

文章编号:1009-3087(2013)01-0138-08

## 协同进化免疫记忆克隆算法

刘 振<sup>1</sup>,胡云安<sup>1</sup>,史建国<sup>2</sup>

(1. 海军航空工程学院 控制工程系, 山东 烟台 264001; 2. 海军航空工程学院 兵器科学与技术系, 山东 烟台 264001)

**摘 要:**为提高免疫克隆算法的收敛性能,基于协调勘探和开采的思想,提出一种新型协同进化免疫记忆克隆算法。将整个进化种群分为免疫克隆种群和免疫记忆种群,将免疫克隆种群又分为2个子种群,在每个免疫克隆子种群内部采用协同竞争模式,动态更新种群规模,并采用适应度共享、动态变异机制和浓度调节机制的克隆扩增以增加种群多样性;在免疫记忆种群中采用混沌精细搜索以及精英交叉机制,提高免疫记忆种群的开采能力。利用基准测试函数进行仿真并与相关方法对比分析,所提出的算法收敛效果优越,能有效地提高全局收敛性能。

**关键词:**免疫克隆;免疫记忆;勘探;开采;协同进化

中图分类号:TP18

文献标志码:A

### Coevolutionary Immune Memory Clonal Algorithm

LIU Zhen<sup>1</sup>, HU Yun-an<sup>1</sup>, SHI Jian-guo<sup>2</sup>

(1. Dept. of Control Eng., Naval Aeronautical and Astronautical Univ., Yantai 264001, China;

2. Dept. of Ordnance Sci. and Technol., Naval Aeronautical and Astronautical Univ., Yantai 264001, China)

**Abstract:** In order to promote the convergence of immune clonal algorithm, based on the coordination of the exploration and exploitation, a new coevolutionary immune memory clonal algorithm was proposed. The whole population was divided into two population named as immune clonal population, which has two sub-populations, and immune memory population. The competitive coevolutionary operation was incorporated in the sub-population of immune clonal population to dynamic update the scale of population, and the fitness sharing, dynamic mutation mechanism and concentration regulation mechanism based clonal reproduction were also incorporated in the sub-population of immune clonal population to increase the diversity of the population. The immune memory population was optimized using the chaotic accurate search and elitist crossover in order to enhance the exploitation ability. Simulation results of benchmark functions showed that the proposed algorithm performs well and can get better results.

**Key words:** immune clonal; immune memory; exploration; exploitation; coevolutionary

免疫克隆算法是受生物免疫系统的启示的一种寻优搜索算法,由 De Castro 在 2002 年首次提出<sup>[1]</sup>,目前已经被广泛应用于组合优化、故障诊断、机器学习等领域。克隆选择算法利用克隆复制、克隆变异及克隆选择操作对个体进行操作。由于其较强的寻优能力以及操作的简便性,已经成为研究的热点。当前对免疫克隆算法的研究和改进主要在以下几个方面:

1) 对免疫克隆算法的进化操作方式的改进,包括采用动态变异方式,采用新的交叉操作方式,文献

[2] 就利用混合变异方式对个体进行更新。

2) 对其结构框架的改进,包括采用并行的结构框架以及采用竞争和协作型的协同进化。例如文献[3]就提出了一种并行的免疫进化机制;文献[4]提出了一种竞争与合作相结合的协同进化人工免疫算法。

3) 从生物系统思想出发进行的改进,包括利用生物信息机制提高算法收敛性,并将受体编辑、克隆记忆以及免疫优势等概念引入到免疫克隆进化算法中。例如文献[5]就提出了一种考虑了生物信息机制中的历史信息和环境信息来综合考虑对群体进行克隆、交叉等操作。

4) 与其他算法的融合,包括免疫克隆算法与量子算法<sup>[6]</sup>的结合及免疫克隆算法与危险理论<sup>[7]</sup>的结合。

收稿日期:2012-07-12

基金项目:国家自然科学基金资助项目(61174031;60674090)

作者简介:刘 振(1983—),男,博士生.研究方向:智能控制理论及应用.

以上算法都在一定程度上提高了免疫克隆算法的全局收敛性,但也存在一定的问题,所采用的交叉和变异方式难免破坏优良染色体的模式,单纯利用免疫克隆算法所产生的群体多样性不高。通过以上的文献可以看到,要想提高克隆选择算法的全局收敛性,必须从种群内部出发,利用种群内部信息,提高算法探索新的解空间能力,同时能够充分开发当前解空间范围内的个体,提高算法寻优过程中的开采能力,即提高算法的勘探和开采能力。文献[8]和[9]较早注意到了勘探和开采对提高算法收敛性能的重要性,分别利用随机种群和进化种群,用以提高遗传算法和粒子群算法的收敛性能,算法的运行步骤略显粗糙。在免疫克隆算法中,将勘探和开采思想相融合的文献还较少,作者对这一方面进行一些探索。

当没有外界信息注入以及没有外部人为操作的前提下,为提高算法的全局收敛性,必须充分利用种群内部的信息。以往的免疫克隆选择算法中,一般都只考虑单一种群的进化,很少考虑种群与种群之间以及种群与环境之间的相互作用,但是在自然界的生物进化机制中,个体与个体之间以及种群与种群之间都存在相互作用的关系,协同进化现象是自然界普遍存在的现象。在免疫克隆选择算法中,每一代的进化都是通过克隆扩增、克隆变异等操作来完成,因此在每一代的进化过程中,利用免疫记忆的优良个体提高算法的开采性能,同时利用其他个体进行当前解空间内的勘探操作,避免进化过程中陷入局部极值或者漏掉解空间内的最优解。因此作者基于上述思想出发,设计了一种新型协同进化免疫记忆克隆算法(coevolutionary immune memory clonal algorithm, CIMCA),以下详细阐述算法的实现过程。

