

分层协同进化免疫算法及其在 TSP 问题中的应用

吴建辉^{1,2}, 章 兢¹, 张小刚¹, 刘朝华¹

(1. 湖南大学电气与信息工程学院, 湖南长沙 410082; 2. 湖南大学信息科学与工程学院, 湖南长沙 410082)

摘要: 为提高人工免疫算法求解 TSP 问题的效率, 借鉴分层和协同进化的思想, 构造了一种基于多子种群免疫进化的两层框架模型, 在此模型的基础上提出了一种基于竞争-合作的分层协同进化免疫算法 (Hierarchical Co-evolution Immune Algorithm, HCIA). HCIA 通过对若干个种群进行低层免疫操作: 局部最优免疫优势、克隆扩增及克隆选择算子、基于改进粒子群优化算法的抗体多样性改善和高层遗传操作: 选择、抗体迁移、变异, 增强优秀抗体实现亲和度成熟的机会, 提高抗体群分布的多样性, 在深度搜索和广度寻优之间取得了平衡. 针对 TSP 实验结果表明, HCIA 具有可靠的全局收敛性及较快的收敛速度.

关键词: TSP; 人工免疫算法; 分层; 协同进化; 粒子群优化算法

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2011) 02-0336-09

Hierarchical Co-Evolution Immune Algorithm and Its Application on TSP

WU Jian-hui^{1,2}, ZHANG Jing¹, ZHANG Xiao-gang¹, LIU Zhao-hua¹

(1. School of Electrical and Information Engineering, Hunan University, Changsha, Hunan 410082, China;

2. School of Information Science and Engineering, Hunan University, Changsha, Hunan 410082, China)

Abstract: In order to solve Traveling Salesman Problem (TSP) more efficient using artificial immune algorithm, using for reference of hierarchical and co-evolutionary idea, a two-floor model based on multiple-population immune evolution as well as Hierarchical Co-evolution Immune Algorithm (HCIA) based on competition-cooperation is put forward. Multiple subpopulations are operated by bottom floor immune operators: local optimization immunodominance, clonal expansion and other clonal selection operators, amelioration of antibody diversity based on improved Particle Swarm Optimization (PSO) algorithm. Multiple subpopulations are also operated by top floor genetic operators: selection, antibody migration, mutation. Through those operators, excellent antibody affinity maturation and diversity of antibody subpopulation distribution was enhanced, the balance between the depth and breadth of the search-optimizing was acquired. Experimental results for TSP indicate that HCIA has a remarkable quality of the global convergence reliability and convergence velocity.

Key words: TSP; artificial immune algorithm; hierarchical; co-evolution; PSO

1 引言

人工免疫系统是抽取生物免疫系统的特点并结合工程应用而描述的一种新兴智能信息处理系统^[1], 目前已被广泛应用于组合优化、网络安全等领域. 受免疫克隆选择学说的启发, 2000年, De Castro等人提出了一种较为简洁的克隆选择算法 (Clonal Selection Algorithm, CSA), 并成功地用于解决数值优化^[2]和 TSP 问题^[3], 文献^[4]提出了求解大规模 TSP 问题的自适应归约免疫算法, 这些算法在求解 TSP 问题中都取得了一定的效果, 但以上人工免疫算法采用基于个体自身亲和度的进化

模式, 而没有考虑其进化的环境和个体之间的复杂联系对个体进化的影响, 所以它在应用中表现出易早熟收敛等缺陷, 很难达到理想的效果. 协同进化算法是借鉴生物学模型的一类新的进化算法, 最早的研究始于 Potter 和 De Jong 的工作^[5]. 借鉴协同进化思想而建立的竞争协同进化模型, 能够很好地描述个体与环境及相互之间的协同行为, 基于此模型改进的克隆算法在一定程度上能够脱离只采用个体适应度控制进化的生物进化框架, 然而纯竞争协同模型随进化过程趋于平衡点易出现协同的弱化和退化现象^[6].

遗传算法(Genetic Algorithm, GA)是模拟自然界的物种进化而形成的一种并行搜索优化智能启发式算法,具有鲁棒性强、便于并行处理等特点,广泛应用于解决搜索和优化问题^[7],但基本遗传算法也存在着容易出现早熟收敛等缺陷.借鉴协同合作思想通过个体迁移等信息交流的手段将遗传算法予以改进,提高解决 TSP 问题的效率.

为进一步提高人工免疫算法求解 TSP 问题的效率,借鉴协同进化和分层的思想,提出了一种基于竞争-合作的分层协同进化免疫算法(HCIA).HCIA 分为上下两层,通过对若干个子种群进行低层免疫操作:局部最优免疫优势、基于种间竞争和种内竞争的克隆扩增等克隆选择算子、基于改进微粒群算法的抗体多样性改善和高层遗传操作:选择、抗体迁移、变异,增强优秀抗体实现亲和度成熟的机会,提高抗体群分布的多样性,在深度搜索和广度寻优之间取得了平衡.针对 TSP 实例的仿真结果表明,HCIA 具有可靠的全局收敛性及较快的收敛速度.

2 HCIA 算法

2.1 基本概念的描述

2.1.1 TSP 问题的描述

TSP 问题是著名的 NP 难问题,它可以简单描述为:已知 N 个城市 $V = \{v_1, v_2, \dots, v_N\}$ 以及任意两城市间的距离 $d(v_i, v_j)$,求一条经过 V 中所有城市一次且仅一次的闭合 $\pi(V = \{v_1, v_2, \dots, v_N\})$,使得总行程 $d = \sum_{i=1}^{N-1} d(v_i, v_{i+1}) + d(v_N, v_1)$ 最小.该问题描述简单但其实际模型在电路板布局, VLSI 芯片设计,车辆调度等优化问题^[8]中有着广泛的应用.

