

# 用于约束多目标优化的免疫记忆克隆算法

尚荣华, 焦李成, 马文萍, 公茂果

(西安电子科技大学智能信息处理研究所和智能感知与图像理解教育部重点实验室, 陕西西安 710071)

**摘 要:** 提出一种用于求解约束多目标优化问题的新算法, 其主要特点是将约束条件转化为一个目标, 并引入免疫克隆和免疫记忆机制, 使抗体种群的演化过程和记忆单元的演化过程并行进行, 更好地实现了抗体间的相互协作, 保证了在演化过程中, 解集从可行域内部和不可行域边缘向着约束最优 Pareto 前端逼近. 从而保证了算法较强的收敛性以及所得最优解较好的多样性. 仿真结果表明, 新算法很好的保持了所得最优解的多样性、均匀性以及较强的收敛性.

**关键词:** 人工免疫系统; 约束多目标优化; 免疫记忆; Pareto 最优

**中图分类号:** TP18 **文献标识码:** A **文章编号:** 0372-2112 (2009) 06-1289-06

## An Immune Memory Clone Algorithm for Constrained Multi-Objective Optimization

SHANG Rong-hua, JIAO Li-cheng, MA Wen-ping, GONG Mao-guo

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

**Abstract:** A new algorithm for constrained multi-objective optimization is presented. The algorithm treats the constraints as an objective and the immune clone and immune memory mechanism are introduced. Therefore, the new algorithm could find the Pareto-optimal solutions from the feasible region and the edge of the infeasible region, which assures both the convergence and diversity of the obtained solutions. Simulation results show that the new algorithm has much better performance in finding a much better spread of solutions, in maintaining a better uniformity of the solutions and in obtaining a better convergence.

**Key words:** artificial immune system; constrained multi-objective optimization; immune memory; Pareto-optimal

### 1 引言

用各种方法求解多目标优化问题是近年来的一个研究热点, 有些是基于进化算法的<sup>[1,2]</sup>, 还有一些方法是基于人工免疫系统的<sup>[3]</sup>, 取得了很好的效果. 但是这些算法主要考虑相互冲突的多个目标间的优化而很少考虑约束条件, 主要原因是约束条件将搜索空间分解成可行和不可行两个区域, 使得多目标优化算法在收敛到最优 Pareto 前端或者保持解的多样性方面存在巨大的困难. 因此, 约束处理是工程优化中的一个关键部分.

现有的一些处理约束多目标优化的方法主要有: 保持可行解法, 罚函数法, 约束锦标赛法, Ray-Tai-Seow 的约束处理方法<sup>[4]</sup>. 惩罚函数法是处理约束条件的最常用方法, 惩罚函数法简单易行, 但在实际操作时, 罚因子的

选取相当困难. 将约束条件转化为一个目标的约束处理方法在约束单目标优化中取得了十分理想的效果<sup>[5]</sup>, 但是将这种约束处理方法应用约束多目标优化中的成果比较少见.

基于人工免疫系统<sup>[6]</sup>的免疫克隆和免疫记忆学说, 本文提出了一种用于求解约束多目标优化问题的新算法 - 免疫记忆克隆约束多目标优化 (IMCCMO) 算法. 其主要特点是将约束条件转化为一个目标, 并引入免疫克隆和免疫记忆机制, 使抗体种群的演化过程和记忆单元的演化过程并行地进行, 更好地实现抗体间的相互协作, 保证在演化过程中, 解集从可行域内部和不可行域的边缘向着最优 Pareto 前端逼近. 更好的保证了所得最优解较好的多样性, 较强的逼近性以及较广的分布. 对 4 个标准测试问题的测试也表明了新算法具有较好的性能.

收稿日期: 2007-03-19; 修回日期: 2008-12-06

基金项目: 国家“863”计划 (No. 2006AA01Z107); 陕西“13115”科技创新工程重大科技专项 (No. 2008ZDKG-37); 国家自然科学基金 (No. 60703107; 60703108; 60803706); 陕西省自然科学基金 (2007F23); 国家教育部博士点基金 (No. 20070701022); 中国博士后科学基金资助项目 (No. 20080431228) 以及教育部长江学者和创新团队支持计划 (No. IRT0654)

## 2 约束多目标优化问题

### 2.1 问题的定义

定义 1 不失一般性,一个包括  $n$  个决策变量,  $k$  个目标函数和  $m$  个约束条件组成的约束多目标优化问题 (CMOP) 定义如下:

$$\left\{ \begin{array}{ll} \min & y = f(x) = (f_1(x), \dots, f_k(x)) \\ \text{subject to} & g(x) = (g_1(x), \dots, g_m(x)) \leq 0 \\ \text{where} & x = (x_1, x_2, \dots, x_n) \in X \\ & X = \{ (x_1, x_2, \dots, x_n) \mid l_i \leq x_i \leq u_i \} \\ & l = (l_1, l_2, \dots, l_n), u = (u_1, u_2, \dots, u_n) \\ & y = (y_1, y_2, \dots, y_n) \in Y \end{array} \right. \quad (1)$$