## 1 协同进化免疫记忆克隆算法实现过程

### 1.1 算法基本思想

早期的免疫记忆克隆算法一般都是保留每一代中的优良个体构成免疫记忆种群,将其和免疫克隆种群进行混合选择提高算法的全局收敛性。采用这种思想的免疫记忆克隆算法对提高算法的寻优能力有限。

基于协调勘探和开采的思想,提出将整个进化种群  $AB(t)$  ( $t = 1, 2, \dots, T_{\max}$ ) 分为 2 部分:免疫克隆种群  $A(t)$  和免疫记忆种群  $B(t)$ , 2 个种群首先利用局部优化算子进行独立进化,然后进行协同免疫选择,从而实现了一次协同进化。免疫克隆种群用于

进行勘探,探索全局范围内的优良解,为保持较好的多样性以及良好的进化宽度,将免疫克隆种群  $A(t)$  划分为 2 个子种群,在各个子种群内部采用协同竞争方式,并分别利用适应度共享、浓度调节机制和动态变异机制对每个子种群进行操作,使得免疫克隆种群能够保持较好的多样性。充分考虑到免疫记忆种群中个体的优越性,在免疫记忆种群内采用混沌精细搜索、精英交叉操作来提高算法的开采性能,从而能够不断探索新的解空间,提高算法的整体收敛性能。当免疫克隆种群和免疫记忆种群各自完成局部进化后,将整个进化种群进行免疫选择。以下详细阐述算法的实现过程。

### 1.2 免疫克隆种群的进化过程

#### 1.2.1 免疫克隆子种群协同进化的实现

将免疫克隆种群  $A(t)$  分为 2 个子种群  $pop_1$  和  $pop_2$  进行协同竞争操作,其子种群规模按照式(1)进行调整<sup>[10]</sup>:

$$\begin{cases} \frac{dN_1}{dt} = r_1 \left(1 - \frac{N_1}{K_2} - \frac{\alpha_1 N_2}{K_1}\right) N_1, \\ \frac{dN_2}{dt} = r_2 \left(1 - \frac{N_2}{K_1} - \frac{\alpha_2 N_1}{K_2}\right) N_2 \end{cases} \quad (1)$$

式中,  $N_1$  和  $N_2$  分别为  $pop_1$  和  $pop_2$  的种群规模,  $K_1$  和  $K_2$  分别为不存在竞争的情形下,每个子种群的最大数目,  $r_1$  和  $r_2$  为 2 个子种群的自然增长率,  $\alpha_1$  和  $\alpha_2$  为竞争系数。在式(1)中,除了竞争系数以外,其余均为常数,以下讨论如何确定竞争系数。

从  $pop_1$  和  $pop_2$  中,分别随机选择若干个体构成竞争集合  $cp_1$  ( $cp_1 < N_1$ ) 和  $cp_2$  ( $cp_2 < N_2$ ),竞争系数按式(2)获得:

$$\alpha_1 = \frac{\sum_{j \in cp_2} \sum_{i \in pop_1} \chi_{ij}}{\sum_{j \in cp_1} \sum_{i \in pop_2} \chi_{ij}}, \quad \alpha_2 = \frac{1}{\alpha_1} \quad (2)$$

在第  $t$  次迭代中,个体  $i \in pop_1$  和  $j \in cp_2$  的适应度满足  $f_i(t) > f_j(t)$  时,  $\chi_{ij} = 1$ , 否则  $\gamma_{ij} = 0$ 。

当  $\frac{dN_i}{dt} > 0$  ( $i = 1, 2$ ) 时,第  $i$  个子种群在竞争中获胜,则对第  $i$  个子种群执行补充操作,随机产生若干个体补充到该种群中。同理,当  $\frac{dN_i}{dt} < 0$  时,删除种群中适应度较差的个体,以提高该子种群的竞争能力。

#### 1.2.2 适应度共享

为提高种群分布的多样性,使得选择机制能够有效的指导进化方向,避免过大的选择压导致的局

部收敛,引入适应度共享机制以增加种群选择的多样性<sup>[11]</sup>。对于2个抗体 $a_i(t)$ 和 $a_j(t)$ ,定义其共享函数为:

$$Sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_s}\right)^2, & d_{ij} < \sigma_s; \\ 0, & \text{其它} \end{cases} \quad (3)$$

$\sigma_s$ 为共享半径,即预先设置的抗体之间的差异值, $d_{ij}$ 为2个抗体之间的距离度量,在给定共享函数后,对规模为 $N$ 的种群,个体 $a_i(t)$ 的适应度为:

$$f_i^*(t) = f_i(t) / \sum_{j=1}^N Sh(d_{ij}) \quad (4)$$

在免疫克隆后的种群中,淘汰相似度较近的个体,设定编码长度为 $L$ 的抗体 $a_i(t) = [a_{i1}(t), a_{i2}(t), \dots, a_{iL}(t)]$ 和 $a_j(t) = [a_{j1}(t), a_{j2}(t), \dots, a_{jL}(t)]$ ,定义其相对距离为: $d_{ij} = \sqrt{\sum_{k=1}^L (a_{ik} - a_{jk})^2}$ ,若 $d_{ij} \leq d_{\max}$ ,则淘汰适应度值较差的个体, $d_{\max}$ 为距离度量阈值, $d_{\max} < \sigma_s$ 。

### 1.2.3 基于浓度调节的克隆扩增

在免疫克隆种群中,为了保持种群多样性,在第 $t$ 次迭代中第 $i$ 个个体的选择概率为<sup>[12]</sup>:

$$p_i(t) = \mu \frac{f_i(t)}{\sum_{j=1}^N f_j(t)} + (1 - \mu) \frac{1}{N} e^{(-C_i(t)/v)} \quad (5)$$

