

## 线性约束非线性函数全局优化算法的研究

张贵军<sup>1</sup>, 俞立<sup>1</sup>, 吴惕华<sup>2</sup>

(1. 浙江工业大学 信息工程学院, 浙江 杭州 310032; 2. 上海交通大学 电子信息与电气工程学院, 上海 200030)

**摘要:** 提出了一种适于处理线性约束条件下非线性规划问题的  $\lambda$  编码稳态遗传算法( $\lambda$ SSGA). 首先对线性可行域进行凸分析后将原优化问题 I 转化为一个仅包含可行域极点信息的等价问题 II. 问题 II 具有小边界的约束条件, 通过采用特定的凸交叉算子、交换变异算子和倒位算子可以保证算法在遗传操作的过程中不会产生无效的编码, 而且能在概率意义上保证  $\lambda$  编码模式在整个可行解空间上充分可达. 其次从理论上推导出了得到线性可行区域全部极点的方法, 证明了问题 I 和问题 II 的等价性. 仿真结果表明  $\lambda$ SSGA 算法在具有较快的收敛速度和精度的同时, 还可以有效地维持群体的多样性, 得到问题全局的最优解.

**关键词:**  $\lambda$  极点编码; 稳态遗传算法; 非线性规划; 线性约束; 全局优化; 凸交叉算子

**中图分类号:** TP301.6

**文献标识码:** A

## Global optimal algorithm for nonlinear programming problems subjected to linear constraints

ZHANG Gui-jun<sup>1</sup>, YU Li<sup>1</sup>, WU Ti-hua<sup>2</sup>

(1. College of Information Engineering, Zhejiang University of Technology, Hangzhou Zhejiang 310032, China;

2. School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200030, China)

**Abstract:** A global optimal algorithm (the  $\lambda$  steady state genetic algorithm,  $\lambda$ SSGA) is presented to solve nonlinear programming problems, which are subjected to linear constraints. By convex analyzing, the primal optimal problem I can be converted to an equivalent problem II, in which only the convex extremes of feasible space are included. The problem II has simpler constraints than problem I. In the evolving of the  $\lambda$ SSGA algorithm, the invalid genetic operating can be avoided by using convex crossover operator, swap mutate operator and inverse operation. It can also explore the entire feasible space in the sense of probability. A method is derived to get all extremes (BFS) of linear constraints, and the equivalence of problem I and problem II is also proven. Finally, the simulation analysis shows that the algorithm not only has fast convergence speed and high precision of solution, but also can maintain the diversity of the population and reach a global optimum of non-concave objective function.

**Key words:**  $\lambda$  extremes encoding; steady state genetic algorithms; non-linear programming; linear constraints; global optimization; convex crossover operator

### 1 引言(Introduction)

实际的优化问题大都可归结为非线性规划模型来求解, 一些传统的基于梯度的非线性优化方法, 如准牛顿法(Quasi-Newton)、广义简约梯度(GRG)等算法通常是寻找一个可行、下降方向, 然后在此方向进行一系列的数值迭代来得到模型的局部最优解<sup>[1,2]</sup>. 这些基于梯度的算法不但要求非线性模型的目标函数和约束条件一级连续可微, 而且对一些非线性度很高的函数, 在实现中采用有限差分计算目标函数的梯度和 Jacobian 矩阵时, 还可能会出现

一些数值计算的病态问题<sup>[3]</sup>.

对于非凸的非线性规划模型, 应用上述梯度算法求解需要解决以下三个方面的问题: 1) 如何确定模型可行空间中局部最优解的个数; 2) 如何确定最初的迭代起点; 3) 如何增强算法在可行空间中的搜索能力. 由于这些算法本质上是一种局部的搜索算法, 为了得到问题的全局最优解, 将不得不选择不同的初始点多次运行算法, 然后从中选择最佳点作为模型的全局最优解<sup>[2,4]</sup>, 这样无论是从理论上还是在具体实现中, 我们都无法保证解的全局最

优性.