其中,目标函数和约束条件都是决策变量的函数,  $x$  为决策变量,  $y$  为目标函数,  $X$  表示决策空间,  $l$  和  $u$  分别为下界和上界,  $Y$  为目标空间。

定义 2 可行集是一个满足约束条件的决策变量的集合,定义如下:

$$X_f = \{ x \in X \mid g(x) \leq 0, \quad i = 1, 2, \dots, m \} \quad (2)$$

定义 3 在约束多目标化中,所有约束 Pareto-最优解的集合  $P_{SF}$  为约束 Pareto-最优解集,定义如下:

$$P_{SF} = \{ x^* \mid \neg \exists x \in X_f: x > x^* \} \quad (3)$$

定义 4  $P_{SF}$  中的解对应的目标函数值组成的集合  $P_{FF}$  称为约束 Pareto-前端,定义如下:

$$P_{FF} = \{ F(x) = (f_1(x), f_2(x), \dots, f_k(x))^T \mid x \in P_{SF} \} \quad (4)$$

### 2.2 约束条件的处理方法

本文没有采用现有的诸如将 Pareto-占优化为约束 Pareto-占优等的一些约束处理方法.而是将约束条件转化为一个目标,具体如下:

令:

$$G_j(x) = \max\{0, g_j(x)\}, \quad 1 \leq j \leq m \quad (5)$$

则,

$$f_{k+1}(x) = G(x) = \sum_{j=1}^m G_j(x) \quad (6)$$

所以,由  $n$  个决策变量,  $k$  个目标函数和  $m$  个约束条件组成的 CMOP 就转化成了  $n$  个决策变量,  $k+1$  个目标函数的非约束多目标优化的问题:

$$\left\{ \begin{array}{ll} \min & y = f(x) = (f_1(x), \dots, f_k(x), f_{k+1}(x)) \\ \text{where} & x = (x_1, x_2, \dots, x_n) \in X \\ & X = \{ (x_1, x_2, \dots, x_n) \mid l_i \leq x_i \leq u_i \} \\ & l = (l_1, l_2, \dots, l_n), u = (u_1, u_2, \dots, u_n) \\ & y = (y_1, y_2, \dots, y_n) \in Y \end{array} \right. \quad (7)$$

对于任一变量  $x$ ,如果  $f_{k+1}(x) = 0$ ,则  $x$  满足约束条件,是可行解;如果  $f_{k+1}(x) > 0$ ,则  $x$  不满足约束条件,是不可行解,且  $f_{k+1}(x)$  的值越大,  $x$  违反约束的程

度越大.基于此,采用如下处理方式:

(1) 将抗体群  $A$  分为可行解集  $X_f$  ( $\{ X_f \mid \forall x \in X_f, f_{k+1}(x) = 0 \}$ ) 与不可行解集  $\bar{X}_f$  ( $\{ \bar{X}_f \mid \forall x \in \bar{X}_f, f_{k+1}(x) > 0 \}$ );

(2) 通过 Pareto-最优解集的概念把可行解集  $X_f$  分为 Pareto-最优解集  $P_s$  和非 Pareto-最优解集  $\bar{P}_s$ ;

(3) 根据不可行解违反约束的程度将不可行解集  $\bar{X}_f$  分为有益的解集  $Q_b$  和非有益的解集  $\bar{Q}_b$ ;

(4) 分别对可行解集合  $X_f$  和有益的解集  $Q_b$  进行相应的免疫操作,以求得高质量 Pareto-最优解。

由于 CMOP 的搜索空间是由可行域与不可行域两部分组成,所以如何有效利用不可行解显得特别重要.本文通过将抗体群  $A$  划分为可行解集和不可行解集,分别从可行解集和不可行解集两个集合向最优的约束 Pareto-前端逼近.所以新算法很好的利用了部分接近可行域边缘的一些不可行解,从而保证了所得解的具有较高的质量。

## 3 用于 CMOP 的免疫记忆克隆算法

### 3.1 免疫记忆克隆

克隆选择原理认为,淋巴细胞除了扩增或分化成浆细胞以外,也能分化成生命期较长的 B 记忆细胞.这些记忆细胞中的信息编码组成免疫系统记忆,使它能学习、记忆蛋白质结构.再次遇到相应的抗原后,为清除已经刺激初次应答的相应的特异抗原,记忆细胞将预先被免疫系统选择出来,并迅速活化、增殖、分化为效应细胞,执行高效而持久的免疫功能。