$\mu$ 和 $v$ 为调节系数,经多次测试,当 $\mu = 0.8$ 及 $v = 12$ 时,克隆扩增能使种群保持良好的多样性, $C_i(t) = \frac{h_i(t)}{N}$ , $h_i(t)$ 为适应度在 $[f_i(t), f_i(t) + \Delta f(t)]$ 个体数目, $\Delta f(t) = (f_{\max}(t) - f_{\min}(t))/3$ , $N$ 为免疫克隆种群规模。当在第 $t$ 次迭代时的免疫克隆种群 $A(t)$ ,经克隆扩增后为: $A(t)' = [A_{n_1}(t), A_{n_2}(t), \dots, A_{n_N}(t)]$ ,其中,

$$n_i = N \sin(\pi \cdot p_i/2) \quad (6)$$

### 1.2.4 动态变异操作

为提高种群多样性,对个体采用动态的自适应变异方式,变异概率设置为:

$$p_m(t) = p_m \cdot e^{(1-T/T_{\max})} \quad (7)$$

柯西变异的变异步长变化较大,适合于进行全局范围内的搜索。柯西分布相比高斯分布,函数分布更加平滑并且概率密度函数均大于0,从而更加有利于进行全局搜索操作。因此在免疫克隆种群中采用柯西变异进行全局范围内的搜索。对于抗体 $a_i(t)$ ,经过柯西免疫后为:

$$a_{ij}'(t) = a_{ij}(t) + \gamma_{ij}'(t) \cdot \zeta_j(t) \quad (8)$$

式中, $\gamma_{ij}'(t) = \gamma_{ij}(t) \cdot e^{(\tau_1 \cdot \eta_i(t) + \tau_2 \cdot \eta_j(t))}$ , $\eta_i(t)$ 和 $\eta_j(t)$ 为常数,均服从高斯分布, $i = 1, 2, \dots, N, j = 1, 2, \dots, L, \tau_1 = 1/(\sqrt{2N}), \tau_2 = 1/(\sqrt{2} \sqrt{N}), \gamma_{ij}(0) = 3 \sqrt{N}$ 。当 $\zeta_j(t)$ 满足柯西分布时,式(8)为柯西变异。

## 1.3 免疫记忆种群进化过程

### 1.3.1 混沌精细搜索机制

为提高免疫记忆种群的开采能力,对免疫记忆个体进行混沌精细搜索。设定在免疫记忆种群中执行精细搜索的某个抗体: $b_i(t) = [b_{i1}(t), b_{i2}(t), \dots, b_{iL}(t)]$ , $i = 1, 2, \dots, M, M$ 为免疫记忆种群规模,令

$$b_{ij}^k(t) = l_j - z_j^k (u_j - l_j) \quad (9)$$

式中, $j = 1, 2, \dots, L, u_j$ 和 $l_j$ 表示 $b_{ij}(t)$ 的取值范围。利用混沌迭代式(10)首先进行混沌迭代搜索操作:

$$z_j^{k+1} = \eta z_j^k (1 - z_j^k) \quad (10)$$

式中, $\eta = 4$ 。当寻找到更优状态时,将混沌变量变换到解空间,如果连续一定迭代次数内最优值不再发生变化,停止混沌迭代,进行混沌精细搜索过程<sup>[13]</sup>,令 $b_{ij}^*$ 表示寻找到最优解, $z_j^*$ 为最优混沌变量。对变量的搜索区间进行缩放: $l_j' = b_{ij}^* - \psi(u_j - l_j)$ , $u_j' = b_{ij}^* + \psi(u_j - l_j)$ , $\psi \in (0, 0.5)$ ,为保证区间缩放的正确性,若 $l_j' < l_j$ ,则令 $l_j' = l_j$ ,若 $u_j' < u_j$ ,则令 $u_j' = u_j$ ,并重新定义最优混沌变量 $z_j^* = \frac{b_{ij}^* - l_j'}{u_j' - l_j'}$ ,同时令:

$$c_j^k = (1 - \beta) z_j^* + \beta z_j^k \quad (11)$$

式中, $\beta = 1 - \left(\frac{t-1}{t}\right)^2$ , $t$ 为迭代次数。以 $c_j^k$ 为新的混沌迭代变量,按照式(12)进行迭代:

$$c_j^{k+1} = \eta c_j^k (1 - c_j^k) \quad (12)$$

当在一定迭代次数内寻找到的最优值不发生变化时,停止混沌精细搜索过程。

### 1.3.2 变异方式的设置

高斯变异适合于进行局部精细的搜索,在免疫记忆种群中采用高斯变异进行局部搜索。将所有个体的变异概率设置为: $p_m(t) = p_m$ ,在免疫记忆种群中采用高斯变异,即式(8)中 $\zeta_j(t)$ 为高斯分布。

### 1.3.3 精英交叉操作

为保持种群内部的局部探索能力,在免疫记忆种群中选择最优解,对免疫记忆种群中的所有个体进行精英交叉操作。假设此时免疫记忆种群最优解为 $b^e(t)$ ,对于免疫记忆种群中的个体以交叉概率 $p_c$

与  $b^e(t)$  进行交叉操作。为避免优良模式的破坏,交叉方式采用单点交叉,即随机选择一个交叉位置,互换个体之间的基因位。

## 2 算法的流程和性能分析

### 2.1 算法流程

通过以上的表述,提出的 CIMCA 流程为:

步骤 1:初始化参数信息,包括种群规模  $Q$ 、染色体长度  $L$ 、循环次数  $T_{\max}$  以及变异概率  $p_m$ 、交叉概率  $p_c$ ,随机初始化抗体种群:

$$AB(0) = \{ab_1(0), ab_2(0), \dots, ab_n(0)\};$$

步骤 2:计算抗体和抗原的亲密度,将整个抗体种群  $AB(t)$  分为免疫克隆种群  $A(t)$  以及免疫记忆种群  $B(t)$ ,即  $AB(t) = \{A(t), B(t)\}$ ;

步骤 3:在免疫克隆种群  $A(t)$  内,进行协同竞争,根据式(1)进行免疫克隆子种群规模的调整,同时采用式(4)进行适应度共享操作调节个体适应度,并淘汰个体相似度高的抗体;

