

# 旅行商问题的人工免疫算法<sup>\*</sup>)

李茂军<sup>1</sup> 舒 宜<sup>2</sup> 童调生<sup>3</sup>

(长沙电力学院电力工程系 长沙 410077)<sup>1</sup> (株州电业局 株州 412000)<sup>2</sup>

(湖南大学电气与信息工程学院 长沙410082)<sup>3</sup>

## An Artificial Immune Algorithm for Travelling Salesman Problem

LI Mao-Jun<sup>1</sup> SHU Yi<sup>2</sup> TONG Tiao-Sheng<sup>3</sup>

(Department of Electrical Engineering, Changsha University of Electrical Power, Changsha, 410077)<sup>1</sup>

(Zhuzhou Electrical Power Bureau, Zhuzhou 412000)<sup>2</sup>

(College of Electrical & Information Engineering, Hunan University, Changsha, 410082)<sup>3</sup>

**Abstract** This paper presents an Artificial Immune Algorithm (AIA) simulating the biological immune systems, and offers its basic principle and approach. Comparing AIA with Genetic Algorithm (GAs) simulating the biological evolution process, the paper points out that the method producing new antibodies in AIA is more versatile than the one producing new individuals in GAs. AIA reflects mechanism of natural selection better than GAs does, as AIA selects effective antibodies from all antibodies by the appetency between an antibody and an antigen and by the repulsion between an antibody and another, while GAs selects new individuals of next colony by the proportion of individual fitness. For Travel Salesman Problem (TSP), this paper brings forward how to describe antibodies artificially, how to produce original antibodies, how to compute the appetency between an antibody and an antigen and the repulsion between an antibody and another, and works out several artificial immune operators producing new antibodies. Simulating examples show that AIA is a very effective method for TSP.

**Keywords** Artificial immune algorithm, Genetic algorithm, Artificial immune operator, Travel salesman problem

## 1 引言

旅行商问题(TSP)是一个典型的有序组合优化问题,可以看成是许多领域内复杂工程优化问题的抽象形式。研究TSP问题的求解方法对解决复杂工程优化问题具有重要的参考价值。对于TSP问题,目前还没有完全有效的求解方法,但是,多年来人们一直在不停地探索。近年来,模拟自然界生物进化过程的求解TSP问题的方法不断见诸文献,但以基于遗传算法(GAs)的居多<sup>[1-5]</sup>。人工免疫系统<sup>[6]</sup>和GAs都属于模仿自然界生物行为的仿生智能信息处理方法。GAs的理论和应用研究已经渗透到了许多领域和学科<sup>[7,8]</sup>,而人工免疫系统却还没有引起足够的重视,其理论和应用研究成果少见有文献报道。本文构造一种TSP问题的人工免疫算法(AIA),并通过仿真实验验证了这种算法的有效性。

## 2 人工免疫算法的基本原理

### 2.1 生物免疫系统和人工免疫系统

生物免疫系统是自然界生物所必备的防御系统,它具有免疫功能的组织、器官和细胞等组成,可以使生物机体免受病原体的侵害。生物免疫系统的免疫功能是通过淋巴细胞的自我调节作用来实现的。在免疫系统中,主要有B和T两类淋巴细胞。B细胞的功能主要是产生抗体,抗体由氨基酸排列组成,氨基酸的不同排列方式形成不同的抗体;T细胞则主要实现免疫调节功能<sup>[9]</sup>。

在构造人工免疫系统时,首先要构造人工抗原和抗体,为

使人工免疫系统具有与生物免疫系统类似的自我调节机制,可以用亲和力来描述抗体和抗原之间匹配程度,用排斥力来描述两个抗体之间的相似程度。在用AIA求解优化问题时,满足约束条件的最优解即是抗原;候选解即是抗体。在人工免疫系统中,一个抗体可以用一个字符串表示。生物抗体由氨基酸的不同排列组成,因此,人工抗体(字符串)中的字符相当于生物抗体中的氨基酸。抗体和抗原之间的亲和力反映了候选解和最优解的接近程度,也即反映候选解对约束条件和目标函数的满足程度;抗体和抗体之间的排斥力反映了不同候选解之间的异同,也即反映了抗体的多样性。保持抗体的多样性可以防止算法陷入局部最优解。

