为交叉概率 C_r 的最小值和最大值,t 和 $t_{\max}$ 分别为当前迭代次数和最大迭代次数。

2.4　算法步骤

综上所述,本文算法的流程如图 2 所示。

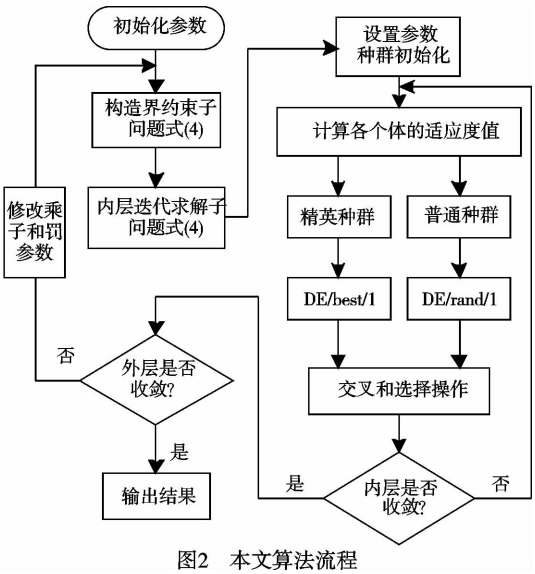

图2　本文算法流程

3　数值实验及分析

为了测试本文算法的性能,从文献[9]的 13 个标准测试问题中选取了 10 个优化问题,即 g01、g02、g04、g05、g06、g07、g08、g09、g10 和 g11 进行测试,各测试问题的具体表达式见文献[9]。将本文算法记为 ALCODE,并与基于可行性规则的差分进化(CHDE)算法[4]、多目标差分进化(MODE)算法[5]、随机排序(SR)算法[10]和集成约束处理技术(ECHT)算法[11]得到的结果进行比较。在比较实验中,ALCODE 算法的参数设置是:种群规模 N 取为每个问题参数变量个数的 10 倍,$C_{r\min}=0$,$C_{r\max}=1$,$F=1.0$。对所有的测试问题适应值评价次数均设置为 350 000 次。其他算法的参数设置分别见其各自的文献。每个测试问题在相同条件下独立运行 30 次实验,记录其最好结果(best)、平均结果(mean)、最差结果(worst)和标准方差(st. dev)。表 1 给出了在上述参数设置下,ALCODE 与 CHDE、MODE、SR 和 ECHT 算法对 10 个标准测试问题的寻优结果比较。

从表 1 可知,本文所提出的 ALCODE 算法对 10 个问题 30 次实验中一致地找到全局最优解,但是对于两个复杂的测试问题 g02 和 g10,虽然找到了全局最优解,还有几次实验陷入了局部最优。与 CHDE 算法相比,ALCODE 算法在 10 个问题上得到的结果明显要优。与 MODE 算法相比,ALCO DE 算法在 10 个问题上得到的结果相当。SR 算法对其中的 6 个问题(g01、g04、g06、g08、g09 和 g11)一致地找到了全局最优解,对其中的 2 个问题(g05、g07)的实验结果也非常接近最优解,而对 2 个非常复杂的测试问题 g02 和 g10,得到的实验结果仍然远离已知的最优解。与 SR 算法相比,ALCODE 算法得到的结果明显要优。与 ECHT 算法相比,ALCODE 算法在 10 个测试问题上取得了与其相似的结果。另外,对各种算法的计算量(计算适应值的次数)比较,ECHT 和 COEA-OED 算法的次数为 240 000 次,CHDE 和 MODE 算法的次数为 348 000 次,ALCODE 和 SR 算法的适应值计算次数均为 350 000 次。基于以上比较和分析,ALCODE 算法要优于或相似于其他算法。

