

求解方程组的正交差分进化算法

封全喜^{1,2} 刘三阳¹ 唐国强² 林亮²

(西安电子科技大学理学院 西安 710071)¹ (桂林理工大学理学院 桂林 541004)²

摘要 首先利用代理约束概念和修正极大熵函数,将非线性方程组等价地转化为无约束优化函数;然后引入平均相似度概念,设计自适应正交交叉算子,利用正交设计产生初始种群,并在此基础上提出了自适应正交差分进化算法,用于求解修正极大熵函数;最后用方程组验证了算法的有效性。

关键词 正交设计,差分进化算法,极大熵,方程组

中图分类号 TP202 **文献标识码** A

Orthogonal Differential Evolution Algorithm for Solving System of Equations

FENG Quan-xi^{1,2} LIU San-yang¹ TANG Guo-qiang² LIN Liang²

(School of Science, Xidian University, Xi'an 710071, China)¹

(College of Science, Guilin University of Technology, Guilin 541004, China)²

Abstract First, nonlinear system of equations was transformed into unconstrained optimization problem by using the concept of surrogate constraints and amended maximum entropy function. Then, the concept of average similarity was introduced to design adaptive orthogonal crossover operator, and orthogonal design was used to generate initial population, and on the basis, adaptive orthogonal differential evolution algorithm was proposed for solving the maximum entropy function. Finally, using equations verified the algorithm.

Keywords Orthogonal design, Differential evolution, Maximum entropy, System of equations

1 引言

在工程实践与理论研究中,很多问题都可以转换为方程组,因而求解方程组是数学与工程应用中的重要研究内容。近年来国内外专家学者对非线性方程组的求解问题做了大量的工作,提出了许多有效的算法,主要分为两类:一是经典算法,这些算法对初始点的选取和方程组具有较高的要求,一般是针对某一类型的方程组^[15];二是近年来兴起的智能算法,其广泛地应用于求解方程组,比如文献[4]将粒子群算法与拟牛顿法结合构造求解非线性方程组的混合粒子群算法、文献[7]引入经典算法和遗传算法相结合求解非线性方程组、文献[5]成功地将差分进化算法应用于求解方程组。这些算法初始种群均是随机产生。

同时,文献[4,5]引入凝聚态函数(极大熵)^[1],将非线性方程组转化为光滑的优化函数,再利用智能算法求解,取得了很好的效果。但文献[2]指出凝聚态函数需要在一定的条件下才能收敛,并给出了一种修正的凝聚态函数,该函数在一般情形下就可以收敛。

差分进化算法是 R. Storn 和 K. Price^[13] 于 1995 年提出的一种随机直接搜索算法。该算法以易用性、稳健性和强大的全局寻优能力在多个领域取得成功,特别适合对不可微、非

凸、非光滑函数进行优化。一般差分进化算法的初始化是随机初始化,随意性太强,在解空间中的取样不均匀,容易导致早熟收敛、计算量大和局部搜索能力差等问题。

因此,本文首先用“极大值”约束,将非线性方程组转化不可微的无约束优化问题,然后利用代理约束概念和最大熵原理导出修正凝聚态光滑函数,以此逼近不可微极大值函数,从而将方程组等价地转化为可微的无约束优化函数;然后引入正交设计初始化差分进化算法的初始种群,并引入平均相似度概念,设计自适应正交交叉算子,提出了自适应正交差分进化算法,用于求解修正凝聚态函数。

2 修正极大熵方法

考虑非线性方程组 $F(x) = 0, x \in R^n$, 其中 $F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T, f_i(x): R^n \rightarrow R$ 是连续可微的实值函数。记向量空间 p -范数为 $\|\cdot\|_p$, 于是非线性方程组转化为求解

$$\min_{x \in R^n} \|F(x)\|_p \quad (1)$$

式中,令 $p \rightarrow \infty$, 则问题转化为

$$\min_{x \in R^n} \max_{1 \leq i \leq m} \{|f_i(x)|\} \quad (2)$$

$$\text{记 } f(x) = \max_{1 \leq i \leq m} \{|f_i(x)|\} \quad (3)$$

到稿日期:2011-06-27 返修日期:2011-09-30 本文受国家自然科学基金项目(60974082, 11101101), 广西区教育厅立项项目(2008112 LX85), 陕西省教育厅规划项目(11JK1051)和湖南省社科联课题(1011044B)资助。

