第33卷 第5期
2010年5月
计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS
Vol. 33 No. 5
May 2010

一种新的基于正交实验设计的约束优化进化算法

蔡自兴 江中央 王 勇 罗一丹

(中南大学信息科学与工程学院 长沙 410083)

摘 要 提出了一种新的基于正交实验设计的约束优化进化算法.新算法的主要特点是:在搜索机制方面,利用正交实验设计方法安排多个父代个体的交叉操作,提出了一种新的多父体正交交叉算子,新的交叉算子能够有效利用多个父代个体所携带的信息产生新的具有代表性的子代个体.此外,利用单形交叉算子对父代种群进行并行搜索,以协调算法的勘探和开采能力.在约束处理技术上,新算法引入了一个衡量个体优、劣的新比较准则.通过13个标准的测试函数验证了算法的通用性和有效性.

关键词 约束优化;进化算法;正交实验设计;约束处理技术;单形交叉算子

中图法分类号 TP18 **DOI号**: 10.3724/SP.J.1016.2010.00855

A Novel Constrained Optimization Evolutionary Algorithm Based on Orthogonal Experimental Design

CAI Zi-Xing JIANG Zhong-Yang WANG Yong LUO Yi-Dan

(*School of Information Science and Engineering, Central South University, Changsha* 410083)

Abstract A novel constrained optimization evolutionary algorithm based on orthogonal experimental design, referred as COEA/OED, is proposed in this paper for constrained optimization problems. The primary features of the algorithm proposed are as follows. As for search mechanism, COEA/OED utilizes orthogonal experimental design method to arrange the crossover operation of several parents and, as a result, a new multi-patent orthogonal crossover operator is proposed, which can effectively make use of the information carried by the parents and generate representative offspring. In addition, the simplex crossover is used to enrich the exploratory and exploitative abilities of the algorithm proposed. As for constraint-handing technique, a novel individual comparison criterion is introduced. COEA/OED is tested on 13 well-known benchmark functions, and the empirical evidence demonstrates that COEA/OED is generic and effective.

Keywords constrained optimization; evolutionary algorithm; orthogonal experimental design; constraint-handing techniques; simplex crossover

1 引 言

在科学、工程和商业等诸多领域中,很多实际问题通过数学建模后都可以转化为约束优化问题(Constrained Optimization Problems,COPs).遗传算法(Genetic Algorithm,GA)是一种借鉴生物界自然选择和自然遗传机制的随机搜索算法,是解决约

收稿日期:2008-05-11;最终修改稿收到日期:2009-07-26.本课题得到国家基础研究项目(A1420060159)、国家自然科学基金(60805027,90820302)、教育部博士点基金(200805330005)和湖南省研究生创新基金(CX2009B039)资助.**蔡自兴**,男,1938年生,教授,博士生导师,主要研究领域为人工智能、计算智能、智能控制.**江中央**,男,1982年生,硕士,主要研究方向为进化计算、全局优化、约束优化.E-mail: zyjiang08@gmail.com.**王 勇**,男,1980年生,博士研究生,讲师,主要研究方向为进化计算、多目标优化、约束优化.**罗一丹**,女,1983年生,硕士,主要研究方向为进化计算、免疫算法.

束优化问题的一种有效方法，目前已被广泛地用于求解约束优化问题[1-2]. 但大量的研究表明，传统的遗传算法也存在许多的不足和缺陷[3-4]，如早熟收敛、计算量大和局部搜索能力差等. 近年来，不少研究者将优化实验设计方法(如正交实验设计、均匀设计、佳点集等)引入到遗传算法中，在一定程度上克服了传统遗传算法的缺陷，表现出良好的搜索性能. Zhang 和 Leung[5] 将正交实验设计引入遗传算法来设计多媒体的多点传输路由. Leung 和 Wang[6] 利用正交表初始化种群和安排两个父代个体的交叉操作，提出了一种量化的正交遗传算法(OGA/Q)求解高维全局优化问题. Wang 和 Dang[7] 利用均匀表设计两个父代个体的交叉算子，并提出了一种基于水平集进化和拉丁方的进化算法(LEA). 张铃和张钹[8] 利用数论中的佳点集理论和方法，对 GA 中的交叉算子进行了重新设计，提出了佳点集遗传算法，该算法在求解全局优化问题、SAT 问题、TSP 问题和背包问题方面取得了满意的效果. 在用 GA 求解约束优化问题时，除了算法本身的搜索能力以外，如何处理约束条件也是得到好的优化效果的关键. 目前常见的约束处理技术有惩罚函数法[9-11]、多目标优化法[12-14] 以及其它方法. 惩罚函数法是利用进化算法处理约束优化问题的常用方法，它允许群体中的个体在一定程度上违反约束条件，但必须对该个体依其违反约束条件的程度进行惩罚以减小它被选择的概率. 惩罚函数法的惩罚参数选取比较困难，而且算法的性能强烈地依赖于所选择的参数. 多目标优化法的主要思想是将约束优化问题转换为多目标优化问题后，利用多目标技术处理约束优化问题，但利用多目标技术处理约束优化问题需要了解多目标处理技术的知识背景，且对群体中的个体进行选择和比较操作时，需要计算 Pareto 前沿，在一定程度上增加了算法的计算开销.

考虑到算法的搜索能力和约束处理技术对求解约束优化问题的性能影响，在搜索策略上，为了充分利用多个父代个体之间所携带的有效信息，产生具有代表性的子代个体，本文利用正交实验设计方法设计了一种新的多父体正交交叉算子(Multi-parent Orthogonal Crossover, MOC)，并采用该算子对群体中的父代个体进行交叉操作. 同时，利用单形交叉算子(Simplex Crossover, SPX)[15] 对父代种群进行并行搜索，来协调算法的勘探和开采能力. 在约束处理技术上，本文将群体的进化过程分为 3 个阶段，并根据群体在 3 个不同阶段的特点采取不同的选择策略，从而有效地引导群体不断向最优解逼近. 数值实验验证了算法的通用性和有效性.

2 约束优化问题的描述

不失一般性，一般的约束优化问题(Constrained Optimization Problems, COPs)可以描述为

$$\begin{aligned} &\text{Minimize} \quad f(\boldsymbol{x}),\ \boldsymbol{x}=(x_1,x_2,\cdots,x_N)\in\mathcal{R}^N \\ &\text{Subject to}\ \ g_j(\boldsymbol{x})\le 0,\ j=1,2,\cdots,l \\ &\qquad\qquad\ \ h_j(\boldsymbol{x})=0,\ j=l+1,\cdots,m \end{aligned} \tag{1}$$

这里，$\boldsymbol{x}\in\Omega\subseteq S$ 为决策向量，Ω 为可行域，S 为决策空间. 一般地，S 为$\mathcal{R}^N$中的 N 维长方体：$l(i)\le x_i\le u(i)$，$l(i)$、$u(i)$ 为常数，$i=1,2,\cdots,N$. $f(\boldsymbol{x})$、$g_j(\boldsymbol{x})$、$h_j(\boldsymbol{x})$均为$\mathcal{R}^N$上的 N 元函数，$f(\boldsymbol{x})$为目标函数，$g_j(\boldsymbol{x})\le 0$ 为第 j 个不等式约束条件，$h_j(\boldsymbol{x})=0$ 为第 j 个等式约束条件，l 表示不等式约束条件的个数，$m-l$ 表示等式约束条件的个数.