2.1.2 抗体及抗原描述

在免疫 CSA 算法求解 N 个城市的 TSP 问题时,抗体 a 对应候选路径,采用实数编码(用城市的序号表示).抗体子种群 Ab 可以表示为 $Ab = [a_1, a_2, \dots, a_m]$, m 为抗体的个数.抗原抽象为 TSP 问题的求解,用当前抗体群中的路径长度最短的抗体看做抗原.

2.1.3 亲和度函数定义

定义 1 抗体与抗原间的亲和度反映了抗体与抗原的匹配程度,亲和度函数定义如下:

$$\text{affinity}(a_i) = \frac{S_0}{\text{dist}(a_i)} \quad (1)$$

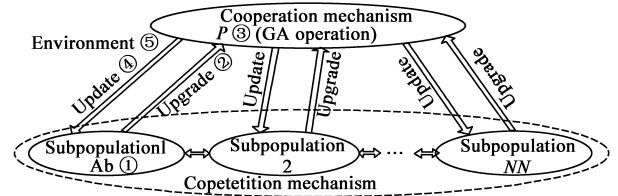
式中, $\text{dist}(a_i)$ 为 Ab 中第 i 个抗体的路径长度, S_0 为相应 TSP 实例的已知最短路径值.

2.2 HCIA 模型及流程

2.2.1 HCIA 模型

借鉴分层和协同进化的思想,设计了 HCIA 模型.

定义 2 HCIA 模型分为上下两层,低层抗体子种群集合 A 和高层抗体子种群 P 按照图 1 中的方式组织而成. $A = \{Ab_1, Ab_2, \dots, Ab_{NN}\}$,把群体分割成 NN 个子种群可以有效维持群体中个体的多样性,能有效避免单种群进化计算的未成熟收敛等缺陷.每个子种群采用竞争协同进化模型能有效提高收敛速度和收敛精度.提升操作是将低层 NN 个子种群提升至高层.高层操作采用基于协同合作的并行遗传算法.更新操作是把高层优化的子种群更新相应的低层子种群.



①低层抗体子种群 Ab ②提升操作 ③高层子种群 P ④更新操作 ⑤环境
图 1 分层竞争-合作型协同进化免疫模型

2.2.2 HCIA 算法流程

HCIA 算法将 CSA、分层、多种群、协同进化、PSO 等概念应用于 TSP 问题,流程如下:

Step1 初始化 NN 个抗体子种群(子种群规模为 m ,取值 30 或 40).

Step2 计算 NN 个抗体子种群中各抗体对应的路径长度,获得抗体种群的最短路径值.

Step3

While(算法终止条件不满足)

For(每个抗体子种群 Ab) //基于竞争协同的低层 LOICSA 操作
计算 Ab 中抗体亲和度并对 Ab 采用 LOIA 操作,
 $Ab' = \text{LOIA}(Ab)$.
克隆扩增, $Ab'' = T_c(Ab')$.
采用 Inver-over 变异算子高频变异, $T_m(Ab''(k))$.
免疫选择 T_s 操作.
基于 IPSO 的抗体多样性改善操作.

End For

If 迭代代数为 l_h 代的整数倍 //基于协同合作的高层遗传操作
依据选择算子选择低层子种群进入高层.
依据抗体迁移算子进行抗体迁移.
依据变异算子进行变异.
变异后各子种群采用最优保存策略.
转到 Step3,并更新相应的低层子种群.

Else

转到 Step3.

End

End While

Step4 输出种群最优抗体及其路径长度值.

2.3 HCIA 算法的低层操作

低层操作采用基于协同竞争模型的局部最优免疫优势克隆选择算法(Local Optimization Immunodominance CSA, LOICSA),操作算子如下:

2.3.1 局部最优免疫优势算法(LOIA)

依据文献[9]抗体免疫优势的相关结论,本文分析每次迭代中最优抗体(最优抗体路径长度最短,广义地讲,它具有免疫优势),从而获得免疫优势,使算法更具有通用性.抗体 LOIA 算子原理参见文献[10].

LOIA 算法流程

依据抗体与抗原间的亲和力大小将 Ab 中抗体按照降序来排序,并分为两部分: $Ab(H) = [a_1, a_2, \dots, a_H]$, $Ab(L) = [a_{H+1}, a_{H+2}, \dots, a_{H+L}]$, $H = L = m/2$. 其中 a_1 的亲和力最大, a_{H+L} 的亲和力最小.

Step1 选取 $Ab(H)$ 中的 H 个高亲和力抗体.

Step2

For(每个被选抗体)

被选抗体和 Ab 中亲和力最高的抗体进行 LOI 操作,得到两个新抗体.

取两个新抗体中亲和力高的抗体,若其亲和力高于被选抗体的亲和力,则替换,否则丢弃新抗体.

End For

Step3 得到新的抗体子种群 Ab' .

2.3.2 基于种间竞争和种内竞争的克隆扩增算子 q_i :

定义 3 假设 Ab 有 m 个抗体,每个抗体有 N 位基因,采用的编码符号集大小为 $|S|$, $S = \{S_1, S_2, \dots, S_{|S|}\}$, 则基因座 j 的信息熵定义为:

$$H_j(m) = - \sum_{i=1}^{|S|} P_{ij} \log_2(P_{ij}) \quad (2)$$

其中, $H_j(m)$ 为第 j 个基因座的总体信息测度, P_{ij} 为 S 中第 i 个符号在 j 上的概率即 $P_{ij} = n_{ij}/m$, n_{ij} 为在 j 上出现第 i 个符号总个数. 子种群平均信息熵定义为:

$$H = \frac{1}{N} \sum_{j=1}^N H_j(m) \quad (3)$$

子种群的平均信息熵越大,种群的分布性越好.