步骤 4:对免疫克隆种群内的抗体进行克隆扩增操作,克隆规模按照式(6)获得,并对个体进行动态柯西变异操作及交叉操作,其中变异概率按照式(7)获得,交叉操作采用单点交叉;

步骤 5:在免疫记忆种群  $B(t)$  内采用混沌精细搜索操作,并采用高斯变异操作及精英交叉;

步骤 6:免疫克隆种群和免疫记忆种群进行混合免疫选择,完成一次种群协同进化;

步骤 7:判断是否满足结束条件,满足则结束,否则转步骤 2。

### 2.2 收敛性分析

传统的免疫克隆算法的进化种群序列为有限齐次马尔科夫过程<sup>[6]</sup>,从作者提出的协调勘探和开采的 CIMCA 流程可以看出,与传统免疫记忆克隆算法的主要区别在于将进化种群分为 2 个种群进行协同进化,在每一个种群内分别采用了有利于勘探和开采的局部操作算子。因此提出的 CIMCA 中的种群序列  $AB(t+1)$  由  $AB(t)$  所决定,因此依然为马尔科夫序列,以下进行详细证明。

**定理 1** CIMCA 的进化种群序列  $AB(t)$  为有限齐次马尔科夫过程。

证明:设问题所需要的精度为  $r$ ,整个状态空间可以表示为  $\Omega = \prod_{i=1}^L \lceil (u_i - l_i)/r \rceil$ ,因此状态空间  $|\Omega|$  是有限的。又由于:

$$AB(t+1) = T_s \circ (T_u \circ A(t) + T_d \circ B(t)) \quad (13)$$

式中,  $T_u$  为免疫克隆种群内的进化算子,包括子种群竞争操作、适应度共享操作、免疫变异、免疫重组及克隆扩增,  $T_d$  为免疫记忆种群内的操作算子,包括混沌精细搜索操作、免疫变异及精英交叉操作,  $T_s$  为免疫克隆种群和免疫记忆种群混合选择操作。又因为:

$$T_q \circ AB(t) = T_u \circ A(t) + T_d \circ B(t) \quad (14)$$

故  $AB(t+1) = T_s \circ T_q \circ AB(t)$ 。可以看到  $T_u, T_d$  均与  $t$  无关,即  $T_q$  也与进化迭代次数无关,  $T_s$  也与  $t$  不相关。因此  $AB(t+1)$  仅由  $AB(t)$  决定,即  $\{AB(t), t \geq 1\}$  是有限齐次马尔科夫过程,证毕。

在给定  $AB(t)$  为有限齐次马尔科夫过程后。令  $s^* = \{ab \mid \max_{ab \in AB(t)} f(ab) = f^*\}$ ,  $s^*$  表示最优解集,  $f^*$  为最优值,依据文献[6]给出定义 1。

**定义 1** 令  $f(t) = \max_{ab_i \in AB(t)} \{f(ab_i), i = 1, 2, \dots, Q\}$ , 当满足  $\lim_{t \rightarrow \infty} P\{f(t) = f^*\} = 1$  时,进化算法以概率 1 收敛。

**定理 2** 提出的协调勘探和开采的 CIMCA 是以概率 1 收敛的。

证明:当种群  $AB(t)$  处于某一状态  $s_i$  时可以表示为  $AB^i(t)$ ,种群  $AB(t)$  的状态转移概率为:

$$P_{ij}(t) = P\{AB^j(t+1)/AB^i(t)\} \quad (14)$$

令  $I = \{i \mid s_i \cap s^* \neq \emptyset\}$ ,  $P_i(t)$  表示种群  $AB(t)$  处于状态  $s_i$  的概率,则  $P(t) = \sum_{i \in I} P_i(t)$ ,按照文献[6]推理方法可以得到  $\lim_{t \rightarrow \infty} P(t) = 0$ ,又因为:

$$\lim_{t \rightarrow \infty} P\{f(t) = f^*\} = 1 - \lim_{t \rightarrow \infty} \sum_{i \notin I} P_i(t) = 1 - \lim_{t \rightarrow \infty} P(t) \quad (15)$$

因此可以得到  $\lim_{t \rightarrow \infty} P\{f(t) = f^*\} = 1$ ,从而由定义 1 可以得出提出的 CIMCA 以概率 1 收敛,证毕。

### 2.3 算法复杂度分析

对提出的 CIMCA 复杂度进行分析,分别对免疫克隆种群以及免疫记忆种群的复杂度进行分析。在免疫克隆种群中,子种群竞争操作和克隆扩增操作的复杂度均为  $O(N \cdot T_{\max})$ ,适应度共享操作、免疫变异和免疫重组操作复杂度均为  $O(N \cdot L \cdot T_{\max})$ ;在免疫记忆种群中,混沌精细搜索操作、免疫变异和精英交叉的复杂度为  $O(M \cdot L \cdot T_{\max})$ ,对免疫克隆种群和免疫记忆种群进行免疫选择的复杂度为  $O(Q \cdot L \cdot T_{\max})$ ,因此 CIMCA 的复杂度为:

$$T_C = O(N \cdot T_{\max}) + O(N \cdot T_{\max}) + O(N \cdot L \cdot T_{\max}) + O(N \cdot L \cdot T_{\max}) + O(M \cdot L \cdot T_{\max}) + O(M \cdot L \cdot T_{\max}) + O(M \cdot L \cdot T_{\max}) + O(Q \cdot L \cdot T_{\max}) \quad (16)$$

在传统的免疫克隆选择算法中,免疫克隆操作的复杂度为  $O(Q \cdot T_{\max})$ ,免疫选择和免疫基因操作复杂度均为  $O(Q \cdot L \cdot T_{\max})$ 。因此所提出的 CIMCA 的复杂度将有所增加。