3 正交实验设计

正交实验设计[6] 是一种解决多因素、多水平实验问题的有效方法，它利用正交表 $L_M(Q^F)$安排少数次实验，就能找到最好或者较好的实验条件，因此它被广泛地用于寻优. $L_M(Q^F)$表示一个具有 F 个因素和 Q 个水平的正交表，其中 L 表示拉丁方，M 表示水平组合数. 对 F 因素、Q 水平的实验问题，若进行全面组合实验，则要做 Q^F组实验，但是当 Q 和 F 很大时，不可能做 Q^F 组实验，而应用正交表 $L_M(Q^F)$来安排正交实验，只需要选择 M 个组合去做实验. 这里 M 一般远小于 Q^F. 我们以正交表 $L_9(3^3)$为例，如表 1，对 3 因素、3 水平的问题而言，若依据正交表 $L_9(3^3)$来进行正交实验，只需做 9 次验，但若进行全面组合实验，则需 $3^3=27$ 次实验. 可见正交实验设计大大减少了实验次数，且因素和水平越大，该方法的优越性越明显.

表 1 正交表 $L_9(3^3)$

实验号	水平		
	1	2	3
1	1	1	1
2	1	2	2
3	1	3	3
4	2	1	2
5	2	2	3
6	2	3	1
7	3	1	3
8	3	2	1
9	3	3	2

表 2　构建正交表 $L_M(Q^F)$ 的伪代码

```
1. select the smallest J fulfilling (Q^J-1)/(Q-1)≥F;
2. if (Q^J-1)/(Q-1)=F, then
   F'=F else F'=(Q^J-1)/(Q-1);
3. construct the basic columns as follows:
   for k=1 to J do
       j=(Q^(k-1)-1)/(Q-1)+1;
       for i=1 to Q^J do
           a_{i,j}=⌊(i-1)/Q^(J-1)⌋ mod Q;
       end for
   end for
4. construct the non-basic columns as follows:
   for k=2 to J do
       j=(Q^(k-1)-1)/(Q-1)+1;
       for s=1 to j-1 do
           for t=1 to Q-1 do
               a_{j+(s-1)(q-1)+t}=(a_s×t+a_j) mod Q;
           end for
       end for
5. increment a_{i,j} by one for all 1≤i≤M and 1≤j≤F';
6. delete the last F'-F columns of L_{Q^J}(Q^{F'}) to get
   L_M(Q^F) where M=Q^J;
```

为了方便，记 $L_M(Q^F)=[a_{i,j}]_{M\times F}$，其中第 i 个组合的第 j 个因素的水平值为 $a_{i,j}$，$a_{i,j}\in\{1,2,\cdots,Q\}$. 设正交表 $[a_{i,j}]_{M\times F}$ 第 j 列为 a_j，若 $j=1,2,(Q^3-1)/(Q-1)+1,\cdots,(Q^{J-1}-1)/(Q-1)+1$，则 a_j 称为基本列，其它的列称为非基本列. 文献[6]给出了创建正交表 $L_M(Q^F)$ 的算法，它首先创建基本的列，然后创建非基本的列，详细描述见表 2. 其中 Q 为素数，且 $M=Q^J$，J 满足式(2).

$$F\le\frac{Q^J-1}{Q-1}\tag{2}$$

4　基于正交实验设计的约束优化进化算法

4.1　多父体正交交叉算子

通过对多个父代个体进行重组，可以交换多个父代个体所携带的基因信息，以引入更多的启发式信息对解空间进行有效搜索，这样增加子代个体的采样范围，使种群在进化过程中保持好的多样性，从而避免早熟收敛. 目前在遗传算法中，广泛使用的交叉算子有单点交叉、两点交叉、均匀交叉、算术交叉、单峰正态分布交叉等. 但是这些算子在子代个体的样本空间中随机产生新的子代个体，搜索具有一定盲目性，在一定程度上降低了算法的搜索效率. 因此，本文利用正交实验设计方法在子代个体的样本空间中产生具有代表性的组合来生成子代个体，以提高搜索效率.

例如，设父代个体 $\boldsymbol{p}_1=(p_{11},p_{12},p_{13},p_{14},p_{15},p_{16})$，$\boldsymbol{p}_2=(p_{21},p_{22},p_{23},p_{24},p_{25},p_{26})$，$\boldsymbol{p}_3=(p_{31},p_{32},p_{33},p_{34},p_{35},p_{36})$. 对 $\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3$ 进行重组，将有 3^6 种可能的组合即子代个体的样本空间大小为 3^6. 我们利用正交实验设计方法，从子代个体的样本空间中产生具有代表性的组合来生成子代个体，从而提高算法的搜索效率. 对 $\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3$ 而言，利用正交实验设计安排交叉操作的具体步骤是：把参与交叉操作的父代个体 $\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3$ 分别作为正交实验的一个水平，分别记为 $\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\boldsymbol{\beta}_3$；然后将每一个个体分成 F 个片段，例如把 $\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3$ 分成 3 个片段，分片的位置如表 3 所示，将每一个片段作为正交实验的一个因素，分别记为 f_1,f_2,f_3. 这样就将对 $\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3$ 的交叉操作转化为 3 因素、3 水平的实验问题，例如因素 f_1 对应的 3 个水平分别为 $f_1(1)=p_{11}\ p_{12}$，$f_1(2)=p_{21}\ p_{22}$，$f_1(3)=p_{31}\ p_{32}$. 最后根据表 2 构造正交表 $L_9(3^3)$ 来安排正交实验设计，如表 4 所示，将产生 9 个子代个体.

表 3　$\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3$ 的因素和水平表

水平	因素		
	f_1	f_2	f_3
$\beta_1(\boldsymbol{p}_1)$	$p_{11}\ p_{12}$	$p_{13}\ p_{14}$	$p_{15}\ p_{16}$
$\beta_2(\boldsymbol{p}_2)$	$p_{21}\ p_{22}$	$p_{23}\ p_{24}$	$p_{25}\ p_{26}$
$\beta_3(\boldsymbol{p}_3)$	$p_{31}\ p_{32}$	$p_{33}\ p_{34}$	$p_{35}\ p_{36}$

