

一般克隆选择算法的收敛性证明^{*}

方贤进¹, 李龙澍²

(1. 安徽理工大学 计算机科学与工程学院, 安徽 淮南 232001; 2. 安徽大学 计算机科学与技术学院, 合肥 230039)

摘要: 克隆选择算法已经广泛应用于计算智能领域, 而针对克隆选择算法理论方面的分析和研究工作却很少。为了丰富克隆选择算法的理论基础, 采用了与研究遗传算法相似的方法, 研究了克隆选择算法的收敛性, 推导出克隆选择算法在求解优化问题时, 收敛到全局最优解的充分条件。因此, 对基于克隆选择算法的各种应用如 BCA 和 CLONALG 算法, 只要检查这些充分条件是否满足就可以证明算法的收敛性。

关键词: 克隆选择算法; 收敛性; 遗传算法; 超变异算子

中图分类号: TP391 **文献标志码:** A **文章编号:** 1001-3695(2010)05-1683-03

doi:10.3969/j.issn.1001-3695.2010.05.020

Convergence proof for generic clonal selection algorithm

FANG Xian-jin¹, LI Long-shu²

(1. School of Computer Science & Engineering, Anhui University of Science & Technology, Huainan Anhui 232001, China; 2. School of Computer Science & Technology, Anhui University, Hefei 230039, China)

Abstract: Clonal selection algorithms have been applied widely in intelligent computation field, but theoretical analysis and research works regarding clonal selection algorithm were lacking relatively. In order to enrich the theoretical underpinning of clonal selection algorithms, this paper investigated the convergence of generic clonal selection algorithms in a similar manner as performed previously for genetic algorithms. The sufficient conditions that generic clonal selection algorithms converge to global optimum for solving optimization problem were derived. Consequently, all kinds of applications based on clonal selection algorithm such as BCA and CLONALG algorithms, their convergence can be proved by examining whether the sufficient conditions are satisfied or not.

Key words: clonal selection algorithm(CSA); convergence; genetic algorithm; hypermutation operator

0 引言

基于克隆选择理论^[1] (clonal selection theory) 的克隆选择算法^[2] (CSA) 是人工免疫系统 (artificial immune system, AIS) 中一个重要的算法, 已经在 AIS 中得到了广泛的应用并取得了一定的成功 (如模式识别和函数优化等领域)。它与进化计算比较所依赖的生物学机理不同, 但克隆选择算法已经表现出很多有益的特性, 如较好地保持了种群的多样性, 从而能够有效地克服诸如早熟等进化计算本身难以解决的问题。

到目前为止, 对克隆选择算法理论方面的分析与研究工作仍然很少, 只有文献[3] 针对一个特定的多目标优化的免疫系统算法 (multi-objective immune system algorithm, MISA), 利用 Markov 链给出了其收敛性的完整证明。文献[4] 对于克隆选择算法的一种简化形式——B 细胞算法 (B cell algorithm, BCA), 描述了 BCA 算法的数学模型——Markov 链模型, 并提出了在 BCA 算法中使用一种变化的超变异算子——连续区域超变异算子 (contiguous region hypermutation operator, CRHO), 并用一种新的方法建立 BCA 算法的状态转换矩阵 (transition matrix), 证明了 BCA 算法是绝对收敛的, 而且这种证明方法没

有像文献[3] 那样为当前最优解 (optimum) 保持一个独立的杰出者记忆集 (elitist memory unit)。

Rudolph 在文献[5] 中针对伪布尔函数最优化问题^[6], 使用 Markov 链理论对进化算法 (EAs) 收敛性问题以及 EAs 第一次遇到最优解的时间界限问题进行了详细的研究。本文也采用 Rudolph 在文献[5] 中使用的类似的方法, 在考虑到一般 CSA 所使用的所有可能的算子的情况下, 研究证明一般 CSA 收敛性的充分条件, 而不是为每个算法 (如 CBA、CLONALG 算法) 都利用 Markov 链设计一个证明。因此, 对基于 CSA 的各种 AIS 应用开发时, 只要检查这些充分条件是否满足即可判断算法的收敛性。

1 进化算法的收敛性度量

对一个给定的优化问题, 一个 EA 被称为收敛到全局最优如果它能确保算法在有限的步骤内找到最优解, 并且这样的解此后一直保持在群体中。

由于一个 EA 的状态转移具有随机性, 确定性收敛的概念不能用于判断这种算法的时间极值行为。两个通常使用的对进化算法的随机收敛性 (stochastic convergence) 的测量是完全收敛 (complete convergence) 和均值收敛 (convergence in