表 1　五种算法对 10 个问题的寻优结果比较

问题	最优值	状态	方法				
			CHDE[4]	MODE[5]	SR[10]	ECHT[11]	ALCODE
g01	-15.000	best	-15.000	-15.000	-15.000	-15.000	-15.0000
		mean	-14.792	-15.000	-15.000	-15.000	-15.0000
		worst	-12.743	-15.000	-15.000	-15.000	-15.0000
g02	-0.803619	best	-0.8036	-0.8033	-0.803515	-0.803619	-0.803619
		mean	-0.7462	-0.8029	-0.781975	-0.803324	-0.803152
		worst	-0.3022	-0.8024	-0.726288	-0.785182	-0.762160
g04	-30665.539	best	-30665.539	-30665.539	-30665.539	-30665.539	-30665.539
		mean	-30592.154	-30665.539	-30665.539	-30665.539	-30665.539
		worst	-29986.214	-30665.539	-30665.539	-30665.539	-30665.539
g05	5126.4967	best	5126.4967	5126.4967	5126.497	5126.4967	5126.4967
		mean	5218.7230	5126.4967	5128.881	5126.4967	5126.4967
		worst	5502.4103	5126.4967	5142.472	5126.4967	5126.4967
g06	-6961.814	best	-6961.814	-6961.814	-6961.814	-6961.814	-6961.814
		mean	-6367.575	-6961.814	-6875.940	-6961.814	-6961.814
		worst	-2236.950	-6961.814	-6350.262	-6961.814	-6961.814
g07	24.306	best	24.306	24.306	24.307	24.306	24.306
		mean	104.599	24.306	24.374	24.308	24.306
		worst	1120.541	24.306	24.642	24.317	24.306
g08	-0.095825	best	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825
		mean	-0.091292	-0.095825	-0.095825	-0.095825	-0.095825
		worst	-0.027188	-0.095825	-0.095825	-0.095825	-0.095825
g09	680.630	best	680.630	680.630	680.630	680.630	680.630
		mean	692.472	680.630	680.656	680.630	680.630
		worst	839.783	680.630	680.763	680.630	680.630
g10	7049.248	best	7049.248	7049.248	7054.316	7049.248	7049.248
		mean	8442.657	7049.253	7559.192	7049.346	7049.336
		worst	15580.37	7049.248	8835.655	7050.390	7049.581
g11	0.7499	best	0.749	0.7499	0.750	0.7499	0.7499
		mean	0.762	0.7499	0.750	0.7499	0.7499
		worst	0.871	0.7499	0.750	0.7499	0.7499

4　结束语

本文采用增广 Lagrange 罚函数法对约束条件进行处理,提出一种新的约束优化差分进化算法。该算法首先将约束优化问题转换为界约束优化问题;然后采用内外两层形式进行迭代,在内层迭代中利用改进的差分进化算法求解一个非线性的界约束优化子问题,Lagrange 乘子和不同的罚参数在外层迭代中修正。数值实验结果表明,该混合算法是一种便于实现、性能稳定、通用性强的算法。

(下转第 1709 页)

(上接第 1675 页)

参考文献:

[1] STORN R, PRICE K. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11(4): 341-359.

[2] WANG Yong, CAI Zi-xing, ZHANG Qing-fu. Differential evolution with composite trial vector generation strategies and control parameters[J]. IEEE Trans on Evolutionary Computation, 2011, 15(1): 55-66.

[3] 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): 439-441.

[4] MEZURA M E, COELLO C A, MORALES E. Simple feasibility rules and differential evolution for constrained optimization[C]//Lecture Notes in Computer Science. Berlin: Springer, 2004: 707-716.

[5] 刘若辰,焦李成,雷峰,等. 一种新的差分进化约束优化算法[J]. 西安电子科技大学学报, 2011, 38(1): 47-53.

[6] 吴亮红,王耀南,周少武,等. 采用非固定多段映射罚函数的非线性约束优化差分进化算法[J]. 系统工程理论与实践, 2007, 27(3): 128-133.

[7] LIANG Xi-ming. Modified augmented Lagrange multiplier methods for large-scale chemical process optimization[J]. Chinese Journal of Chemical Engineering, 2001, 9(2): 167-172.

[8] 龙文. 求解优化问题的混合进化算法及其应用[D]. 长沙: 中南大学, 2011.

[9] 张铃,张钹. 佳点集遗传算法[J]. 计算机学报, 2001, 24(9): 917-922.

[10] RUNARSSON T P, YAO Xin. Stochastic ranking for constrained evolutionary optimization[J]. IEEE Trans on Evolutionary Computation, 2000, 4(3): 284-294.

[11] MALLIPEDDI R, SUGANTHAN P N. Ensemble of constraint handling techniques[J]. IEEE Trans on Evolutionary Computation, 2010, 10(4): 311-338.