表 4　对 $\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3$ 进行正交实验设计的方案和结果

实验号	水平			子代个体
	f_1	f_2	f_3	
	1	2	3	
1	$1(p_{11}\ p_{12})$	$1(p_{13}\ p_{14})$	$1(p_{15}\ p_{16})$	$p_{11}\ p_{12}\ p_{13}\ p_{14}\ p_{15}\ p_{16}$
2	$1(p_{11}\ p_{12})$	$2(p_{23}\ p_{24})$	$2(p_{25}\ p_{26})$	$p_{11}\ p_{12}\ p_{23}\ p_{24}\ p_{25}\ p_{26}$
3	$1(p_{11}\ p_{12})$	$3(p_{33}\ p_{34})$	$3(p_{35}\ p_{36})$	$p_{11}\ p_{12}\ p_{33}\ p_{34}\ p_{35}\ p_{36}$
4	$2(p_{21}\ p_{22})$	$1(p_{13}\ p_{14})$	$2(p_{25}\ p_{26})$	$p_{21}\ p_{22}\ p_{13}\ p_{14}\ p_{25}\ p_{26}$
5	$2(p_{21}\ p_{22})$	$2(p_{23}\ p_{24})$	$3(p_{35}\ p_{36})$	$p_{21}\ p_{22}\ p_{23}\ p_{24}\ p_{35}\ p_{36}$
6	$2(p_{21}\ p_{22})$	$3(p_{33}\ p_{34})$	$1(p_{15}\ p_{16})$	$p_{21}\ p_{22}\ p_{33}\ p_{34}\ p_{15}\ p_{16}$
7	$3(p_{31}\ p_{32})$	$1(p_{13}\ p_{14})$	$3(p_{35}\ p_{36})$	$p_{31}\ p_{32}\ p_{13}\ p_{14}\ p_{35}\ p_{36}$
8	$3(p_{31}\ p_{32})$	$2(p_{23}\ p_{24})$	$1(p_{15}\ p_{16})$	$p_{31}\ p_{32}\ p_{23}\ p_{24}\ p_{15}\ p_{16}$
9	$3(p_{31}\ p_{32})$	$3(p_{33}\ p_{34})$	$2(p_{25}\ p_{26})$	$p_{31}\ p_{32}\ p_{33}\ p_{34}\ p_{25}\ p_{26}$

在 N 维的实数空间中，设参与重组的 Q 个父代个体为 $\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_Q$，其中 $\boldsymbol{p}_i=(p_{i1},p_{i2},\cdots,p_{iN})$，且 $i\in\{1,2,\cdots,Q\}$. 将每一个参与重组的父代个体看作是正交实验设计的一个水平，即 Q 个水平；然后将每一个父代个体分成 F 个片段，每一个片段作为正交实验设计的一个因素，即 F 个因素. 这样就将对 Q 个父代个体的重组问题转化为 Q 水平、F 因素的实验设计问题. 然后构造正交表 $L_M(Q^F)$，安排正交实验设计将产生 M 个子代个体，其具体的操作过程见算法 1.

算法 1. 多父体正交交叉.

1. 将参与重组的 Q 个父代个体分别作为正交实验设计的一个水平,记第 i 个水平为 $\boldsymbol{\beta}_i, i\in\{1,2,\cdots,Q\}$,则 $\boldsymbol{\beta}_i=\boldsymbol{p}_i$.

2. 设 $\boldsymbol{x}=(x_1,x_2,\cdots,x_N)$ 为参与重组的任一个体,将个体 $\boldsymbol{x}$ 分成 F 个片段,将每一个片段作为正交实验设计的一个因素,具体的分片操作方法如下:首先随机产生 $F-1$ 个整数,即 $k_1,k_2,\cdots,k_{F-1}$,且满足 $1<k_1<k_2<\cdots<k_{F-1}<N$;然后把个体 $\boldsymbol{x}=(x_1,x_2,\cdots,x_N)$ 分成 F 个片段,其中每一片段表示个体 $\boldsymbol{x}$ 的一个因素:

$$\begin{cases}f_1=(x_1,\cdots,x_{k_1})\\ f_2=(x_{k_1+1},\cdots,x_{k_2})\\ \cdots\\ f_F=(x_{k_{F-1}},\cdots,x_N)\end{cases} \tag{3}$$

令 $k_0=0, k_F=N$,因此第 $j(j=1,2,\cdots,F)$ 个因素 f_j 的 Q 个水平可以表示为

$$\begin{cases}f_j(1)=(\beta_{1,k_{j-1}+1},\beta_{1,k_{j-1}+2},\cdots,\beta_{1,k_j})\\ f_j(2)=(\beta_{2,k_{j-1}+1},\beta_{2,k_{j-1}+2},\cdots,\beta_{2,k_j})\\ \cdots\\ f_j(Q)=(\beta_{Q,k_{j-1}+1},\beta_{Q,k_{j-1}+2},\cdots,\beta_{Q,k_j})\end{cases} \tag{4}$$

3. 构造正交表 $L_M(Q^F)=[a_{i,j}]_{M\times F}$,根据正交表 $L_M(Q^F)$,安排实验,产生 M 个子代个体:

$$\begin{cases}(f_1(a_{1,1}),f_2(a_{1,2}),\cdots,f_F(a_{1,F}))\\ (f_1(a_{2,1}),f_2(a_{2,2}),\cdots,f_F(a_{2,F}))\\ \cdots\\ (f_1(a_{M,1}),f_2(a_{M,2}),\cdots,f_F(a_{M,F}))\end{cases} \tag{5}$$

4.2 单形交叉算子

单形交叉算子(Simplex Crossover, SPX)[15]是目前常用的实数交叉算子之一,并广泛地用于求解各类优化问题.单形交叉算子基于均匀分布来产生后代个体,且单形交叉算子运算前后,群体中个体向量的均值应保持不变.m 个独立的父体向量($\boldsymbol{x}_i, i=1,2,\cdots,m$)形成了一个单形,子代个体的产生按照以下步骤进行:(1)按一定的比率 ε 将单形沿各个方向($\boldsymbol{x}_i-\boldsymbol{o}$,这里 $\boldsymbol{o}$ 是 m 个向量的中心,即 $\boldsymbol{o}=\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i$,$\varepsilon$ 为扩张比)进行扩张得到一个新的单形;(2)从新的单形中随机选择一个点作为后代个体.例如,在两维空间中由 3 个点 $\boldsymbol{x}_1$,$\boldsymbol{x}_2$ 和 $\boldsymbol{x}_3$ 构成的一个单形.将这个单形以 $(1+\varepsilon)(\varepsilon\geq 0)$ 的比例向各个方向进行扩张,令 $\boldsymbol{o}=\frac{1}{3}\sum_{i=1}^{3}\boldsymbol{x}_i$,$\boldsymbol{y}_i=(1+\varepsilon)(\boldsymbol{x}_i-\boldsymbol{o})$,则由 $\boldsymbol{y}_1$,$\boldsymbol{y}_2$ 和 $\boldsymbol{y}_3$ 形成了一个新的单形.在新单形中随机取一点 $\boldsymbol{z}$,则 $\boldsymbol{z}=k_1\boldsymbol{y}_1+k_2\boldsymbol{y}_2+k_3\boldsymbol{y}_3+\boldsymbol{o}$,其中 k_1,k_2 和 k_3 为 $[0,1]$ 中均匀分布的随机数且满足 $k_1+k_2+k_3=1$.

4.3 个体优劣的新比较准则

类似于文献[16],本文将种群的进化过程分为以下 3 个阶段:种群中只有不可行解;种群中既有可行解又有不可行解;种群中只有可行解.

(1)当种群中只有不可行解的时候,应使整个种群尽快地向可行域边界逼近,而个体违反约束程度反映个体到可行域边界的距离,因此本文根据个体违反约束程度的大小来对个体进行选择操作.选择违反约束程度小的个体,进入下一代种群.

