

1,2 , 2

<sup>1,2</sup>, WU Xiao-ming<sup>2</sup>

### 1 概述

人工免疫系统(AIS)是运用生物免疫系统抵抗外来抗原的生命规律来解决实际问题的模拟系统。对AIS的研究重点多数是与其演进周期的3个关键步骤(选择、亲和力成熟和免疫记忆)相关的算法及应用<sup>[1]</sup>。虽然目前存在很多基于AIS的软件,但基本上是解决某一特定问题的,软件的可重用性较差。文献[2]提出了一种通用的AIS功能模型ARTIS,该模型描述了AIS应该具备的特征,如:分布式计算,多样性,容错,动态学习和自适应等,对进一步设计可重用的人工免疫软件系统具有良好的指导作用;Suzuki等基于AIS网络模型设计了一种具有网络可伸缩结构的软件系统框架iNet<sup>[3]</sup>,把系统分解为可视化包、图管理包、免疫组件管理包和持久化和交换包等4个部分,主要使用设计模式<sup>[4]</sup>提高模块的可重用性;严悍等从AIS软件架构设计的角度出发,提出了一种的AIS通用基因模型(GeGeM),该模型主要是对免疫系统中的基因部分进行了分层的结构设计,即元基因、基因序列和基因数据集3层,且下层依赖上层的实现<sup>[5]</sup>。由此可见,随着AIS在各个领域的广泛应用,其软件系统的通用性也逐渐得到重视。为了适应在这种开放计算环境中的变化,就必须能够动态地调整AIS中免疫组件的自身行为和属性等,即动态调节这些软件实体的软件结构。而具有这种能力的软件体系结构之一就是元层次体系结构(MLA)<sup>[6]</sup>。

鉴于上述理由,本文设计了一种基于元层次体系结构的AIS抽象软件框架。该框架设计的基本思路如下:根据MLA把AIS划分为免疫元层和免疫组件层,分别对应于MLA的元层次和基本层次。免疫元层是AIS动态变化部分的抽象层,

包含描述免疫组件层对象的元对象,主要有:抗体元,记忆细胞元和节点通信元等,并且通过元对象协议来管理这些元对象;免疫组件层是AIS的应用逻辑实现层,相对免疫元层是非动态的,与之相对应的包含抗体,记忆细胞和节点通信等组件,并可以提供用户接口。

### 2 AIS基本模型

AIS基本模型如图1所示,基本算法流程如图1(a)所示。运用模块化的设计思想,AIS可以划分为抗体管理、记忆细胞管理、动态性管理和网络管理等几个主要模块。这种按功能的划分方式在各种免疫系统中比较常见,如图1(b)所示。

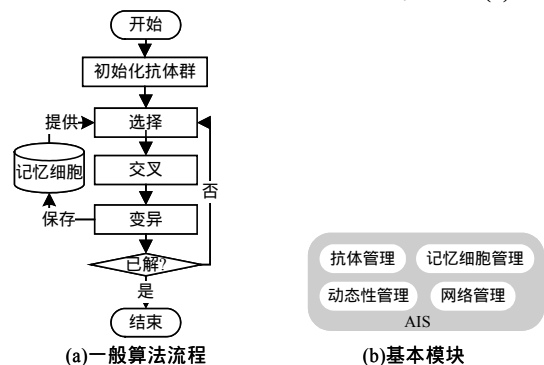


图1 AIS基本模型

基金项目:国家自然科学基金资助项目(30670538)

作者简介:杨铁军(1979-),男,博士研究生,主研方向:免疫计算,医学图像处理;吴效明,教授、博士、博士生导师

收稿日期:2007-01-20 E-mail:yattie@163.com

### 3 AIS 软件框架

#### 3.1 MLA 剖析

MLA也可以称为映像模式(reflection),为动态改变软件系统的结构和行为提供了一种机制<sup>[6]</sup>。MLA基本上划分为两层:元层次对基本层次的组件的属性、行为等进行描述并使软件含自述信息,元对象协议(Meta-Object Protocol, MOP)是元层次的外部接口,提供了对其自身进行修改的可控映像组件;基本层次是软件系统的应用逻辑实现部分,建立在元层次之上。通过MOP对元层次进行的修改,将会影响到对应的基本层次的属性和行为。元层次体系结构如图2所示。