### 3.2 免疫记忆克隆约束多目标优化算法

受生物免疫系统免疫记忆和免疫克隆机理的启发,针对 CMOP,本文设计的免疫记忆克隆约束多目标优化算法主要由免疫克隆重组操作、免疫克隆变异操作以及抗体群更新操作三个子算法组成。

#### 3.2.1 免疫克隆重组操作

免疫基因操作是生物学中抗体亲和力成熟的原动力,主要包括克隆重组操作和克隆变异.本文采用多父体单形杂交算子,来实现克隆重组操作.多父体单形杂交算子使用  $n+1$  个父体向量重组产生后代,具体的操作步骤如下:

在  $R^n$  中,  $n+1$  个独立的父体向量  $x_i$  ( $i = 1, 2, \dots, n$ ) 形成  $R^n$  空间的一个单形 (simplex);

计算  $n+1$  个向量的中心  $o: o = \frac{1}{n+1} \sum_{i=1}^{n+1} x_i$ ;

将单形沿各个方向 ( $x_i - o$ ) 以一定比例扩张;

从扩张后的单形中随机地取一点即为一个后代。

如图 1 所示,  $x^{(1)}$ ,  $x^{(2)}$ ,  $x^{(3)}$  为二维空间中的 3 个向量, 这 3 个向量形成一个单形, 把这个单形以比例  $(1 + \alpha)$  向外扩张 (称为扩张比率), 令  $\alpha = 1/3(x^{(1)} + x^{(2)} + x^{(3)})$ ,  $y^{(j)}$

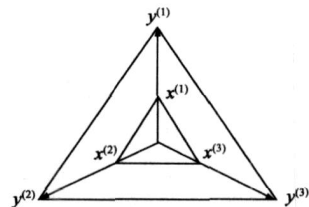


图1 二维三父体单形杂交

$= (1 + \alpha)(x^{(j)} - \alpha)$ ,  $(j = 1, 2, 3)$ , 由  $y^{(1)}$ ,  $y^{(2)}$ ,  $y^{(3)}$  产生一个新的单形. 在新单形中随机取一点  $z$ ,  $z = k_1 \cdot y^{(1)} + k_2 \cdot y^{(2)} + k_3 \cdot y^{(3)}$ ,  $k_1, k_2, k_3$  为  $[0, 1]$  中的 3 个随机数, 满足  $k_1 + k_2 + k_3 = 1$ ,  $z$  即为一个三父代单形杂交产生的后代.

### 3.2.2 免疫克隆变异操作

在用于约束多目标优化的免疫记忆克隆算法中, 克隆变异操作采用的是非一致性变异<sup>[7]</sup>, 即在单一抗体周围产生一个变异的解的群体, 利用局部搜索增加了所得解向着最优 Pareto-前端的逼近能力. 从而很好的保证了所得的解与最优 Pareto-前端的最小距离.

### 3.2.3 抗体群更新操作

多目标优化的一个很重要的评价指标就是解分布的均匀性, 约束多目标优化更是如此, 所以, 在保证所得解与最优 Pareto-前端的最小距离以及解分布的广度的情况下, 也要尽可能的保证解分布的均匀性. 抗体群更新操作主要通过将比较密集的地方的解删除, 保证解分布的均匀性.

### 3.2.4 免疫记忆克隆约束多目标优化

结合免疫克隆重组操作、免疫克隆变异操作以及抗体群更新操作, 本文给出了用于解决 CMOP 的新的人工免疫系统算法 - 免疫记忆克隆约束多目标优化算法 (IMCCMO). 该算法的主要步骤如下:

**算法:** 免疫记忆克隆约束多目标优化算法

**Step1:** 设定算法终止条件, 初始化进化代数  $it = 0$ ; 给定抗体群规模  $N$ , 有益解集规模  $N_0$ , 克隆比例  $R$ , 最大迭代次数  $g_{max}$ ; 随机产生初始抗体群:  $A(it) = \{a_1(it), a_2(it), \dots, a_N(it)\} \quad I^N$ ; 从  $A(it)$  中选取  $N_0$  个抗体构成记忆单元:

$$M(it) = \{m_1(it), m_2(it), \dots, m_{N_0}(it)\} \quad I^{N_0};$$

**Step2:** 结合单纯形法对  $A(it)$  进行重组操作  $T_r^c$ :

$$R(it) = T_r^c(A(it)) = \{r_1(it), r_2(it), \dots, r_{r_0}(it)\} \quad I^{r_0},$$

其中  $r_0 = \lfloor N/3 \rfloor$ ; 令:

$$\begin{aligned} B(it) &= R(it) \cup A(it) \\ &= \{a_1(it), a_2(it), \dots, a_N(it)\} \\ &\quad \{r_1(it), r_2(it), \dots, r_{r_0}(it)\} \\ &= \{b_1(it), b_2(it), \dots, b_{r_0+N}(it)\} \quad I^{r_0+N} = I^r \end{aligned}$$