收稿日期: 2009-10-11; 修回日期: 2009-11-15 基金项目: 安徽省高等学校省级自然科学基金重点研究项目 (KJ2010A093)

作者简介: 方贤进 (1970-), 男, 安徽舒城人, 副教授, 博士, 主要研究方向为智能计算与信息安全 (xfjfang@aust.edu.cn); 李龙澍 (1957-), 男, 教授, 博导, 主要研究方向为智能软件。

mean)^[7]。文献[5]中还提到了概率收敛(convergence in probability)和以概率 1 收敛(convergence with probability 1)的概念。这些概念之间的关系是:完全收敛意味着以概率 1 收敛,以概率 1 收敛和均值收敛意味着概率收敛。

定义 1 设 X 是一个随机变量,并且 $(X_i; t > 0)$ 是一个随机变量序列。那么序列 X_i 被称为是完全收敛到 X ,如果对任何 $\varepsilon > 0$,有:

$$\lim_{t \rightarrow \infty} \sum_{i=0}^t P(|X_i - X|) > \varepsilon < \infty \quad (1)$$

定义 2 设 X 是一个随机变量,并且 $(X_i; t > 0)$ 是一个随机变量序列。那么序列 X_i 被称为是均值收敛到 X ,如果有:

$$\lim_{t \rightarrow \infty} E[|X_i - X|] = 0 \quad (2)$$

根据文献[5],有下面的关于 EAs 收敛性的定义。

定义 3 对最优化问题,设 $X_i; t \geq 0$ 是由进化算法生成的群体序列,一个 EA 被称为完全收敛到全局最优值 f^* ,如果非负随机序列 $D_i = f^* - F_i$ 完全收敛到零。其中 F_i 是指在时间步 t 时群体中最优个体的适应度值,即 $F_i = \max\{f(X_{i,1}), \dots, f(X_{i,n})\}$, f 是目标函数 $f: X \rightarrow \mathbb{R}$, f^* 定义为 $f^* = \max\{f(x); x \in X\}$ 。

定义 4 对最优化问题,设 $X_i; t \geq 0$ 是由进化算法生成的群体序列,一个 EA 被称为均值收敛到全局最优值 f^* ,如果非负随机序列 $D_i = f^* - F_i$ 均值收敛到零。其中 F_i 是指在时间步 t 时群体中最优个体的适应度值,即 $F_i = \max\{f(X_{i,1}), \dots, f(X_{i,n})\}$, f 是目标函数 $f: X \rightarrow \mathbb{R}$, f^* 定义为 $f^* = \max\{f(x); x \in X\}$ 。

2 一般克隆选择算法

在不同的应用中 CSA 会有所差别,如在文献[2]中,当将 CLONALG 用于模式识别时,要保持独立的记忆单元和抗原种群,而将 CLONALG 用于多模函数优化时则不需要这两个群体;再如,在 BCA^[8]中使用的一种特殊超变异算子——连续区域超变异算子(CRHO)。因此为了对一般 CSA 收敛性进行研究,本文首先描述了一般克隆选择算法的每一代行为。该描述具有一般性,囊括了 CSA 中的所有算子及其变化的算子。其克隆选择算法的行为如下:

```

Clonal Selection Algorithm( $T_\beta$ )// $T_\beta$  是算法参数
t: = 0;
P(1): = Initialize Pop();
while(- Termination Condition( )) do
    Evaluate(P(1));
    P(clo): = Cloning(P(1));
    P(hyp): = Hypermutation(P(clo));
    Evaluate(P(hyp));
    (Pa(t), Pa(hyp)): = Aging(P(1), P(hyp), Tβ);
    P(t+1): = Selection(Pa(t), Pa(hyp));
    t: = t + 1;
end while
    
```

以下是对克隆选择算法中一些最常见的算子及其变化的描述。

1) 克隆算子(cloning operator)

克隆算子生成一个新的群体 $P^{(clo)}$,该群体是当前群体(其大小为 n)中个体的拷贝。通常用的克隆算子是静态克隆算子(static cloning operator)^[9],只是每个 B 细胞被克隆 dup 次,因而产生长度为 $Nc = n \times dup$ 的中间群体 $P^{(clo)}$;第二种是按比例