封全喜(1980—),男,博士生,讲师,主要研究方向为智能优化、鲁棒优化, E-mail: fqx9904@163.com; 刘三阳(1959—),男,博士,教授,主要研究方向为最优化理论方法及其应用; 唐国强(1971—),男,博士,副教授,主要研究方向为随机优化算法及其理论分析。

则求解原非线性方程组问题转化为求解优化问题 $\min_{x \in R^n} f(x)$, 其中函数 $f(x)$ 不可微。引入修正极大熵函数^[2]

$$G_p(x) = \frac{1}{p} \ln \left(\sum_{i=1}^m e^{pf_i(x)} \right) / m \quad (4)$$

式中, $p > 0$, m 为约束个数。该函数又称为修正凝聚态函数, 是对文献[1]凝聚态函数的改进。当 $p \rightarrow \infty$ 时, $G_p(x)$ 在整个空间上一致逼近函数。

命题 1^[1] 对 $\forall x \in R^n$, $p > 0$ 成立, $f(x) \leq G_p(x) \leq f(x) + \ln m/p$ 。

命题 2^[2] 对 $\forall x \in R^n$, $G_p(x)$ 随正数 p 增加而增加, 且 $f(x) = \lim_{p \rightarrow \infty} G_p(x)$ 。

由上两个命题可知, 可令 p 取足够大, 然后用可微函数 $G_p(x)$ 代替不可微函数 $f(x)$, 从而将原不可微优化问题 $\min_{x \in R^n} f(x)$ 转化为可微优化问题 $\min_{x \in R^n} G_p(x)$ 。根据命题 1 和命题 2, 非线性方程组问题可以等价地转化为无约束优化问题 $\min_{x \in R^n} G_p(x)$ 。

3 自适应正交差分进化算法设计

文献[6]指出, 在群体进化过程中, 使用固定的正交表安排父代个体的交叉操作, 由于因素分割位置的随机性, 当参与交叉的父代个体之间的部分基因间的距离很近时, 容易产生很多冗余个体, 导致子代群体间的多样性较差, 特别是种群进化后期尤为明显。为了使正交交叉产生的子代个体能均匀分布在解空间中, 文献[6]引入相似度概念, 设计了自适应正交交叉算子, 与遗传算法相结合, 提出了自适应正交遗传算法, 取得了很好的收敛效果。该算子给相似度设定固定的下限 Δ , 从满足下限的基因中选择一部分作为因素的分割位置点。然而算法设计中, 选取合适的 Δ 有一定困难。当 Δ 较小时, 种群进化早期个体间的相似度较低, 几乎所有的基因距离均大于 Δ , 运行效果与一般正交交叉算子近似; 当 Δ 较大时, 在种群进化的后期, 个体间的相似度极高, 如果问题解的精度要求较高, 种群中可能出现部分个体所有基因间的距离均小于 Δ , 因素分割位置点集为空集, 从而导致交叉无法进行。

因此, 在相似度概念的基础上, 引入平均相似度概念, 设计了一种新的自适应正交交叉算子。该算子根据平均相似度动态地调节下限 Δ , 从满足相似度要求的基因集合中随机选取基因作为因素分割位置点, 这样既能避免进化早期部分相似度极高的基因进入因素分割点集, 又能避免进化后期出现因素分割位置点集为空集的情形, 同时保持了算法的随机特性。

3.1 正交试验设计^[10]