(1) 种间竞争机制

定义 4 第 k 个子种群 Ab_k 的竞争力指数定义为:

$$F(Ab_k) = \left(l_1 \cdot \frac{\text{maff}(Ab_k)}{\frac{1}{NN} \sum_{i=1}^{NN} \text{maff}(Ab_i)} + l_1 \cdot \frac{H(Ab_k)}{\frac{1}{NN} \sum_{i=1}^{NN} H(Ab_i)} + l_2 \cdot \frac{\frac{1}{NN} \sum_{i=1}^{NN} \text{MITL}(Ab_i)}{\text{MITL}(Ab_k)} \right) \quad (4)$$

式中, $\text{maff}(Ab_k)$ 为 Ab_k 中所有抗体亲和度的平均值, $H(Ab_k)$ 为 Ab_k 的平均信息熵, $\text{MITL}(Ab_k)$ 为 Ab_k 中所有抗体的最短路径值 MITL (Minimal value of Tour Length). $F(Ab_k)$ 为 Ab_k 的竞争力指数,它是反映种群亲和力、多样性、最短路径值的综合指数. 与其它子种群相比,若 $\text{maff}(Ab_k)$ 、 $H(Ab_k)$ 、 $\text{MITL}(Ab_k)$ 越优,则 Ab_k 的竞争力指数值越大. l_1, l_2 为相应权重系数,其自适应变化公式如下:

$$l_1 = l_1 \cdot (1 + \frac{It}{H_m} \cdot C_1) \quad (5)$$

$$l_2 = l_2 \cdot (1 - \frac{It}{H_m} \cdot C_2) \quad (6)$$

式(5)、(6)中, $C_1 = C_2 = 0.01$, It 为当前进化代数, H_m 为最大迭代代数. 在 $It = 1$ 时, $l_1 = 1, l_2 = 4$. 在进化初期,各子种群的 MITL 相差较大, l_2 取较大值可突出 MITL 对竞争力指数的影响;随着进化的不断进行,各子种群 MITL 的差距不断减少,通过 l_1, l_2 自适应变化,逐渐加大亲和力均值和平均信息熵对竞争力指数的影响.

Ab_k 的竞争奖励倍数 Cb_k 公式如下:

$$Cb_k = L \cdot F(Ab_k) \quad (7)$$

其中 L 为竞争奖励放大系数. 由式(7)可知, $F(Ab_k)$ 越大,则 Cb_k 也越大.

随进化过程趋于平衡点(各子种群的竞争力指数值相差不大)时,会出现协同的弱化和退化现象^[6] (Cb_k 相差不大),所以在进化中后期,将各子种群依据竞争力指数值按照降序来排序,并将 Cb_k 公式改为:

$$Cb_k = L \cdot F(Ab_k) + h_k \quad (8)$$

子种群 Ab_k 的 $F(Ab_k)$ 越大,排位越靠前时,竞争奖励常数 h_k 值也越大 ($h_k > 0$),因而 Cb_k 也越大.

(2) 种内竞争机制

在克隆扩增之前将子种群 Ab'_k 中的抗体依据亲和力大小按照降序来排序,并分为两部分: $Ab'_k(H) = [a'_1, a'_2, \dots, a'_H]$, $Ab'_k(L) = [a'_{H+1}, a'_{H+2}, \dots, a'_{H+L}]$. $H = L = m/2$, a'_1 的亲和力最大, a'_{H+L} 的亲和力最小. 为了减少计算量只选择 $Ab'_k(H)$ 中抗体克隆扩增.

(3) 基于种间竞争和种内竞争机制的克隆扩增算子 q_i

$Ab'_k(H)$ 中抗体的克隆扩增算子 q_i 公式如下:

$$q_i = \text{round}(\frac{Cb_k}{i} + b) \quad (9)$$

式中, i 为 $Ab'_k(H)$ 中抗体所对应的位置,抗体 a'_1 的亲和力最高,则 $i = 1$,克隆倍数最大. $\text{round}(\ast)$ 为取整, b 取 20. 当 Cb_k 越大时, $Ab'_k(H)$ 中抗体的克隆扩增倍数也越大, Ab_k 越有机会在竞争中胜出.

2.3.3 基于动态变异概率的高频变异算子

(1) 动态变异概率

为了提高算法的收敛速度,通过下式动态改变变异概率 p .

$$p = p \times (1 - \frac{It}{H_m} \times 0.01) \quad (10)$$

(2) 高频变异算子

针对 TSP 问题,本文变异采用 Inver_over 变异算子. 实验已经证明该变异算子^[8] 优于传统的交叉算子,

如:PMX,OX,CX 等.

2.3.4 免疫选择 T_s^c

$S_i(k) = T_s^c(a_i(k)) = \max\{\text{affinity}(a_i'(k))\}$, 然后 $S_i(k) \cup a_i(k) \rightarrow a_i(k+1)$, $a_i'(k)$ 表示第 k 个子种群中抗体 $a_i(k)$ 经克隆扩增、变异后所得到的抗体. 通过局部择优实现了子种群的压缩, 同时保证了子种群的最优解不会变差.

2.3.5 基于 IPSO 的抗体多样性改善机制

PSO 算法是通过群体粒子间的协作与竞争产生的群体智能优化搜索算法, 由 Kennedy 和 Eberhart 受鸟群觅食行为的启发于 1995 年提出. PSO 参数结构简单, 无需复杂的调整, 但基本 PSO 也存在早熟收敛现象. PSO 已经应用到函数优化、信号处理、TSP 问题等领域^[11].

(1) 粒子选择策略

① 最优选择法

当 Ab_k 中有 Ct ($Ct > 1$) 个路径长度相同最优抗体

If $\text{MITL}(l, k) = \text{MITL}(l)$ & $\text{MITL}(l - l_p, k) = \text{MITL}(l)$ & $\text{MITL}(l) > S_0$

Ab_k 的最优抗体陷入局部最优, 将 Ct 个最优抗体都作为粒子.

Else

Ab_k 的最优抗体未陷入局部最优, 将 $Ct - 1$ 个最优抗体都作为粒子.

