

修正免疫克隆约束多目标优化算法*

尚荣华⁺, 焦李成, 胡朝旭, 马晶晶

(智能感知与图像理解教育部重点实验室(西安电子科技大学), 陕西 西安 710071)

Modified Immune Clonal Constrained Multi-Objective Optimization Algorithm

SHANG Rong-Hua⁺, JIAO Li-Cheng, HU Chao-Xu, MA Jing-Jing

(Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education (Xidian University), Xi'an 710071, China)

+ Corresponding author: E-mail: rhshang@mail.xidian.edu.cn

Shang RH, Jiao LC, Hu CX, Ma JJ. Modified immune clonal constrained multi-objective optimization algorithm. *Journal of Software*, 2012, 23(7): 1773-1786 (in Chinese). <http://www.jos.org.cn/1000-9825/4108.htm>

Abstract: This paper proposes a modified immune clonal constrained multi-objective algorithm for constrained multi-objective optimization problems. By introducing a new constrained handling strategy to modify the objective values of individuals, the proposed algorithm optimizes the individuals with the modified objective values and stores the non-dominated feasible individuals in an elitist population. In the optimization process, the algorithm not only preserves the non-dominated feasible individuals, but also utilizes the infeasible solutions with smaller constrained violation values. Meanwhile the new algorithm introduces the overall cloning strategy to improve the distribution diversity of the solutions. The proposed algorithm has been tested on several popular constrained test problems and compared with the other two constrained multi-objective optimization algorithms. The results show that the optimal solutions of the proposed algorithm are more diverse than the other two algorithms and better in terms of convergence and uniformity.

Key words: constrained multi-objective optimization; immune clonal; constrained handling strategy; constrained violation value; non-dominated solutions

摘要: 针对约束多目标优化问题,提出修正免疫克隆约束多目标优化算法.该算法通过引进一个约束处理策略,用一个修正算法对个体的目标函数值进行修正,并对修正后的目标函数值采用免疫克隆算法进行优化,用一个精英种群对可行非支配解进行存储.该算法在优化过程中,既保留了非支配可行解,也充分利用了约束偏离值小的非可行解,同时引进整体克隆策略来提高解分布的多样性.通过对约束多目标问题的各项性能指标的测试以及和对比算法的比较可以看出:该算法在处理约束多目标优化测试问题时,所得解的多样性得到了一定的提高.同时,解的收敛性和均匀性也得到了改进.

关键词: 约束多目标优化;免疫克隆;约束处理策略;约束偏离值;非支配解

中图法分类号: TP18 文献标识码: A

* 基金项目: 国家自然科学基金(61001202, 61003199); 中国博士后科学基金(201104658, 20090451369, 200801426, 20080431228); 陕西省自然科学基金基础研究计划(2009JQ8015, 2010JQ8023); 国家教育部博士点基金(20100203120008, 20090203120016, 200807010003); 高等学校学科创新引智计划(B07048); 教育部“长江学者和创新团队发展计划”(IRT1170)

收稿时间: 2011-03-11; 修改时间: 2011-05-06; 定稿时间: 2011-08-09

不失一般性,首先给出约束多目标优化问题的一个标准形式^[1]:

$$\begin{aligned} \min \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, q \\ & h_j(\mathbf{x}) = 0, j = q + 1, \dots, m \\ & \mathbf{x} = (x_1, \dots, x_n) \in \mathbf{X} \\ & \mathbf{X} = \{(x_1, x_2, \dots, x_n) \mid l_i < x_i < u_i\} \\ & \mathbf{l} = (l_1, l_2, \dots, l_n), \mathbf{u} = (u_1, u_2, \dots, u_n) \end{aligned} \quad (1)$$

其中, $\mathbf{x}=(x_1, \dots, x_n)$ 是 n 维决策向量, x_j 是第 j 个决策变量; \mathbf{X} 为决策变量空间, \mathbf{u} 和 \mathbf{l} 为决策变量的上下界; $\mathbf{F}(\mathbf{x})$ 为目标函数向量, $f_i(\mathbf{x})$ 为第 i 个目标函数值; $g_i(\mathbf{x}) \leq 0 (i=1, \dots, q)$ 为第 i 个不等式约束, $h_j(\mathbf{x})=0 (j=q+1, \dots, m)$ 为第 j 个等式约束. 通常, 等式约束将通过公式(2)转换为不等式约束.

$$|h(\mathbf{x})| - \varepsilon \leq 0 \quad (2)$$

其中, ε 为一个小的松弛度值, 根据情况通常取 0.001 或者 0.0001. 当 $\mathbf{x} \in \mathbf{X}$ 满足所有的约束条件 $g_i(\mathbf{x}) \leq 0 (i=1, \dots, q)$ 和 $h_j(\mathbf{x})=0 (j=q+1, \dots, m)$ 时, 称 \mathbf{x} 为可行解, 所有可行解的集合记为 \mathbf{X}_f .

过去几年中, 一些学者已经提出了许多经典的约束多目标优化算法. 下面首先对约束条件的处理方法进行介绍, 之后回顾几种经典的约束多目标优化算法.

通常, 对约束条件的处理有两种方法: 约束偏离值方法和约束偏离度方法.

1) 约束偏离值方法^[1-9]

约束偏离值因为实现简单, 使用最多. 标准约束偏离值的计算方式如公式(3)所示:

$$C(\mathbf{x}) = \sum_{j=1}^q w_j \frac{c_j(\mathbf{x})}{c_j^{\max}} \quad (3)$$

其中, $c_j(\mathbf{x})$ 为第 j 个约束偏离值, 见公式(4); c_j^{\max} 为第 j 个约束偏离值的最大值, 见公式(5); w_j 为第 j 个约束函数的加权值, 通常 $w_j = \frac{1}{q}$; $C(\mathbf{x})$ 为个体 \mathbf{x} 的约束偏离值.

$$c_j(\mathbf{x}) = \begin{cases} \max(0, g_j(\mathbf{x})), & j = 1, \dots, q \\ \max(0, |h_j(\mathbf{x})| - \varepsilon), & j = q + 1, \dots, m \end{cases} \quad (4)$$

$$c_j^{\max} = \max_{\mathbf{x}} c_j(\mathbf{x}) \quad (5)$$

对约束偏离值的处理方法分为两种: 1) 将约束偏离值加到每个个体的目标函数值上, 从而将约束多目标优化问题简化为非约束多目标优化问题进行处理; 2) 将约束偏离值看作一维目标函数值, 从而使目标函数维数成为 $k+1$ 维, 之后对新的目标函数空间进行非约束多目标优化.

2) 约束偏离度方法^[10]

这种方法用的较少. 用变量 v 标记个体 \mathbf{x} 不满足的约束条件的个数, 处理 v 的方法类似于处理上述方法 1) 中约束偏离值的方法.

接下来, 主要介绍几种已有的约束多目标优化算法.