设一般试验系统有 J 个因素, 每个因素有 Q 个水平, 全面组合试验需做 Q^J 组试验。当 Q 和 J 很大时, 不可能完成。正交设计利用正交表 $L_M(Q^J)$ 安排 M 次试验, 能较好地从一个解空间中抽出 M 个具有代表性的样本点。

例如: 正交表 $L_9(3^4)$, 对 4 个因素, 3 个水平, 依据正交表只需做 9 次试验, 但若进行全面组合试验, 则需 $3^4 = 81$ 次试验, 大大减少了试验次数。

记 $L_M(Q^F) = [a_{ij}]_{M \times N}$, a_{ij} 为 i 个水平 j 个因素, $a_{ij} \in \{1, 2, \dots, Q\}$, Q 和 J 根据具体情况选取, 一般有关系

$$M \leq \frac{Q^F - 1}{Q - 1} \quad (5)$$

3.2 自适应正交交叉算子

设 $x_1 = (x_{11}, x_{12}, \dots, x_{1n})$, $x_2 = (x_{21}, x_{22}, \dots, x_{2n})$ 为参与交叉操作的一对父个体, x_1, x_2 确定的可行空间为 $[l_{low}, u_{up}]$, 其中

$$\begin{cases} l_{low} = [\min(x_{11}, x_{21}), \min(x_{12}, x_{22}), \dots, \min(x_{1n}, x_{2n})] \\ u_{up} = [\max(x_{11}, x_{21}), \max(x_{12}, x_{22}), \dots, \max(x_{1n}, x_{2n})] \end{cases} \quad (6)$$

定义 1^[6] 令 $\delta_i = |x_{1i} - x_{2i}|$, 其中 $i = 1, 2, \dots, n$, 称 δ_i 为 x_1, x_2 的第 i 维相似度。

定义 2 令 $\delta = (\delta_1 + \delta_2 + \dots + \delta_n)/n$, 称 δ 为 x_1, x_2 的平均相似度。

记 $B = \{i | \delta_i > \alpha \cdot \delta, i = 1, 2, \dots, n\}$, 称 B 为因素分割位置点集, 其中 α 为常数。该集合用于调节满足相似度要求的位置点的个数(可根据实际情况调节), 以防止相似度极高的基因作为因素分割位置点, 导致种群多样性较差。 α 取值越大, 集合 B 中的元素个数越少, 能够选取作为因素分割的基因个数越少, 分割越均匀, 但随机性相对较差; 反之分割均匀性较差, 随机性较好。一般取值 0.55 或 0.15。

例如: 设参与交叉操作的两个父代个体为 $x_1 = (2, 4, 3, 3)$ 和 $x_2 = (1, 3, 3, 4)$, 下限 $\Delta = 0.05$, 似度 $\delta_1 = \delta_2 = 1, \delta_3 = 0.1$, 显然均大于 0.05。不妨设因素个数为 2, 可能出现分割

$$\begin{cases} x_1 = (2, 4, \vdots, 3, 3) \\ x_2 = (1, 3, \vdots, 3, 4) \end{cases} \quad (7)$$

而平均近似相似度 $\delta = 0.7$, 不妨取 $\alpha = 0.55$, 有 $\delta_1, \delta_2 > \alpha \cdot \delta$, 新设计的自适应正交交叉算子相应的分割为

$$\begin{cases} x_1 = (2, \vdots, 4, 3, 3) \\ x_2 = (1, \vdots, 3, 3, 4) \end{cases} \quad (8)$$

显然后一种分割式(8)的效果比前一种分割式(7)好。

自适应正交交叉算子:

步骤 1 设 $x_1 = (x_{11}, x_{12}, \dots, x_{1n})$, $x_2 = (x_{21}, x_{22}, \dots, x_{2n})$ 为参与交叉操作的一对父个体, 其对应的解可行空间为 $[l_{low}, u_{up}]$, 利用公式

$$\beta_j = \begin{cases} \min(x_{1i}, x_{2i}), & j=1 \\ \min(x_{1i}, x_{2i}) + (j-1)(|x_{1i} - x_{2i}|/(Q-1)), & 2 \leq j \leq Q-1 \\ \max(x_{1i}, x_{2i}), & j=Q \end{cases} \quad (9)$$