End

其中 $\text{MITL}(l, k)$ 为当前第 l 代时 Ab_k 的 MITL 值, $\text{MITL}(l)$ 为第 l 代时所有子种群的 MITL 值, l_p 为局部最优设定代数.

② 随机选择法

在 Ab_k 的次优抗体中随机选择 nb 个抗体作为粒子.

文献[5]通过理论分析和数字试验表明, 综合两种选择方法选择相应的粒子, 可以发挥两者各自的优势, 弥补各自的不足.

(2) 改进的 PSO 算法

在 PSO 算法中, 粒子群中的粒子在一个 N 维空间中搜索, 粒子的位置表示问题的一个解. 粒子通过不断调整自己的位置 X 来搜索新解. 每个粒子搜索到的最好解记作 P_{id} , 整个粒子群经历过的最好位置记作 P_{gd} . 每个粒子都有一个速度, 记作 V .

$$V'_{id} = W \cdot V_{id} + C_1 \cdot R_1 \cdot (P_{id} - X_{id}) + C_2 \cdot R_2 \cdot (P_{gd} - X_{id}) \quad (11)$$

式中, W 为惯性权重, V_{id} 表示第 i 个粒子第 d 维上的速度, 限制 V_{id} 为 $[-V_{\max}, V_{\max}]$. C_1, C_2 为加速系数, 一般取 2.0. R_1, R_2 为 $(0, 1)$ 之间的随机数.

粒子移动的下一个位置为

$$X'_{id} = X_{id} + V'_{id} \quad (12)$$

X_{id} 表示第 i 个粒子第 d 维上的位置.

通过引入交换子和交换序^[12]的概念, 将式(11)进行改造应用到 TSP 问题中. 基本 PSO 算法中的粒子仅根据自身的个体极值和全局极值来更新速度和位置, 并没有考虑其它粒子的信息, 因此在解空间的搜索是单向的. 在本文改进的 PSO (Improved PSO) 算法中, 将粒子群中所有粒子的路径长度按升序进行排序, 选取前 h 个粒子的信息来修正每个粒子下一次迭代的行动策略. 这样, 整个粒子群在解空间的搜索是多方向性的, 搜索过程更均匀, 能有效提高算法的精度和全局收敛能力. 重新构造的速度算式如下:

$$V'_{id} = WV_{id} \oplus \alpha (P_{id} - X_{id}) \oplus \left(\prod_{j=1}^h \beta_j (P_{gd_j} - X_{id}) \right) \quad (13)$$

式(13)的参数说明:

① 惯性权值 W 采用以下递减策略能在全局搜索和局部搜索之间取得较佳平衡^[13], 公式如下:

$$W = (W_{start} - W_{end}) (It / It_m)^2 + (W_{end} - W_{start}) (2It / It_m) + W_{start} \quad (14)$$

式(14)中, $W_{start} = 0.6$, $W_{end} = 0.4$, 当 $It = 1$ 时, $W = W_{start}$, 当 $It = It_m$ 时, $W = W_{end}$.

② α, β_j ($\alpha, \beta_j \in [0, 1]$) 为随机数, $\alpha (P_{id} - X_{id})$ 表示基本交换序 $(P_{id} - X_{id})$ 中的所有交换子以概率 α 保留.

$$\prod_{j=1}^h \beta_j (P_{gd_j} - X_{id}) = \beta_1 (P_{gd_1} - X_{id}) \oplus \beta_2 (P_{gd_2} - X_{id}) \oplus \cdots \oplus \beta_h (P_{gd_h} - X_{id}) \quad (15)$$

式(15)中, P_{gd_h} 表示排序后第 h 个粒子的解, $\beta_j (P_{gd_j} - X_{id})$ 表示基本交换序 $(P_{gd_j} - X_{id})$ 中的所有交换子以概率 β_j 保留. 由此可以看出, α 的值越大, $(P_{id} - X_{id})$ 保留的交换子就越多, P_{id} 的影响就越大; 同理, β_j 的值越大, $(P_{gd_j} - X_{id})$ 保留的交换子就越多, P_{gd_j} 的影响就越大.

(3) 基于 IPSO 的抗体多样性改善机制

当 $It > It_d$ (It_d 为多样性操作代数) 且 $Ct > 1$ 时, 每隔 It_i 代进行基于 IPSO 的抗体多样性改善操作. 流程如下:

Step1 根据粒子选择策略选择相应的抗体组成粒子群, 建立粒子记忆库, 初始化粒子的速度和位置.

Step2 用式(13)、(12)更新粒子速度和位置.

Step3 将上述粒子的路径长度按照升序来排序, 前三分之一的粒子保留, 中间三分之一的粒子杂交, 后面三分之一的粒子变异.

① 杂交操作

选取中间三分之一的粒子依据杂交概率与前三分之一的粒子进行杂交操作, 用产生的孩子粒子代替父母粒子, 以保持子种群的粒子数目不变.

$$X'_{jd} = X_{jd} \oplus X_{kd} \quad (16)$$

其中, X_{jd} 是中间三分之一粒子中第 j 个粒子第 d 维位置, X_{kd} 是前三分之一粒子中第 k 个粒子第 d 维位置.

$$V'_j = \frac{V_j + V_k}{|V_j| + |V_k|} |V_j| \quad (17)$$

式中, $|V_j|$ 为速度长度, $|V_j| = \sqrt{V_{j1}^2 + V_{j2}^2 + \dots + V_{jN}^2}$, $|V_k|$ 第 k 个粒子的速度长度.

通过遗传交叉算子操作可以增加粒子多样性,充分利用群体中优良粒子的特性,加快群体的收敛速度.

②变异操作

变异操作选择的是一种重新初始化的方式,即是对选中要进行变异的粒子,随机按照一定的变异概率对部分维数进行初始变异,以保持和提高种群的多样性,防止出现算法陷入早熟收敛.