遗传算法(GAs)是 Holland 教授提出的一种并行的全局优化算法<sup>[5]</sup>. 基于模式定理的遗传算法利用遗传算子来产生群体中的最优个体, 可适用于非连续的离散变量模型; 此外, 遗传算法还具有很强的空间搜索能力<sup>[6]</sup>, 可以在理论上保证解的全局性.

遗传算法一般来说适于处理仅有小边界约束条件的非线性规划问题, 因为这样可以使算法有较小规模的搜索空间, 经过选择、交叉、变异等遗传操作后可产生合法的解, 从而在概率意义上保证解空间的可达性. 总的来说, 变量边界范围和约束条件两方面的因素制约了遗传算法在数值函数优化计算方面的应用. 遗传算法的搜索空间随变量边界范围的增大呈指数增长, 当变量的边界范围很大时, 搜索空间的增大将使算法无法保证在一定运行时间内得到非线性规划问题的满意解, 而约束条件的存在则使遗传操作产生大量无效的解空间, 减弱染色体的遗传性状, 降低产生更好解的能力. 这两方面的因素都将极大增加算法的计算时间, 导致遗传算法后期收敛速度减慢.

针对上述优化算法存在问题, 作者提出了一种求解凸(线形)约束条件下非线性规划问题的全局优化算法—— $\lambda$  编码稳态遗传算法( $\lambda$ SSGA). SSGA 是一种改进的遗传算法, 在每一代进化过程中只是在概率意义上替换指定替换数目或替换概率的个体, 而不是群体中的所有个体.  $\lambda$ SSGA 算法首先将原优化问题 I 转化为包含凸极点优化问题 II. 可以证明, 二者在数学上是等价的. 然后对问题 II 变量  $\lambda$  (即凸区域极点系数)进行编码, 而不是对原变量进行编码. 这样不但可大大降低遗传算法的搜索空间(0~1 之间), 而且通过设计特定的初始化算子、交叉算法和变异算子还可以避免在遗传操作中产生无效的解. 仿真结果证明了  $\lambda$  编码遗传算法比一般编码的遗传算法有更好的收敛速度和精度.

## 2 理论基础(Theory basis)

考虑下述线性约束条件下优化问题 I 的一般形式:

$$\max f(\mathbf{x}), \quad (1)$$

$$\text{s.t. } \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0. \quad (2)$$

其中  $f$  是任意非线性函数,  $\mathbf{A}$  是  $m \times n$  维矩阵,  $\mathbf{x}$  是  $n$  维决策列向量,  $\mathbf{b}$  是  $m$  维列向量. 为了计算上需要, 不失一般性, 假设  $\mathbf{b}$  的每个分量是非负数. 下面分析如何得到给定线性凸集的全部极点.

### 2.1 凸集极点(Extreme points of convex set)

为分析需要, 先引入下列两个定理:

**定理 1**  $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0$  的可行域是凸集<sup>[1]</sup>.

**定理 2** 令  $K = \{\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0\}$ ,  $\mathbf{A}$  是  $m \times n$  维矩阵,  $\text{rank}(\mathbf{A}) = m$ , 则  $K$  的极点集与  $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0$  的基本可行解等价<sup>[1]</sup>.

由定理 1 可知, 式(2)的可行域是凸集, 再由定理 2 可知, 其极点集可通过相应代数形式的基本可行解 BFS 来得到. 记

$$\mathbf{A} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n).$$

将  $\mathbf{A}$  分解成  $(\mathbf{B}, \mathbf{N})$  (可能经过列变换), 使得其中  $\mathbf{B}$  是基矩阵,  $\mathbf{N}$  是非基矩阵, 设

$$\mathbf{x}^{(0)} = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{b} \\ 0 \end{bmatrix}$$

是基本可行解, 现在分析如何从该基本可行解出发, 找到其它所有的基本可行解.