### 2.2 人工免疫算法的基本步骤

(1)输入问题的目标函数和约束条件,作为AIA的抗原。

(2)确定抗体的编码方式。与GAs的染色体一样,AIA的抗体也用字符串表示。一个字符串(抗体)只能代表一个候选解;但一个候选解允许有一个以上的字符串(抗体)相对应。在确定抗体的编码方式时,应尽量使字符串(抗体)和候选解之间形成一一对应的关系,以缩小抗体空间(串空间),提高搜索效率。

(3)产生初始抗体。通常可以在解空间中随机产生 $N$ 个候选解作为初始抗体。 $N$ 为抗体群中抗体的数目。

(4)计算亲和力。构造抗体的亲和力函数 $f(B)$ 。 $f(B)$ 反映抗体B和抗原G之间的匹配程度。 $f(B)$ 越大说明抗体B和抗原G之间的匹配得越好。

(5)计算排斥力。构造抗体与抗体之间的排斥力函数 $f$

<sup>\*</sup>) 国家教育部博士点基金资助项目(98053205)。

$(B_1, B_2), f(B_1, B_2)$  反映抗体  $B_1$  与抗体  $B_2$  之间的相似程度。 $f(B_1, B_2)$  越大说明抗体  $B_1$  与抗体  $B_2$  之间的差距越大。计算抗体群中所有抗体与最佳抗体(当前抗体群中与抗原匹配得最好的抗体)之间的排斥力。

(6) 产生新的抗体。构造适当的人工免疫算子。抗体通过人工免疫算子的作用产生新的抗体。

(7) 计算新抗体的亲和力和排斥力。若新抗体中有与抗原相匹配的抗体,或已满足预定的停机条件则停机。否则转下一步。停机条件的设定方法与 GAs 的相同。

(8) 抗体选择。按照“优胜劣汰”的自然选择机制,在原有的  $N$  个有效抗体和新产生的若干个抗体中选择出  $N$  个与抗原匹配得较好的抗体构成新的抗体群,转(6)。

### 2.3 人工免疫算法与遗传算法的比较

从上述基本步骤可以看出, AIA 和 GAs 有相似之处, 这使 AIA 的研究, 尤其是其理论研究在很大程度上可以借鉴 GAs 的某些研究成果。但 AIA 和 GAs 相比又有其独特之处, 主要表现在以下三方面:

(1) AIA 中新抗体的产生和 GAs 中新个体的繁殖不同。GAs 模拟自然界生物的双亲繁殖方式, 主要由交叉算子繁殖新的个体。虽然在遗传操作过程中也可以采取单亲繁殖方式, 构造出单亲遗传算法(PGA)<sup>[9]</sup>, 但是 PGA 的遗传算子仍然隐含了传统 GAs 的遗传算子的功能<sup>[10]</sup>, 因此 PGA 与传统 GAs 相比并无本质区别。GAs 中后代个体继承了前代个体的大部分遗传特征。AIA 模拟自然界生物免疫系统的功能, 新抗体的产生方式不受生物的双亲繁殖方式的影响, 新抗体中的字符不是由原抗体“遗传而来”, 新抗体的产生比 GAs 中新个体的产生要灵活得多。在构造人工免疫算子时, 一个新抗体中的字符可以来源于一个或几个原有抗体, 也可以与任何原有抗体无关, 即 AIA 中的新抗体并不一定继承原抗体中的字符。

(2) AIA 中亲和力的计算和 GAs 中适应度的计算不同。GAs 中个体适应度体现了个体对环境的适应能力, 与个体所代表的解的目标函数值直接相关。GAs 的选择策略依据个体适应度的大小, 按照“优胜劣汰”的原则, 选择出较优的个体用于繁殖后代。当群体中个体的适应度相差不大时, “优胜劣汰”的效果不明显, 搜索效率会降低。AIA 中抗体和抗原之间的亲和力体现了抗体与抗原之间的匹配程度, 相当于 GAs 中个体的适应度值与最优个体的适应度值的相对偏差的倒数(参见 3.2.3 节)。依据抗体和抗原之间的亲和力来选择有效抗体能更好地体现“优胜劣汰”的原则, 特别是当待选抗体相差不大时, “优胜劣汰”的效果更明显, 搜索效率会更高。