### 3 基准函数仿真分析

为了验证算法的正确性,同时对算法的收敛性能、收敛速度以及参数进行验证对比分析,选择基准函数进行仿真分析。

$$f_1(x) = (4 - 2 \cdot 1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_1^2)x_2^2,$$

$$x_i \in [-3, 3];$$

$$f_2(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2, x_i \in [-2.048, 2.048];$$

$$f_3(x) = \sum_{i=1}^n x_i^2, n = 2, x_i \in [-5.12, 5.12];$$

$$f_4(x) = \sum_{i=1}^5 i \cos[(i+1)x_1 + i] + \sum_{i=1}^5 i \cos[(i+1)x_2 + i],$$

$$x_i \in [-10, 10];$$

$$f_5(x) = 418.9829n - \sum_{i=1}^n x_i \sin(\sqrt{|x_i|}),$$

$$n = 30, x_i \in [-500, 500];$$

$$f_6(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)],$$

$$n = 30, x_i \in [-5.12, 5.12];$$

$$f_7(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{i}\right),$$

$$n = 30, x_i \in [-600, 600];$$

$$f_8(x) = -20 \left( -0.02 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp\left( \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) +$$

$$20 + e, n = 30, x_i \in [-30, 30];$$

$$f_9(x) = 1 - \left[ \exp\left(-0.5 \sum_{i=1}^n x_i^2\right) \right],$$

$$n = 30, x_i \in [-600, 600];$$

$$f_{10}(x) = \sum_{i=1}^n x_i^2 \left( \sum_{i=1}^n \frac{i}{2} x_i \right)^2 + \left( \sum_{i=1}^n \frac{i}{2} x_i \right)^4,$$

$$n = 30, x_i \in [-5.12, 5.12];$$

$$f_{11}(x) = \sum_{i=1}^n (x_i - i)^2, n = 30, x_i \in [-30, 30];$$

$$f_{12}(x) = x_1^2 + 100000 \sum_{i=2}^n x_i^2, n = 30, x_i \in [-10, 10]。$$

进化种群规模  $Q = 200$ ,免疫克隆种群规模为  $N = 120$ ,免疫记忆种群规模为  $M = 80$ , $K_1 = K_2 = 300$ , $cp_1$  和  $cp_2$  取免疫克隆子种群规模的 10%, $p_m = 0.1$ , $p_c = 0.7$ 。

### 3.1 低维函数仿真分析

对提出的 CIMCA,以 4 个基准函数  $f_1 \sim f_4$  为例,在低维空间范围内进行测试,维数均为 2。将提出的 CIMCA 与免疫克隆选择算法 (ICSA)、文献 [14] 提出的免疫记忆克隆选择算法 (IMCSA) 以及文献 [15] 提出的混沌免疫进化算法 (CIEA) 进行仿真对比和分析。表 1 为算法独立运行 20 次的统计结果,其中每次运行迭代次数为 100。

表 1 低维基准函数仿真对比结果

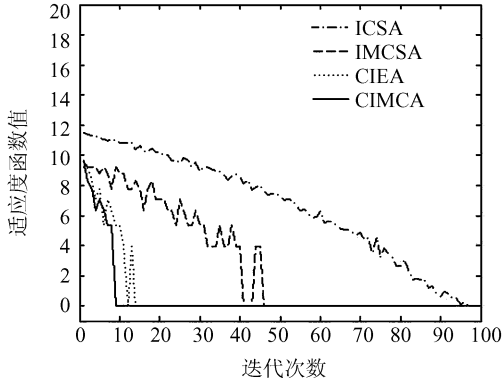
Tab.1 Comparison results for the low dimension functions

函数	算法	均值	标准差
$f_1$	ICSA	-1.02	$2.45 \times 10^{-2}$
	IMCSA	-1.03	$2.11 \times 10^{-2}$
	CIEA	-1.03	$1.82 \times 10^{-2}$
	CIMCA	-1.03	$1.17 \times 10^{-2}$
$f_2$	ICSA	$1.01 \times 10^{-1}$	$6.29 \times 10^{-2}$
	IMCSA	$9.61 \times 10^{-2}$	$3.74 \times 10^{-3}$
	CIEA	$5.80 \times 10^{-3}$	$4.15 \times 10^{-3}$
	CIMCA	$1.91 \times 10^{-3}$	$2.87 \times 10^{-3}$
$f_3$	ICSA	$4.82 \times 10^{-1}$	$8.71 \times 10^{-2}$
	IMCSA	$2.81 \times 10^{-2}$	$2.60 \times 10^{-3}$
	CIEA	$3.26 \times 10^{-3}$	$2.09 \times 10^{-3}$
	CIMCA	$2.05 \times 10^{-3}$	$7.14 \times 10^{-4}$
$f_4$	ICSA	$-1.69 \times 10^2$	1.43
	IMCSA	$-1.72 \times 10^2$	$4.89 \times 10^{-1}$
	CIEA	$-1.84 \times 10^2$	$3.23 \times 10^{-1}$
	CIMCA	$-1.86 \times 10^2$	$3.09 \times 10^{-3}$

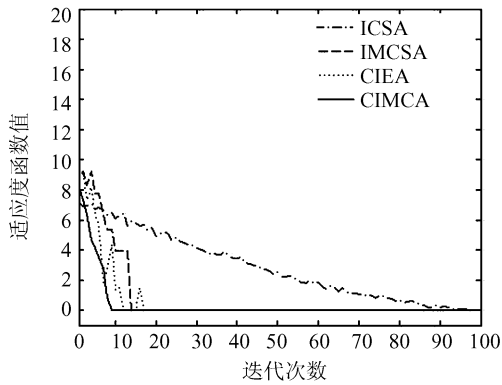
由表 1 可知,提出的算法能有效地提高 4 个基准函数的收敛性能,在均值和标准差方面均优于其他 3 种进化算法。本文算法优于其他几种进化算法主要是因为采用协同进化机制增加了全局勘探和局部开采的能力,从而有利于算法能够找到更加可靠的解。同时也注意到由于文献 [14] 只是采用免疫记忆策略保留每一次运行过程中的最优解,并没有充分利用免疫记忆个体进行二次局部开采,因此该算法在很多情况下寻优结果劣于文献 [15] 所提出的利用混沌机制进行局部搜索的免疫进化算法。作者提出 CIMCA 能够充分协调勘探和开采能力,故优于其它几种相关算法,充分说明了利用种群优良的免疫记忆个体进行局部开采并利用克隆种群个体进行全局勘探的重要性。