设  $\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix}$  是任意一个可行解, 则由式(2)得到

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N. \quad (3)$$

在原来的  $n - m$  个非基变量中, 使得  $n - m - 1$  个变量仍然取 0 值, 而令一个非基变量  $x_k$  增大, 即取正值, 从而实现我们的目的.  $x_k$  由 0 变为正数后得到方程组(2)的解:

$$\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{p}_k x_k = \bar{\mathbf{b}} - \mathbf{y}_k x_k, \quad (4)$$

即

$$\mathbf{x}_B = \begin{bmatrix} x_{B_1} \\ x_{B_2} \\ \vdots \\ x_{B_m} \end{bmatrix} = \begin{bmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \vdots \\ \bar{b}_m \end{bmatrix} = \begin{bmatrix} y_{1k} \\ y_{2k} \\ \vdots \\ y_{mk} \end{bmatrix} x_k, \quad (5)$$

$$\mathbf{x}_N = (0, \dots, 0, x_k, 0, \dots, 0)^T.$$

一般来说, 当  $\mathbf{A}$  是  $m \times n$  矩阵,  $\text{rank}(\mathbf{A}) = m$  时, 基本可行解的个数不会超过  $C_n^m$ , 因此应适当选取下标  $k$ , 从而保证可以得到式(2)全部的基本可行解, 为此, 定义了欧氏范数意义下的度量标准:

$$D_{k \min} = \min\{c_B \mathbf{B}^{-1} \mathbf{p}_j - 1 \mid c_B \mathbf{B}^{-1} \mathbf{p}_j - 1 > 0, j \in \mathbb{R}\}, \quad (6)$$

$$D_{k \max} = \max\{c_B \mathbf{B}^{-1} \mathbf{p}_j - 1 \mid c_B \mathbf{B}^{-1} \mathbf{p}_j - 1 > 0, j \in \mathbb{R}\}. \quad (7)$$

其中  $c_B$  为单位行向量,  $\mathbb{R}$  为非基变量下标集.

式(6)可以保证极点向度量意义上逐渐减小的方向进基, 直至达到度量意义下的最小极点; 式(7)可以保证极点向度量意义上逐渐增加的方向进基, 直至达到度量意义下的最大极点. 这样初始基下标

$k$  先按(6)式选取,直至  $D_{k \min} \leq 0$ ;然后初始基重新按(7)式选取,直至  $D_{k \max} \geq 0$ . 这样就可以得到度量意义下凸集极点的所有基本可行解.

$x_k$  的取值必须大于0,当  $y_k < 0$  时,  $x_k$  可以取任意正值,这时该凸集是无界区域;当  $y_k > 0$  时,受到可行性的限制,它不能无限增大. 对某个  $i$ , 当  $y_{ik} \leq 0$  时,  $x_k$  取任何正值,总成立  $x_{B_i} \geq 0$ , 而当  $y_{ik} > 0$  时,为保证

$$x_{B_i} = \bar{b}_i - y_{ik} \geq 0, \quad (8)$$

就必须取值为

$$x_k = \min \left\{ \frac{\bar{b}_i}{y_{ik}} \mid y_{ik} > 0 \right\} = \frac{\bar{b}_r}{y_{rk}}. \quad (9)$$

$x_k$  取值  $\bar{b}_r/y_{rk}$  后,原来的基变量  $x_{B_r} = 0$ , 得到了一个新的可行解:

$$x = (x_{B_1}, \dots, x_{B_{r-1}}, 0, x_{B_{r+1}}, 0, \dots, x_k, 0, \dots, 0)^T.$$

这个解一定是基本可行解,这是因为原来的基  $B = (p_{B_1}, p_{B_2}, \dots, p_{B_r}, \dots, p_{B_m})$  中的  $m$  个列是线性无关,其中不包括  $p_k$ , 由于  $y_k = B^{-1}p_k$ , 故

$$p_k = By_k = \sum_{i=1}^m y_{ik} p_{B_i}. \quad (10)$$

即  $p_k$  是向量组  $p_{B_1}, \dots, p_{B_r}, \dots, p_{B_m}$  的线性组合,且系数  $y_{rk} \neq 0$ . 因此用  $p_k$  取代  $p_{B_r}$  后,得到的向量组  $p_{B_1}, \dots, p_{B_k}, \dots, p_{B_m}$  也是线性无关的. 因此新的可行解  $x$  的正分量对应列线性无关,故  $x$  是基本可行解.