将第 $i(i = 1, 2, \dots, n)$ 维离散化为 Q 个水平 $\beta_{i1}, \beta_{i2}, \dots, \beta_{iQ}, i = 1, 2, \dots, n$ 。

步骤 2 计算相似度、平均相似度及其分割位置点集合 B , 设解向量分割为 J 个因素。记集合 B 的元素个数为 $|B|$ 。

1) 若 $|B| \leq J-1$, 则令 $J-1 = |B|$, 即分割为 $|B|$ 个因素, 集合 B 的元素均为因素的分割位置点;

2) 若 $|B| > J-1$, 则从 B 中随机选取 $J-1$ 个数为对应的分割位置点。

记位置分割点为 $1 \leq k_1 \leq k_2 \leq \dots \leq k_{J-1}$, 则解向量分割为

$$\begin{cases} f_1 = (x_1, \dots, x_{k_1}) \\ f_2 = (x_{k_1+1}, \dots, x_{k_2}) \\ \dots \\ f_F = (x_{k_{J-1}+1}, \dots, x_n) \end{cases} \quad (10)$$

步骤3 根据正交表 $L_M(Q')$ 和因素分割产生 M 个子代个体。

3.3 种群初始化

在优化问题求解前,没有任何关于全局最优解的位置,因此初始种群均匀地分布在可行解空间,有助于算法逃避欺骗问题,有效地寻找问题的全局最优解。

正交试验设计是研究多因素、多水平的一种试验方法,根据正交阵,将很少数量的组合均匀地分布在整个解空间的所有可能组合中,是一种较好地产生初始种群的方法。为了算法更好地收敛,当解空间很大时,初始种群最好是从更多的个体中选取。正交阵 $L_M(Q')$ 中的 M 依赖于 Q 和 J ,而 Q 和 J 不能随意增加。为了克服这个问题,通常将解空间分成 S 个子空间,然后再分布在其上利用正交阵产生初始种群,具体分解方法见文献[10]。

3.4 自适应正交差分进化算法

本文所提自适应正交交叉差分进化算法是在基于正交交叉差分进化算法^[1]的基础上,引入正交种群初始化和自适应正交交叉算子进行改进。

算法步骤如下:

(1) 种群初始化:设置算法参数,将可行解空间分割成 S 个子空间,在每个子空间上利用正交阵产生子种群,然后从 S 个子种群的所有个体中选取 NP 个个体组成初始种群。

(2) 随机产生整数 $k \in \{1, 2, \dots, NP\}$ 。

(3) 变异操作:对于种群中的每个个体,首先从种群中随机选择3个个体 $x_{r_1}, x_{r_2}, x_{r_3}$ 且 $r_1 \neq r_2 \neq r_3$, 则

$$v_i = x_{r_1} + \text{rand} * (x_{r_2} - x_{r_3}) \quad (11)$$

$$v_i = x_{r_1} + F * (x_{r_2} - x_{r_3}) \quad (12)$$

(4) 交叉操作:对第 k 个个体,利用式(11)生成的 v_i 与 x_k 进行自适应正交交叉运算,选择最优个体作为尝试解 u_k ;其他个体使用式(12)生成的 v_i 再根据式(13)产生尝试解 u_k (其中 P_c 为交叉概率, j_{rand} 为一随机数)。

$$u_{ij} = \begin{cases} v_{ij}, & \text{rand} < P_c \text{ 或 } j = j_{\text{rand}} \\ x_{ij}, & \text{否则} \end{cases} \quad (13)$$

(5) 选择操作:

$$x_i' = \begin{cases} u_i, & f(u_i) < f(x_i) \\ x_i, & \text{否则} \end{cases} \quad (14)$$