其中,  $r = r_0 + N$ ;

**Step3:** 对  $B(it) \cup M(it)$  进行克隆操作  $T_c^c$ :

$$\begin{aligned} Z(it) &= T_c^c([B(it) \cup M(it)]) \\ &= [T_c^c(b_1(it)), \dots, T_c^c(b_r(it)), T_c^c(m_1(it)), \dots, \\ &\quad T_c^c(m_{N_0}(it))] \\ &= \{b_{1l}(it), \dots, b_{1R}(it), \dots, b_{rl}(it), \dots, b_{rR}(it), \\ &\quad \dots, \\ &\quad m_{1l}(it), \dots, m_{1R}(it), \dots, m_{N_0l}(it), \dots, m_{N_0R}(it)\} \\ &= \{z_1(it), \dots, z_l(it)\} \quad I^l (l = (r + N_0) * R) \end{aligned}$$

**Step4:** 对  $Z(it)$  进行克隆变异操作  $T_m^c$ :

$$\begin{aligned} V(it) &= T_m^c(Z(it)) \\ &= \{z_1(it), z_2(it), \dots, z_r(it)\} \\ &= \{v_1(it), v_2(it), \dots, v_r(it)\} \quad I^r \end{aligned}$$

**Step5:** 计算第  $it$  代的  $k + 1$  个目标函数在  $V(it)$  上的目标函数值:

$$\begin{aligned} N^V(it) : \{f(V(it))\} &= \{f_1(V(it)), \dots, f_{k+1}(V(it))\} \\ &= \{f_1(v_1(it)), \dots, f_1(v_n(it)), \dots, f_{k+1}(v_1(it)), \\ &\quad \dots, f_{k+1}(v_n(it))\} \end{aligned}$$

**Step6:** 对抗体群中的每一个抗体, 根据第  $k + 1$  个函数的目标值是否为零, 将抗体群  $V(it)$  划分为可行解集  $X_f(it)$  和不可行解集  $\tilde{X}_f(it)$ ;

**Step7:** 根据 Pareto-占优的概念把可行解集  $X_f(it)$  划分为 Pareto-占优集  $P_s(it)$  和非 Pareto-占优集  $\tilde{P}_s(it)$ ;

**Step8:** 根据不可行解违反约束的程度将不可行解集  $\tilde{X}_f(it)$  分为有益的解集  $Q_b(it) = \{q_1(it), \dots, q_{N_0}(it)\} \quad I^{N_0}$  和非有益的解集  $\tilde{Q}_b(it)$ ;

**Step9:** 对  $P_s(it)$  执行抗体群更新操作  $T_d^c$ : 若  $P_s(it)$  中抗体数目大于  $N$ , 则用抗体群更新操作删除  $P_s(it)$  中多余的抗体, 直到  $P_s(it)$  中抗体的数目为  $N$ ;

**Step10:** 对记忆单元进行学习操作  $T_l^c$ : 如果  $Q_b(it)$  中存在一个抗体 (记为  $q_i(it)$ ), 其违反约束的程度小于  $M(it)$  中的某一个抗体 (记为  $m_j(it)$ ), 则将  $q_i(it)$  添加到记忆单元  $M(it)$  中, 同时将  $M(it)$  中的  $m_j(it)$  删除; 直到  $Q_b(it)$  中每一个抗体违反约束的程度都大于或者等于  $M(it)$  中的抗体违反约束的程度;

**Step11:** 如果迭代次数  $it$  满足终止条件, 则输出  $P_s(it)$ ; 否则, 令  $A(it + 1) = P_s(it)$ ;  $M(it + 1) = M(it)$ ;  $a_i(it + 1) = a_i(it)$  ( $i = 1, 2, \dots, N$ );  $m_j(it + 1) = m_j(it)$  ( $j = 1, 2, \dots, N_0$ ); 则:

$$\begin{aligned} A(it + 1) &= \{a_1(it + 1), a_2(it + 1), \dots, a_N(it + 1)\} \quad I^N; \\ M(it + 1) &= \{m_1(it + 1), m_2(it + 1), \dots, m_{N_0}(it + 1)\} \quad I^{N_0}; \end{aligned}$$

令  $it = it + 1$ , 返回 Step2.