(3) AIA 中抗体和抗体之间的排斥力体现了抗体与抗体之间的相似程度。在 AIA 的运行过程中计算抗体和抗体之间的排斥力, 并在选择操作时使选出的有效抗体中排斥力相同或相近的抗体数量控制在一个适当的范围以内, 即保持抗体群中有效抗体的多样性, 可以防止算法陷入局部收敛。

## 3 旅行商问题的人工免疫算法

### 3.1 旅行商问题的描述

TSP 问题就是要寻找一条遍历  $l$  个城市的最短路径, 在数学上可以描述为以下优化问题:

$$\min T_d = \sum_{i=1}^{l-1} d(c_i, c_{i+1}) + d(c_l, c_1), \quad (1)$$

$$st. c_i \in C, C = \{c_1, c_2, c_3, \dots, c_l\}, i = 1, 2, 3, \dots, l,$$

$$c_i, c_j \in \{1, 2, 3, \dots, l\}, \text{ and } c_i \neq c_j, i \neq j, i, j = 1, 2, 3, \dots, l.$$

其中  $C$  为城市集合,  $c_i$  为城市编号,  $i = 1, 2, 3, \dots, l$ ;  $d(c_i, c_j)$  为编号为  $c_i$  和  $c_j$  的两城市之间的距离, 且有  $d(c_i, c_j) = d(c_j, c_i)$ 。

### 3.2 旅行商问题的人工免疫算法

人工免疫算法的基本步骤已在上一节给出, 本小节只说明用 AIA 求解 TSP 问题时的几个特殊问题。

3.2.1 抗体编码方式 对于有  $l$  个城市的旅行商问题, 从其中的一个城市出发, 遍历其余  $(l-1)$  个城市且每个城市只去一次的路径有  $(l!/2l)$  条<sup>[2]</sup>。对这  $l$  个城市编号, 其号码分别为  $1, 2, 3, \dots, l$ , 并且把商人所在城市即出发城市编为第 1 号, 其它城市可随意编号。把这  $l$  个城市的编号任意排列成一个长度为  $l$  的字符串都可以形成一个抗体。因此抗体空间包含  $l!$  个抗体。而解空间只包含  $(l!/2l)$  个可行解。这  $l!$  个抗体只代表  $(l!/2l)$  个可行解。因此人工免疫算法要在抗体空间中搜索到与抗原相匹配的抗体, 比在解空间中搜索到最优解更难。为了缩小抗体空间, 提高搜索效率, 本文提出的 TSP 问题抗体编码方式将每个抗体(字符串)的第一个字符固定为出发城市的编号 1。这样, 每个抗体只有  $(l-1)!$  个字符可任意排列, 抗体空间就只包含  $(l-1)!$  个抗体。

3.2.2 初始抗体的产生与预处理 在一般情况下, 可按某种抗体编码方式, 随机产生  $N$  个抗体作为初始抗体, 构成初始抗体群。为了提高算法的搜索效率, 本文提出的旅行商问题人工免疫算法先对每一个初始抗体进行预处理, 然后才开始算法的迭代计算。设初始抗体  $A = (c_1, c_2, c_3, \dots, c_l)$ , 其中  $c_1 = 1$  (即  $c_1$  代表商人所在城市的编号), 对  $A$  进行预处理的步骤是:

(1) 随机取正整数  $r (1 \leq r \leq l)$ 。若  $r = c_1 = 1$ , 则令  $A' = (c_1, c_2, c_3, \dots, c_l) = A$ , 转(3); 若  $r = c_k \neq 1 (1 < k \leq l)$ , 则转下一步。

(2) 对初始抗体  $A$  中的各个字符依次循环左移位  $(k-1)$  次, 每次移位时, 使  $c_{i+1}$  移到  $c_i (i = 1, 2, 3, \dots, l-1)$  的位置, 且使  $c_1$  移到  $c_l$  的位置。设移位以后初始抗体  $A$  变为  $A'$ , 则  $A' = (c_k, c_{k+1}, \dots, c_l, c_1, c_2, \dots, c_{k-1})$ 。令  $A' = (c'_1, c'_2, c'_3, \dots, c'_l)$ , 其中  $c'_1 = c_k$ , 余类推。