式中, $f(x_i)$ 为解 x_i 的函数值, x_i' 为新种群中的个体。

(6) 终止条件:最大迭代次数。

4 数值实验及结果

为了验证本文所提算法的性能,从文献[4, 7-9, 12, 14]中选择部分具有代表性的非线性方程组进行仿真模拟,并将其有效性与其他算法的进行比较。在试验中,本文算法的参数设置如下:参数 $p=100$, $J=2$,初始化的水平数 $Q=11$,正交交叉的水平数 $Q=3$,所有问题的初始种群均为30,初始化的子空间数均为2,迭代代数分别为300, 1000, 600, 1000,交叉概率 $CR=0.8$,尺度标量 $F=0.8$,4个问题的仿真次数分别是50, 100, 50, 50。同时为了更好地比较算法性能,对问题1、2,根据求得的解计算对应方程组值的绝对值的最大值,即 $f(x) = \max\{f_i(x)\}$,并比较它们的精度。

本文首先利用上述参数设置,用下面4个方程组做50次

仿真,分析参数 α 对方程根收敛性的影响。设参数 α 可能取值为0.15, 0.35, 0.55, 0.75, 0.95,通过仿真模拟发现当参数 α 取0.15和0.55时,算法求根效果最佳,因此本文中取 $\alpha=0.55$ 。

问题1

$$\begin{cases} f_1(x) = (x_1 - 5x_2)^2 + 40\sin^2(10x_3) = 0 \\ f_2(x) = (x_2 - 2x_3)^2 + 40\sin^2(10x_1) = 0 \\ f_3(x) = (3x_1 + x_3)^2 + 40\sin^2(10x_2) = 0 \end{cases} \quad (15)$$

问题2^[7, 9, 12] 解方程组 $f(x) = x$, 方程组的解为 $(-1, 1, -1, 1, -1, 1)^T$ 。

$$\begin{cases} f_1(x) = -0.75 - x_2^2 x_4 x_6 / 4 \\ f_2(x) = -0.405e^{1+x_1 x_2} + 1.405 \\ f_3(x) = x_4 x_6 / 2 - 1.5 \\ f_4(x) = 0.605e^{1-x_3^2} + 0.395 \\ f_5(x) = x_2 x_6 / 2 - 1.5 \\ f_6(x) = x_1 x_5 \end{cases} \quad (16)$$

表1为本算法求解问题1、2的结果。由表1知,本算法求解问题1的成功率稍低于文献[4],但50次仿真所得平均解的对应方程组值的最大值即 $f(x)$ 值比文献(说明:文献中的方程组最大值是根据文献中的解而求得)中的精度高很多。问题2的求解成功率与文献一样,均求解成功,本算法所得平均解的方程最大值的精度也远好于文献所求得平均解对应的方程最大值的精度。图1、图2为本算法求解问题1、2的迭代收敛图。

表1 问题1、2的收敛情况比较

问题	仿真次数	本算法成功率	文献成功率	本算法最大绝对值	文献最大绝对值
问题1	50	0.98	1	2.15E-015	1.00E-005
问题2	100	1	1	8.91E-010	1.00E-005