在文献[1]中, Venkatraman 等人将约束优化问题分为两个步骤进行优化: 第 1 步寻找可行解, 忽略目标函数值, 仅根据每个个体的约束偏离值对其进行等级分配, 当种群中出现有可行解时进入第 2 步; 第 2 步将约束值空间和目标函数值空间合为一个空间, 将问题转化为无约束的多目标问题进行优化. 该方法比较容易理解和实现, 但当约束函数条件较多时, 优化的目标函数空间迅速变大, 从而严重影响了算法的性能. 在文献[2]中, Deb 等人在 NSGA-II 中采用一种新的选择策略, 分为 3 种情况: 1) 两个个体都为可行解时, 选取非支配的个体; 2) 两个个体都为非可行解时, 选取约束偏离值小的个体; 3) 两个个体中, 当一个为可行解, 一个为非可行解时, 选取可行解. 这种选择策略中, 所有的可行解支配所有的非可行解. Li 等人在文献[3]中采用了同样的选择策略, 并通过修正 PSO 算法来解决约束多目标优化问题. 文献[4]中, Geng 等人提出了一个对非支配解的修正算子 SBMS. 修正非支配个体时, 需要选择一个离其最近的可行非支配参考点, 同时, 在 NSGA-II 和约束选择策略的基础上提出了

SMBS_MOEA 算法.Chafekar 等人在文献[5]中提出两个算子:1) OEGADO 算子,将约束多目标优化问题转换为约束单目标优化问题,每个目标函数都有独立种群,用 GA 优化各个目标函数的种群;2) OSGADO 算子,用 GA 依次优化同一种群中的各个目标函数.在文献[6]中,Young 分配给每个个体两个非支配等级:目标函数空间的非支配等级 R_O 和约束函数空间的非支配等级 R_C .每个个体的混合非支配等级如公式(6)所示,其中, α 为可行解在种群中所占的比例.

$$R_B = \alpha R_O + (1 - \alpha) R_C \quad (6)$$

在文献[7]中,Liu 首先给每个个体分配两个变量: $R(x)$ 和 $C(x)$. $R(x)$ 为个体 x 在种群中被支配的抗体的个数加 1, $C(x)$ 为公式(3)所求出的约束偏离值.之后为每个个体赋予一个适应度值,如公式(7)所示.其中,当种群中非支配解的个数大于所要得到种群的个数时,则 $w_1=1, w_2=0$;如果不大于所要得到的种群的个数,则 $w_1=0, w_2=1$.此时,所有的支配解进入下一代进化.

$$fitness = \frac{w_1}{R(x)} + \frac{w_2}{C(x)} \quad (7)$$

在文献[8]中,Jaddan 用罚函数法将个体在目标函数空间和约束函数空间的两个非支配等级加入个体的目标函数空间中,同时加入了个体的约束偏离值,并引入一个自适应选择变量.在文献[9]中,Singh 在模拟退火算法的基础上,用一种新的选择策略:1) 当 x_{old} 为非可行解、 x_{new} 为可行解时, x_{new} 优于 x_{old} ;2) 当 x_{old} 为可行解、 x_{new} 为非可行解时, x_{new} 被赋予一个被接受的概率;3) 当 x_{new} 和 x_{old} 都为非可行解时,选择约束偏离值小的个体.在文献[10]中,Isaacs 等人将每个个体违反约束的个数合并到目标函数空间中,在每一代中,将解分为可行解集 S_f 和非可行解集 S_{inf} .在合并后的目标函数空间中,对每个解集中的每个个体进行非支配排序.在进行选取下一代时,保留一定的选择份额给非可行解.

以上算法为约束多目标优化问题的解提供了多种选择思路:在处理约束条件时,从最初的可行解支配所有非可行解,到约束偏离值小的非可行解也可能支配可行解;使用的进化算法,从 GA,NSGA-II 再到 PSO 等.但是,各种算法求出的最优解集的多样性和逼近性仍有待提高.文献[11]中,Woldesenbet 等人提出的约束多目标进化算法采用了新的约束处理策略,通过一定数目的惩罚项对目标函数值进行修正,并从修正后的目标函数值中选取一定数目的非支配解集作为父代,然后进行下一代优化.同时,还有一个精英种群来储存进化过程中出现的非支配可行解.这种方法巧妙地解决了可行解和非可行解的选择问题,在进化种群中,既保留了可行非支配解,又保留了约束偏离值较小且目标函数值较小的非可行解.

免疫克隆算法^[12,13]在求解多目标优化问题时,每次选取一定数目的非支配解集进入下一代优化,并且通过克隆选择策略使得解集能够保持好的多样性和收敛性.由于文献[11]中所提出的约束处理策略能够很好地应用到免疫克隆算法中,所以本文根据文献[11]中的约束处理策略,提出一种修正免疫克隆约束多目标优化算法.

然而,文献[11]中的约束处理策略也存在不足之处,即通过该约束处理策略,在种群中无可行非支配解时,该策略就已经使种群朝着搜索可行解的方向进化.因此,本文没有采取两步的策略.这一点在第 1 节从理论分析和实验两个方面已给出了说明.同时,免疫克隆算法在求解约束多目标优化问题时,大多数文献的做法是将已求得的可行非支配解进行克隆和变异等操作.而本文算法中,针对约束多目标优化问题,通过种群 *ModNonPop* 存储修正后的非支配解,通过种群 *FeaNonPop* 存储可行非支配解集,在进行克隆操作时,我们经过大量的实验发现,如果将种群 *FeaNonPop* 也进行克隆,优化过程中的可行非支配解集规模几乎将成倍上升,给后面的选择更新操作造成了很大的压力,从而使算法花费大量时间在选择更新操作上.所以,本文并没有将种群 *FeaNonPop* 选入克隆操作,从而提高了本文算法的执行效率和搜索能力.通过对约束多目标问题的测试,并与文献[11]中的算法和 NSGA-II 进行对比,表明本文算法在处理约束多目标优化测试问题时,所得解的多样性、收敛性和均匀性均得到了很大提高,说明本文算法具有很好的多样性和均匀性保持能力以及较强的收敛能力.

本文第 1 节详细介绍本文所提出的免疫克隆约束多目标优化算法.第 2 节给出约束多目标经典测试函数,之后用本文算法、NSGA-II 约束多目标优化算法和文献[11]中约束多目标优化算法优化测试问题,并对测试结果进行对比分析.第 3 节总结本文的工作.

1 免疫克隆约束多目标优化算法

本文提出的修正免疫克隆约束多目标优化算法仍采用免疫克隆算法的框架,同时在优化过程中采用约束处理策略对目标函数值进行修正.并且,设置一个大小为 N 的精英种群,用来存储可行非支配解集 $FeaNonPop$.本文算法的主要目的是选取可行非支配解集 $FeaNonPop$ 和目标函数值修正后的非支配解集 $ModNonPop$.

文献[11]中的算法等采用两个步骤:1) 寻找可行解;2) 寻找可行非支配解.本文算法并没有采用这种方法,原因如下:

- i) 理论分析:通过对以下本文所采用的约束处理策略的分析,当种群中无可行解时,约束处理策略给种群施加一个朝着可行解优化的趋势,使种群寻找可行解,这本身就替代了以上所提到的第 1)步;
- ii) 实验结果表明:每次实验的第 1 步都能找到可行解,从而跳过第 1)步.