对低维函数进行进一步分析,为节省篇幅,以  $f_2$

和  $f_3$  函数为例进行说明,4 种算法的对比仿真结果如图 1 所示。从图 1 可以看出,由于引入免疫记忆机制,所以 IMCSA 收敛效果优于 ICSA,显示出免疫记忆的优越性。CIEA 以及 CIMCA 都采用了局部的搜索机制,因而算法收敛性能都优于传统的 ICSA 和 IMCSA。由于本文算法在采用局部搜索的同时也采用了子种群竞争,因而收敛效果优于 CIEA。



(a)  $f_2$  函数对比结果



(b)  $f_3$  函数对比结果

图 1 低维函数的仿真对比结果

Fig. 1 Comparison results for low dimension functions

### 3.2 高维函数仿真分析

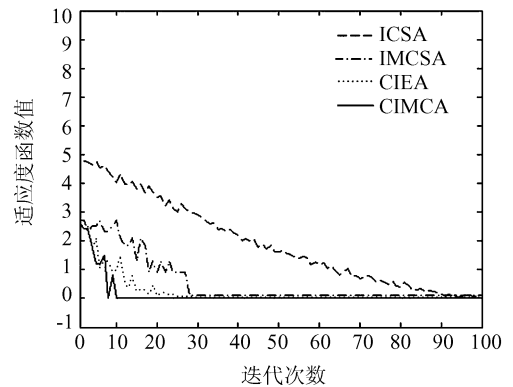
利用 4 个高维函数  $f_5 \sim f_8$ , 对提出的 CIMCA 在高维空间范围内进行测试,维数均为 30。利用提出的 CIMCA 与 ICSA、IMCSA 及 CIEA 进行对比仿真分析,表 2 为 4 种算法独立运行 20 次的仿真对比结果。从表 2 可以看出,提出的 CIMCA 在均值和标准差均优于其它几种算法。

从图 2 的迭代收敛曲线可以看出,利用 ICSA 和 IMCSA 都存在陷入局部极值问题的危险,同时也注意到利用作者提出的 CIMCA 在优化 30 维的  $f_6$  和  $f_8$  函数时能迅速收敛到全局极值附近,但传统的 ICSA 收敛速度较慢,收敛效果也劣于本文算法,同时也可以看到,在函数维数较高的情形下,ICSA 和 IMCSA 由于在多样性保持上的劣势,此时易陷入局部极值,从图 2(b)可以清晰地看到这一点。

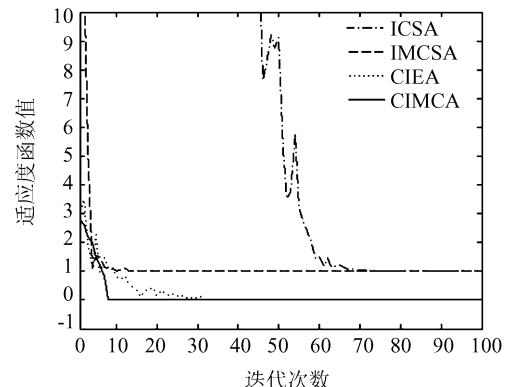
表 2 高维函数仿真对比结果

Tab. 2 Comparison results for the high dimension functions

函数	算法	均值	标准差
$f_5$	ICSA	$4.56 \times 10^{-1}$	$3.04 \times 10^{-1}$
	IMCSA	$2.50 \times 10^{-1}$	$4.36 \times 10^{-2}$
	CIEA	$6.71 \times 10^{-3}$	$6.28 \times 10^{-3}$
	CIMCA	$1.39 \times 10^{-4}$	$8.43 \times 10^{-4}$
$f_6$	ICSA	$7.01 \times 10^{-1}$	$5.64 \times 10^{-1}$
	IMCSA	$8.97 \times 10^{-2}$	$7.41 \times 10^{-3}$
	CIEA	$6.13 \times 10^{-2}$	$5.29 \times 10^{-3}$
	CIMCA	$2.89 \times 10^{-2}$	$5.21 \times 10^{-3}$
$f_7$	ICSA	$3.69 \times 10^{-1}$	$2.7 \times 10^{-1}$
	IMCSA	$6.29 \times 10^{-2}$	$3.16 \times 10^{-2}$
	CIEA	$2.37 \times 10^{-3}$	$1.15 \times 10^{-3}$
	CIMCA	$6.89 \times 10^{-4}$	$2.25 \times 10^{-4}$
$f_8$	ICSA	$8.13 \times 10^{-1}$	$1.38 \times 10^{-1}$
	IMCSA	$6.81 \times 10^{-1}$	$4.36 \times 10^{-2}$
	CIEA	$2.43 \times 10^{-2}$	$8.14 \times 10^{-3}$
	CIMCA	$7.32 \times 10^{-3}$	$3.34 \times 10^{-3}$



(a)  $f_6$  函数对比结果



(b)  $f_8$  函数对比结果

图 2 高维函数的仿真对比结果

Fig. 2 Comparison results for high dimension functions

### 3.3 收敛速度仿真分析

利用  $f_9$  和  $f_{10}$  函数,对提出的 CIMCA 进行收敛速度的仿真比较和分析。每种算法独立运行 20 次,当每次得到的最优结果与理论最优值之差小于  $10^{-3}$  时,视为算法收敛,将每种算法得到的结果进行统计如表 3 所示。

表 3 收敛速度对比结果

Tab.3 Comparison results for the convergence speed

函数	算法	运行时间/s		收敛次数
		均值	标准差	
$f_9$	ICSA	4.21	1.16	6
	IMCSA	4.56	1.09	11
	CIEA	7.45	0.89	13
	CIMCA	6.98	0.58	16
$f_{10}$	ICSA	5.34	1.55	6
	IMCSA	5.78	1.47	9
	CIEA	9.32	1.34	14
	CIMCA	7.73	1.22	15