通过上述分析可知,分别按式(6),(7)确定出基变量,按式(9)确定进基变量,或者得出式(2)无界的结论,或者得到全部的基本可行解,即凸区域所有的极点.

## 2.2 优化问题 II (Optimization problem II)

**定理 3<sup>[7]</sup>** 设  $S = \{Ax = b, x \geq 0\}$  为非空多面集,则有

1) 极点集非空,且存在有限个极点  $x^{(1)}, \dots, x^{(k)}$ .

2) 极方向集合为空集的充要条件是  $S$  有界. 若  $S$  无界,则存在有限个极方向  $d^{(1)}, \dots, d^{(l)}$ .

3)  $x \in S$  的充要条件是

$$x = \sum_{j=1}^k \lambda_j x^{(j)} + \sum_{j=1}^l \mu_j d^{(j)}, \quad (11)$$

$$\sum_{j=1}^k \lambda_j = 1, \quad (12)$$

其中  $\lambda_j \geq 0, j = 1, \dots, k; \mu_j \geq 0, j = 1, \dots, l$ .

由定理 3 可知道,若已知线性凸区域极点集,则

该区域里的所有点都可以用区域顶点的凸组合表示. 设约束条件式(2)有界,通过上述方法得到极点集为

$$(x^1, x^2, \dots, x^l).$$

则可行域中任意一点就可以表示成

$$x = \lambda_1 x^1 + \lambda_2 x^2 + \dots + \lambda_l x^l. \quad (13)$$

代入式(1),原优化问题可转化为优化问题 II:

$$\max f(\lambda), \quad (14)$$

$$\sum_{j=1}^l \lambda_j = 1, \lambda_j \geq 0. \quad (15)$$

显然,原优化问题 I 和优化问题 II 是等价的,对  $\lambda$  进行编码,求解优化问题 II,然后利用式(13)就可以得到原优化问题 I 的最优解. 优化问题 II 不但具有很小的搜索空间,而且通过适当的选择初始化算子、交叉算子以及变异算子还可以避免在遗传操作中产生无效的编码空间,这样就可大大地提高遗传算法的收敛速度和精度,在较短的时间内可得到原优化问题的全局满意解.

## 3 $\lambda$ 稳态遗传算法 ( $\lambda$ SSGA)

Goldberg 总结出了基本的遗传算法 SGA, SGA 只使用选择算子、交叉算子和变异算子这三种基本遗传算子,在遗传进化操作过程中,需要在概率意义上替换群体 (Population) 中所有的个体 (Individuals), 这样 SGA 算法就不可避免涉及到大量的交叉和遗传操作,增加算法的运行时间. 虽然交叉和变异增大会增加群体的多样性,但同时也降低了遗传算法的离线性能 (收敛性能) 和在线性能 (动态性能).

为了使遗传算法具有更好的在线和离线性能,本文中设计了一种最优替代策略稳定状态遗传算法 SSGA. 这里所说的稳态是指每一代在进化过程中只是在概率意义上替换指定替换数目或替换概率的个体,并不是群体中的所有个体,可在一定的程度上加快算法的收敛速度.

对于优化问题 II,如果采用常规的初始化算子、交叉算子及变异算子将会产生大量的无效编码空间 (解空间),虽然采用惩罚函数方法可以将约束函数转化为无约束的问题,但是运算结果证明虽然经过很长时间的进化,该方法也得不到很好的结果<sup>[8]</sup>. 通过对试验数据的分析,可以看出由于这些约束条件进一步增加了无效染色体的数目,故使收敛速度减慢,更重要的是在交叉和变异过程中由于产生太多的无效染色体而使群体的有效进化模式减少,从而

不能使个体进化到最优的状态.故必须设计合理的初始化算子、交叉算子和变异算子,使之成为“有效”的,在这里“有效”指的是经遗传算子操作后产生的染色体编码所对应的解空间是满足方程约束条件的.