克隆算子(proportional cloning operator),如用于模式识别的 CLONALG 算法^[2]中,B 细胞克隆的数量与其同抗原的亲合力成正比;第三种称为概率克隆(probabilistic cloning)^[10],B 细胞从当前种群中选择是依赖其克隆选择率 P_c 的大小。

2) 超变异算子(hypermutation operator)

超变异算子是对克隆种群 $P^{(clo)}$ 中的每个个体执行超变异。对个体执行单一变异的变异次数 M 称为随机变异潜能。通过下面的几种超变异算子可以看出,超变异算子是不同于遗传算法中的概率变异的。最常见的超变异算子有以下几种^[11]:

a) 静态超变异(static hypermutation)。变异次数 M 独立于适应度函数 f ,因此每个 B 细胞受体在每个时间步(经过一次单一变异)要经历至多 $M_t(x) = c$ 次变异。其中 c 是一个参数。 $\alpha = M(x)/L$ 称为变异率(mutation rate)。

b) 按正比例超级变异(proportional hypermutation)。一步变异中变异次数与适应度函数成正比,对每个 B 细胞 x 至多进行 $M_p(x) = (E^* - f(x)) \times c \times L$ 次变异。其中 E^* 为当前实例中(如一个最小化问题)适应度函数的最小值, c 是参数, L 为个体编码长度。

c) 按反比超级变异(inversely proportional hypermutation)。变异的数量与适应度函数值成反比,特别地,在时间步 t 该算子至多执行 $M_i(f(x)) = ((1 - \frac{E^*}{f(x)}) \times (c \times L) + (c \times L))$ 次变异。其中 E^* 是问题的最优解。

d) 凸超变异(convex hypermutation)。B 细胞 x 的每个基因 x_i 根据超变异率 p_h 执行凸组合超变异: $x_i' = (1 - \beta)x_i + \beta x_k$ 。其中 β 是一个随机数, $\beta \in \{0, \dots, 1.0\}$, x_k 是从 x 中随机选择的其他等位基因(allele)。该算子类似于概率变异中的均匀变异。

e) 超大变异(hypermacromutation)。其也就是 B 细胞算法中的连续区域超体变异算子 CRHO。变异数量独立于适应度函数和参数 c ,在这种情况下随机选择串中的两个位置 i, j 满足 $(i + 1) \leq j \leq L$,算子至多在范围 $[i, j]$ 内执行 $M_m(x) = j - i + 1$ 次变异。

3) 衰老算子(aging operator)

Aging 算子消除种群中老的个体,避免早熟的产生。一般使用的是静态纯衰老算子(static pure aging operator),设定一个参数 T_β 表示 B 细胞允许在群体中存在的最大代数。当一个 B 细胞的年龄达到 $T_\beta + 1$ 时,该 B 细胞就从当前群体中被删除,而不管其适应度值是多少。在克隆扩展期间,一个克隆的 B 细胞的年龄继承了其父代的年龄,超变异之后,只有那些获得了更高适应度的克隆的 B 细胞其年龄才被设置为 $age = 0$ 。如果该算子结合杰出者选择机制(elitist mechanism),那么可以将每一代群体中亲和力最高的个体设置为 $age = 0$ 。

4) 选择算子(selection operator)

在 CSA 中,该算子通常将群体中的最差个体用新随机产生的个体替换,目的是引入多样性。当然也有关于该算子的一些变化,如无冗余选择算子,该算子消除同样个体的拷贝使之存活到下一代。

为了使克隆选择算法收敛性研究尽量具有通用性,考虑到了一般克隆选择算法中的每个算子及其在不同应用中各个算子的一些变化,本文采用前文代码中的方案,但对超变异算子

提出了一种约定:本文使用一种更为简单的超变异算子,即随机地对群体中的个体选择 $r \leq L$ 位 (L 是候选解的长度) 独立地进行变异,该过程称为一步单变异 (single mutation),并且在一般情况下,为了尽可能保持超变异算子的通用性,每次所选择的变异位 (mutation digit) 不必是完全不同的,但实际上在一步超大变异 (macromutation) 中,通常要避免对同一个 bit 位的变异 (对二进制编码)。例如,如果随机选择 $r=2$ (即对个体的两个 bit 位进行变异),那么如果随机选择了同一 bit 位两次,则变异后个体仍然保持不变。因此下面的讨论中,约定超变异算子对个体的一步变异只对不同的 bit 位进行。