(2)Runarsson 和 Yao[17]认为,通常情况下平衡可行解与不可行解的比例是约束处理技术的关键问题之一.文献[18]使违反约束程度最小的不可行解以一定的概率继续生存,以此来增加群体的多样性.实验结果表明这种多样性机制对提高算法的性能是有效的.事实上,增加群体中违反约束条件最小的不可行解的生存概率,不仅可以提高群体的多样性,而且还可以引导群体中其它不可行解迅速地逼近可行域边界.因此,当种群中既有可行解又有不可行解时,使可行解和不可行解保持一定的比例,同时增加违反约束条件最小的不可行解的生存概率,可以维持种群的多样性,有利于群体向最优解逼近.

(3)当种群中只有可行解时,适应度值大小反映个体与最优解的距离,为了使种群以较快的速度收敛到最优解,我们应根据个体适应度值的大小来进行选择操作.

基于以上考虑,当种群处于不同进化阶段时,本文遵循以下原则来比较和选择个体:

(1)种群中只有不可行解时,只比较个体违反约束程度的大小,违反约束条件小的个体占优.

(2)种群中既有可行解又有不可行解时,为了使可行解与不可行解保持一定的比列,以更好地搜索可行的最优解,同时为了对群体中的可行解和不可行解进行简单直接比较,我们采取了以下处理方式.

令

$$G_j(\boldsymbol{x})=\begin{cases}\max\{0,g_j(\boldsymbol{x})\}, & 1\leq j\leq l\\ |h_j(\boldsymbol{x})|, & l+1\leq j\leq m\end{cases} \tag{6}$$

表示个体 $\boldsymbol{x}$ 与第 j 个约束条件的距离,则

$$G(\boldsymbol{x})=\sum_{j=1}^{m}G_j(\boldsymbol{x}) \tag{7}$$

表示个体 $\boldsymbol{x}$ 违法约束条件的程度.

设 $\boldsymbol{x}_{\min}$,$\boldsymbol{x}_{\max}$ 分别为当前群体可行解中适应度值最小和最大的个体,$\boldsymbol{x}_{uf}$ 为当前群体不可行解中违反约束程度最小的个体.

$$f'(\boldsymbol{x})=\begin{cases}f(\boldsymbol{x}), & \boldsymbol{x}\text{ 为可行解}\\ \max\{f(\boldsymbol{x}_{\min})+\eta^{*}(f(\boldsymbol{x}_{\max})-f(\boldsymbol{x}_{\min})),f(\boldsymbol{x})\}, & \\ & \boldsymbol{x}\text{ 为不可行解}\end{cases}\tag{8}$$

其中，η 为当前群体中不可行解的百分比.

$$G'(\boldsymbol{x})=\begin{cases}0, & \boldsymbol{x}\text{ 为可行解}\\ |G(\boldsymbol{x})-G(\boldsymbol{x}_{uf})|, & \boldsymbol{x}\text{ 为不可行解}\end{cases}\tag{9}$$

$$F(\boldsymbol{x})=f'(\boldsymbol{x})+G'(\boldsymbol{x})\tag{10}$$

$F(\boldsymbol{x})$ 为个体 $\boldsymbol{x}$ 转化后的适应度值，对求最小值的问题来说，$F(\boldsymbol{x})$ 数值小的个体占优. 当群体中不可行解的比例较小时，由式(8)知，不可行解转换后的函数适应度值小，被选择进入下一代的概率增加，这样提高了群体中不可行解比例，反之不可行解被选择概率减少. 因此通过这种自适应的动态调节机制可以有效地调节群体中可行解与不可行解的比例. 此外，由式(9)可知，通过取消对群体中违反约束程度最小的不可行解的惩罚，可以增加其进入下一代种群的概率，这样引导其它不可行解快速地逼近可行域.

(3) 种群中只有可行解时，比较个体目标函数值的大小. 对求最小值问题来说，函数值小的个体占优.

4.4 算法步骤

设种群 P 的规模为 n，N 为个体变量的维数，t 为进化代数，采用实数编码，$\lfloor w\rfloor$ 表示取实数 w 的整数部分. 整个算法实现的具体步骤如下：

1. 生成初始种群

随机生成初始种群 $P_0=\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n\}$，$P_0$ 中的每一个个体满足 $\boldsymbol{x}_i\in S$，$i\in\{1,2,\cdots,n\}$，令 $t=0$；

2. 交叉操作

2.1. 把群体 P_t 中的 n 个个体随机分成 $\lfloor n/\lambda\rfloor$ 子类，每一个子类中有 λ 个个体，P_t 剩余的个体加入到种群 P_t^1. 构造正交表 $L_M(\lambda^{N-1})$，利用多父体正交交叉算子以 p_0 的概率对每一个子类中的个体进行交叉操作产生新的子代个体. 然后再从每一个子类的父代个体和子代个体中，根据 4.3 节的个体优劣比较准则，选出 λ 个最好的个体进入到种群 P_t^1；

2.2. 把群体 P_t^1 中的 n 个个体随机分成 $\lfloor n/u\rfloor$ 子类，每一个子类中有 u 个个体，P_t^1 剩余的个体加入到种群 P_t^2. 利用 SPX 算子以 p_1 的概率对每一个子类中的个体进行交叉操作产生新的子代个体. 然后再从每一个子类的父代个体和子代个体中，根据 4.3 节的个体优劣比较准则，选出 u 个最好的个体进入种群 P_t^2.

3. 变异操作

种群 P_t^2 的任一个体 $\boldsymbol{x}_i=(x_{i,1},x_{i,2},\cdots,x_{i,N})$，以概率 p_m 参与变异操作. 具体操作是：① 产生一个随机整数 $j\in[1,N]$ 和一个随机数实数 $r\in[0,1]$；② 令 $x_{i,j}=l_j+r(u_j-l_j)$. 群体 P_t^2 经变异后生成的新的子代种群 P_t^3.

4. 种群更新

从 P_t^2 和 P_t^3 中，根据 4.3 节的个体优劣比较准则，选出 n 个最好的个体进入到下一代种群 P_{t+1} 中.

5. 判断是否满足终止条件.

若满足终止条件，则输出最优解；否则 $t=t+1$，同时转步 2.

5 数值实验

5.1 实验结果

为了评估新算法(COEA/OED)的性能，我们选取文献[16-19]中采用的 13 个复杂的测试函数进行实验研究. 函数 g02 属于高维多峰函数，主要用来考察算法的搜索能力. 函数 g03、g05、g11、g13 主要考察算法的约束处理能力，其它几个函数主要用来考察算法的综合能力. 实验中，种群规模 $n=100$，$\lambda=3$，$u=3$，多父体正交交叉操作的概率 $p_0=0.1$，单形交叉操作的概率 $p_1=0.8$，扩张比 $\varepsilon=6$，变异概率 $p_m=0.1$，适应度函数值评价次数为 240000 次. 对等式约束优化问题，我们采用文献[9]中提出的新方法. 该方法将等式约束条件 $h(\boldsymbol{x})=0$ 转化为不等式约束条件 $|h(\boldsymbol{x})-e|<\delta$ 来处理，其中 δ 是一个很小的正常数，偏差值 e 随着种群代数的递增逐渐下降，其表达式如下：$e(t+1)=e(t)/C$，其中 C 为常数，t 为进化代数. 本文对所有等式约束优化问题取 $C=1.0165$，$e(0)=2.0$，$\delta=1\text{E}-4$. 所有实验都在 MATLAB 2007 中完成(需要 MATLAB 代码可与作者联系)，对于每一个测试函数，在相同的条件下独立运行 30 次，记录其最好的结果(best)、中间结果(median)、平均结果(mean)和解的标准差(std. dev). 表 5 列出了新算法 COEA/OED 对 13 个测试函数实验的结果.