适当的初始化算子是保证后续遗传操作产生有效编码的前提,因为通过定制适当的交叉和变异算子之后,算法就不会产生无效染色体,这样“实现”就不必在每次遗传操作之后进行编码有效性判断,从而可加快算法的计算速度.初始化算子产生的解既要求满足式(15),又要保证群体中的个体尽可能均匀分布在解空间中.这里采用的初始化策略是:假设凸集有  $L$  个极点,随机选取  $L-2$  个  $0 \sim 1$  之间的随机数,之后应用快速排序算法<sup>[9]</sup>对其进行升序排列,相应的取值为对应的间距,这样可生成满足上述条件的初始解.

变异算子采用交换操作和倒位操作来实现.交换操作为随机交换染色体基因座上两个基因;倒位操作的具体过程是:1) 在个体编码串中随机指定两个基因座之后的位置为倒位点;2) 以倒位概率  $P = 0.5$  颠倒这两个倒位点之间的基因排列顺序.倒位操作改变了个体编码串的部分基因排列顺序,其目的主要是为了能够使遗传算法更有利于保存和生成较好的模式,因为有时对形成一种较好模式有积极意义的基因在个体的编码串中隔离的较远,而使用倒位算子就有可能使它们逐渐接近,从而形成更好的个体.上述变异操作都不会产生非法的染色体.

基于传统的单点或两点交叉都可能产生大量的非法染色体,故本文中设计了一种凸组合(线性)交叉算子来实现交叉操作,每次交叉操作只产生一个子个体.由于式(15)是一个凸集,设  $S$  为  $n$  维空间中一个凸集,由凸集的定义可知对  $S$  中任意两点  $x^{(1)}$ ,  $x^{(2)}$  及任一实数  $\alpha \in [0, 1]$  都有

$$\alpha x^{(1)} + (1 - \alpha)x^{(2)} \in S. \quad (16)$$

这样随机产生一个  $[0, 1]$  之间随机数,按式(16)可以得到一个新的点,同时能满足约束条件.

上述初始化算子、交叉算子、变异算子等遗传操作都不会破坏解空间的合法性,因此在具体的实现中不必在每次遗传操作之后测试解的有效性,从而加快实现的计算速度.更重要的是在概率意义上保证了解空间是完全可达的,从而在理论上保证算法结果的全局性.此外 SSGA 选择算子采用确定式采样选择策略(Deterministic Sampling);适应度函数采

用线性尺度变换策略.

#### 4 仿真实例(Simulation implementing)

作者采用 Standard C++ (ISO-14882)<sup>[9]</sup> 语言编写了基于 Microsoft Visual C++ 环境下  $\lambda$ SSGA 算法的实现程序.考虑下列测试函数,该函数在可行区域内有 3 个局部最优解,其中有 1 个为全局最优解,见图 1.

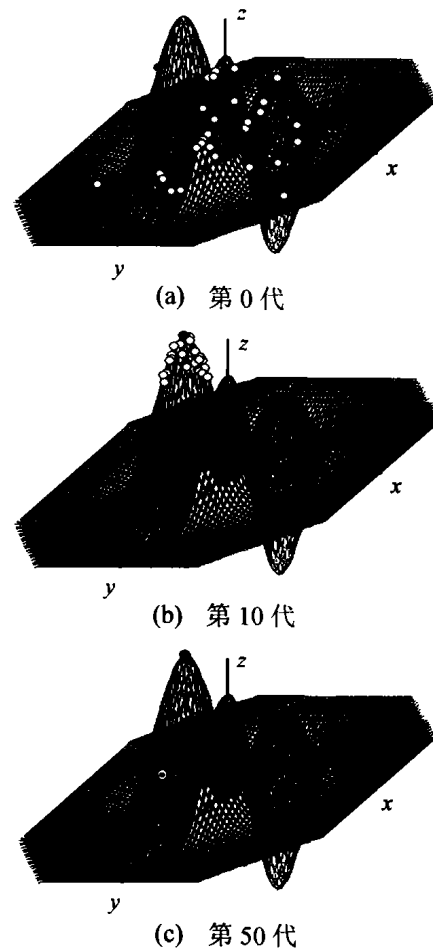


图1  $\lambda$ SSGA 进化过程

Fig. 1  $\lambda$ SSGA evolving procedure

$$\begin{aligned} \max Z = & 3(1 - x_1)^2 e^{(-x_1^2 - (x_2+1)^2)} - \\ & 10(0.2x_1 - x_1^3 - x_2^5) e^{(-x_1^2 - x_2^2)} - \\ & \frac{1}{3} e^{((-x_1+1)^2 - x_2^2)}, \end{aligned} \quad (17)$$