### 3.2.5 算法的时间复杂度分析

给定问题的目标数  $r$ , 抗体群规模  $N$ , 有益解集规模  $N_0$ , 克隆比例  $R$ , 非支配解抗体规模  $Nn$ , 每一代最后要保留的非支配解的个数  $N$ , 最大迭代次数  $g_{\max}$ . 则初始化抗体群, 并计算各抗体的目标函数值的时间复杂度为:  $O(N)$ , 初始化记忆单元的时间复杂度为:  $O(r * N + N_0)$ ; 构造非支配解集的时间复杂度为:  $O(r * (N + N_0)^2)$ , 克隆重组操作的时间复杂度为:  $O(Nn)$ , 对非支配解集对应的抗体群进行克隆的时间复杂度与克隆变异的时间复杂度均为:  $O((Nn + \lfloor Nn/3 \rfloor) * R)$ ; 抗体群更新操作的时间复杂度为:  $O(Nn^2)$ ; 对记忆单元的学习操作的时间复杂度最差为:  $O(N_0^2)$ . 因此, 算法总的时间复杂度最差为:

$$\begin{aligned} & O(N) + O(rN + N_0) + g_{\max} * [O(r(N + N_0)^2) + O(Nn) \\ & + 2(O((Nn + \lfloor Nn/3 \rfloor) * R)) + O(Nn^2) + O(N_0^2)] \\ & = O(g_{\max} * (r(N + N_0)^2 + (R + N)Nn^2 + N_0^2)) \end{aligned}$$

## 4 仿真实验及其结果比较

### 4.1 评价指标

为定量的刻画所得解与最优 Pareto 前端的逼近程度, 本文采用了两集合覆盖<sup>[8]</sup>这一评价指标, 至于解的分布的均匀性和广度, 则用所得到的 Pareto 前端形象地表示. 该评价指标描述如下:

指标: 假设  $X, X \subseteq X$  是目标空间中的两个非劣解集. 函数 按以下定义:

$$(X, X) \triangleq \frac{| \{ a \in X : \exists a \in X : a \triangleright a \} |}{|X|} \quad (14)$$

其中  $\triangleright$  表示 Pareto 不劣于.

由定义可知, 指标  $(X, X) \in [0, 1]$ ,  $(X, X) = 1$  表示对于集合  $X$  中的所有点均能在集合  $X$  中找到至少一个点不劣于它, 相反,  $(X, X) = 0$  表示集合  $X$  中的所有点, 均能在集合  $X$  中找到至少一个点不劣于它. 但是,  $(X, X)$  和  $(X, X)$  必须同时考虑, 因为二者之间没有确定的等价关系.

### 4.2 测试问题

本文使用文献[1]中 4 个标准的测试函数来检验新算法. 第一个测试问题的非约束 Pareto 前端的一部分是不可行的, 其约束 Pareto 前端是由第一个约束条件边界和一些非约束 Pareto 前端的某些部分连接而成; 第二个测试问题的约束 Pareto 最优解集是非约束 Pareto 最优解集的子集; 第三个问题 TNK 是由 Tanaka 等人提出的, 其 Pareto 前端是非连续的, 完全落在第一个约束条件的边界上; 第四个问题 WATER 是一个由 5 个目标、7 个约束条件组成的约束多目标问题, 由于具有 5 个目标, 所以很难讨论约束条件对于非约束 Pareto 前端的影响.

### 4.3 IMCCMO 的参数设置

本文算法中的主要参数有抗体群规模  $N$ , 最大迭

代次数  $g_{\max}$ , 记忆单元 (有益解集) 规模  $N_0$ , 克隆比例  $R$ , 非支配解抗体规模  $Nn$ , 每一代最后要保留的非支配抗体的个数  $N$ . 一般来讲, 如果不考虑算法的时间复杂度,  $g_{\max}$ 、 $N$  和  $N_0$  越大越好. 算法中, 令  $g_{\max} = 150$ ,  $N = 300$ ,  $N_0 = 30$ ,  $Nn$  由问题的性质和其他参数决定. 与之相比, 参数  $R$  对算法性能的影响较为复杂. 基于以前的工作, 将  $R$  取为 4.

### 4.4 测试结果及性能分析

新算法对 4 个标准的约束多目标优化问题进行了测试. 前三个问题 CONSTR, SRN 和 TNK 的测试结果如图 2~4 所示; 对于第四个问题 WATER, 给出了所得到的 Pareto 前端分别在任意两个目标空间的投影, 一共是十幅图, 分别表示了 Pareto 前端在  $f_1 - f_2, f_1 - f_3, f_1 - f_4, f_1 - f_5, f_2 - f_3, f_2 - f_4, f_2 - f_5, f_3 - f_4, f_3 - f_5, f_4 - f_5$  上面的投影, 如图 5 所示. 表 1 给出了 5 个目标函数值的变化范围.