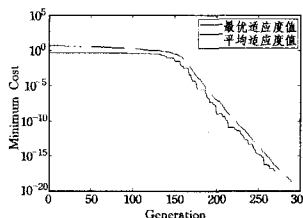


图1 问题1的迭代收敛图

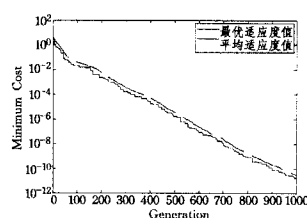


图2 问题2的迭代收敛图

问题3^[8, 14] 解稀疏非线性方程组,方程组的维数为10,解空间为 $[-2, 2.0]^{10}$ 。

$$\begin{cases} f_1(x) = x_1 - 0.25428722 - 0.18324757x_3x_4x_9 \\ f_2(x) = x_2 - 0.37842197 - 0.16275499x_1x_6x_{10} \\ f_3(x) = x_3 - 0.27162577 - 0.16955071x_1x_2x_{10} \\ f_4(x) = x_4 - 0.19807914 - 0.15585316x_1x_6x_7 \\ f_5(x) = x_5 - 0.44166728 - 0.19950920x_3x_6x_7 \\ f_6(x) = x_6 - 0.14654113 - 0.18922793x_5x_8x_{10} \\ f_7(x) = x_7 - 0.42937161 - 0.21180486x_2x_5x_8 \\ f_8(x) = x_8 - 0.07056438 - 0.17081208x_1x_6x_7 \\ f_9(x) = x_9 - 0.34504906 - 0.19612740x_6x_8x_{10} \\ f_{10}(x) = x_{10} - 0.42651102 - 0.21466544x_1x_4x_8 \end{cases} \quad (17)$$

问题4^[8, 14] 解稀疏非线性方程组,方程组的维数为10,解空间为 $[-2, 2.0]^{10}$ 。

(下转第194页)

[11] Kamal A H M, Zhu X Q, Narayanan R. Gene Selection for Microarray Expression Data with Imbalanced Sample Distributions [C] // Proceedings of 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing. Shanghai: IEEE Press, 2009, 3-9

[12] Breiman L. Random Forest[J]. Machine Learning, 2001, 45: 5-32

[13] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: Synthetic minority over-sampling technique[J]. Journal of Artificial Intelligence Research, 2002, 16: 321-357

[14] Han H, Wang W Y, Mao B H. Borderline-smote: a new over-sampling method in imbalanced data sets learning[J]. Lecture Notes in Computer Science, 2005, 3644: 878-887

[15] Liu A, Ghosh J, Martin C. Generative oversampling for mining

imbalanced datasets[C] // Proceedings of the Seventh International Conference on Data Mining. Las Vegas, Nevada, USA: CSREA Press, 2007: 66-72

[16] Uriarte R D, Andres S A D. Gene selection and classification of microarray data using random forest[J]. BMC Bioinformatics, 2006, 7: 3

[17] Alon U, Barkai N, Notterman D A, et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by Oligonucleotide array[J]. Proceedings of the National Academy of Sciences, 1999, 96(12): 6745-6750

[18] Shipp M A, Ross K N, Tamayo P, et al. Diffuse Large B-Cell Lymphoma Outcome Prediction by Gene Expression Profiling and Supervised Machine Learning[J]. Nature Medicine, 2002, 8(1): 68-74

(上接第 189 页)

$$\begin{cases} f_1(x) = x_1^2 - 0.25428722 - 0.18324757x_3^2x_4^2x_5^2 \\ f_2(x) = x_2^2 - 0.37842197 - 0.16275499x_1^2x_6^2x_{10}^2 \\ f_3(x) = x_3^2 - 0.27162577 - 0.16955071x_1^2x_2^2x_{10}^2 \\ f_4(x) = x_4^2 - 0.19807914 - 0.15585316x_1^2x_6^2x_7^2 \\ f_5(x) = x_5^2 - 0.44166728 - 0.19950920x_3^2x_6^2x_7^2 \\ f_6(x) = x_6^2 - 0.14654113 - 0.18922793x_3^2x_8^2x_{10}^2 \\ f_7(x) = x_7^2 - 0.42937161 - 0.21180486x_2^2x_8^2x_9^2 \\ f_8(x) = x_8^2 - 0.07056438 - 0.17081208x_1^2x_6^2x_7^2 \\ f_9(x) = x_9^2 - 0.34504906 - 0.19612740x_3^2x_8^2x_{10}^2 \\ f_{10}(x) = x_{10}^2 - 0.42651102 - 0.21466544x_1^2x_4^2x_8^2 \end{cases} \quad (18)$$