3 一般克隆选择算法的收敛性

一般来说,在某些假设的条件下进化算法的收敛性已经得到证明,下面以进化算法收敛的充分条件为基础来研究一般 CSA 收敛的充分条件。

一个普通的进化算法的单一迭代过程可描述如下:

$$\forall i \in \{1, \dots, m\} : x'_i = \text{mut}(\text{reco}(\text{mat}(x_1, \dots, x_n)))$$

$$(y_1, \dots, y_n) = \text{sel}(x_{m(1)}, \dots, x_{m(q)}, x'_1, \dots, x'_m) \quad (3)$$

其中: $(x_1, \dots, x_n) \in X^n$ 表示当前父代种群,并且

$$\text{mat}: X^n \rightarrow X^p, \text{reco}: X^p \rightarrow X, \text{mut}: X \rightarrow X, \text{sel}: X^k \rightarrow X^n \quad (4)$$

分别表示杂交过程、重组算子、变异算子和选择算子。对任何优化问题,文献[5]给出了在下列条件下,一个进化算法完全收敛及均值收敛到全局最优。

条件 1 群体中的每一个体能以概率 $p(p > 0)$ 使用一步变异改变到其他任一个体。

条件 2 群体中的最优个体以概率 $p = 1$ 存活在每一代中。

以上两个条件可以用数学语言描述如下:

$$\forall x, y \in X, P\{y = \text{mut}(x)\} \geq \delta_m > 0 \quad (5)$$

$$P\{v_n^*(\text{sel}(x_1, \dots, x_k)) = v_k^*(x_1, \dots, x_k)\} = 1 \quad (6)$$

此处 v_i^* 返回由 i 个个体组成的群体中的最优个体,其定义如式(7)。

$$v_n^*(x_1, \dots, x_n) = \max\{f(x_i) : i = 1, \dots, n\} \quad (7)$$

如果只有条件 1 成立,则能证明 EA 能在有限时间步内以概率 1 接触到全局最优值,而不管它的初始状态是什么。但是这并不意味着它是收敛的,因为不能保证在找到最优值之后该最优值能永远地保留在群体中。如果条件 2 也成立,就能证明 EA 能收敛到全局最优。

采用与进化算法相似的方法,一般 CSA 的单一迭代过程也可以描述如下:

$$\forall i \in \{1, \dots, m\} : x'_i = \text{hyp}(\text{clone}(x_1, \dots, x_n))$$

$$(x''_1, \dots, x''_k) = \text{aging}(x_1, \dots, x_n, x'_1, \dots, x'_m)$$

$$(y_1, \dots, y_n) = \text{sel}(x''_1, \dots, x''_k) \quad (8)$$

在实际使用 CSA 时,可以使用也可以不使用 aging 算子,就像 EAs 中的 crossover 算子一样。另外通过前面的定义可知 aging 算子实际上参与了进化过程中的选择机制,因为它根据个体的年龄决定了该个体是否存活到下一代中,这意味着将以上的条件 1 应用到一般克隆选择算法时只需要考虑超变异算子 (因为 cloning 算子不改变个体),而 aging 算子则需要在条件 2 中加以考虑。因此针对一般克隆选择算法而言,条件 1 和条件 2 可以用式(5')和(6')加以描述。

$$\forall x, y \in X, P\{y = \text{hyp}(x)\} \geq \delta_m > 0 \quad (5')$$

$$P\{v_n^*(\text{aging}(x_1, \dots, x_n, x'_1, \dots, x'_m)) = v_k^*(x''_1, \dots, x''_k)\} = 1$$

$$P\{v_n^*(\text{sel}(x''_1, \dots, x''_k)) = v_k^*(y_1, \dots, y_n)\} = 1 \quad (6')$$

定理 对优化问题,不管它的初始值是什么,一般克隆选择算法完全收敛及均值收敛到全局最优,只要使用了基于杰出者选择的 aging 算子。

证明 为了证明该定理成立,只要证明一般克隆选择算法满足式(5')和(6')即可。

由于 cloning 和 aging 算子都不改变存在的个体,也不产生不同的个体,只有超变异算子和选择算子可能第一次产生群体中的最优解。

设长度为 L 的 bit 串为搜索空间的一个点,并通过向量 $\{0, 1\}^L$ 来表示。如果群体中的一个个体与最优解相比时,有 c 个 bit 不匹配,也就是说有 $L - c$ 个 bit 匹配,那么该个体利用一步超变异算子到达全局最优的概率为 (根据约定一步变异中的 c 个 bit 变异都是在不同的 bit 位上进行)

$$P_c^{(L)} = \frac{c!}{L^c} \times \frac{1}{L} \quad (9)$$

其中:式(9)的第一项表示从 L^c 个可能的选择中选择 c 个 bit 位的排列,这个概率还要乘以实际随机选择要进行变异的 c 个 bit 位的概率 $1/L$ 。