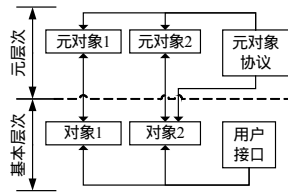


图2 元层次体系结构

#### 3.2 框架设计

由上分析知,该AIS被称为MLA-AIS(Meta-Level Architecture Based Artificial Immune System),可以划分为两层:免疫元层和免疫组件层。免疫元层包含描述免疫组件对象的元对象,主要有:抗体元,记忆细胞元和节点通信元等,并且通过元对象协议来管理这些元对象;免疫组件层与之相对应的有抗体、记忆细胞和节点通信等组件,并提供用户接口。该框架如图3所示。

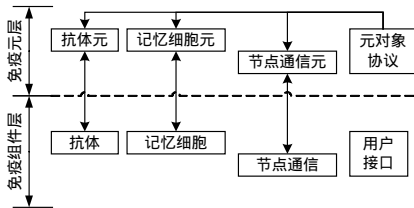


图3 MLA-AIS 框架

#### 3.3 免疫组件层

该层是AIS的功能实现层。使用统一建模语言(UML)进行建模的结果如图4所示。

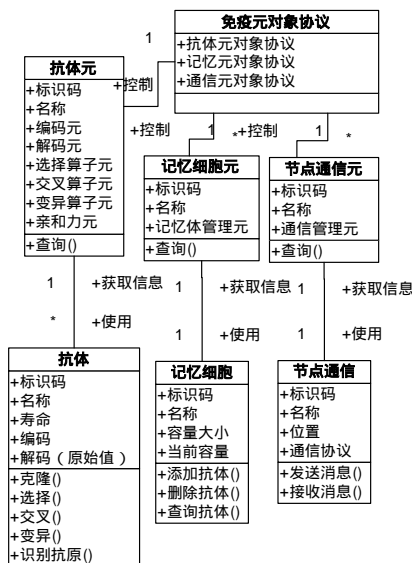


图4 MLA-AIS 的主要组件

抗体组件是系统的核心,它包含了抗体所有的属性和行为,如抗体属性主要有标识码、名称等,方法主要有初始化、克隆增值等;记忆细胞组件用来保存已经成熟的抗体,加快二次免疫应答;节点通信组件是一个通信管理单元,它与免疫网络中其他节点交换信息,如交换记忆细胞、抗体浓度等信息。该层是AIS的通用应用逻辑部分,是相对“静态”的。

#### 3.4 免疫元层

免疫元层是MLA-AIS的动态因素管理层,主要包括抗体元、记忆细胞元和节点通信元等。由AIS的基本模型分析得知,在抗体的编码、选择、变异等过程中,可以根据抗原的特征进行动态调整。如抗体的编码的长度、方式、基因数量等,这些动态属性和行为构成了抗体元中的编码元;变异算子元负责管理与变异算子相关的动态部分,如变异率调整,变异位选择等。因为整个免疫过程都是围绕抗体进行的,所以抗体元较为复杂,它还包括解码元、选择算子元等,如图4所示。

另外,记忆细胞元定义了记忆细胞如何保存成熟的抗体和淘汰不合格的抗体、记忆体的容量等;节点通信元则为节点在网络中通信提供了位置信息和多种传输协议等。

免疫元对象协议(Immune Meta-Object Protocol, IMOP)是管理免疫元的外部接口,每一个免疫元对象都创建一个MOP对象,如图4所示。

#### 3.5 算法工具箱与扩展

算法工具箱提供了各种经典的免疫算法,但是这些算法按照框架的结构进行了分解,如克隆选择算子、负选择算子等加入到选择算子工具箱中,编码工具箱中有二进制编码、实数编码和字符编码等。另外,系统还提供算法接口供外部扩展。

### 4 基于基因编码的自适应免疫算法

该算法运用生物特异性免疫的基本原理,对变异算子进行改进,使得变异遵循“高适应者低变异,低适应者高变异”的规律,以提高算法的搜索效率。同时,该算法的动态特性也要求软件的实现结构能够动态调整。鉴于实验部分将使用MLA-AIS实现该算法,下面重点介绍它的动态特性。