由表 2 可以看出,本算法能 100% 求出方程(3)、(4)的解,50 次解对应的最大方程值的平均值精度均很高。由此可以看出本文所提的自适应正交差分进化算法是一种求解非线性方程组的有效算法。

表 2 问题 3、4 的收敛情况

问题	仿真次数	本算法成功率	本算法最大绝对值	平均求解时间
问题 3	50	1	1.3197e-008	11.7
问题 4	50	1	2.3193e-008	17.6

结束语 本文分析了极大熵收敛性对函数形式具有一定的依赖性,而修正极大熵(凝聚态函数)收敛性不依赖于函数形式的特点,通过代理约束将方程组转化为不可微的无约束优化问题,然后利用修正极大熵将其转化为可微的无约束优化问题。同时,针对文献[6]自适应正交交叉算子的不足之处,设计了新的自适应正交交叉算子,该算子不受问题收敛精度的影响,自适应调节相似度的下界,以避免个别相似度极大的基因作为因素的分割位置而影响算法的收敛速度,从而提出了自适应正交交叉差分进化算法,用于求解方程组对应的修正极大熵函数。通过求解 4 个不同特点的方程组,并比较其他算法求解结果,表明了该算法求解非线性方程组的有效性。

参 考 文 献

[1] 李兴斯. 解非线性规划的凝聚函数法[J]. 中国科学: A 辑, 1991,

12: 1283-1288

[2] 颜世建. 关于解非线性规划的一个修正凝聚函数法的注记[J]. 南京师大学报, 2002, 25(2): 93-96

[3] Yang Yan, Zhou Yong-quan, Gong Qiao-qiao. Hybrid Artificial Glowworm Swarm Optimization Algorithm for Solving System of Nonlinear Equations [J]. Journal of Computational Information Systems, 2010, 6(10): 3431-3438

[4] 雍龙泉. 基于极大熵微粒群混合算法的非线性方程组求解[J]. 海军工程大学学报, 2009, 21(3): 28-32

[5] 陈海霞, 杨铁贵. 基于极大熵差分进化混合算法求解非线性方程组[J]. 计算机应用研究, 2010, 27(6): 2028-2030

[6] 江中央, 蔡自兴, 王勇. 求解全局优化问题的混合自适应正交遗传算法[J]. 软件学报, 2010, 21(6): 1296-1307

[7] 田巧玉, 古钟壁, 周新志. 基于混合遗传算法求解非线性方程组[J]. 计算机技术与发展, 2007, 17(3): 10-12

[8] Hentenryck P V, Mcallester D, Kapur D. Solving polynomial systems using a branch and prune approach [J]. SIAM J. NUMER. ANAL., 1997, 34(2): 797-827

[9] Sonia K. Extension of the Lanczos and CGS methods to systems of nonlinear equations [J]. Journal of Computational and Applied Mathematics, 1996, 69: 181-190

[10] Leung Y-W, Wang Yu-ping. An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization [J]. IEEE Transactions on Evolutionary Computation, 2001, 5(1): 41-53

[11] Wang Yong, Cai Zi-xing, Zhang Qing-fu. Enhancing the Exploration Ability of Differential Evolution through Orthogonal Crossover [R]. working report, 2011

[12] 罗亚中, 袁端才, 唐国金. 求解非线性方程组的混合遗传算法[J]. 计算力学学报, 2005, 22(1): 109-114

[13] Storn R, Price K. Differential Evolution-A Simple and Efficient Heuristic for global Optimization over Continuous Spaces [J]. Journal of Global Optimization, 1997, 11: 341-359

[14] 李耀辉, 薛继伟, 冯勇. 基于预处理和区间计算的非线性方程组实根求解[J]. 四川大学学报, 2005, 36(5): 86-93

[15] 段治健, 杨永, 马欣荣, 等. 求解带状线性方程组的一种并行算法[J]. 计算机科学, 2010, 37(3): 242