为了对比，我们将 COEA/OED 的运行结果与其它 4 个国内外较好的约束算法实验结果进行了比较. 它们分别是文献[10]中的 SAFF 算法、文献[18]中的 SMES 算法、文献[19]中的 RY 算法、文献[20]中的 AIRCES 算法. 表 6 列出了 5 种算法比较的结果，这些实验结果均来自相关算法的参考文献，NA 表示原文中没有可获得的结果.

表 5 COEA/OED 算法对 13 个标准测试函数独立运行 30 次的结果

函数	运行结果					
	optimal	best	median	mean	worst	std. dev
g01	−15.000	−15.000	−15.000	−15.000	−15.000	0.0E+00
g02	−0.803619	−0.803619	−0.791021	−0.790908	−0.761532	7.9E−03
g03	1.00	1.00	1.00	1.00	1.00	5.8E−09
g04	−30665.539	−30665.539	−30665.539	−30665.539	−30665.539	1.2E−11
g05	5126.498	5126.498	5126.498	5126.498	5126.498	7.2E−12
g06	−6961.814	−6961.814	−6961.814	−6961.814	−6961.814	3.9E−11
g07	24.306	24.306	24.306	24.306	24.306	1.1E−05
g08	0.095825	−0.095825	0.095825	0.095825	0.095825	1.6E−17
g09	680.630	680.630	680.630	680.630	680.630	3.5E−12
g10	7049.248	7049. 521	7070.424	7072.167	7155.754	3.7E+01
g11	0.75	0.75	0.75	0.75	0.75	2.8E−07
g12	−1.000	−1.000	−1.000	−1.000	−1.000	0.0E+00
g13	0.0539498	0.0539498	0.0539498	0.1437424	0.4397026	1.6E−01

表 6 5 个算法(COEA/OED、SAFF[10]、SMES[18]、RY[19]和 AIRCES[20])对 13 个标准测试函数实验结果比较(NA 表示原文没有提供实验结果)

Function/optimal	status	运行结果				
		COEA/OED	SAFF	SMES	RY	AIRCES
g01/−15.000	best	−15.000	−15.000	−15.000	−15.000	−15.000
	mean	−15.000	−15.000	−15.000	−15.000	−15.000
	worst	−15.000	−15.000	−15.000	−15.000	−15.000
	std. dev	0.0E+00	0.0E+00	0.0E+00	0.0E+00	NA
g02/−0.803619	best	−0.803619	−0.80297	−0.803601	−0.803515	−0.803575
	mean	−0.790908	−0.79010	−0.785238	−0.781975	−0.779465
	worst	−0.761532	−0.76043	−0.751322	−0.726288	−0.716312
	std. dev	7.9E−03	1.2E−02	1.7E−02	2.0E−02	NA
g03/−1.000	best	−1.000	−1.000	−1.000	−1.000	−1.000
	mean	−1.000	−1.000	−1.000	−1.000	−1.000
	worst	−1.000	−1.000	−1.000	−1.000	−0.999
	std. dev	5.8E−09	7.5E−05	2.1E−04	1.9E−04	NA
g04/−30665.539	best	−30665.539	−30665.50	−30665.539	−30665.539	−30665.539
	mean	−30665.539	−30665.20	−30665.539	−30665.539	−30665.539
	worst	−30665.539	−30663.30	−30665.539	−30665.539	−30665.539
	std. dev	1.2E−11	4.9E−01	0.0E+00	2.0E−05	NA
g05/5126.498	best	5126.498	5126.989	5126.599	5126.497	NA
	mean	5126.498	5432.080	5174.492	5128.881	NA
	worst	5126.498	6089.430	5304.167	5142.472	NA
	std. dev	7.2E−12	3.9E+03	5.0E+01	3.5E+00	NA
g06/−6961.814	best	−6961.814	−6961.800	−6961.814	−6961.814	−6961.814
	mean	−6961.814	−6961.800	−6961.284	−6875.940	−6695.987
	worst	−6961.814	−6961.800	−6952.482	−6350.262	−6030.333
	std. dev	3.9E−11	0.0E+00	1.9E+00	1.6E+02	NA
g07/24.306	best	24.306	24.48	24.327	24.307	NA
	mean	24.306	26.58	24.475	24.374	NA
	worst	24.306	28.40	24.843	24.642	NA
	std. dev	1.1E−05	1.1E+00	1.3E−01	6.6E−02	NA
g08/−0.095825	best	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825
	mean	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825
	worst	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825
	std. dev	1.6E−17	0.0E+00	0.0E+00	2.6E−17	NA
g09/680.630	best	680.630	680.64	680.632	680.630	680.630
	mean	680.630	680.72	680.643	680.656	680.652
	worst	680.630	680.87	680.719	680.763	680.801
	std. dev	3.5E−12	5.9E−02	1.6E−02	3.4E−02	NA
g10/7049.248	best	7049. 521	7061.34	7051.903	7054.316	NA
	mean	7072.167	7627.89	7253.047	7559.192	NA
	worst	7155.754	8288.79	7638.366	8835.655	NA
	std. dev	3.7E+01	3.7E+02	1.4E+02	5.3E+02	NA

（续　表）

Function/optimal	status	运行结果				
		COEA/OED	SAFF	SMES	RY	AIRCES
g11/0.75	best	0.75	0.750	0.75	0.750	0.750
	mean	0.75	0.750	0.75	0.750	0.750
	worst	0.75	0.750	0.75	0.750	0.750
	std. dev	2.8E−07	0.0E+00	1.5E−04	8.0E−05	NA
g12/−1.000	best	−1.000	−1.000	−1.000	−1.000	−1.000
	mean	−1.000	−1.000	−1.000	−1.000	−1.000
	worst	−1.000	−1.000	−1.000	−1.000	−1.000
	std. dev	0.0E+00	0.0E+00	0.0E+00	0.0E+00	NA
g13/0.0539498	best	0.0539498	NA	0.053986	0.053957	0.53950
	mean	0.1437424	NA	0.166385	0.067543	0.054716
	worst	0.4397026	NA	0.468294	0.216915	0.440825
	std. dev	1.6E−01	NA	1.8E−01	3.1E−02	NA

Farmani 和 Wright[10] 提出的 SAFF 算法实质是一种自适应惩罚函数法. SMES[18] 算法使群体中违反约束程度最小的不可行解以一定的概率进入到下一代种群中,这对维持种群的多样性具有很重要的作用. Runarsson 和 Yao[19] 提出的随机排序法是目前最经典的约束优化算法之一,它通过定义概率 p_f 来协调目标函数和惩罚函数对个体适应度的影响,并采用冒泡排序的方法来实现选择操作. 文献[20]提出了一种新的用于解决约束优化问题的人工免疫响应进化策略,表现出良好的收敛速度和求解精度.