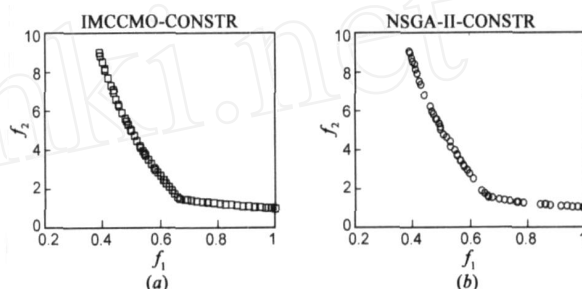


图2 IMCCMO和NSGA-II关于CONSTR的实验结果

图 2 给出了算法 IMCCMO 和算法 NSGA-II 关于 CONSTR 的实验结果, 从图中可以看出, 算法 IMCCMO 找到的解分布比较均匀, 而且分布的范围比较广; 算法 NSGA-II 找到的解分布的范围也比较广, 但是找到的解的多样性稍微差一些, 分布也不太均匀. 说明, 对于该测试问题, 与算法 NSGA-II 相比, 算法 IMCCMO 所得解的多样性和均匀性较好.

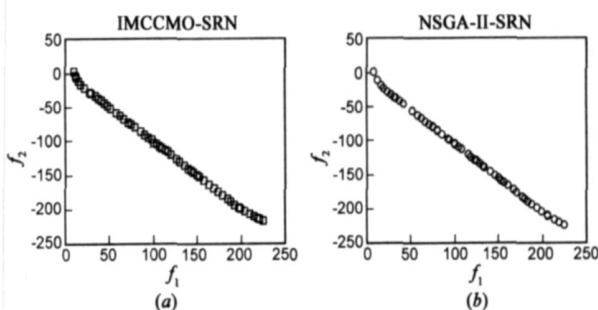


图3 IMCCMO和NSGA-II关于SRN的实验结果

图 3 给出了算法 IMCCMO 和算法 NSGA-II 关于 SRN 的实验结果, 从图中可以看出, 两种算法找到的解分布的范围都比较广, 但是与算法 IMCCMO 相比, 算法 NSGA-II 所得解的分布的均匀性和多样性较差.

对于测试问题 TNK, 由于 Pareto 前端是非连续的,

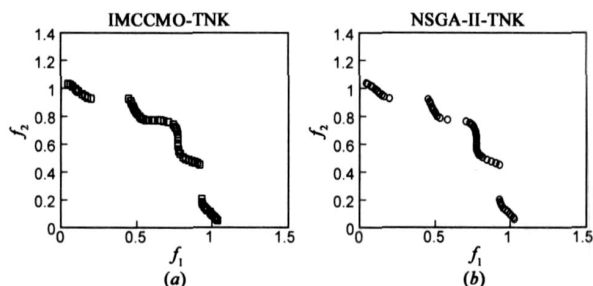


图4 IMCCMO和NSGA-II关于TNK的实验结果

所以能找到一个分布较广的解集,是衡量一个算法很重要的一个方面.从图4可以看出,算法IMCCMO找到的解范围分布的比较广,并且在 $f_2 = 0.8$ 附近的点连续性较强,从而说明了与算法NSGA-II相比,算法IMCCMO找到的解分布的比较均匀,而且分布的范围也比较广,从而保证了所得解很好的多样性和均匀性.

对于测试问题WATER,由于其目标函数值比较大,故采用如下的方法<sup>[1]</sup>标准化各个目标函数值: $f_1/8$  ( $10^4$ ),  $f_2/1500$ ,  $f_3/3$  ( $10^6$ ),  $f_4/6$  ( $10^6$ ),  $f_5/8000$  图5给出了算法IMCCMO和NSGA-II关于WATER的实验结果.算法NSGA-II的结果在图5中对角线的上方,算法IMCCMO得到结果在图5中对角线的下方.图中,任意一幅图的坐标可以通过观察对应的对角线上的方框图得到.例如,在第一行第三列的那幅图的横坐标为 $f_1$ ,纵坐标为 $f_3$ ,又因为这幅图在对角线的上方,所以这幅图是由NSGA-II得到的结果;为了便于比较,将该幅图对应的由算法IMCCMO得到那幅图在第三行第一列表示.第三行第一列的这幅图的横坐标是 $f_3$ ,而纵坐标是 $f_1$ .