#### 4.1 抗体编码

对抗体的基因编码就是对解的编码过程。根据Perelson形态空间理论,抗体和抗原都可以看作是 $L$ 维形态空间 $S$ 中的一个点,并且可以用向量 $m$ 来表示:

$$m = \langle m_1, m_2, \dots, m_L \rangle, m \in S^L \quad (1)$$

其中,  $m_i$  表示形态空间中的第  $i$  个属性,可以是任何类型。

一般按照以下原则对抗体进行编码:

- (1)编码过程简单,编码适合于计算机操作。
- (2)编码符合基因的特征,方便实施各种算子。

抗体编码定义如下:

抗体编码  $enc(m)$  一般可以表示为包含编码方式  $code$  和编码串  $encstr$  的二元组,即  $enc(m) = \langle code, encstr \rangle$ 。其中,  $encstr = \langle str_1, str_2, \dots, str_L \rangle$ ,  $str_i$  与  $m_i$  对应,即  $str_i \leftrightarrow m_i$ 。

#### 4.2 变异率分布函数

变异率分布与基因的编码密切相关,可以根据基因编码来确定基因的变异率分布。变异率分布一般满足以下法则:

Rule 1 基因的的稳定部位变异率一般较低,非稳定部位变异率一般较高。

Rule 2 基因段的变异率与亲和力成反比。

根据 Rule 1 可以定义编码基因段  $str_i$  的变异率  $mr_i$  :

$$mr_i = \begin{cases} k_1 p, i \in set_1, k_1 \in (0,1] \\ k_2 p, i \in set_2, k_2 \gg k_1 \end{cases} \quad (2)$$

其中,  $p$  是抗体的经验变异率;  $k_1$  表示基因稳定部位的变异常数;  $k_2$  是非稳定部位的变异常数;  $set_1$  是稳定基因段序号集合;  $set_2$  是非稳定基因段序号集合。  $k_2$  取值远大于  $k_1$ 。

根据 Rule 2 可以定义每个基因段变异率与亲和力的关系即  $mr_i \propto k_3 \cdot \frac{1}{a}$ , 其中,  $k_3$  是比例系数;  $a$  是亲和力大小。所以, 式(2)可以修正为

$$mr_i = \begin{cases} \frac{k_1 k_3 p}{a}, i \in set_1, k_1 \in (0,1] \\ \frac{k_2 k_3 p}{a}, i \in set_2, k_2 \gg k_1 \end{cases} \quad (3)$$

式(3)即为编码基因的变异率分布函数, 因为  $mr_i$  是变异概率, 所以如果  $mr_i > 1$ , 则取  $mr_i \approx 1$ 。

### 4.3 自适应变异算子

与传统算法使用经验变异率进行随机变异不同, 改进的变异算子根据变异率分布函数进行变异率动态调整, 提高搜索效率。

## 5 算法实现

下面使用 MLA-AIS 来实现该免疫算法:

(1)主程序使用免疫组件层组件定义该算法的基本流程, 也就是传统算法流程, 如图 1(a)所示。

(2)使用 AbMOP 操作抗体元, 使其编码元和变异算子元参数设置符合本算法的编码和变异操作, 其他没有设置的免疫元将使用缺省的配置。

(3)抗体在初始化过程中使用编码元进行基因编码, 即  $enc(m) = \langle code, encstr \rangle$ , 此时编码将按照本算法的方式进行。

(4)在实施变异算子过程中, 变异算子元获取免疫组件层的抗体编码, 并根据式(3)计算变异率分布函数, 以完成自适应变异。

(5)系统工具箱如果没有提供本算法的编码方法和变异算子, 需要事先通过系统扩展实现。

## 6 实验与分析

### 6.1 实验验证

分别用经典的CLONALG<sup>[7]</sup>免疫算法和本文算法求多模态函数多峰值的实验来验证MLA-AIS的可用性。该函数在文献[7]中用来验证CLONALG算法在优化问题中的可用性, 其定义如下:

$$f(x) = 2^{-2((x-0.1)/0.9)^2} \cdot \sin^6(5\pi x), x \in [0,1] \quad (4)$$

用本文算法求解式(4)峰值的主要步骤描述如下: 多个峰值代表抗原集, 峰值对应的自变量  $x$  则代表抗体。抗体取值精度到万分位(即小数点后 4 位), 每一位采用二进制的 BCD 码, 即  $encstr = \langle Bcd_1, Bcd_2, Bcd_3, Bcd_4 \rangle$ ,  $Bcd_{i_4}$  分别表示抗体取值的十、百、千和万分位, 并且  $set_1 = \{1,2\}$ ,  $set_2 = \{3,4\}$ 。根据式(3), 变异率分布函数计算如下:

$$mr_i = \begin{cases} \frac{k_1 k_3 p}{f(x)}, i = 1,2 \\ \frac{k_2 k_3 p}{f(x)}, i = 3,4 \end{cases} \quad (5)$$