Step4 计算当代每个粒子的路径长度值,若优于其初始的路径长度值,则更新粒子记忆库.当达到设定代数后转到 Step5,否则返回 Step2.

Step5 使用粒子记忆库中的最优 $C_t - 1$ (或 C_t) 个粒子更新子种群中 $C_t - 1$ (或 C_t) 个相同抗体.若 $C_t - 1$ (或 C_t) 个抗体中仍有 N_r 个没有改变,则可以采用如下替换策略中的一个:

①从一个随机产生的规模为 R 的候选抗体群 Ab^R 中选择 N_r 个亲和度较高的抗体来取代.

②对没有改变的每个抗体采用 Inver_over 变异操作进行基因位的随机互换,直到与原来的抗体不同(抗体路径长度值不同).

通过基于 IPSO 的抗体多样性引入机制操作,使得 $C_t - 1$ (或 C_t) 个路径相同最优抗体予以改变(优化或加入随机抗体),极大改善子种群中抗体的多样性.

2.4 HCIA 算法的高层操作

HCIA 算法的高层操作是采用基于粗粒度模型的协同合作并行遗传算法,这里的协同合作是对同一问题采用多个子种群独立地进行演化并通过个体迁移等手段达到信息交流、合作的目的,这样有利于各子种群共享进化的成果,同时给各子种群引进新的个体,有效避免早熟现象.高层操作算子如下:

2.4.1 选择算子 低层 NN 个子种群提升至高层,组成高层子种群 P .

2.4.2 抗体迁移算子 迁移拓扑确定了子种群之间个体的迁移路径,本文采用环形迁移拓扑结构见图 2.

在粗粒度模型中,迁移策略主要控制参数如下:

(1)迁移选择 迁移抗体的选择策略:

①最优抗体 选取子种群中的最优抗体,但在有些情况下,这种策略显得太“贪

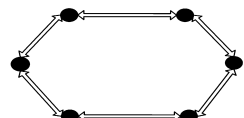


图2 环状拓扑结构的粗粒度模型并行遗传算法的基本结构

婪”.

②随机选择 从子种群中随机选取抗体.

③“平均”抗体 选择具有平均亲和度的抗体.

(2)迁移规模 按照迁移抗体的选择策略选择相应的抗体,迁移抗体数即迁移规模为子种群抗体数的 10% ~ 20%.

(3)迁移替换 用选择的抗体相互替代子种群中亲和度最低的抗体.

(4)迁移率 迁移率决定了个体迁移的时间间隔,通常,迁移规模越大,迁移率就越小.本算法的迁移率 $It_h = 20$.

2.4.3 变异算子 依照高层变异概率 P_{mh} 确定需变异抗体,采用基本位变异方法,即将需变异的抗体用随机产生抗体来代替,以增加抗体的多样性.

2.4.4 最优保存算子 经过抗体迁移和变异后,采用了最优保存策略,将原来子种群的最优抗体予以保留,这样可以防止算法运行中抗体出现退化,减缓收敛速度.

3 算法收敛性分析

引理 1 文献[14]研究表明基于免疫克隆选择策略算法的种群序列 $\{A(n), n \geq 0\}$ 是有限齐次马尔可夫链,并且以概率 1 收敛.

定理 1 基于 LOICSA 的抗体子种群优化解序列 $\{A(n), n \geq 0, n$ 为迭代代数 $\}$ 是有限齐次马尔可夫链.

证明 与免疫克隆选择策略算法一样,LOICSA 的状态变化均在有限空间中进行,只是其矢量的分量是离散实数,因此种群是有限的.由于 $A(k+1) = T(A(k)) = T_s^c \circ T_m^c \circ T_e^c \circ T_l(A(k))$ (k 为迭代代数),其中 T_l , T_e^c , T_m^c , T_s^c 分别代表局部最优免疫优势,克隆扩增,克隆变异,免疫选择; T_l 的操作是针对特定抗体的一个动态调整局部最优过程,与 k 无关;由引理 1 可知, T_e^c , T_m^c , T_s^c 也与 k 无关,因此 $A(k+1)$ 仅与 $A(k)$ 有关,即 $\{A(n), n \geq 0\}$ 是有限齐次马尔可夫链.

定义 5 对于任意的初始分布均

$$\lim_{k \rightarrow \infty} \sum_{s_j \cap X^k \neq \Phi} P\{A_k^i\} = 1 \quad (18)$$

则称算法以概率 1 收敛.

定理 2 LOICSA 是以概率 1 收敛于全局最优解.

证明 上述算法 Ab 初始规模为 m ,克隆后中间种群的规模为 Nm (通过 LOICSA 的相关算子操作后,得到优秀抗体群 m'_0),初始种群中的全部近似解看成是状态空间 $S1 := X^m$ 中的一个点,其中 X 为抗体子种群,而将中间种群中的全部近似解看成是状态空间 $S2 := X^{Nm}$ 中的一个点,当没有必要区分 S^1 和 S^2 时,用 S 表示状态

空间, $s_i \in S$ 表示 S 中的第 i 个状态. 用 X^* 表示由 $f(x)$ ($f(x)$ 是 X 上的亲和度函数) 在 S^2 上取最优值的所有抗体组成, $X^* = \{x \in X | f(x) = \max_{x_i \in X} f(x_i)\}$. x_i 为抗体 x_i 克隆扩增并变异后的抗体集. 记 $I = \{i | s_i \cap X^* \neq \Phi\}$, 设随机过程 $\{A_k\}$ 的转移概率为 $p_{ij}(k) = p \left\{ \frac{A_{j,k+1}}{A_k^i} \right\}$, 讨论 $p_{ij}(k)$ 的 2 种特殊情况:

① 当 $i \in I, j \notin I$ 时, 由免疫选择可知, 下一代抗体是取本代抗体和其克隆变异后最优抗体这两个中的较好者, 保证抗体不会变差, 所以 $p_{ij}(k) = 0$.