作为式(9)的扩展,如果字符集的基数为 K ,搜索空间的每个点用一个向量 $\{0, 1, 2, \dots, K-1\}^L$ 表示,则式(9)变为

$$P_c^{(L)} = \frac{c!}{L^c} \times \frac{1}{(K-1)^c} \times \frac{1}{L} \quad (9')$$

此处, $\frac{1}{K-1}$ 为一步变异中的一位变异到最优解相应基因位值的概率 (均匀变异)。由于 $P_c^{(L)}$ 总是为正,满足式(5'),也就说明一般克隆选择算法满足条件 1。

要证明条件 2 也成立,就要证明一旦最优解被找到,对种群执行的任何一个算子都不会再丢失全局最优解。

Cloning 算子只产生个体的拷贝,不会改变任何个体的值,因此它不会丢失最优解。

Hypermutation 算子只对 cloning 算子所产生的中间种群 $P^{(\text{clo})}$ 执行操作,也不改变由其他任何算子 (包括它自己) 所产生的个体。

Aging 算子尽管删除了老的个体,但由于每一代种群的最优候选解的 age 被设置成 0,它不可能丢失最优解。

至于 selection 算子,因为其删除了最小亲和力的个体,取而代之的是随机新生成的个体,所以最优解不会失去,即便使用无冗余 selection 算子。如果群体中有多个最优解,那么也至少有一个最优解存活下来。

由此可见,一般克隆选择算法中的所有算子满足条件 2。

定理证毕。

4 结束语

根据本文的方法,绝大多数克隆选择算法的收敛性能够被证明,只要其中的算子 (通常是超变异或选择算子) 能满足条件 1 并且也能满足条件 2 (通常只考虑验证 aging 算子的杰出者机制,而不去考虑选择算子)。可以用本文的方法来证明以前用 Markov 链证明 B 细胞算法的收敛性^[4] (下转第 1689 页)

间间隔 σ 出现情况(c)或(d),优先保障子问题1的工作时间;同时可与客户商量适当放宽子问题2的承诺完成时间;同理,插入子问题3的调度结果。综合三个子问题,得到案例的最终调度如图3所示。

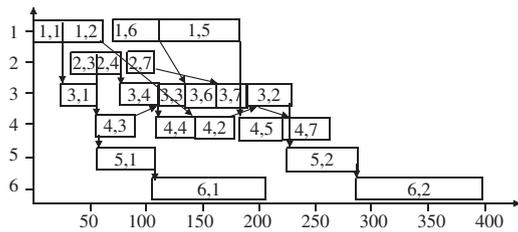


图3 案例最终调度

从甘特图得出目标函数值为 $\sum T_j = 90$, $\sum \omega_j C_j = 2\ 143$ 。相对于当前企业中常用的 FCFS 规则计算出的目标函数值 $\sum T_j = 107$, $\sum \omega_j C_j = 2\ 520$, 本文提出的调度算法优化了目标函数值。以上程序均用 C++ 编写,在奔腾双核、主频 1.73 GHz 的计算机上进行实验仿真, CPU 运算时间小于 0.05 ms。

5 结束语

汽车维修车间的优化调度是 Job-shop 调度理论在企业的实际应用。本文针对当前企业中主要采取 FCFS 规则进行调度,导致较长的客户等待时间和较低的设备利用率等问题,在优化调度理论的基础上,对这一调度问题和算法进行了研究,分析了这一实际问题的特征,建立了相应的优化调度模型,设计了分解法与约束引导的启发式算法相结合的调度算法。最后以实例分析验证了算法的可行性,通过对比仿真结果展示了本文所用算法相对于 FCFS 规则在优化目标函数值上的优越性。

参考文献:

[1] VILCOT G, BILLAUT J C. A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem[J]. *European Journal of Operational Research*, 2008, 190(2): 398-411.

(上接第 1685 页)问题。由于 B 细胞算法没有使用 aging 算子,使用了超变异算子的一种变化——连续区域超体变异算子 (CRHO)。显然 CRHO 满足条件 1,因而可获得 BCA 收敛性的证明。

参考文献:

[1] BURNET F M. The clonal selection theory of acquired immunity[M]. Cambridge: Cambridge University Press, 1959.