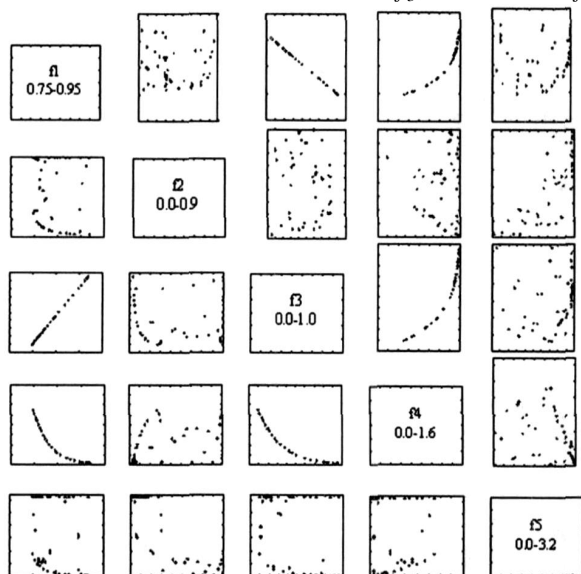


图5 IMCCMO和NSGA-II关于WATER的实验结果.对角线的上方为NSGA-II得到的结果,对角线的下方为IMCCMO得到的结果;将第 $(i, j)$  ( $i < j$ , IMCCMO得到结果)个图与第 $(j, i)$  (NSGA-II得到的结果)个图进行比较;坐标及其变化范围都在对角线上的图中表示

若将这两幅图作比较,须将第三行第一列的这幅图旋转90度,这就与第一行第三列的那幅图的坐标一致了.

从图5可以看出,对于大多数的目标函数,算法IMCCMO找到的解的分布要明显好于算法NSGA-II找到的解.例如由算法IMCCMO得到的由 $f_2 - f_1, f_3 - f_2, f_5 - f_1, f_5 - f_2, f_5 - f_3, f_5 - f_4$ 相交得到的图非常清晰,而由算法NSGA-II得到的结果不太清晰,这说明,由算法NSGA-II得到的结果向着最优Pareto前端的收敛性较差.而对于由 $f_1 - f_3, f_1 - f_4$ 以及 $f_3 - f_4$ 相交得到的图可以看出,算法IMCCMO得到的结果的分布要明显好于算法NSGA-II得到的结果,从而说明算法IMCCMO具有较强的保持均匀性的能力.

同时,对于该问题, Pareto-前端上的各个目标函数值的上下界的变化范围如表1所示.

表1 WATER测试问题 Pareto-前端上的各个目标函数值的上下界的变化范围

算法	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$
IMCCMO	0.798 -	0.027 -	0.095 -	0.031 -	0.001 -
	0.918	0.900	0.951	1.096	3.125
NSGA-II	0.798 -	0.027 -	0.095 -	0.031 -	0.001 -
	0.920	0.900	0.951	1.110	3.124

从表1可以看出,对于目标函数 $f_2$ 和 $f_3$ ,算法IMCCMO得到的范围与算法NSGA-II得到的范围一致,而对于目标函数 $f_1$ 和 $f_4$ ,算法IMCCMO得到的范围要略小于由算法NSGA-II得到的范围,对于目标函数 $f_5$ ,算法IMCCMO得到的范围要略大于由算法NSGA-II得到的范围.表明这两种算法在这个评价指标上区别不大,从而说明,这两种算法在得到解分布的广度上非常相近.

图6给出了两种算法关于指标独立运行30次的结果的统计盒图<sup>[9]</sup>.盒图在判断数据的分布方面具有重要的作用,能形象地表示出数据的分布情况.盒图中间是一个盒子,盒子的上方为上四分位数,下方为下四分位数,而中间的一条横线为中位数;有向上和向下延伸的虚线表示了数据的展布和形状,虚线的顶端称为须(whisker),须以外的点称为边远数据点(outliers).

图6中,  $X^{\text{IMCCMO}}$ 表示由算法IMCCMO得到的非劣解集,  $X^{\text{NSGA-II}}$ 表示由算法NSGA-II得到的非劣解集.图6(a)给出了四个测试问题的关于  $(X^{\text{IMCCMO}}, X^{\text{NSGA-II}})$  的

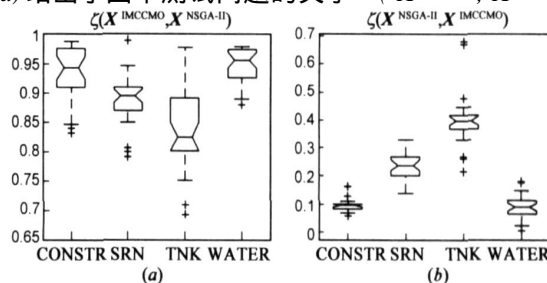


图6 算法IMCCMO和算法NSGA-II关于指标的统计结果的盒图

测试结果;图 6(b) 给出了四个测试问题关于 ( $X^{\text{NSGA-II}}$ ,  $X^{\text{IMCCMO}}$ ) 的测试结果.