② 当 $i \notin I, j \in I$ 时, $p_{ij}(k) \geq 0$.

记 $p\{A_k^i\}$ 为 $p_i^k, p_k = \sum_{i \notin I} p_i(k)$, 则由马尔可夫链性质知:

$$p_{k+1} = \sum_{i \in S} \sum_{j \notin I} p_i(k) p_{ij}(k)$$

$$= \sum_{i \in I} \sum_{j \notin I} p_i(k) p_{ij}(k) + \sum_{i \notin I} \sum_{j \notin I} p_i(k) p_{ij}(k)$$

有

$$\sum_{i \notin I} \sum_{j \notin I} p_i(k) p_{ij}(k) + \sum_{i \notin I} \sum_{j \in I} p_i(k) p_{ij}(k) = \sum_{i \notin I} p_i(k) = p_k$$

因此:

$$\sum_{i \notin I} \sum_{j \notin I} p_i(k) p_{ij}(k) = p_k - \sum_{i \notin I} \sum_{j \in I} p_i(k) p_{ij}(k),$$

则

$$0 \leq p_{k+1} \leq p_k - \sum_{i \notin I} \sum_{j \in I} p_i(k) p_{ij}(k) \leq p_k \leq 1$$

于是有

$$\lim_{k \rightarrow \infty} p_k = 0,$$

则

$$1 \geq \lim_{k \rightarrow \infty} \sum_{s_i \cap X^* \neq \Phi} p_i(k) \geq \lim_{k \rightarrow \infty} \sum_{i \in I} p_i(k) = 1 - \lim_{k \rightarrow \infty} p_k = 1$$

可证明式(18)概率 1 收敛.

定理 3 HCIA 是以概率 1 收敛于全局最优解.

证明 不妨设问题 P 为 HCIA 优化问题, f^* 为其全局最优适应值, 即 $f^* = \max\{f(x) | x \in P\}$, 则可令 l 时刻总体进化种群表示为 NN 个子种群的集合 $\{Ab_1(l), Ab_2(l), \dots, Ab_{NN}(l)\}$, 优秀抗体集合为 $\{f_1^*(l), f_2^*(l), \dots, f_s^*(l)\}$, 假设 $Z(l) = \max\{f_1^*(l), f_2^*(l), \dots, f_s^*(l)\}$, 则待证

$$\lim_{l \rightarrow \infty} p(Z(l) = f^*) = 1 \quad (19)$$

根据 LOI 的保留与更新规则可知, 在进化过程中若某子种群收敛于 f^* , 则肯定全局收敛于 f^* , 即有 HCIA 算法全局收敛的概率大于其任意子种群收敛于全局最优解的概率. 对于任意的子种群 Ab^* , 有

$$\lim_{l \rightarrow \infty} p(Z(l) = f^*) \geq \lim_{l \rightarrow \infty} p(\max\{f(x) | x \in Ab^*(l)\} = f^*) \quad (20)$$

HCIA 算法的低层 LOICSA 以概率 1 收敛于全局最优值(定理 2 已证明), 而根据 HCIA 算法的高层算子及精英保留规则可知, 高层操作不影响低层 LOICSA 的全局收敛性. 由只要存在某一种群 Ab , 其种内最优适应值等于全局最优适应值 f^* , 即 $f^* = \max\{f(x) | x \in Ab^*\}$ 而且 HCIA 中的成熟种群即为趋于收敛的种群. 因此若 Ab^* 在进化过程中成为成熟种群, 则趋于收敛到其种内最优解 f^* , 即有

$$\lim_{l \rightarrow \infty} p[\max\{f(x) | x \in Ab^*(l)\} = f^* / Ab^*] = 1$$

又对于子种群 Ab^* , $\lim_{l \rightarrow \infty} p(Ab^*) = 1$.

综上所述, 有

$$1 \geq \lim_{l \rightarrow \infty} p(Z(l) = f^*)$$

$$\geq \lim_{l \rightarrow \infty} p(\max\{f(x) | x \in Ab^*(l)\} = f^*)$$

$$\geq \lim_{l \rightarrow \infty} p[\max\{f(x) | x \in Ab^*(l)\} = f^* / Ab^*]$$

$$\times \lim_{l \rightarrow \infty} p(Ab^*) = 1 \times 1 = 1$$

即 $\lim_{l \rightarrow \infty} p(Z(l) = f^*) = 1$

4 仿真实验

4.1 算法比较

为了更好地说明 HCIA 算法的有效性, 选用国际上通用的 TSPLIB 测试库中多个实例进行测试. 试验仿真环境: Windows XP 系统, 1.66GHZ 主频的 INTEL 处理器, 760MB 内存, 仿真软件 MATLAB7.0. 实例 1 是单种群 IDIA (Immunodominance Immune Algorithm)、单种群 LOICSA 算法、低层无种间竞争高层只有经典遗传算法的多种群 MLOICSA 算法 (Multi-population LOICSA)、HCIA 四种算法针对 Tsp225 进行 10 次独立测试, 具体数据见表 1. 四个算法的截止代数数为城市数的 3 倍 (675 代), 变异初始概率 $P_m = 0.78$. LOICSA 算法抗体数为城市数 225, MLOICSA、HCIA 的抗体种群规模为 40×10 (子种群有 40 个抗体, 共有 10 个子种群). IDIA 的数据来自文献 [14]. 图 3 是针对 Tsp225 问题各算法收敛过程对比图, 从图 3 可知, HCIA 算法的收敛速度及最优解均优于 IDIA、LOICSA、MLOICSA 算法, 其原因在于协同进化及多子种群的应用.