5.2　实验结果分析及算法性能评价

从表 5 可以看出,除 g10 外,新算法(COEA/OED)在其它的 12 个测试函数中达到了最优解.

就寻优精度而言,从表 6 可以看出,对于高维多峰测试函数 g02,只有 COEA/OED 算法找到了精确的最优解;同 SAFF 算法相比,COEA/OED 算法在函数 g02、g04、g05、g06、g07、g09、g10 的最优结果、平均结果和最差结果方面明显优于 SAFF 算法,对于函数 g01、g03、g08、g11、g12,COEA/OED 算法取得了与 SAFF 算法相似的结果;同 SMES 算法相比,COEA/OED 算法在函数 g02、g05、g07、g09、g10、g13 的最优结果、平均结果和最差结果方面均占优,对于函数 g01、g03、g04、g06、g08、g11、g12, COEA/OED 算法取得了与 SMES 算法相似的结果;同 RY 算法相比,COEA/OED 算法在函数 g02、g07、g10 的最优结果、平均结果和最差结果方面优于 RY 算法,对于函数 g01、g03、g04、g06、g08、g11、g12, COEA/OED 算法取得了与 RY 算法相似的结果;同 AIRCES 算法相比,COEA/OED 算法在函数 g02 的最优结果、平均结果和最差结果方面均优于 AIRCES 算法,对于函数 g06、g09,COEA/OED 和 AIRCES 算法都同时找到了最优解,但 COEA/OED 算法得到平均结果和最差结果方面优于 AIRCES 算法,对函数 g01、g03、g04、g08、g11、g12、g13,COEA/OED 和 AIRCES 算法得到了相似的结果.

此外,就计算代价而言,对同一测试问题,SAFF 算法所需适应值函数评价次数为 1400000 次,RY、AIRCES 算法需评价 350000 次,而 COEA/OED 和 SMES 算法只需要评价 240000 次.

从以上比较研究可以看出,无论是在寻优精度还是在计算代价上,COEA/OED 算法整体优于其它 4 种算法,表现了良好的寻优性能.

6　结　论

全局搜索能力和约束处理技术是利用进化算法求解约束优化问题最为关键的两个问题. 针对这些问题,本文提出了一种新的基于正交实验设计的约束优化进化算法. 新算法具有以下优点:

(1) 利用正交实验设计方法设计了一种新的多父体正交交叉算子,新的交叉算子能够有效利用多个父代个体所携带的信息产生新的具有代表性的子代个体,以更好地搜索解空间和维持种群的多样性,使种群在进化的过程中尽可能收敛到最优解.

(2) 针对群体进化过程中所处不同阶段的特点,利用不同的比较个体优、劣的准则来指导选择操作,使群体逐步地向可行的最优解逼近.

(3) 通过对 13 个测试函数的优化结果表明,针对不同类型的约束优化问题,新算法参数比较固定,不需要针对不同的优化问题来过多地调整实验参数,有利于算法的推广和应用. 对该算法进行完善和推广是本文的进一步工作.

参 考 文 献

[1] Coello C C A. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. Computation Methods in Applied Mechanics and Engineering, 2002, 191(11-12): 1245-1287

[2] Wang Yong, Cai Zi-xing, Zhou Yu-Ren at al. Constrained optimization evolutionary algorithms. Journal of Software, 2009, 20(1): 11-29(in Chinese)
(王勇,蔡自兴,周育人等. 约束优化进化算法. 软件学报, 2009, 20(1): 11-29)

[3] Davis L. Handbook of Genetic Algorithm. New York: van Nostrand, Reinhold, 1991

[4] Chen Guo-Liang, Wang Xi-Fa, Zhuang Zhen-Quan et al. Genetic algorithm and its applications. Beijing: People's Post & Telecommunications Publishing House, 1996(in Chinese)
(陈国良,王熙法,庄镇泉等. 遗传算法及其应用. 北京: 人民邮电出版社, 1996)

[5] Zhang Q F, Leung Y W. Orthogonal genetic algorithm for multimedia multicast routing. IEEE Transactions on Evolutionary Computation,1999, 3(1): 53-62

[6] Leung Y W, Wang Yu-Ping. An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Transactions on Evolutionary Computation, 2001, 5(1): 41-53

[7] Wang Yu-Ping, Dang Chuang-Yin. An evolutionary algorithm for global optimization based on level-set evolution and latin squares. IEEE Transactions on Evolutionary Computation, 2007, 11(5): 579-595

[8] Zhang Ling, Zhang Bo. Good point set based genetic algorithm. Chinese Journal of Computers, 2001, 24(9): 917-922 (in Chinese)
(张铃,张钹. 佳点集遗传算法. 计算机学报, 2001, 24(9): 917-922)

[9] Hamida S B, Schoenauer M. ASCHEA: New results using adaptive segregational constraint handling//Proceedings of the Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press, 2002: 884-889

[10] Farmani R, Wright J A. Self-adaptive fitness formulation for constrained optimization. IEEE Transactions on Evolutionary Computation, 2003, 7(5): 445-455

[11] Yu J X, Yao X, Choi C, Gou G. Materialized view selection as constrained evolutionary optimization. IEEE Transactions on Systems,Man and Cybernetics(C), 2003, 33(4): 458-467

[12] Zhou Yu-Ren, Li Yuan-Xiang, Wang Yong, Kang Li-Shan. A Pareto strength evolutionary algorithm for constrained optimization. Journal of Software, 2003, 14(7): 1243-1249(in Chinese)
(周育人,李元香,王勇,康立山. Pareto 强度值进化算法求解约束优化问题. 软件学报, 2003, 14(7): 1243-1249)

[13] Cai Zi-Xing, Wang Yong. A multiobjective optimization-based evolutionary algorithm for constrained optimization. IEEE Transactions on Evolutionary Computation, 2006,10 (6): 658-675

[14] Wang Yong, Cai Zi-Xing, Guo Guan-Qi, Zhou Yu-Ren. Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. IEEE Transactions on Systems, Man, and Cybernetics (B), 2007, 37(3): 560-575

[15] Tsutsui S, Yamamura M, Higuchi T. Multi-parent recombination with simplex crossover in real-coded genetic algorithms//Proceedings of the Genetic and Evolutionary Computing Conference. San Mateo: Morgan Kaufmann Publishers, 1999: 657-664

[16] Wang Yong, Cai Zi-Xing, Zhou Yu-Ren, Zeng Wei. An adaptive trade-off model for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 80-92

[17] Runarsson T P, Yao X. Search biases in constrained evolutionary optimization. IEEE Transactions on Systems, Man, Cybernetics (C), 2005, 35(2): 233-243

[18] Efrén Mezura-Montes, Coello C C A. A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Transactions on Evolution Computation, 2005, 9(1): 001-017