[2] De CASTRO L N, Von ZUBEN F J. Learning and optimization using the clonal selection principle[J]. *IEEE Trans on Evolutionary Computation*, 2002, 6(3): 239-251.

[3] VILLALOBOS-ARIAS M, COELLO-COELLO C A, HERNANDEZ-LERMA O. Convergence analysis of a multi-objective artificial immune system algorithm[C]//Proc of Lecture Notes in Computer Science. Berlin: Springer, 2004: 226-235.

[4] CLARKE E, HONE A, TIMMIS J. A Markov chain model of the B-cell algorithm[C]//Proc of the 4th International Conference on Artificial Immune Systems. Berlin: Springer, 2005: 318-330.

[5] RUDOLPH G. Finite Markov chain results in evolutionary computation: a tour d' horizon[J]. *Fundamenta Informaticae*, 1998, 35(1-

[2] GAO Jie, SUN Lin-yan, GENB M. A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems[J]. *Computers & Operations Research*, 2008, 35(9): 2892-2907.

[3] BALAS E, LENSTRA J K, VAZACOPOULOS A. The one-machine problem with delayed precedence constraints and its use in job shop scheduling[J]. *Management Science*, 1995, 41(1): 94-109.

[4] McCORMICK S T, PINEDO M L. Scheduling n independent job on uniform machines with both flow time and makespan objectives: a parametric analysis[J]. *Journal of Computing*, 1995(7): 63-77.

[5] ELKAMEL A, MOHINDRA A. A rolling horizon heuristic for reactive scheduling of batch process operation[J]. *Engineering Optimization*, 1999, 31(6): 763-792.

[6] LIU Min, HAO Jing-hua, WU Cheng. A prediction based iterative decomposition algorithm for scheduling large-scale job shops[J]. *Mathematical and Computer Modeling*, 2008, 47(3-4): 411-421.

[7] KELLY J D, ZYNGIER D. Hierarchical decomposition heuristic for scheduling: coordinated reasoning for decentralized and distributed decision-making problems[J]. *Computers and Chemical Engineering*, 2008, 32(11): 2684-2705.

[8] SZWARC W. Decomposition in single machine scheduling[J]. *Annals of Operations Research*, 1998, 83(1): 271-287.

[9] CHENG C C, SMITH S F. Applying constraint satisfaction techniques to job shop scheduling[J]. *Annals of Operations Research*, 1997, 70(1): 327-357.

[10] NUIJTEN W P M, AARTS E H L. A computational study of constraint satisfaction multiple capacitated job shop scheduling[J]. *European Journal of Operational Research*, 1996, 90(2): 269-284.

[11] GARRIDO A, ONAINDIA E, SAPENA O. Planning and scheduling in an e-learning environment: a constraint-programming-based approach[J]. *Engineering Applications of Artificial Intelligence*, 2008, 21(5): 733-743.

[12] SALIDO M A, GARRIDO A, BARTAK R. Introduction: special issue on constraint satisfaction techniques for planning and scheduling problems[J]. *Engineering Applications of Artificial Intelligence*, 2008, 21(5): 679-682.

4): 67-89.

[6] HAMMER P L, RUDEANU S. Boolean methods in operations research and related areas[M]. Berlin: Springer, 1968.

[7] BACK T, FOGEL D B, MICHALEWICZ Z. Handbook of evolutionary computation[M]. Amsterdam: IOP/Oxford University Press, 1997.

[8] KELSEY J T J. Immune inspired somatic contiguous hypermutation for function optimisation[C]//Proc of Genetic and Evolutionary Computation Conference. Berlin: Springer, 2003.

[9] NICOSIA V C G. An immunological approach to combinatorial optimization problems[C]//Proc of Advances in Artificial Intelligence, IBERAMIA(2002). Berlin: Springer, 2002: 361-370.

[10] CUTELLO V, NICOSIA G, PAVONE M. A hybrid immune algorithm with information gain for the graph coloring problem[C]//Proc of GECCO2003. Chicago, IL: Springer, 2003: 171-182.

[11] CUTELLO V, NARZISI G, NICOSIA G, et al. Clonal selection algorithms: a comparative case study using effective mutation potentials[C]//Proc of ICARIS 2005. [S. l.]: Springer, 2005: 13-28.

[12] KOEHLER H A G J. Stopping criteria for finite length genetic algorithms[J]. *Inform Journal on Computer*, 1996, 8(2): 183-191.