表 1 为 IDIA、LOICSA、MLOICSA、HCIA 四种算法针

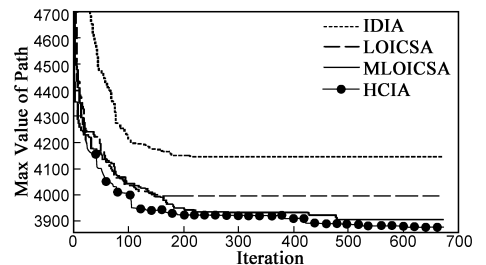


图3 Tsp225的收敛过程对比

对 TSP 实例的测试结果.表 1 列出了在 10 次测试结果中达到最优值次数 N^{best} ,最长路径值 MTL(Maximal value of Tour Length),路径均值 METL(Mean value of Tour Length),平均截止代数 MEI(Mean End Iteration),平均百分误差 $\sigma = \frac{\sum_{i=1}^{10} (S_{Ti} - S_0)}{10S_0} \times 100\%$ (其中 S_{Ti} 是第 i 次的最短路径)等参数.对于表 1 中的 TSP 实例,对于 MLOICSA、HCIA 算法,不带“*”表示抗体种群规模为 30×10 ,带“*”表示为 40×10 .算法截止代数为城市数的

3 倍.算法退出条件为达到截止代数或者找到最优值或者为相同路径值的迭代次数达到预设代数 It_n . IDIA 无 MEI 数据,用“-”表示.从表 1 可知:①基于 LOI 算子的 LOICSA 算法性能明显优于 IDIA 算法,从而验证了 LOI 算子的有效性;②基于多种群的 MLOICSA、HCIA 算法性能明显优于基于单种群的 IDIA、LOICSA 算法,从而验证了多种群的有效性;③HCIA 算法的性能也明显优于 MLOICSA,从而验证了协同进化算法的有效性;④实验也发现随着城市数目增加,会使算法性能有所下降,尤其对于 Pr1002 算法性能下降较多.

表 1 其他实例测试

Problem S_0	Method	IDIA	LOICSA	MLOICSA	HCIA	Problem S_0	Method	IDIA	LOICSA	MLOICSA	HCIA
att48 33524	N^{best}	1	6	9	10	* chn144 30354.3	N^{best}	0	0	0	0
	MTL	34076	33701.2618	33588.3434	33523.7085		MTL	31797.0936	31027.1676	30946.0536	30641.9261
	MITL	33524	33523.7085	33523.7085	33523.7085		MITL	30824.4895	30516.3054	30383.0643	30382.1477
	METL	33666	33562.7882	33530.1720	33523.7085		METL	31297.8266	30771.0061	30604.0768	30436.7360
	MEI	-	192	39.3	37.4		MEI	432	576	467.5	454.2
	$\sigma(\%)$	0.42	0.12	0.019	0		$\sigma(\%)$	3.049	1.37	0.82	0.272
	* Tsp225 3859	N^{best}	0	0	0		0	* A280 2586.8	N^{best}	0	0
MTL		4144.2	3997.4192	3938.9447	3941.8159	MTL	2815.2		2668.7999	2662.7500	2616.7534
MITL		3898.4	3925.6913	3878.3564	3870.0629	MITL	2621.6		2592.9980	2586.7696	2586.7696
METL		4020.5	3954.3721	3915.6573	3904.9706	METL	2704.4		2639.7102	2609.581	2594.8484
MEI		-	675.0	515.1	569.0	MEI	-		840	781.2	696.3
$\sigma(\%)$		4.18	2.471	1.468	1.191	$\sigma(\%)$	4.55		2.05	0.88	0.311
* Pcb442 5078.35		N^{best}	0	0	0	0	* Pa561 19330.8		N^{best}	10	10
	MTL	5468.83	5276.6	5213.9856	5180.2181	MTL		16790.83	15638.1011	15559.2922	15516.5636
	MITL	5184.49	5118.0	5115.6345	5113.2660	MITL		15739.69	15445.5290	15441.8438	15319.5058
	METL	5278.40	5193.0	5170.3142	5134.4309	METL		16146.23	15521.4352	15485.1120	15407.5970
	MEI	-	1326	688.1	1177.9	MEI		-	1683	1420	1679.9
	$\sigma(\%)$	3.94	2.26	1.811	1.104	$\sigma(\%)$		0	0	0	0
	* Gr666 3952.54	N^{best}	10	10	10	10		* Pr1002 259068	N^{best}	0	0
MTL		3290.6	3199.6368	3168.9853	3167.1139	MTL	269499		268082.2268	267419.6856	267083.9260
MITL		3200.6	3135.6174	3128.6477	3126.2800	MITL	269343		266362.6579	264658.4849	264468.1456
METL		3275.3	3176.0864	3149.2949	3142.8657	METL	269355		267341.6859	265884.0714	265311.9387
MEI		-	1998	1881.6	1925.6	MEI	-		3006	2484.8	2638.8
$\sigma(\%)$		0	0	0	0	$\sigma(\%)$	3.97		3.194	2.631	2.410

4.2 抗体个数 m 对 HCIA 算法性能影响

表 2 为子种群中抗体数 m 对 HCIA 性能影响的分析.表 2 中种群规模(Population Size, PS) $m' = m \times NN$, NN 为子种群个数. T 为 10 次测试结果的平均运行时间,单位是 s.从表 2 可知,无论 m 取何值,HCIA 的解均明显优于 MLOICSA.总体而言,随着抗体个数增加,算法性能尤其是均值有所提高,但计算时间也增加较多,因此, m 取 30 或 40 时较优.从表 2 也可知,由于抗体多样性改善以及协同进化机制的运用,HCIA 算法的计算时间多于 MLOICSA 算法.

4.3 子种群个数 NN 对 HCIA 算法性能影响

表 3 为子种群个数 NN 对 HCIA 性能影响的分析.从表 3 可知,无论 NN 取何值,HCIA 的解均明显优于

MLOICSA.总体而言,随着子种群个数增加,算法性能尤其是均值有所提高,但计算时间也增加较多,因此, NN 取 10 时较优.