通过以上分析,本文直接采用免疫克隆算法的框架进行优化.同时,在不同的情况下,通过约束处理策略自适应地调整种群进化趋势,并针对约束多目标优化问题的特点,设计了克隆、交叉、变异和选择等算子.

1.1 克隆操作

克隆算子在本文算法中对解的多样性分布和逼近性起着重要的作用,如文献[12,13].本文算法在对父代抗体种群进行克隆时也采取整体克隆的方法,对父代的优势抗体种群进行 n 倍克隆, $Pop=\{Pop, \dots, Pop\}$.这样省去了为父代中的每个抗体分配适应度值的操作,使得算法更加简单.本文主要考虑两种克隆方法:

1) 克隆非支配解集 $ModNonPop$ 的同时,将精英种群中的可行非支配解集 $FeaNonPop$ 复制进克隆后的种群中,如式(8)所示:

$$Pop = \underbrace{\{ModNonPop, \dots, ModNonPop\}}_n \underbrace{\{FeaNonPop, \dots, FeaNonPop\}}_k \quad (8)$$

其中,将非支配解集 $ModNonPop$ 克隆 n 倍,将可行非支配解集 $FeaNonPop$ 克隆 k 倍.

2) 仅仅克隆非支配解集 $ModNonPop$.此时,精英种群只起到一种存储可行非支配解集 $FeaNonPop$ 的作用,该克隆过程如式(9)所示:

$$Pop = \underbrace{\{MonNonPop, \dots, MonNonPop\}}_n \quad (9)$$

通过大量的实验结果表明:方法 1)将可行非支配解集 $FeaNonPop$ 克隆进种群 Pop ,对优化结果帮助不大.同时,由于可行非支配解集 $FeaNonPop$ 的引入,使得优化过程中的可行非支配解集规模几乎成倍上升,给后面的选择更新操作造成了很大的压力,从而使算法花费大量时间在选择更新操作上,即使当 k 为 1 时也是如此.因此采用方法 2),仅仅考虑非支配解集 $ModNonPop$.具体操作过程见表 1.

Table 1 Flow chart of the clone operation

表 1 整体克隆算子

算子: 整体克隆算子.
n_c : 克隆倍数
N : $MonNonPop$ 种群大小
Begin
$i=1; k=1;$
While $i \leq n_c$
For $j=1:1:N$
$Pop(k)=MonNonPop(j);$
$k=k+1;$
End
$i=i+1;$
End
$Pop.size=k;$
End

1.2 交叉、变异操作

本文采用文献[14]中模拟二进制交叉(SBX cross-over)算子(如公式(10)所示)和多项式变异(polynomial

mutation)算子(如公式(12)所示).

$$a'_{ik} = \begin{cases} 0.5[(1 + \beta_k)a_{ik} + (1 - \beta_k)a_{jk}], & \text{if } r(0,1) \geq 0.5 \\ 0.5[(1 - \beta_k)a_{ik} + (1 + \beta_k)a_{jk}], & \text{if } r(0,1) < 0.5 \end{cases} \quad (10)$$

其中,

$$\beta_k = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & \text{if } u(0,1) \geq 0.5 \\ [2(1-u)]^{-\frac{1}{\eta_c+1}}, & \text{if } u(0,1) < 0.5 \end{cases} \quad (11)$$

其中, $a_{ik}, a_{jk}(i \neq j, k=1, \dots, n)$ 是抗体 i, j 的第 k 个决策变量, r, u 是分布在 $[0, 1]$ 之间的随机数.

$$v'_k = v_k + \delta(u_k - l_k) \quad (12)$$

其中,

$$\delta = \begin{cases} [2u + (1 - 2u)(1 - \delta_1)^{\eta_m+1}]^{\frac{1}{\eta_m+1}}, & \text{if } u \leq 0.5 \\ 1 - [2(1 - u) + 2(u - 0.5)(1 - \delta_2)^{\eta_m+1}]^{\frac{1}{\eta_m+1}}, & \text{if } u > 0.5 \end{cases} \quad (13)$$

其中, $\delta_1 = (v_k - l_k) / (u_k - l_k), \delta_2 = (u_k - v_k) / (u_k - l_k), u$ 是 $[0, 1]$ 间分布的随机数, u_k 和 l_k 分别为 v_k 的上界和下界.

交叉变异操作见表 2.

Table 2 Flow chart of the crossover and mutation operations

表 2 交叉变异算子

算子. 交叉变异算子.
p_c : 交叉概率
p_m : 变异概率
N : Pop 种群大小
Begin
//交叉操作
For $i=1:2:N-1$
For $j=1:2:N-1$
If $random(\cdot) < p_c$
对个体 $Pop(i)$ 和 $Pop(i+1)$ 的第 j 维变量进行交叉操作;
End
End
End
//变异操作
For $i=1:1:N$
For $j=1:1:N$
If $random(\cdot) < p_m$
对个体 $Pop(i)$ 的第 j 维变量进行变异操作;
End
End
End
End

1.3 选择操作

本文算法的选择操作不像 NSGA-II 那样对种群每个个体分配非支配等级,然后根据非支配等级进行个体选择,而是和文献[12,13]中一样,直接在种群中选取非支配抗体,这样有利于简化算法.本文算法包括两个选择操作:选取可行非支配解集 $FeaNonPop$ 和选取目标函数值修正后的非支配解集 $ModNonPop$.这两个选择操作如第 1.3.1 节和第 1.3.2 节所示.

1.3.1 选取可行非支配解集 $FeaNonPop$

首先从当代进化种群 Pop 中根据每个个体的约束偏离值选取可行解集 $FeasiblePop$,之后将 $FeaNonPop$ 中的可行非支配解也复制进可行解集 $FeasiblePop$ 中,在可行解集 $FeasiblePop$ 中根据每个个体的目标函数值选取可行非支配解集 $FeaNonPop$.表 3 给出选取可行非支配解集的主要步骤.