其中,  $a = f(x)$ ;  $p=0.008$ ;  $k_1=1$ ;  $k_2=100$ ;  $k_3=0.5$ 。

CLONALG算法的变异算子采用缺省的变异算子(随机变异), 变异率取 0.008, 其他过程基本相同。经过多次实验, 图 5 是分别运用这两种算法求式(4)峰值的结果, 其中,  $g$  表示进化代数;  $Ag_1$ 是CLONALG算法求得的抗体集;  $Ag_2$ 是本文算法求得的抗体集。由图 5 可知, 本文算法较CLONALG算法具有更高的搜索效率; 同时也证明了使用MLA-AIS能快速构建各种AIS。总之, 基于该框架可以方便地进行各种免疫算法验证、比较等工作, 还可以扩展该框架开发各种应用系统。

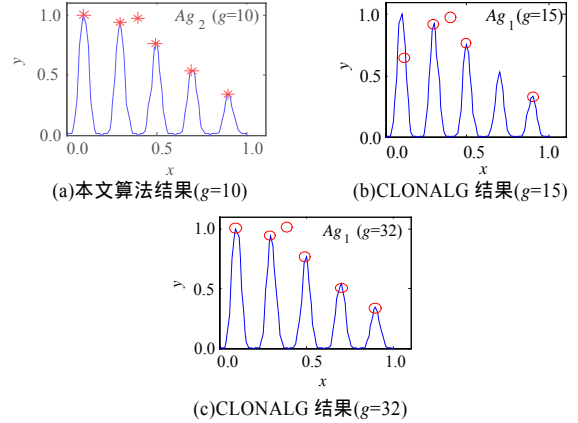


图 5 实验结果比较

### 6.2 相关工作比较

MLA-AIS 相对于其他结构的 AIS 具有以下 3 个优点:

(1)动态元结构和静态结构相分离, 并通过 IMOP 动态修改动态元。不同的用户只需要关注其感兴趣的动态元, 不需要完全重建或修改系统, 可重用性更高。

(2)对比层次结构的AIS<sup>[5,8]</sup>, 不是下层为上层提供服务, 而是基本层次和元层次交互合作, 使系统在运行期也能动态改变其结构, 适应性更好。

(3)对比使用各种设计模式的AIS<sup>[4]</sup>, 在更高的体系结构层次进行动态调整, 具有更大的灵活性, 也可以加入合理的设计模式。

## 7 结束语

综上所述, 元层次体系结构较好地解决了 AIS 中动态因素的管理, 是一种可行的 AIS 软件框架。MLA-AIS 通过免疫元层分离出可变的、动态的免疫属性和免疫行为, 而把相对“静止”的通用特征、方法和流程等“固化”到免疫组件层。两层相互协作, “静态”层使用“动态”层提供的元信息进行动态实现, 并向元层次提供该实现信息; “动态”层通过 IMOP 对自身进行修改, 并动态影响基本层次。这样, 不需要修改已有的免疫组件层对象就能实现各种特定的免疫操作, 提高了 AIS 的可重用性、可扩展性和适应性。

### 参考文献

- [1] 焦李成, 杜海峰. 人工免疫系统进展与展望[J]. 电子学报, 2003, 31(10): 1540-1548.
- [2] Hofmeyr S A, Forrest S. Architecture for an Artificial Immune System[J]. Evolutionary Computation, 2000, 8(4): 443-473.
- [3] Suzuki J, Yamamoto Y. iNet: an Extensible Framework for Simulating Immune Network[C]//Proc. of IEEE International Conference on Systems, Man, and Cybernetics. [S. l.]: IEEE Press, 2000.

(下转第 42 页)