由表 3 的统计结果也可以看出,本文算法的平均运行时间并不占优,只是略优于 CIEA,运行时间的标准差却小于其它几种算法,收敛到最优解次数也明显多于其它几种算法。这主要是因为作者提出的 CIMCA 将进化种群个体分为 2 个种群,并执行独立的进化操作,从而增加了算法复杂性,同时也增加了系统的开销,导致每次迭代过程运行的时间增加,故平均运行时间比 ICSA 和 IMCSA 略有增加,但算法运行时间的标准差更小,因而更加稳定。

### 3.4 参数影响分析

#### 3.4.1 免疫克隆种群和免疫记忆种群比例影响分析

改变免疫克隆种群规模  $N$  和免疫记忆种群规模  $M$ ,分析进行勘探和开采的种群比例对 CIMCA 算法的影响。以  $f_{11}$  函数为例,独立运行 20 次,每次迭代运行次数为 1 000。将得到最优值的均值、标准差和平均运行时间进行统计,结果如表 4 所示。

从表 4 可以看出,进行勘探和开采的种群规模比例不宜设置得过大或过小,即当  $N:M$  较大时,算法较多的种群个体进行勘探,虽然算法的收敛速度相对较快,但收敛得到的均值却不好,当  $N:M$  较小时,此时算法收敛效果相对较优,但算法收敛时间较长,当  $N:M = 5:5$  取得的效果最优,但运行时间较之  $N:M = 6:4$  时要长,故在收敛效果基本接近的情形下,建议算法的种群比例取为  $6:4$ 。

表 4 种群规模对算法的影响对比结果

Tab.4 Comparison results for the scale of population

$N:M$	$f_{11}$ 函数		
	均值	标准差	平均运行时间/s
9:1	$8.85 \times 10^{-4}$	$7.93 \times 10^{-4}$	5.04
8:2	$7.95 \times 10^{-4}$	$6.25 \times 10^{-4}$	4.68
7:3	$7.32 \times 10^{-4}$	$6.71 \times 10^{-4}$	4.79
6:4	$6.76 \times 10^{-4}$	$5.34 \times 10^{-4}$	4.67
5:5	$6.58 \times 10^{-4}$	$5.95 \times 10^{-4}$	5.60
4:6	$6.82 \times 10^{-4}$	$6.07 \times 10^{-4}$	5.63
3:7	$7.12 \times 10^{-4}$	$6.23 \times 10^{-4}$	6.74
2:8	$7.43 \times 10^{-4}$	$6.45 \times 10^{-4}$	6.45
1:9	$7.58 \times 10^{-4}$	$7.22 \times 10^{-4}$	7.19

#### 3.4.2 子种群竞争和混沌搜索对算法的影响分析

为了分析免疫克隆种群内部竞争方式对全局勘探能力的影响,以及免疫记忆种群的混沌精细搜索机制对局部开采的影响,将不采用 1.2.1 节的子种群竞争的算法和不采用 1.3.1 节的混沌精细搜索机制后的算法以及作者提出 CIMCA,以  $f_9$   $f_{10}$  和  $f_{12}$  函数为例进行仿真分析。将不采用子种群竞争的算法记为 CIMCA<sub>wpc</sub>,将不采用混沌精细搜索机制的算法记为 CIMCA<sub>wcs</sub>,各种算法独立运行 20 次,每次运行迭代次数为 100,统计结果如表 5 所示,其中  $f_{12}$  函数的迭代收敛曲线如图 3 所示。

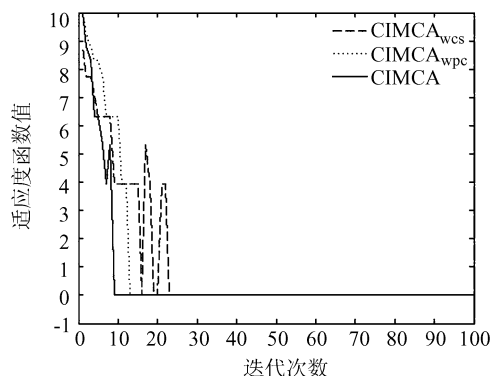


图 3  $f_{12}$  函数对比结果

Fig.3 Comparison of simulation result for  $f_{12}$

从表 5 的统计结果可以看出,当在免疫克隆种群内不采用子种群竞争后,算法在全局勘探能力上有所下降,从而导致算法的均值和标准差都有所降低,当不采用混沌精细搜索后,算法的平均运行时间明显缩短,说明混沌精细搜索过程在整个算法运行过程中占用了较长的时间,同时也注意到混沌精细搜索机制对算法的收敛精度有明显提高。从表 5 中还可以看到 CIMCA<sub>wpc</sub> 所得到的结果要优于 CIMCA<sub>wcs</sub> 所得到的结果,即说明了混沌精细搜索机制在提高本文算法性能上优于种群竞争机制。同时也注意到当采用子种群竞争和混沌精细搜索机制后,即

提出的 CIMCA, 仿真函数的收敛精度明显提高。从表 5 还可以看到本文算法的运行时间略大于 CIMCA<sub>wpc</sub> 和 CIMCA<sub>wcs</sub>, 反应出进化算法的性能指标很难同时达到最优, 从一个侧面印证了进化算法研究领域著名的“*No Free Lunch*”定理的正确性。