Table 3 Flow chart of the selection of feasible non-dominated population**表 3** 可行非支配解集的选取操作

算子. 可行非支配解集的选取.	
1)	从种群 <i>Pop</i> 中选取可行解集 <i>FeasiblePop</i> ;
2)	将父代 <i>FeaNonPop</i> 中的个体复制进种群 <i>FeasiblePop</i> 中;
3)	从种群 <i>FeasiblePop</i> 中选取可行非支配解集 <i>FeaNonPop</i> .
步骤 3)的操作细节如下:	
<i>N:FeasiblePop</i> 种群的大小	
Begin	
将种群 <i>FeasiblePop</i> 中所有个体设为非支配解;	
For <i>i</i> =1:1: <i>N</i>	
For <i>j</i> =1:1: <i>N</i>	
If <i>FeasiblePop</i> (<i>j</i>)不是删除状态且 <i>i</i> != <i>j</i>	
If <i>FeasiblePop</i> (<i>i</i>)被 <i>FeasiblePop</i> (<i>j</i>)支配	
设定 <i>FeasiblePop</i> (<i>i</i>)为支配解;	
Break ;	
Else if <i>FeasiblePop</i> (<i>i</i>)与 <i>FeasiblePop</i> (<i>j</i>)相同	
设定 <i>FeasiblePop</i> (<i>i</i>)为删除状态;	
End	
End	
End	
End	
从 <i>FeasiblePop</i> 中选取非支配解.	
End	

1.3.2 选择目标函数值修正后的非支配解集 *ModNonPop*

首先,根据文献[11]中的约束处理策略,通过距离值(distance value)和惩罚项(two penalties)对当代进化种群 *Pop* 中每个个体的目标函数值进行修正,如公式(14)所示.然后,根据修正后的目标函数值,从种群 *Pop* 中选取非支配解集 *ModNonPop*.具体选择操作与表 3 类似.

$$f'_i(\mathbf{x}) = d_i(\mathbf{x}) + p_i(\mathbf{x}) \quad (14)$$

其中, $d_i(\mathbf{x})$ 为个体 \mathbf{x} 的第 i 维目标函数的距离值, $p_i(\mathbf{x})$ 为个体 \mathbf{x} 的第 i 维目标函数的惩罚项, $f'_i(\mathbf{x})$ 为个体 \mathbf{x} 修正后的第 i 个目标函数值.

以下简要介绍距离值和惩罚项的计算方法.

1) 距离值^[11]

根据种群中是否存在可行解,可以将距离值的求解分为两种情况:1) 不存在可行解时,距离值为非可行解的约束偏离值;2) 存在可行解时,距离值是目标函数和约束偏离值共同作用的结果.下面为距离值的数学计算公式:

$$d_i(\mathbf{x}) = \begin{cases} C(\mathbf{x}), & \text{if } \alpha = 0 \\ \sqrt{\tilde{f}_i(\mathbf{x})^2 + C(\mathbf{x})^2}, & \text{otherwise} \end{cases} \quad (15)$$

其中, $d_i(\mathbf{x})$ 为个体 \mathbf{x} 的第 i 维目标函数 $f_i(\mathbf{x})$ 的距离值, $C(\mathbf{x})$ 为个体 \mathbf{x} 的约束偏离值, α 为可行解的个数在种群中所占的比例. $\tilde{f}_i(\mathbf{x})$ 为个体 \mathbf{x} 的第 i 维目标函数 $f_i(\mathbf{x})$ 的归一化值,归一化过程如下所示:

$$\tilde{f}_i(\mathbf{x}) = \frac{f_i(\mathbf{x}) - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \quad (16)$$

其中, f_i^{\min} 为种群中第 i 个目标函数值的最小值, f_i^{\max} 为种群中第 i 个目标函数值的最大值.距离值的作用是:

1) 对可行解不进行惩罚,仅对非可行解进行惩罚;2) 在非可行解中,约束偏离值大的个体受到的惩罚就越大.

2) 惩罚项^[11]

可行解的惩罚项值为 0,下面讨论针对的都是非可行解.假定个体 \mathbf{x} 为非可行解,其惩罚项 $p_i(\mathbf{x})$ 的求解公式如式(17)所示:

$$p_i(\mathbf{x}) = (1 - \alpha)C'(\mathbf{x}) + \alpha\tilde{f}_i(\mathbf{x}) \quad (17)$$

其中, α 为可行解在种群中所占的比例; $\tilde{f}_i(\mathbf{x})$ 为个体 \mathbf{x} 的第 i 维目标函数 $f_i(\mathbf{x})$ 的归一化值; $C'(\mathbf{x})$ 为修正的约束偏离值,如式(18)所示:

$$C'(\mathbf{x}) = \begin{cases} 0, & \text{if } \alpha = 0 \\ C(\mathbf{x}), & \text{otherwise} \end{cases} \quad (18)$$

其中, $C(\mathbf{x})$ 为个体 \mathbf{x} 的约束偏离值.在 $\alpha=0$ 即种群中没有可行解时,惩罚项值为 0.此时,仅靠距离值对目标函数值进行修正;当 $\alpha>0$ 时,如果 α 较小即种群中可行解的个数较少,则惩罚项侧重于对约束函数空间的惩罚,有利于种群中的非可行解朝可行解的方向进化;如果 α 较大即种群中可行解的个数较多,则惩罚项侧重于对目标函数空间的惩罚,有利于种群中个体朝目标函数值最优的方向进化^[11].

1.4 算法的主要流程

本节给出修正免疫克隆约束多目标优化算法的主要流程,见表 4.

Table 4 Flow chart of modified immune clonal constrained multi-objective optimization algorithm

表 4 修正免疫克隆约束多目标优化算法的主要流程

算法. 修正免疫克隆约束多目标优化算法.	
Begin	
1)	初始化种群 Pop ,并设定迭代参数 $iger=1$;
2)	计算种群 Pop 的目标函数值、约束偏离度值;
3)	根据步骤 2)中的信息,对种群 Pop 中每个个体求解修正后的目标函数值;
4)	从种群 Pop 中选取可行非支配解集 $FeaNonPop$,并更新 $FeaNonPop$ 的大小;
5)	从种群 Pop 中选取非支配解集 $ModNonPop$,并更新 $ModNonPop$ 的大小;
	$iger=iger+1$;
While $iger \leq max_ger$	
6)	克隆非支配解集 $ModNonPop$,使 $Pop=\{ModNonPop, \dots, ModNonPop\}$;
7)	对种群 Pop 进行交叉、变异;之后合并以前的非支配解集 $ModNonPop$,使得 $Pop=\{Pop, ModNonPop\}$;
8)	进行步骤 2),步骤 3)~步骤 5)进行相同的操作.
	$iger=iger+1$;
End	
End	

1.5 算法的复杂度分析

设在每一代进化中,种群 $FeaNonPop$ 和 $ModNonPop$ 的规模都为 N ,克隆倍数为 c ,变量的维数为 n ,约束维数为 m ,目标函数维数为 k ,则:

- 在每次克隆种群 $ModNonPop$ 所用的复杂度为 $O(cN)$;
- 交叉操作所需复杂度为 $O(ncN/2)$;
- 变异操作所需复杂度为 $O(ncN)$;
- 计算种群 Pop 目标函数值和约束偏离值的时间复杂度为 $O(ncN)$;
- 合并种群 $ModNonPop$ 所需复杂度为 $O(nN+mN+kN)$;
- 修正种群 Pop 中个体的目标函数值所需复杂度为 $O(3k(c+1)N+2m(c+1)N)$;
- 选取并更新可行非支配解集所需复杂度为 $O((2c+6+kc+2k)N+k(c+2)^2N^2+(c+2)(k+1)N\log_2((c+2)N))$;
- 选取并更新非支配解集所需复杂度为 $O((k+1)(c+1)N+N+k(c+1)^2N^2+(k+1)(c+1)N\log_2((c+1)N))$.