[19] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation, 2000, 4(3): 284-294

[20] Gong Mao-Guo, Jao Li-Cheng, Du Hai-Feng et al. A novel evolutionary strategy based on artificial immune response for constrained optimizations. Chinese Journal of Computers, 2007, 30(1): 37-47(in Chinese)
(公茂果,焦李成,杜海峰等. 用于约束优化的人工免疫响应进化策略. 计算机学报, 2007, 30(1): 37-47)

附 录. 约束优化测试函数.

g01

Minimize: $f(\boldsymbol{x})=5\sum_{i=1}^{4}x_i-5\sum_{i=1}^{4}x_i^2-\sum_{i=5}^{13}x_i$

subject to:

$g_1(\boldsymbol{x})=2x_1+2x_2+x_{10}+x_{11}-10\le 0$,

$g_2(\boldsymbol{x})=2x_1+2x_3+x_{10}+x_{12}-10\le 0$,

$g_3(\boldsymbol{x})=2x_2+2x_3+x_{11}+x_{12}-10\le 0$,

$g_4(\boldsymbol{x})=-8x_1+x_{10}\le 0$,

$g_5(\boldsymbol{x})=-8x_2+x_{11}\le 0$,

$g_6(\boldsymbol{x})=-8x_3+x_{12}\le 0$,

$g_7(\boldsymbol{x})=-2x_4-x_5+x_{10}\le 0$,

$g_8(\boldsymbol{x})=-2x_6-x_7+x_{11}\le 0$,

$g_9(\boldsymbol{x})=-2x_8-x_9+x_{12}\le 0$,

$0\le x_i\le 1(i=1,2,\cdots,9)$, $0\le x_i\le 100(i=10,11,12)$, $0\le x_{13}\le 1$.

已知最优解为 $\boldsymbol{x}^*=(1,1,1,1,1,1,1,1,1,3,3,3,1)$,

$f(\boldsymbol{x}^*)=15$. 约束条件 $g_1, g_2, g_3 g_7, g_8$ 和 g_9 活跃.

g02

Maximize：$f(\boldsymbol{x})=\left|\dfrac{\sum_{i=1}^{n}\cos^4(x_i)-2\prod_{i=1}^{n}\cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}}\right|$

subject to：

$g_1(\boldsymbol{x})=0.75-\prod_{i=1}^{n} x_i \le 0$,

$g_2(\boldsymbol{x})=\sum_{i=1}^{n} x_i-7.5n \le 0$,

$0\le x_i\le 10(i=1,2,\cdots,n), n=20$.

全局最优解未知，目前公布的最好结果为 $f(\boldsymbol{x}^*)=0.803619$. 约束条件 g_1 几乎活跃($g_1=-10^{-8}$).

g03

Maximize：$f(\boldsymbol{x})=(\sqrt{n})^n\prod_{i=1}^{n} x_i$

subject to：

$h(\boldsymbol{x})=\sum_{i=1}^{n} x_i^2-1=0, 0\le x_i\le 1(i=1,2,\cdots,n), n=10$.

已知最优解为 $x_i^*=1/\sqrt{n}(i=1,2,\cdots,n), f(\boldsymbol{x}^*)=1$.

g04

Minimize：$f(\boldsymbol{x})=5.3578547x_3^2+0.8356891x_1x_5+37.293239x_1-40792.141$

subject to：

$g_1(\boldsymbol{x})=85.334407+0.0056858x_2x_5+0.0006262x_1x_4+0.0022053x_3x_6-92\le 0$,

$g_2(\boldsymbol{x})=-85.334407-0.0056858x_2x_5-0.0006262x_1x_4+0.0022053x_3x_6\le 0$,

$g_3(\boldsymbol{x})=80.51249+0.0071317x_2x_5+0.0029955x_1x_2+0.0021813x_3^2-110\le 0$,

$g_4(\boldsymbol{x})=-80.51249-0.0071317x_2x_5-0.0029955x_1x_2-0.0021813x_3^2+90\le 0$,

$g_5(\boldsymbol{x})=9.300961+0.0047026x_3x_5+0.0012547x_1x_3+0.0019085x_3x_4-25\le 0$,

$g_6(\boldsymbol{x})=-9.300961-0.0047026x_3x_5-0.0012547x_1x_3-0.0019085x_3x_4+20\le 0$,

$78\le x_1\le 102$, $33\le x_2\le 45$, $27\le x_i\le 45(i=3,4,5)$.

已知最优解为 $\boldsymbol{x}^*=(78, 33, 29.995256025682, 45, 36.775812905788)$, $f(\boldsymbol{x}^*)=-30665.539$. 约束条件 g_1 和 g_6 活跃.

g05

Minimize：$f(\boldsymbol{x})=3x_1+0.000001x_1^3+2x_2+(0.000002/3)x_2^3$

subject to：

$g_1(\boldsymbol{x})=-x_4+x_3-0.55\le 0$,

$g_2(\boldsymbol{x})=-x_3+x_4-0.55\le 0$,

$h_3(\boldsymbol{x})=1000\sin(-x_3-0.25)+1000\sin(-x_4-0.25)+894.8-x_1=0$

$h_4(\boldsymbol{x})=1000\sin(x_3-0.25)+1000\sin(x_3-x_4-0.25)+894.8-x_2=0$

$h_5(\boldsymbol{x})=1000\sin(x_4-0.25)+1000\sin(x_4-x_3-0.25)+1294.8=0$

$0\le x_1\le 1200$, $0\le x_2\le 1200$, $-0.55\le x_3\le 0.55$, $-0.55\le x_4\le 0.55$.

已知最优解为 $\boldsymbol{x}^*=(679.9453, 1026.067, 0.1188764, -0.3962336)$, $f(\boldsymbol{x}^*)=5126.4981$.

g06

Minimize：$f(\boldsymbol{x})=(x_1-10)^3+(x_x-20)^3$

subject to：

$g_1(\boldsymbol{x})=-(x_1-5)^2-(x_2-5)^2+100\le 0$,

$g_2(\boldsymbol{x})=(x_1-6)^2+(x_2-5)^2-82.81\le 0$,

$13\le x_1\le 100, 0\le x_2\le 100$.

已知最优解为 $\boldsymbol{x}^*=(14.095, 0.84296)$, $f(\boldsymbol{x}^*)=-6961.81388$. 两个约束条件均活跃.

g07

Minimize：$f(\boldsymbol{x})=x_1{}^2+x_2{}^2+x_1x_2-14x_1-16x_2+(x_3-10)^2+4(x_4-5)^2+(x_5-3)^2+2(x_6-1)^2+5x_7^2+7(x_8-11)^2+2(x_9-10)^2+(x_{10}-7)^2+45$

subject to：

$g_1(\boldsymbol{x})=-105+4x_1+5x_2-3x_7+9x_8\le 0$,

$g_2(\boldsymbol{x})=10x_1-8x_2-17x_7+2x_8\le 0$,

$g_3(\boldsymbol{x})=-8x_1+2x_2+5x_9-2x_{10}-12\le 0$,

$g_4(\boldsymbol{x})=3(x_1-2)^2+4(x_2-3)^2+2x_3^2-7x_4-120\le 0$,

$g_5(\boldsymbol{x})=5x_1^2+8x_2+(x_3-6)^2-2x_4-40\le 0$,

$g_6(\boldsymbol{x})=x_1^2+2(x_2-2)^2-2x_1x_2+14x_5-6x_6\le 0$,

$g_7(\boldsymbol{x})=0.5(x_1-8)^2+2(x_2-4)^2+3x_5^2-x_6-30\le 0$,

$g_8(\boldsymbol{x})=-3x_1+6x_2+12(x_9-8)^2-7x_{10}\le 0$,

$-10\le x_i\le 10(i=1,\cdots,10)$.