对于测试问题 CONSTR 和 WATER,从图 6(a) 可以看出,30 次运行中, ( $X^{\text{IMCCMO}}$ ,  $X^{\text{NSGA-II}}$ ) 的值基本上都大于 0.9,这说明,算法 IMCCMO 得到的最优解集在很大程度上支配着由算法 NSGA-II 得到的最优解集;从图 6(b) 也可以看出 ( $X^{\text{NSGA-II}}$ ,  $X^{\text{IMCCMO}}$ ) 的值基本上集中在 0.1 附近,说明算法 NSGA-II 得到的最优解集基本上不能支配由算法 IMCCMO 得到的最优解集.因此对于测试问题 CONSTR 和 WATER,与算法 NSGA-II 相比,算法 IMCCMO 具有较强的收敛性,能较好的向着约束最优 Pareto 前端逼近.而对于 SRN 和 TNK,从图 6(a) 可以看出,30 次运行中, ( $X^{\text{IMCCMO}}$ ,  $X^{\text{NSGA-II}}$ ) 的值基本上集中在 0.8 和 0.9 之间,这说明,算法 IMCCMO 得到的最优解集在一定程度上支配着由算法 NSGA-II 得到的最优解集;从图 6(b) 也可以看出 ( $X^{\text{NSGA-II}}$ ,  $X^{\text{IMCCMO}}$ ) 的值基本上集中在 0.2 和 0.4 之间,说明算法 NSGA-II 得到的最优解集较弱地支配着由算法 IMCCMO 得到的最优解集.因此对于这两个测试问题,与算法 NSGA-II 相比,算法 IMCCMO 具有一定的优势,能较好的向着约束最优 Pareto 前端逼近.从而可以得出,对于这 4 个测试问题,与算法 NSGA-II 相比,算法 IMCCMO 找到的最优解能更好地逼近约束最优 Pareto 前端,具有较强的收敛性.

## 5 结论

基于免疫记忆克隆算法,本文提出一种用于解决 CMOP 的人工免疫系统新算法—免疫记忆克隆约束多目标优化(IMCCMO)算法.该算法并行采用抗体种群的演化过程和记忆单元的演化过程.抗体种群是与抗原作用(解空间搜索)的基本种群,强调全局搜索;而记忆单元主要是在抗体种群的基础上,对接近于可行域边缘的不可行解进行的局部搜索,一方面是记忆搜索过程,另一方面通过记忆学习,保持种群的多样性.另外,正是由于存在记忆单元,才更好地实现了抗体间的相互协作,更好的保证了在演化过程中,使得解集能够从可行域内部和不可行域的边缘向着最优解逼近,从而更好的保证了所得最优解的多样性以及很好的逼近性.用 4 个标准的测试函数对算法进行了测试,并将实验结果与算法 NSGA-II 进行了比较.实验结果表明,算法 IMCCMO 在所得最优解分布的均匀性、宽广性以及收敛性等方面都表现了较明显的优势.

## 参考文献:

- [1] K Deb, A Pratap, S Agarwal, T Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II [J]. IEEE Transac-

tions on Evolutionary Computation, 2002, 6(2): 182 - 197.

- [2] E Zitzler, L Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach [J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257 - 271.
- [3] R H Shang, W P Ma. Immune clonal MO algorithm for ZDT problems [A]. Proceedings of 3rd International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '06) [C]. Berlin: Springer-Verlag, 2006. 100 - 109.
- [4] E Zitzler, L Thiele. A simple multi-membered evolution strategy to solve constraint optimization problems [J]. IEEE Transactions on Evolutionary Computation, 2005, 9(1): 1 - 17.
- [5] Z X Cai, Y Wang. A multiobjective optimization based evolutionary algorithm for constrained optimization [J]. IEEE Transaction on Evolutionary Computation, 2006, 10(6): 658 - 675.
- [6] L. C. Jiao, L. Wang. A novel genetic algorithm based on immunity [J]. IEEE Transaction on System, Man and Cybernetic, 2000, 30(5): 552 - 561.
- [7] 潘正君, 康立山, 陈毓屏. 演化计算 [M]. 北京: 清华大学出版社, 1998.
- [8] E Zitzler, K Deb, and L Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results [J]. Evolutionary Computation, 2000, 8(2): 173 - 195.
- [9] M. Tanaka. GA-based decision support system for multicriteria optimization [A]. Proceedings of the International Conference on systems, Man and Cybernetics 2 [C]. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Incorporated, 1995. 1556 - 1561.

## 作者简介:



尚荣华 女, 1979 年 11 月生于河北阜城, 西安电子科技大学电子工程学院讲师. 主要研究方向包括: 工程优化、人工智能、智能信息处理、数据挖掘和人工免疫系统.

E-mail: rhshang@mail.xidian.edu.cn



焦李成 男, 1959 年 10 月生于陕西白水, 西安电子科技大学电子工程学院教授, 博士生导师, IEEE 高级会员. 主要研究方向包括: 信号和图像处理、非线性电路与系统理论、智能信息处理、数据挖掘和人工免疫系统.

E-mail: lchjiao@mail.xidian.edu.cn