2 实验和分析

2.1 测试函数

本文选取 Deb 等人在文献[15]中提出的 6 个经典的最小化测试问题 CTP2~CTP7,这类问题比较容易陷入局部最优,因而很难收敛到真正的 Pareto 前端,且这些测试函数随约束问题和约束中参数取值的不同相应的测试难度也有所不同.在下面的测试问题中,CTP4 测试难度较大.6 个测试问题的数学模型如式(19)所示:

$$\begin{cases} \min \mathbf{F}(\mathbf{x}) = \min \begin{Bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{Bmatrix} = \min \begin{Bmatrix} x_1 \\ c(\mathbf{x}) \left[1 - \frac{f_1(\mathbf{x})}{c(\mathbf{x})} \right] \end{Bmatrix} \\ g(\mathbf{x}) = \cos(\theta)[f_2(\mathbf{x}) - e] - \sin(\theta)f_1(\mathbf{x}) \geq a \mid \sin\{b\pi[\sin(\theta)[f_2(\mathbf{x}) - e] + \cos(\theta)f_1(\mathbf{x})]\}^d \end{cases} \quad (19)$$

其中,

$$c(\mathbf{x}) = 1 + \sum_{i=2}^5 [x_i^2 - 10\cos(2\pi x_i) + 10] \quad (20)$$

其中,变量范围 $x_1 \in [0,1], -5 \leq x_i \leq 5, i=2, \dots, 10$, 变量 θ, a, b, c, d, e 的取值见表 5.

Table 5 Variables set of test problems CTP2~CTP7

表 5 测试问题 CTP2~CTP7 的变量设置

测试问题	θ	a	b	c	d	e
CTP2	-0.20π	0.20	10	1	6	1
CTP3	-0.20π	0.10	10	1	0.5	1
CTP4	-0.20π	0.75	10	1	0.5	1
CTP5	-0.10π	0.10	10	2	0.5	1
CTP6	0.10π	40	0.5	1	2	-2
CTP7	-0.05π	40	5	1	6	0

CTP2 和 CTP7 最优解为间断的连续区域,CTP3~CTP5 最优解主要考虑几个离散点,CTP6 的最优解为连续区域.

2.2 性能度量

本文采用 3 项测试指标来度量算法的性能,具体如下:

1) 逼近程度度量指标:世代距离(generational distance,简称 GD^[16])

世代距离测量所得到的 Pareto 前端 PF_{known} 和真实前端 PF_{true} 之间的距离,世代距离的值越小,说明所求得的 Pareto 前端 PF_{known} 越接近 PF_{true} ,结果就越好.世代距离的数学表达式如公式(21)所示:

$$G = \frac{1}{n} \left(\sum_{i=1}^n d_i^p \right)^{1/p} \quad (21)$$

其中, n 是 PF_{known} 中个体的个数, $p=2, d_i$ 是 PF_{known} 中第 i 个个体的目标函数向量到 PF_{true} 中最近的那个个体的欧氏距离.

2) 分布性度量指标:空间度量(the spacing,简称 S^[16])

空间度量指标用于度量 Pareto 前端 PF_{known} 分布的均匀性,其值越小,说明 PF_{known} 上解分布得越均匀.空间度量指标的数学表达式如公式(22)所示:

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (22)$$

其中, $d_i = \min_j (|f_2^i(\mathbf{x}) - f_1^j(\mathbf{x})| + |f_2^i(\mathbf{x}) - f_2^j(\mathbf{x})|), i, j = 1, \dots, n, \bar{d}$ 是 d_i 的均值, n 是 PF_{known} 中个体的个数.

3) 多样性度量指标:修正后的最大展布(MS' ^[17])

最大展布用于测量所求的 Pareto 前端 PF_{known} 覆盖真实前端 PF_{true} 的程度, MS' 的值越大,说明 PF_{known} 覆盖 PF_{true} 的范围就越大,所求的解集就越好.最大展布 MS' 的数学表达式如公式(23)所示:

$$MS' = \sqrt{\frac{1}{M} \sum_{i=1}^M \left[\frac{\min[(PF_{\text{true}})_i, (\overline{PF_{\text{known}}})_i] - \max[(PF_{\text{true}})_i, (PF_{\text{known}})_i]}{(PF_{\text{true}})_i - (PF_{\text{true}})_i} \right]^2} \quad (23)$$

其中, $(\overline{PF_{\text{known}}})_i$ 和 $(PF_{\text{known}})_i$ 是 PF_{known} 中第 i 个目标函数的最大值和最小值.同样, $(PF_{\text{true}})_i$ 和 $(\underline{PF_{\text{true}}})_i$ 是 PF_{true} 中第 i 个目标函数的最大值和最小值.

由于有些测试问题的 Pareto 前端是几个间断的点,此时,如果再采用空间度量指标和最大展布去测量所求

得解集 PF_{known} 分布的均匀性和多样性,已显得不太合适,所以本文采用文献[18]中 Xiao 提出的方法,把测试问题分为 3 组,对不同组的测试问题采用不同的评价指标.

2.3 仿真结果及其分析

与约束单目标优化算法^[19,20]和无约束的多目标优化算法^[21]不同,约束多目标优化算法更为复杂,求解更加困难.为了说明本文算法的性能,将本文算法与 NSGA-II 约束多目标优化算法^[2](以下简称 NSGA-II 算法)和文献[11]中的算法(以下简称“文献[11]算法”)两种优秀的算法进行对比.NSGA-II 算法和文献[11]中算法参数的设置如下:初始种群大小为 100,交叉概率为 0.9,交叉分布指数为 15,变异概率为 0.1,变异分布指数为 20.本文算法的参数设置如下:初始种群个数为 400,交叉概率为 0.9,交叉分布指数为 15,变异概率为 0.1,变异分布指数为 20,最大的可行非支配解集 $FeaNonPop$ 大小为 100,修正非支配解集 $ModNonPop$ 大小为 100.为了选择合适的进化代数,本文测试了 3 种算法在不同的测试函数上的 GD 指标,代数范围在[1000~10000],每隔 1 000 代统计 1 次,独立运行 50 次,取其均值.以 CTP2 测试函数为例,如图 1 所示.

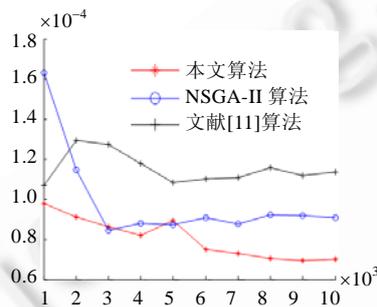


Fig.1 Average value of GD metric for CTP2 test function

图 1 CTP2 测试函数的 GD 均值

由图 1 可以看出,3 种算法在进化到 6 000 代时,GD 指标趋于稳定.根据该方法,为了得到每种算法在每个测试函数上的最优解,这里设定 3 种算法迭代次数均为 10 000 次,对每个测试问题均独立运行 50 次.仿真结果如图 2 和图 3 所示.