表 5 种群竞争和混沌搜索对算法的影响对比结果

Tab. 5 Comparison results for population competitive and chaos search

函数	算法	均值	标准差	平均时间/s
$f_9$	CIMCA <sub>wpc</sub>	$6.01 \times 10^{-2}$	$4.14 \times 10^{-2}$	0.48
	CIMCA <sub>wcs</sub>	$3.74 \times 10^{-1}$	$5.80 \times 10^{-2}$	0.35
	CIMCA	$7.19 \times 10^{-3}$	$2.87 \times 10^{-3}$	0.59
$f_{10}$	CIMCA <sub>wpc</sub>	$2.97 \times 10^{-2}$	$1.71 \times 10^{-2}$	0.64
	CIMCA <sub>wcs</sub>	$4.17 \times 10^{-2}$	$3.26 \times 10^{-2}$	0.38
	CIMCA	$9.40 \times 10^{-3}$	$5.01 \times 10^{-4}$	0.76
$f_{12}$	CIMCA <sub>wpc</sub>	$3.43 \times 10^{-2}$	$2.71 \times 10^{-2}$	0.32
	CIMCA <sub>wcs</sub>	$6.26 \times 10^{-2}$	$5.13 \times 10^{-2}$	0.29
	CIMCA	$7.82 \times 10^{-3}$	$7.65 \times 10^{-4}$	0.45

## 4 结 论

提出一种能够协调勘探和开采的协同进化免疫记忆克隆算法, 主要贡献在于:

1) 在传统免疫克隆选择算法的框架之上, 借鉴勘探和开采的思想, 将进化种群分为免疫克隆种群和免疫记忆种群分别进行协同进化。

2) 在免疫克隆种群内部通过子种群竞争、适应度共享和动态变异操作, 并利用基于浓度调节的扩增操作, 增加种群多样性, 提高算法的勘探能力; 在免疫记忆种群内进行混沌精细搜索以及精英交叉操作, 提高算法的开采能力。

3) 对算法的利用基准函数进行了验证, 并对算法的收敛速度、参数选取以及影响因素进行了仿真分析。

作者提出的算法在收敛性能上相比较之前的算法有了一定的提高, 但算法的收敛时间还有待提高, 另外对算法的时间复杂度以及收敛性分析方面还有待深入, 这将是下一步重点研究的内容。

### 参考文献:

[1] De Castro L N, Von Zuben F J. Learning and optimization using the clonal selection principle[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(3): 239 - 251.

[2] Chen J Y, Lin Q Z, Ji Z. A hybrid immune multi-objective optimization algorithm[J]. European Journal of Operational Research, 2010, 204(2): 294 - 302.

[3] Zhao J, Liu Q L, Wang W, et al. A parallel immune algorithm for traveling salesman problem and its application on

cold rolling scheduling[J]. Information Sciences, 2011, 181(7): 1212 - 1223.

[4] Liu Chaohua, Zhang Jing, Zhang Yingjie, et al. Competitive cooperative coevolutionary immune-dominant clone selection algorithm for solving the traveling salesman problem[J]. Control Theory & Applications, 2010, 27(10): 1322 - 1330. [刘朝华, 章兢, 张英杰, 等. 竞争合作型协同进化免疫算法及其在旅行商问题中的应用[J]. 控制理论与应用, 2010, 27(10): 1322 - 1330.]

[5] Liu Shirong, Zhang Botao. Quantum Immune clonal algorithm based on biologic information mechanism[J]. Pattern Recognition and Artificial Intelligence, 2008, 21(5): 592 - 597. [刘士荣, 张波涛. 采用生物信息机制的量子免疫克隆算法[J]. 模式识别与人工智能, 2011, 24(3): 392 - 398.]

[6] Jiao L C, Li Y Y, Gong M G, et al. Quantum-inspired immune clonal algorithm for global optimization[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2008, 38(5): 1234 - 1253.

[7] Xu Bin, Zhuang Yi. Adaptive immune algorithm based on danger theory[J]. Journal of Sichuan University: Engineering Science Edition, 2011, 43(3): 123 - 132. [许斌, 庄毅. 基于危险理论的自适应免疫算法[J]. 四川大学学报: 工程科学版, 2011, 43(3): 123 - 132.]

[8] Jiang Rui, Luo Yupin, Hu Dongcheng, et al. A genetic algorithm by coordinating exploration and exploitation—convergence properties and performance analyses[J]. Chinese J Computers, 2001, 24(12): 1234 - 1241. [江瑞, 罗予频, 胡东成, 等. 一种协调勘探和开采的遗传算法: 收敛性及性能分析[J]. 计算机学报, 2001, 24(12): 1234 - 1241.]

[9] Tao Xinmin, Xu Jing, Yang Libiao, et al. Particle-swarm algorithm coordinating the exploration and the exploitation[J]. Control Theory & Applications, 2010, 27(5): 636 - 640. [陶新民, 徐晶, 杨立标, 等. 一种协调勘探和开采能力的粒子群算法[J]. 控制理论与应用, 2010, 27(5): 636 - 640.]

[10] Gu J W, Gu M Z, Cao C W, et al. A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem[J]. Computers & Operations Research, 2010, 37(5): 927 - 937.

[11] Goldberg D E, Richardson J. Genetic algorithms with sharing for multimodal function optimization[C]//Proceedings of the 2nd International Conference on Genetic Algorithms. Hillsdale: L Lawrence Erlbaum Associate, 1987: 41 - 49.

[12] Cheng L J, Ding Y S, Hao K R. An ensemble kernel classifier with immune clonal selection algorithm for automatic discriminant of primary open - angle glaucoma[J]. Neurocomputing, 2012, 83(15): 1 - 11.

[13] Yang Haidong, E Jiaqiang. An adaptive chaos immune optimization algorithm with mutative scale and its application[J]. Control Theory & Applications, 2009, 26(10): 1069 - 1074. [杨海东, 鄂加强. 自适应变尺度混沌免疫优化算法及其应用[J]. 控制理论与应用, 2009, 26(10): 1069 - 1074.]

[14] Dong W S, Shi G M, Zhang L. Immune memory clonal selection algorithms for designing stack filters[J]. Neurocomputing, 2007, 70(4): 777 - 783.

[15] Guo Z L, Wang S A, Zhuang J. A novel immune evolutionary algorithm incorporating chaos optimization[J]. Pattern Recognition Letters, 2006, 27(1): 1 - 8.