(3) 首先取  $c'_m = c'_1$ , 令  $C = \{c'_2, c'_3, \dots, c'_l\}$ , 若对  $C$  中任一元素  $c'_n$ , 都有  $d(c'_m, c'_n) \leq d(c'_m, c'_1)$ ,  $c'_n \in C$ , 则把  $c'_n$  置于  $A'$  中  $c'_{m+1}$  的位置, 此时  $A'$  变为  $A^{(m)}$ , 令  $A^{(m)} = (c_1^{(m)}, c_2^{(m)}, c_3^{(m)}, \dots, c_l^{(m)})$ 。然后从  $C$  中删除元素  $c'_n$ , 再取  $c'_m = c'_2 (m)$ , 重复上述步骤, 直到  $C$  中的元素全部被删除为止。设这一步完成以后初始抗体  $A$  变为  $B$ 。

(4) 对初始抗体  $B$  中的各个字符依次循环右移位若干次, 直到抗体中第一个字符为 1 为止, 移位方法与(2)中的类似, 但方向相反。

上述初始抗体预处理方法是考虑到旅行商问题的任何一条路径都是闭合路径, 从任一城市出发, 要到达的下一个城市选择为未到过的城市中距该城市最近的一个。图 1 是一个 6 个城市的旅行商问题初始抗体预处理方法示意图。设某初始抗体  $A = (1, 3, 4, 6, 2, 5)$ , 城市 1 为商人首先出发城市。首先在 6 个城市中随机选择城市 4 作为当前出发城市, 对  $A$  左移位 2 次, 得  $A' = (4, 6, 2, 5, 1, 3)$ 。在其余的 5 个城市  $(1, 2, 3, 5, 6)$  中, 城市 5 距城市 4 最近, 则选择城市 5 作为下一个要到达的城市; 再以城市 5 作为当前出发城市, 在未到过的 4 个城市  $(1, 2,$

3,6)中,城市6距城市5最近,则选择城市6作为下一个要到达的城市;再以城市6作为当前出发城市,在未到达过的3个城市(1,2,3)中,城市2距城市6最近,则选择城市2作为下一个要到达的城市;再以城市2作为当前出发城市,在未到达过的2个城市(1,3)中,城市1距城市2最近,则选择城市1作为下一个要到达的城市;再以城市1作为当前出发城市,此时未到达过的城市只剩下城市3,则下一个要到达的就是城市3。这样,得到一个抗体  $B=(4,5,6,2,1,3)$ ,再用右移位方法使抗体  $B$  变为  $(1,3,4,5,6,2)$ 。经过上述预处理的初始抗体与抗原的亲合力较大,这样的抗体经人工免疫算子的作用能产生更好的新抗体的概率较大,算法搜索效率较高。

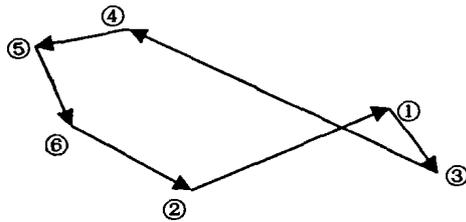


图1 初始抗体预处理方法示意图

3.2.3 亲和力和排斥力的计算 抗体与抗原之间的亲和力反映抗体与抗原之间的匹配程度。对于旅行商问题,可定义抗体  $B$  与抗原  $G$  之间的亲和力为

$$App(B) = 1 / (T_B - T_G) \quad (2)$$

其中  $T_B, T_G$  分别为抗体  $B$  和抗原  $G$  对应的旅行路线的总长度,  $T_G$  也是所求的最短路线的总长度。因在计算结束之前往往不知道  $T_G$  的大小,无法按(2)式计算抗体与抗原之间的亲和力,可以将(2)式修改为

$$App(B) = 1 / (T_B - T) \quad (3)$$

其中  $T$  为一适当的正数,且  $T < T_G$ 。按式(3)计算抗体与抗原之间的亲和力需作除法运算,如果再将(3)式修改为

$$App(B) = T_M - T_B \quad (4)$$

其中  $T_M$  为较大的正数,并要求  $T_M$  大于任意抗体对应的旅行路线的总长度,则可避免在计算抗体与抗原之间的亲和力时作除法运算。

抗体与抗体之间的排斥力反映抗体与抗体之间的差距。对于 TSP 问题,可定义抗体  $B_1$  与抗体  $B_2$  之间的排斥力为

$$Rep(B_1, B_2) = |T_{B_1} - T_{B_2}| \quad (5)$$

其中  $T_{B_1}, T_{B_2}$  分别为抗体  $B_1$  和抗体  $B_2$  对应的旅行路线的总长度。

3.2.4 人工免疫算子 本文构造 TSP 问题 AIA 的几种人工免疫算子如下(假定抗体按3.2.1中所述方法编码,即将每个抗体的第一个字符(即字符串的第一个字符)固定为出发城市的编号1)。