在图 2 中,对于 CTP2~CTP5 这 4 个测试问题,阴影部分为可行目标空间.测试问题 CTP2 的 Pareto 前端都是由许多不连续区域的点构成,该问题难度较大.由图 2 可以看出,对于 CTP2,3 种算法都能较好地收敛于 Pareto 前端,且多样性和均匀性都比较好.而测试问题 CTP3~CTP5 的 Pareto 前端由一系列离散的点构成,相对于 CTP2 问题难度更大.如何全面地找到这些点,是测试算法准确度的重要指标.对比图 2 可以看出,对于 CTP3~CTP5 这 3 种测试问题,3 种算法都能较好地收敛于 Pareto 前端.然而与其他两种算法相比,本文算法找到的离散解的个数明显多于 NSGA-II 算法和文献[11]算法所找到的解,说明本文算法具有较强的搜索能力以及较好的保持多样性的能力.

图 3 为 3 种算法关于 CTP6 和 CTP7 的仿真结果,图 3 中阴影部分为这两个测试问题的可行目标空间.由图 3 可以看出,其可行域是一系列不连续的带状区域.在求解 CTP6 时,很容易陷入局部最优,找到的解容易收敛于 Pareto 前面上方的曲线上;在求解 CTP7 时,不仅很容易丢失部分可行域,而且也容易陷入局部最优.比较图 3 中 3 种算法关于 CTP6 的仿真结果可以得出:对于解的多样性和收敛性,3 种算法的结果相似,但是文献[11]算法找到的解的均匀性最差,NSGA-II 次之,本文算法均匀性最好.对于测试问题 CTP7,3 种算法都较好地收敛到最优约束 Pareto 前端.为进一步对比几种算法的性能,根据第 2.2 节给出的评价指标对各测试问题的测试结果进行测试,并得出 50 次独立运行结果的平均值和标准差,见表 6~表 8.