$$\begin{aligned} \text{s.t.} \quad & \begin{cases} -x_1 + x_2 - 6 \leq 0, \\ -x_1 - x_2 - 6 \leq 0, \\ -x_1 + x_2 - 6 \leq 0, \\ -x_1 - x_2 - 6 \leq 0. \end{cases} \quad (18) \\ & -4 \leq x_1 \leq 4, -4 \leq x_2 \leq 4. \end{aligned}$$

由式(6),(7)可循环得到约束条件和边界条件组成凸区域的 8 个极点:

$$x^1 = (-2, +4), x^2 = (+2, +4),$$

$$x^3 = (+4, +2), x^4 = (+4, -2),$$

$$x^5 = (+2, -4), x^6 = (-2, -4),$$

$$x^7 = (-4, -2), x^8 = (-4, +2).$$

由式(13)可将分量  $x_1, x_2$  分别表示为

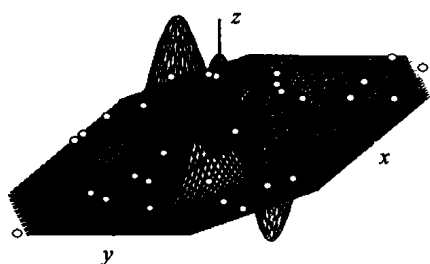
$$x_1 = -2\lambda_1 + 2\lambda_2 + 4\lambda_3 + 4\lambda_4 + 2\lambda_5 - 2\lambda_6 - 4\lambda_7 - 4\lambda_8, \quad (19)$$

$$x_2 = -4\lambda_1 + 4\lambda_2 + 2\lambda_3 - 2\lambda_4 - 4\lambda_5 - 4\lambda_6 - 2\lambda_7 + 2\lambda_8. \quad (20)$$

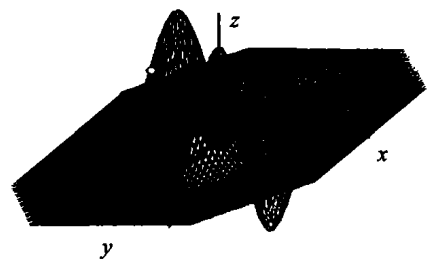
则优化问题式(17), (18)可转化为优化问题 II:

$$\max z = f(\lambda_1, \lambda_2, \dots, \lambda_8), \quad (21)$$

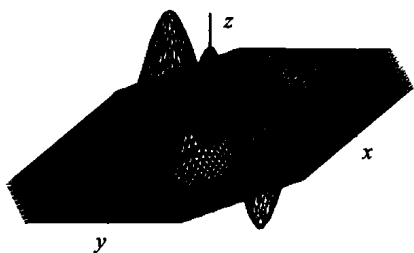
$$\begin{aligned} \text{s.t. } & \sum_{j=1}^8 \lambda_j = 1, \lambda_j \geq 0. \\ & j = 1, 2, \dots, 8. \end{aligned} \quad (22)$$



(a) 第0代



(b) 第10代



(c) 第50代

图2  $x$ SSGA 进化过程

Fig. 2  $x$ SSGA evolving procedure

应用 SSGA 算法对上述转化优化问题进行优化,得到全局最优的  $\lambda$  向量,再按式(19), (20)进行计算即可得到原优化问题的全局最优解.算法的参数设置如下:交叉概率 0.90;变异概率 0.01;群体数 30;进化代数 50;每代替换个体数 5.