(1)字符换位算子,可分为单对字符换位算子和多对字符换位算子。

定义1 单对字符换位操作是对抗体  $A=(c_1, c_2, c_3, \dots, c_l)$ ,随机取两个正整数  $i, j(1 < i, j \leq l, i \neq j)$ ,以一定的概率  $p$ ,交换抗体  $A$  中一对字符  $c_i, c_j$  的位置;多对字符换位操作是预先确定一个正整数  $u$ ,对抗体  $A=(c_1, c_2, c_3, \dots, c_l)$ ,随机取一个正整数  $r(1 \leq r \leq u)$ ,再随机取  $r$  对字符作字符换位操作。

(2)字符移位算子,可分为单个字符移位算子和多个字符移位算子。

定义2 单个字符移位操作是对抗体  $A=(c_1, c_2, c_3,$

$\dots, c_l)$ ,随机取两个正整数  $i, j(1 < i, j \leq l, i \neq j)$ ,从  $A$  中取出一个字符串  $A_1, A_1=(c_i, c_{i+1}, \dots, c_{j-1}, c_j)$ ,以一定的概率  $p$ ,依次往左(或往右)移动字符串  $A_1$  中的各个字符,最左(或最右)边的一个字符则移动到最右(或最左)边的位置;多个字符串换位操作是预先确定一个正整数  $u$ ,随机取一个正整数  $r(1 \leq r \leq u)$ ,再在抗体  $A$  中随机取  $r$  个字符串作字符串移位操作。

(3)字符串逆转算子,可分为单个字符串逆转算子和多个字符串逆转算子。

定义3 单个字符串逆转操作是对抗体  $A=(c_1, c_2, c_3, \dots, c_l)$ ,随机取两个正整数  $i, j(1 < i, j \leq l, i \neq j)$ ,从  $A$  中取出一个字符串  $A_1, A_1=(c_i, c_{i+1}, \dots, c_{j-1}, c_j)$ ,以一定的概率  $p$ ,使字符串  $A_1$  中的各个字符首尾倒置;多个字符串逆转操作是预先确定一个正整数  $u$ ,随机取一个正整数  $r(1 \leq r \leq u)$ ,再在抗体  $A$  中随机取  $r$  个字符串作字符串逆转操作。

(4)字符重组算子。

定义4 字符重组操作是在抗体  $A=(c_1, c_2, c_3, \dots, c_l)$  中,随机取一个字符串  $A_1, A_1=(c_i, c_{i+1}, \dots, c_{j-1}, c_j)$ ,  $(1 \leq i, j \leq l, i \neq j, |i-j| < l-1)$ ,以一定的概率  $p$ ,使字符串  $A_1$  中字符重新排序。重新排序的方法与初始抗体预处理方法类似,即:保持  $c_i$  的位置不变,把城市  $c_{i+1}, c_{i+2}, \dots, c_{j-1}, c_j$  中距城市  $c_i$  最近的一个调整  $c_{i+1}$  的位置,使  $A_1$  变为  $A_1^{(1)}, A_1^{(1)}=(c_i, c_{i+1}^{(1)}, \dots, c_{j-1}^{(1)}, c_j^{(1)})$ ;再保持  $c_i, c_{i+1}^{(1)}$  的位置不变,把城市  $c_{i+2}, c_{i+3}, \dots, c_{j-1}^{(1)}, c_j^{(1)}$  中距城市  $c_{i+1}^{(1)}$  最近的一个调整  $c_{i+2}^{(1)}$  的位置,使  $A_1^{(1)}$  变为  $A_1^{(2)}, A_1^{(2)}=(c_i, c_{i+1}^{(1)}, c_{i+2}^{(2)}, \dots, c_{j-1}^{(2)}, c_j^{(2)})$ ;...;重复上述步骤,直到  $A_1$  中的所有字符都按上述方法被重新排序。