已知最优解为 $\boldsymbol{x}^*=(2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$, $f(\boldsymbol{x}^*)=24.3062091$. 约束条件 g_1, g_2, g_3, g_4, g_5 和 g_6 活跃.

g08

Minimize：$f(\boldsymbol{x})=\dfrac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)}$

subject to：

$g_1(\boldsymbol{x})=x_1^2-x_2+1\le 0$,

$g_2(\boldsymbol{x})=1-x_1+(x_2-4)^2\le 0$,

$0\le x_1\le 10, 0\le x_2\le 10$.

已知最优解为 $\boldsymbol{x}^*=(1.2279713, 4.2453733)$, $f(\boldsymbol{x}^*)=0.095825$. 最优解位于可行域内.

g09

Minimize：$f(\boldsymbol{x})=(x_1-10)^2+5(x_2-12)^2+x_3^4+3(x_4-11)^2+10x_5^6+7x_6^2+x_7^4-4x_6x_7-10x_6-8x_7$

subject to：

$g_1(\boldsymbol{x})=-127+2x_1^2+3x_2^4+x_3+4x_4^2+5x_5\le 0$,

$g_2(\boldsymbol{x})=-282+7x_1+3x_2+10x_3^2+x_4-x_5\le 0$,

$g_3(\boldsymbol{x})=-196+23x_1+x_2^2+6x_6^2-8x_7\le 0$,

$g_4(\boldsymbol{x})=4x_1^2+x_2^2-3x_1x_2+2x_3^2+5x_6-11x_7\le 0$,

$-10\le x_i\le 10(i=1,2,\cdots,7)$.

已知最优解为 $\boldsymbol{x}^*=(2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.1038131, 1.594227)$, $f(\boldsymbol{x}^*)=680.630573$.

约束条件 g_1 和 g_4 活跃.

g10

Minimize: $f(\boldsymbol{x})=x_1+x_2+x_3$

subject to:

$g_1(\boldsymbol{x})=-1+0.0025(x_4+x_6)\le 0$,

$g_2(\boldsymbol{x})=-1+0.0025(x_5+x_7-x_4)\le 0$,

$g_3(\boldsymbol{x})=-1+0.01(x_8-x_5)\le 0$,

$g_4(\boldsymbol{x})=-x_1x_6+833.33252x_4+100x_1-83333.333\le 0$,

$g_5(\boldsymbol{x})=-x_2x_7+1250x_5+x_2x_4-1250x_4\le 0$,

$g_6(\boldsymbol{x})=-x_3x_8+1250000+x_3x_5-2500x_5\le 0$,

$100\le x_1\le 10000, 1000\le x_i\le 10000(i=2,3)$,

$10\le x_i\le 1000(i=4,\cdots,8)$.

已知最优解为 $\boldsymbol{x}^*=(579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979)$, $f(\boldsymbol{x}^*)=7049.3307$. 约束条件 g_1, g_2 和 g_3 活跃.

g11

Minimize: $f(\boldsymbol{x})=x_1^2+(x_2-1)^2$

subject to:

$h(\boldsymbol{x})=x_2-x_1^2=0$,

$-1\le x_1\le 1, -1\le x_2\le 1$.

已知最优解为 $\boldsymbol{x}^*=(\pm 1/\sqrt{2}, 1/2)$, $f(\boldsymbol{x}^*)=0.75$.

g12

Maximize: $f(\boldsymbol{x})=(100-(x_1-5)^2-(x_2-5)^2-(x_3-5)^2)/100$

subject to:

$g(\boldsymbol{x})=(x_1-p)^2+(x_2-q)^2+(x_3-r)^2-0.0625\le 0$,

$0\le x_i\le 10(i=1,2,3)$, $p,q,r=1,2,\cdots,9$.

可行域由 9^3 个离散的球体组成. 向量 (x_1,x_2,x_3) 可行当且仅当存在 p,q,r 满足上述约束条件. 已知最优解为 $\boldsymbol{x}^*=(5,5,5)$, $f(\boldsymbol{x}^*)=1$. 最优解位于可行域内.

g13

Minimize: $f(\boldsymbol{x})=e^{x_1x_2x_3x_4x_5}$

subject to:

$h_1(\boldsymbol{x})=x_1^2+x_2^2+x_3^2+x_4^2+x_5^2-10=0$,

$h_2(\boldsymbol{x})=x_2x_3-5x_4x_5=0$,

$h_2(\boldsymbol{x})=x_1^3+x_2^3+1=0$,

$-2.3\le x_i\le 2.3(i=1,2)$, $-3.2\le x_i\le 3.2(i=3,4,5)$.

已知最优解为 $\boldsymbol{x}^*=(-1.717143, 1.595709, -1.827247, 0.7636413, -0.763645)$, $f(\boldsymbol{x}^*)=0.0539498$.

CAI Zi-Xing, born in 1938, professor, Ph.D. supervisor. His research interests include artificial intelligence, intelligent computation and intelligent control.

JIANG Zhong-Yang, born in 1982, M.S.. His research interests include evolutionary algorithm, global optimization and constrained optimization.

WANG Yong, born in 1980, Ph.D. candidate, lecturer. His research interests include evolutionary computation, multiobjective optimization and constrained optimization.

LUO Yi-Dan, born in 1983, M.S.. Her research interests include evolutionary algorithm, immune algorithm.

Background

This work is supported in part by the National Basic Scientific Research Founds under grant No. A1420060159, in part by the National Natural Science Foundation of China under grant Nos. 60805027 and 90820302, the Research Fund for the Doctoral Program of Higher Education under grant No. 200805330005, and the Graduate Innovation Fund of Hunan Province under grant No. CX2009B039.

Constrained optimization problems (COPs) are very commonly faced in many science and engineering disciplines, such as the navigation of the multi-mobile agents. To solve COPs, evolutionary algorithms (EAs) and constraint-handling techniques are two basic aspects. The orthogonal design is an experimental design method, which has been widely applied in scientific research, manufacturing, agricultural experiments and quality management. This method, which can be used for planning experiments, provides an efficient way to find a near-best sample for the multi-factor experiments. In recent years, as some major steps of a GA (such as the crossover operator) can be considered to be "experiments", orthogonal design is used to solve global optimization problems with good performance. Nowadays, there is no report about using orthogonal method to design multi-patent crossover operator to solve constrained optimization problems effectively. This paper proposes a new multi-patent orthogonal crossover operator and new constraint-handling technique based on constrained optimization evolutionary algorithm, and provides new tools for practical applications.