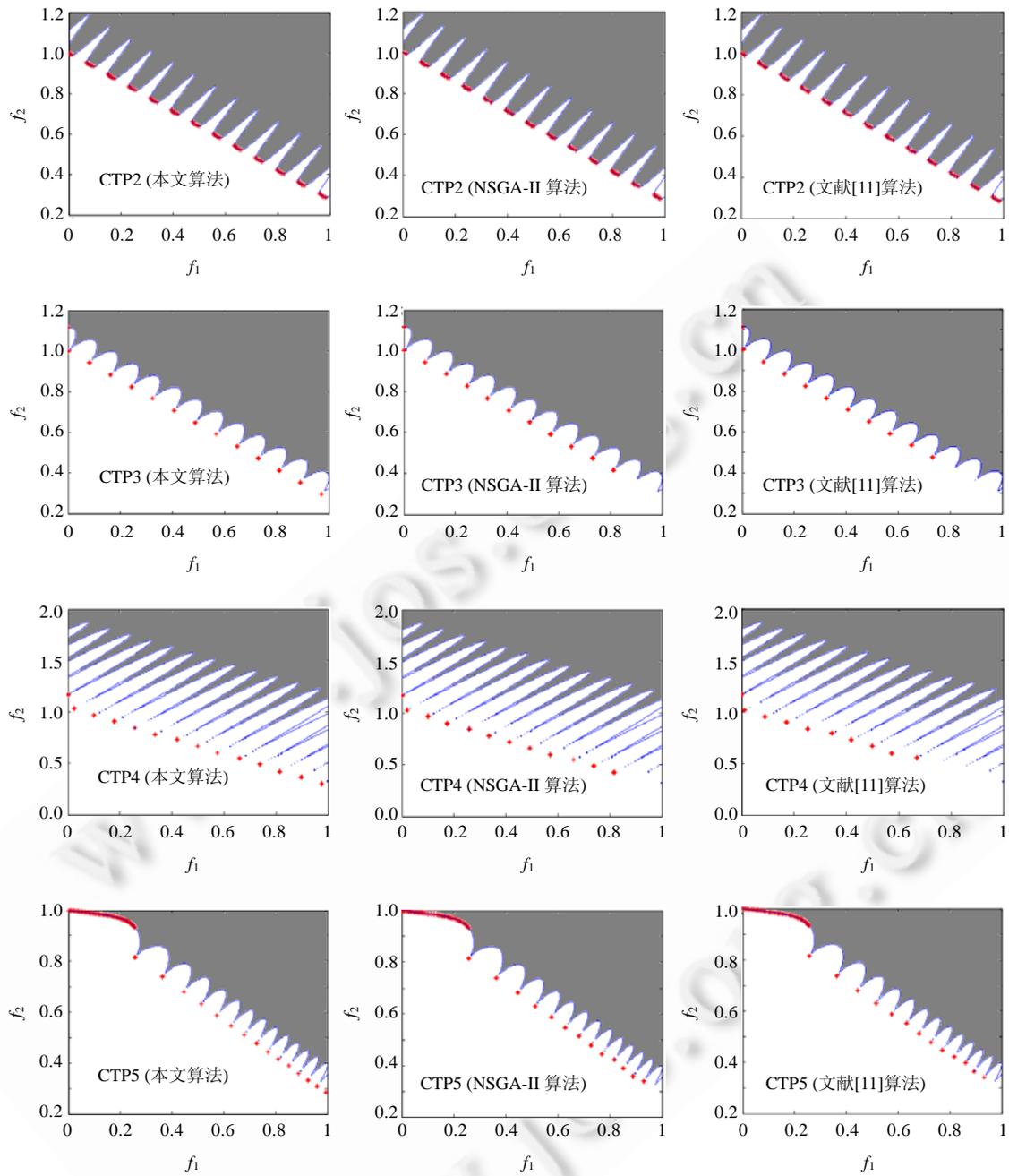


Fig.2 Simulation results of three algorithms on CTP2~CTP5

图2 3种算法关于CTP2~CTP5的仿真结果

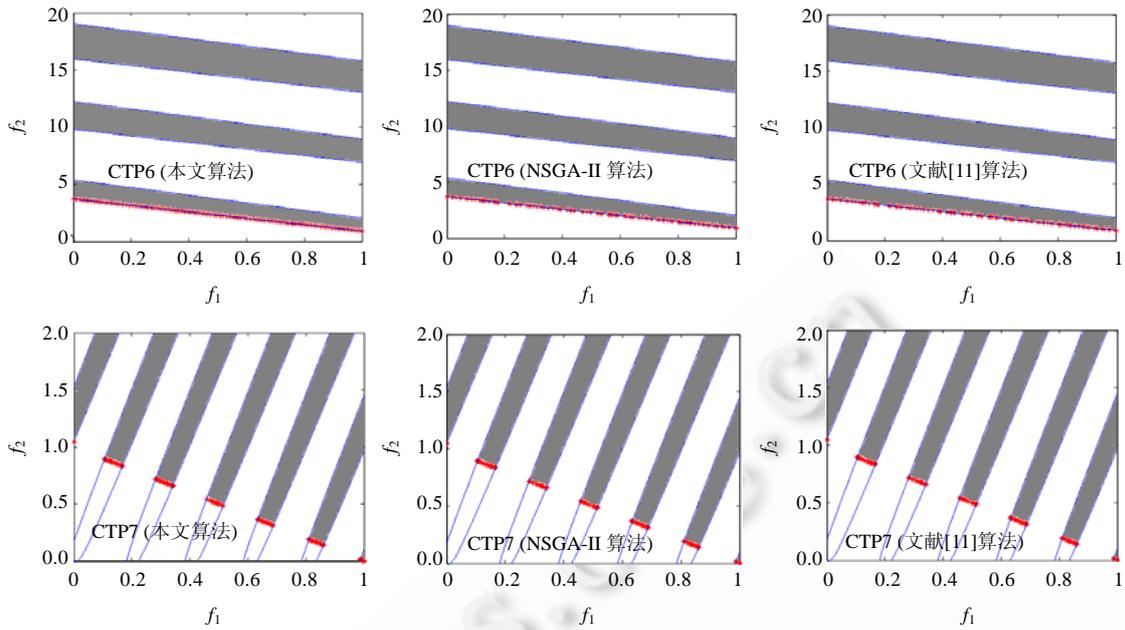


Fig.3 Simulation results of three algorithms on CTP6 and CTP7

图 3 3 种算法关于 CTP6 和 CTP7 的仿真结果

Table 6 Performance metrics of the first group CTP test problems

表 6 第 1 组 CTP 测试问题的各项性能指标

测试问题	算法	世代距离		空间分布		离散区域的个数	
		Mean	S.D.	Mean	S.D.	Mean	S.D.
CTP2	NSGA-II	3.56E-04	1.33E-05	0.002 5	1.52E-04	13	0
	文献[11]算法	1.14E-04	8.53E-06	0.003 0	1.30E-04	13	0
	本文算法	8.48E-05	3.56E-06	0.001 7	1.66E-04	13	0
CTP7	NSGA-II	9.90E-05	4.59E-06	0.002 3	3.68E-04	7	0
	文献[11]算法	3.70E-03	1.01E-07	0.002 6	1.60E-04	7	0
	本文算法	9.38E-05	2.39E-06	0.001 2	1.01E-04	7	0

Table 7 Performance metrics of the second group CTP test problems

表 7 第 2 组 CTP 测试问题的各项性能指标

测试问题	算法	世代距离		离散点的个数	
		Mean	S.D.	Mean	S.D.
CTP3	NSGA-II	2.40E-03	7.12E-04	13.58	0.758 4
	文献[11]算法	2.60E-03	7.43E-04	9.9	2.901 4
	本文算法	2.50E-03	6.05E-04	14	0
CTP4	NSGA-II	2.50E-03	7.81E-04	12.3	1.865 4
	文献[11]算法	4.50E-03	1.10E-03	7.38	2.954 7
	本文算法	3.00E-03	4.51E-04	13.82	0.437 5
CTP5	NSGA-II	3.04E-04	8.79E-05	13.52	2.401
	文献[11]算法	4.07E-04	1.92E-04	9.5	4.136 7
	本文算法	2.32E-04	1.45E-05	14.88	0.385 4

Table 8 Performance metrics of the third group CTP test problems**表 8** 第 3 组 CTP 测试问题的各项性能指标

测试问题	算法	世代距离		空间分布		最大展布	
		Mean	S.D.	Mean	S.D.	Mean	S.D.
CTP6	NSGA-II	8.48E-04	6.51E-05	0.0135	9.55E-04	1	9.78E-07
	文献[11]算法	5.56E-04	5.19E-05	0.0131	1.10E-03	1	1.96E-02
	本文算法	6.54E-04	2.32E-05	0.0042	7.00E-04	1	1.98E-07

从表 6 可以看出,对测试问题 CTP2 和 CTP7,新算法的世代距离和空间分布的均值都明显小于 NSGA-II 算法和文献[11]算法所得的结果,说明新算法优化所得的解集更接近真实 PF_{true} ,同时也说明本文算法所得的解集分布的均匀性比 NSGA-II 算法和文献[11]算法要好.3 种算法每次都能找到所有的离散区域.对 CTP2 测试问题,虽然文献[11]算法得到的解集空间分布的标准差最小,但其均值最大.尤其是对 CTP7 测试问题而言,虽然文献[11]算法得到的世代距离的标准差明显低于本文算法和 NSGA-II 算法,但其世代距离的均值明显高于其余两种算法.

从表 7 可以看出,对 CTP3~CTP5 测试问题,本文算法每次运行都能找到离散点的个数明显多于 NSGA-II 算法和文献[11]算法;尤其是对 CTP3 问题,本文算法每次运行都能找到全部的离散点,说明本文算法在多样性保持方面要明显优于 NSGA-II 算法和文献[11]算法.同时可以看出,NSGA-II 算法所得结果要优于文献[11]算法,说明 NSGA-II 算法在多样性保持方面要优于文献[11]算法.同时,从 CTP3 和 CTP4 的世代距离的均值可以看出,NSGA-II 算法的世代距离最小,说明 NSGA-II 所得解最接近真实解集 PF_{true} .文献[11]算法在这 3 个测试问题的表现最差.对 CTP5 测试问题,本文算法所得解不仅最接近真实解集 PF_{true} ,而且多样性保持也最好.同时,由图 2 也可以看出,在 CTP5 的连续解区域,本文算法所得解分布的均匀性也是最好的.

从表 8 可以看出,对 CTP6 测试问题,文献[11]算法所得解集的世代距离的均值最小,说明文献[11]算法所得解最接近真实解集 PF_{true} ,本文算法次之,NSGA-II 最差.本文算法空间分布的均值明显要小于 NSGA-II 算法和文献[11]算法,说明本文算法所得解分布的均匀性最好.从最大展布的标准差可以看出,本文算法和 NSGA-II 算法几乎每次都能找着解的边缘,而文献[11]算法的稳定稍差.

3 总结与展望

本文首先简要介绍了已有的几种约束多目标优化算法,并根据已有算法的缺点提出了修正免疫克隆约束多目标优化算法.通过引入约束处理策略,对种群中的个体的每一维目标函数值进行修正,并根据修正后的目标函数值选择优势个体种群进入下一代进行优化.最后,通过对 CTP2~CTP7 问题进行测试,并与 NSGA-II 算法和文献[11]算法进行比较,结果表明,本文算法在优化解集的多样性、真实解集的逼近性以及解分布的均匀性上都得到了很大的提高.

约束多目标优化目前所存在的问题仍然是约束处理策略的选取.支配可行解和非可行非支配解之间的选取问题仍然是一个亟待解决的首要问题,是否根据不同的问题选取不同的策略,或者让约束处理策略根据问题的不同自适应地去选取不同的策略,仍是今后的研究方向.