(5)优质字符串的保留。

定义5 如果若干个抗体与抗原之间的亲和力都很大,且这些抗体包含了一个相同的字符串,则称这个字符串为优质字符串,简称优质串。

定义6 如果抗体中存在优质字符串,则在抗体产生过程中以概率  $p_0$  使该优质串不受破坏,即把该优质串当成一个字符看待,称为优质字符串的保留。

## 4 仿真实验研究

为了验证 AIA 的有效性,本文对一个50个城市和一个100个城市的 TSP 问题进行了仿真实验研究,并将人工免疫算法的计算结果与遗传算法的计算结果进行了比较。

对于一个50个城市的旅行商问题,人工免疫算法取抗体群规模  $N=50$ ,人工免疫算子的作用概率分别为  $p_c=0.2, p_e=0.3, p_r=0.4, p_i=0.5, p_0=0.5, u_c=u_e=u_r=[l^{1/4}]$ ,  $[\cdot]$  表示取整运算,每次抗体选择时保留与抗原匹配得最好的抗体,重复进行20次计算,每次计算随机产生不同的初始抗体,并对初始抗体进行预处理,迭代5000次,计算结果如表1所示。表1中也给出了同一问题用 GAs 的计算结果。

再考虑一个100个城市的旅行商问题,人工免疫算法取抗体规模  $N=80$ ,人工免疫算子的作用概率同上,每次抗体选择时也保留与抗原匹配得最好的抗体,重复进行20次计算,每次计算随机产生不同的初始抗体,并对初始抗体进行预处理,迭代10000次,计算结果如表2所示。表2中也给出了同一问题用遗传算法的计算结果。

从表1和表2可以看出,本文提出的旅行商问题的人工免疫算法是非常有效的。50个城市的问题20次计算中人工免疫算法有7次重复搜索到了最好解,而遗传算法只有4次重复搜

(下转第89页)

断增大时,该算法时间开销并不是很大,而且其变化曲线近似由一条直线变成一条抛物线。

简而言之,我们设计的并在 Beowulf PCs Cluster 上实现的高度混合的遗传/模拟退火算法是一个成功率高、非常健壮的随机优化算法。应用该算法解决复杂的非线性 N 维函数全局优化问题将会是十分快速、有效的。(限于篇幅,有关本算法和其它全局随机优化算法的比较工作以后将另文给出,本文暂不予以考虑。)

致谢:美国佛吉尼亚理工大学 D. Chen 博士对本文提出了很好的修改建议,对此本文作者表示衷心的感谢。

### 参考文献

1 Esin O, Linet O. Parallel Simulated Annealing Algorithms in Global Optimization. Journal of Global Optimization, 2001, 19: 27~50

2 Hamma B S, Viitanen S, Torn A. Parallel Continuous Simulated Annealing for Global Optimization, presented at the NATO advance study institute - Algorithms for Continuous Optimization: The State of the Art, II Ciocco-castelvecchio Pascoli, Italy, 1993  
 3 Chen H, Flann N S, Watson D W. Parallel Genetic Simulated Annealing: A Massively Parallel SIMD Algorithm. IEEE Trans. on Parallel and Distributed Systems, 1998, 9: 126~136  
 4 Kimura K, Taki K. Time-homogeneous parallel annealing algorithm: [Report TR-673]. Institute for New Generation Computer Technology, Tokyo, Japan, 1991  
 5 Mahfoud S W, Goldberg D E. Parallel recombinative simulated annealing: A genetic algorithm: [IlliGAL Report No. 92002] . University of Illinois, Urbana, IL, 1992  
 6 Cetin B C, Barhen J, Burdick W. Terminal Peppeler Unconstrained Subenergy Tunneling (TRUST) for Fast Global Optimization. Journal of Optimization Theory and Applications, 1993, 77 (1): 97~127

(上接第82页)

索到了最好解;100个城市的问题20次计算中人工免疫算法有4次重复搜索到了最好解,而遗传算法只有2次重复搜索到了最好解。人工免疫算法的平均计算结果和最差计算结果都比