$\lambda$ SSGA 算法中群体多样性性能指标定义为

$$\text{Div}_{\text{pop}} = \frac{2 \cdot \sum_{i=1}^N \sum_{j=i+1}^N d(i, j)}{N \cdot (N - 1)}. \quad (23)$$

在上式中,  $N$  表示为群体规模,  $d(i, j)$  表示为第  $i$  个体与第  $j$  个体(即染色体)之间的欧氏距离,即

$$d(i, j) = \sqrt{\sum_{m=1}^l (i_m - j_m)^2}. \quad (24)$$

其中  $l$  为染色体长度,  $i_m$  和  $j_m$  分别为  $i$  和  $j$  染色体第  $m$  个显式基因. 群体多样性计算复杂度为  $O(N^2)$ , 即在每一代进化中都需要额外的  $0.5N(N-1)$  次计算. 在实际的计算中可以通过关闭多样性选项极大提高算法计算速度.

从图1,图3可以看出,  $\lambda$  编码 SSGA 算法具有较快的收敛速率,群体规模为 30 情况下,在进化到 20 代左右时群体中所有的个体基本上都收敛到了全局最优化解的附近,同时还能保持群体中的多样性(见图4),从而可以增强算法的全局搜索特性;而由图2,图5可看出  $x$  编码 SSGA 算法在同样的条件下,直到 50 代也没有进化到全局最优解.从图6中可以看到群体在 10 代以后多样性趋于 0,这都表明由于约束条件的存在,交叉产生优良的模式不能有效地遗传到下一代,算法更多的依靠变异操作来产生优良模式(从图6中看出,在 10, 20, 30 代附近的有效变异导致了相对好的解产生),这样最终导致算法的收敛速度减慢或者是在相当长的时间内收敛不到全局最优化解.

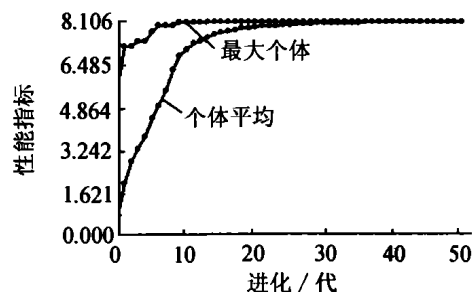


图3  $\lambda$  编码最优个体及个体平均值曲线

Fig. 3 Best individual and mean value of population by using  $\lambda$ SSGA

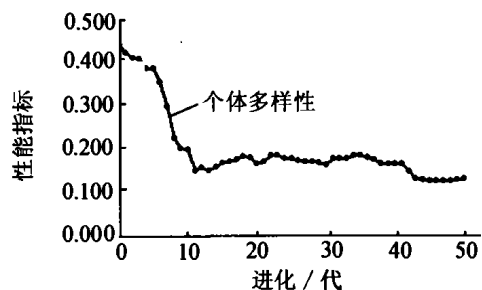


图4  $\lambda$  编码个体多样性曲线

Fig. 4 Diversity of population by using  $\lambda$ SSGA

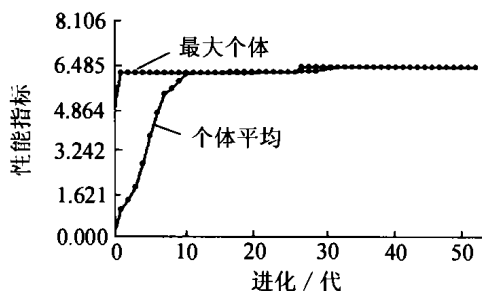


图5  $\lambda$  编码最优个体及个体平均值曲线

Fig. 5 Best individual and mean value of population by using  $\lambda$ SSGA

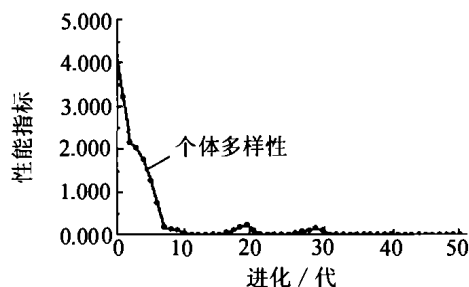


图6  $\lambda$  编码个体多样性曲线

Fig. 6 Diversity of population by using  $\lambda$ SSGA

$\lambda$ SSGA 算法得到测试函数的最优解为 (0.000966, 1.579330), 最优目标值为 8.105275; 应用 CONOPT Solver<sup>[4]</sup> 通过选取不同起点, 得到最好的解为 (0.000474, 1.580535), 最优目标值为 8.105417, 误差仅为 0.0018%, 这说明  $\lambda$  编码 SSGA 算法的收敛精度是很高的。