References:

- [1] Venkatraman S, Yen GG. A generic framework for constrained optimization using genetic algorithms. IEEE Trans. on Evolutionary Computation, 2005,9(4):424-435. [doi: 10.1109/TEVC.2005.846817]
- [2] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation, 2002,6(2):182-197. [doi: 10.1109/4235.996017]
- [3] Li LD, Yu XH, Li XD, Guo W. A modified PSO algorithm for constrained multi-objective optimization. In: Zhou WL, ed. Proc. of the 2009 3rd Int'l Conf. on Network and System Security. 2009. 462-467. [doi: 10.1109/NSS.2009.72]
- [4] Geng HT, Song QX, Wu T, Liu JF. A multi-objective constrained optimization algorithm based on infeasible individual stochastic binary-modification. In: Trajkovic L, ed. Proc. of the IEEE Int'l Conf. on Intelligent Computing and Intelligent Systems (ICIS 2009). New York: IEEE, 2009. 89-93. [doi: 10.1109/ICICISYS.2009.5357931]

- [5] Chafekar D, Xuan J, Rasheed K. Constrained multi-objective optimization using steady state genetic algorithms. In: Cantú-Paz E, ed. Proc. of the Genetic and Evolutionary Computation Conf. Berlin: Springer-Verlag, 2003. 813–824.
- [6] Young N. Blended ranking to cross infeasible regions in constrained multiobjective problems. In: Proc. of the 2005 Int'l Conf. on Computational Intelligence for Modelling, Control and Automation, and Int'l Conf. Intelligent Agents, Web Technologies and Internet Commerce. 2005. 191–196. <http://www.computer.org/portal/web/csdl/doi/10.1109/CIMCA.2005.1631467> [doi: 10.1109/CIMCA.2005.1631467]
- [7] Liu CA. New multi-objective constrained optimization evolutionary algorithm. In: Pan JS, Shi P, Shieh CS, eds. Proc. of the 3rd Int'l Conf. on Innovative Computing Information and Control (ICICIC 2008). 2008. 320–320. [doi: 10.1109/ICICIC.2008.387]
- [8] Jaddan OA, Rao CR, Ragamani L. Solving constrained multi-objective optimization problems using non-dominated ranked genetic algorithm. In: Imae J, Kimura A, eds. Proc. of the 3rd Asia Int'l Conf. on Modelling & Simulation (AMS 2009). New York: IEEE, 2009. 113–118. [doi: 10.1109/AMS.2009.38]
- [9] Singh HK, Isaacs A, Ray T, Smith WF. A simulated annealing algorithm for constrained multi-objective optimization. In: Esma A, ed. Proc. of the Evolutionary Computation. New York: IEEE, 2008. 1655–1662. [doi: 10.1109/CEC.2008.4631013]
- [10] Isaacs A, Ray T, Smith WF. Blessings of maintaining infeasible solutions for constrained multi-objective optimization problems. In: Esma A, ed. Proc. of the Evolutionary Computation. New York: IEEE, 2008. 2780–2787. [doi: 10.1109/CEC.2008.4631171]
- [11] Woldeesenbet YG, Yen GG, Tessema BG. Constrained handling in multiobjective evolutionary optimization. IEEE Trans. on Evolutionary Computation, 2009,13(3):514–525. [doi: 10.1109/TEVC.2008.2009032]
- [12] Shang RH, Jiao LC, Ma WP. Immune clonal multi-objective optimization algorithm for constrained optimization. Journal of Software, 2008,19(11):2943–2956 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/20081116.htm> [doi: 10.3724/SP.J.1001.2008.02943]
- [13] Shang RH, Jiao LC, Gong MG, Ma WP. An immune clonal algorithm for dynamic multi-objective optimization. Journal of Software, 2007,18(11):2700–2711 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/2700.htm> [doi: 10.1360/jos182700]
- [14] Deb K, Beyer HG. Self-Adaptive genetic algorithms with simulated binary crossover. Evolutionary Computation, 2001,9(2): 197–221.
- [15] Deb K, Pratap A, Meyarivan T. Constrained test problems for multi-objective evolutionary optimization. In: Zitzler E, Deb K, Thiele L, Coello Coello CA, Corne D, eds. Proc. of the Evolutionary Multi-Criterion Optimization. LNCS 1993, Berlin: Springer-Verlag, 2001. 284–298. [doi: 10.1007/3-540-44719-9_20]
- [16] Van Veldhuizen DA, Lamont GB. On measuring multiobjective evolutionary algorithm performance. In: Proc. of the Congress on Evolutionary Computation (CEC 2000). Piscataway: IEEE Press, 2000. 204–211. [doi: 10.1109/CEC.2000.870296]
- [17] Goh CK, Tan KC. An investigation on noisy environments in evolutionary multiobjective optimization. IEEE Trans. on Evolutionary Computation, 2007,11(3):354–381. [doi: 10.1109/TEVC.2006.882428]
- [18] Xiao HS, Zu JW. A new constrained multiobjective optimization algorithm based on artificial immune systems. In: Proc. of the 2007 IEEE Int'l Conf. on Mechatronics and Automation. 2007. 3122–3127. [doi: 10.1109/ICMA.2007.4304060]
- [19] Zhou YR, Li YX, Wang Y, Kang LS. A Pareto strength evolutionary algorithm for constrained optimization. Journal of Software, 2003,14(7):1243–1249 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/14/1243.htm>
- [20] Lin D, Li MQ, Kou JS. A GA-based method for solving constrained optimization problems. Journal of Software, 2001,12(4): 628–632 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/20010420.htm>
- [21] Gong MG, Jiao LC, Yang DD, Ma WP. Research on evolutionary multi-objective optimization algorithms. Journal of Software, 2009,20(2):271–289 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3483.htm> [doi: 10.3724/SP.J.1001.2009.00271]

附中文参考文献:

- [12] 尚荣华,焦李成,马文萍.免疫克隆多目标优化算法求解约束优化问题.软件学报,2008,19(11):2943–2956. <http://www.jos.org.cn/1000-9825/20081116.htm>

- [13] 尚荣华,焦李成,马文萍.免疫克隆算法求解动态多目标优化问题.软件学报,2007,18(11):2700-2711. <http://www.jos.org.cn/1000-9825/18/2700.htm> [doi: 10.1360/jos182700]
- [19] 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. <http://www.jos.org.cn/1000-9825/14/1243.htm>
- [20] 林丹,李敏强,寇纪淞.基于遗传算法求解约束优化问题的一种算法.软件学报,2001,12(4):628-632. <http://www.jos.org.cn/1000-9825/20010420.htm>
- [21] 公茂果,焦李成,杨咚咚,马文萍.进化多目标优化算法研究.软件学报,2009,20(2):271-289. <http://www.jos.org.cn/1000-9825/3483.htm> [doi: 10.3724/SP.J.1001.2009.00271]



尚荣华(1979—),女,河北阜城人,博士,副教授,CCF 会员,主要研究领域为自然计算,工程优化,智能信息处理.



焦李成(1959—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为自然计算,数据挖掘,图像处理,智能信息处理.



胡朝旭(1986—),男,硕士生,主要研究领域为自然计算,多目标优化.



马晶晶(1983—),女,博士生,主要研究领域为智能计算,图像处理.