遗传算法的好。两个问题的最差计算结果与最好结果的相对偏差都在5%以内。可见,本文提出的算法有较好的全局搜索能力。

表1 50个城市旅行商问题的计算结果

计算数		1	2	3	4	5	6	7	8	9	10
计算结果	AIA	11369	11554	11874	11519	11369	11369	11409	11513	11555	11369
	GAs	11519	11369	11555	11369	11688	11673	11874	11369	11821	11905
相对偏差	AIA	0%	1.63%	4.44%	1.32%	0%	0%	0.35%	1.27%	1.64%	0%
	GAs	13.2%	0%	1.64%	0%	2.81%	2.67%	4.44%	0%	3.98%	4.71%
计算数		11	12	13	14	15	16	17	18	19	20
计算结果	AIA	11427	11554	11369	11453	11555	11369	11483	11520	11369	11519
	GAs	11782	11717	11554	11483	11818	11926	11703	11453	11369	11892
相对偏差	AIA	0.51%	1.63%	0%	0.74%	1.64%	0%	1.00%	1.33%	0%	1.32%
	GAs	3.63%	3.06%	1.63%	1.00%	3.95%	4.90%	2.94%	0.74%	0%	4.60%

表2 100个城市旅行商问题的计算结果

计算数		1	2	3	4	5	6	7	8	9	10
计算结果	AIA	15842	15753	15840	15716	15769	16106	15826	15716	16225	15987
	GAs	15826	15795	15987	15842	15716	15840	16108	15973	16286	16423
相对偏差	AIA	0.80%	0.24%	0.79%	0%	0.40%	2.48%	0.70%	0%	3.24%	1.72%
	GAs	0.70%	0.50%	1.72%	0.80%	0%	0.79%	2.49%	1.64%	3.63%	4.50%
计算数		11	12	13	14	15	16	17	18	19	20
计算结果	AIA	15753	15883	15716	16320	15788	15748	15716	16030	15795	15882
	GAs	15877	15905	16225	16239	16320	15716	15753	15788	15716	16030
相对偏差	AIA	0.24%	1.06%	0%	3.84%	0.46%	0.20%	0%	2.00%	0.50%	1.06%
	GAs	1.02%	1.20%	3.24%	3.33%	3.84%	0%	0.24%	0.46%	0%	2.00%

结束语 本文针对旅行商问题提出了一种模拟生物免疫系统的AIA,并将其与模拟生物进化过程的GAs进行了比较,指出AIA的新抗体产生方法比GAs的新个体产生方法要灵活得多,并且AIA依据抗体与抗原之间亲和力以及抗体与抗体之间的排斥力来选择抗体比GAs依据个体适应度值来选择个体能更好地体现“优胜劣汰”的自然选择机制。AIA比GAs具有更强的全局搜索能力,是继GAs以来又一种具有广阔应用前景的非数值计算优化方法。

### 参考文献

1 Lin W, Delgado-frias J G. Hybrid Newton-Raphson Genetic Algorithm for the Traveling Salesman Problem. Cybernetics and Systems, 1995, 26(4): 387~412  
 2 Chien-ying Lu, Delgado-frias J G, Lin W. A Clustering and Ge-

netic Scheme for Large TSP Optimization Problem. Cybernetics and Systems, 1998, 29(2): 137~157  
 3 孟繁桢,胡云昌,徐慧,等.旅行商问题的遗传算法.系统工程理论与实践,1997,17(9):15~21  
 4 赵赫,杜端甫.遗传算法求解旅行推销员问题时算子的设计与选择.系统工程理论与实践,1998,18(2):62~65  
 5 梁艳春,冯大鹏,周春光.遗传算法求解旅行商问题时的基因片段保序.系统工程理论与实践,2000,20(4):7~10  
 6 丁永生,任立红.人工免疫系统:理论与应用.模式识别与人工智能,2000,13(1):52~59  
 7 席裕庚,恽卫民.遗传算法综述.控制理论与应用,1996,13(6):697~708  
 8 文幼宇,刘沛,程时杰.遗传算法及其在电力系统中的应用(上、下).电力系统自动化,1996,20(10):58~60,20(11):60~65  
 9 李茂军,童调生.单亲遗传算法及其全局收敛性分析.自动化学报,1999,25(1):68~72  
 10 李茂军,朱陶业,童调生.单亲遗传算法与传统遗传算法的比较研究.系统工程,2001,19(1):61~65