## 5 结束语 (Conclusions)

本文中提出了一种适于求解线性约束条件下非线性目标函数的  $\lambda$ SSGA 全局优化算法, 并从理论上推导出了如何得到线性区域全部极点的方法以及证明了原问题和转化问题的等效性。大量的仿真表明, 该算法不仅能得到问题的全局最优解, 而且能在保证较高精度的同时还加快了全局收敛速度。算法具有以下的特点:

- 1)  $\lambda$ SSGA 算法是基于遗传算法的框架进行优化的, 因此在理论上和实现上都可以保证得到解的全局最优性;
- 2)  $\lambda$ SSGA 算法编码方法可将原约束条件转化为小边界的约束条件, 从而减小搜索空间, 有利于提高解的精度;
- 3)  $\lambda$ SSGA 算法采用特定凸交叉算子以及交换变

异算子、倒位算子, 从而保证在遗传操作过程中不会产生无效的编码, 并在概率意义上实现整个可行解空间上充分可达, 这样就可维持在维持较高收敛速度的同时能有效地保持群体的多样性;

4) 非线性规划问题中的目标函数并不要求是连续可微的, 而且目标函数也可以是非凹函数;

5) 通过对非线性约束条件进行凸逼近处理后, 算法可进一步适用于非线性约束的情形。

## 参考文献 (References):

- [1] 陈宝林. 最优化方法理论与算法 [M]. 北京: 清华大学出版社, 1989.  
(CHEN Baolin. *Optimal Theory and Algorithms* [M]. Beijing: Tsinghua University Press, 1989.)
- [2] AVRIEL M. *Nonlinear Programming Analysis and Methods* [M]. Englewood Cliffs: Prentice-Hall, 1976.
- [3] 颜庆津. 数值分析 [M]. 北京: 北京航空航天大学出版社, 1991.  
(YAN Qingjin. *Numerical Analysis* [M]. Beijing: BeiHang University Press, 1991.)
- [4] ARNE D. *Conopt* [EB/OL]. Bagsvaerd, Denmark: ARKI Consulting and Development. <http://www.gams.com/solvers/conopt.pdf>.
- [5] 周明. 遗传算法原理及应用 [M]. 北京: 国防工业出版社, 1999.  
(ZHOU Ming. *Genetic Algorithms: Theory and Applications* [M]. Beijing: National Defence Industry Press, 1999.)
- [6] RENDERS J M, FLASSE S P. Hybrid methods using genetic algorithms for global optimization [J]. *IEEE Trans on System, Man, and Cybernetics*, Part B, 1996, 26(2): 243 - 258.
- [7] BAZARAS M S, JARVIS J J. *Linear Programming and Network Flows* [M]. New York: Wiley, 1977.
- [8] BECKWITH S F, WONG K P. A genetic algorithm approach for electric pump scheduling in water supply systems [C]// *IEEE Int Conf on Evolutionary Computation*. Perth, Australia: [s. n.], 1995(1): 21 - 26.
- [9] BJARNE S. *The C++ Programming Language (Special Edition)* [M]. Beijing: China Machine Press, 2002.

## 作者简介:

张贵军 (1974—), 男, 浙江工业大学信息工程学院讲师, 2000年在太原理工大学获工学硕士学位, 2004年在上海交通大学获工学博士学位, 主要研究兴趣是优化调度, 及算法软件的开发等, E-mail: zgj@zjut.edu.cn;

俞立 (1961—), 男, 浙江工业大学信息工程学院教授, 博士生导师, 主要研究领域为鲁棒控制等, E-mail: lyu@zjut.edu.cn.

吴锡华 (1940—), 上海交通大学电子信息与电气工程学院教授, 博士生导师, 研究方向为过程控制。