5 结论

本文将免疫、协同进化、分层、PSO 等概念应用于 TSP 问题,提出了 HCIA 算法.通过对若干个子种群进行基于协同竞争的低层 LOICSA 算法操作以及基于改进微粒群算法的抗体多样性改善操作,避免了抗体群被少数亲和度最高的抗体占满,增强优秀抗体实现亲和度成熟的机会;通过基于协同合作的高层并行遗传算法操作有利于各子种群共享进化的成果,避免早熟现象,在深度搜索和广度寻优之间取得了平衡.

仿真试验表明 HCIA 算法针对所列举的 TSP 实例非常有效,与 IDIA、LOICSA、MLOICSA 相比,全局搜索能力更强,效率更高.HCIA 算法中不同参数设置时的性能

分析也表明 HCIA 具备良好的全局收敛可靠性.当求解大规模 TSP 问题时如何避免 HCIA 运行时间过长,是需要进一步研究和解决的问题.

表 2 子种群中抗体个数对 HCIA 性能的影响

Problem S ₀	PS	m' = 20 * 10	m' = 30 * 10	m' = 40 * 10	m' = 50 * 10	MLOICSA	Problem S ₀	PS	m' = 20 * 10	m' = 30 * 10	m' = 40 * 10	m' = 50 * 10	MLOICSA
	$\frac{m'}{m_0 * NV}$							$\frac{m'}{m_0 * NV}$					
Eil76 545.32	N ^{best}	5	8	8	9	2	Rd100 7910.4	N ^{best}	8	8	9	9	7
	MTL	548.3569	549.7780	548.3569	546.8864	550.7486		MTL	7944.3219	7911.3049	7911.3049	7911.3049	7959.0996
	MITL	544.3691	544.3691	544.3691	544.3691	544.3691		MITL	7910.3962	7910.3962	7910.3962	7910.3962	7910.3962
	METL	545.8934	545.4395	544.8697	544.6208	547.3938		METL	7917.1814	7910.5780	7910.4871	7910.4871	7922.0517
	MEB	270.9	210.4	224.6	222.5	256.2		MEB	180.9	157.9	128.4	158.1	165.9
	T(s)	275.5	200.8	307.5	362.8	274.4		T(s)	222.4	286.5	253.8	337.1	267.1
	σ(%)	0.105	0.022	0	0	0.415		σ(%)	0.086	0.002	0.001	0.001	0.147
	N ^{best}	3	5	6	7	2		Gr202 549.99	N ^{best}	10	10	10	10
MTL	6148.9047	6125.0712	6125.1932	6125.0712	6153.3945	MTL	492.2027		490.2243	490.3873	489.3853	492.3238	
MITL	6110.7219	6110.7219	6110.7219	6110.7219	6110.7219	MITL	487.3794		487.2567	487.1769	487.8589	488.4269	
METL	6123.9958	6115.4313	6115.2412	6115.0267	6127.5864	METL	489.3760		488.9684	488.6827	488.5814	489.8160	
MEB	380.8	365.0	269.1	277.4	304.7	MEB	597.9		603.3	516.6	566.7	471.5	
T(s)	528.2	658.3	589.4	521.7	499.4	T(s)	1652.9		2006.8	1769.9	2441.1	1163.5	
σ(%)	0.214	0.074	0.071	0.068	0.274	σ(%)	0		0	0	0	0	

表 3 子种群数对 HCIA 性能的影响

Problem S ₀	PS	m' = 40 * 6	m' = 40 * 10	m' = 40 * 14	MLOICSA	Problem S ₀	PS	m' = 40 * 6	m' = 40 * 10	m' = 40 * 14	MLOICSA
	$\frac{m'}{m_0 * NV}$						$\frac{m'}{m_0 * NV}$				
Chn144 30354.3	N ^{best}	0	0	0	0	Ch150 6532.3	N ^{best}	0	0	1	0
	MTL	30625.1014	30641.9261	30492.2396	30946.0536		MTL	6594.8395	6581.5312	6573.1734	6619.6180
	MITL	30383.0643	30382.1477	30382.1477	30383.0643		MITL	6548.7568	6548.7568	6530.9025	6549.8940
	METL	30464.5389	30436.7360	30416.3047	30604.0768		METL	6563.7353	6559.7852	6556.2890	6566.4061
	MEB	486.9	454.2	521.2	467.5		MEB	514.9	464.7	445.2	434.2
	T(s)	727.0	1281.5	1730.1	967		T(s)	819.1	1170.2	1702.6	1119.4
	σ(%)	0.363	0.272	0.204	0.82		σ(%)	0.481	0.421	0.369	0.522
	N ^{best}	0	0	0	0		A280 2586.8	N ^{best}	1	1	1
MTL	15866.1757	15843.4044	15850.0408	15883.9919	MTL	2621.2202		2616.7534	2604.4792	2662.7500	
MITL	15818.6977	15818.4734	15809.4258	15809.4258	MITL	2586.7696		2586.7696	2586.7696	2586.7696	
METL	15841.8382	15827.8256	15827.7800	15842.4325	METL	2603.2841		2594.8484	2594.3436	2609.581	
MEB	573.4	572.3	570.1	565.5	MEB	640.6		696.3	686.6	781.2	
T(s)	1278.1	2110.3	2900.9	1362.7	T(s)	2421.1		3654.9	5927.5	3856.8	
σ(%)	0.210	0.121	0.121	0.214	σ(%)	0.637		0.311	0.292	0.88	

参考文献:

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

[2] 郭一楠, 王辉, 程健. 自适应免疫克隆选择文化算法[J]. 电子学报, 2010, 38(4): 966 - 972.

Guo Yinan, Wang Hui, Cheng Jian. Adaptive immune clonal selection cultural algorithm[J]. Acta Electronica Sinica, 2010, 38(4): 966 - 972. (in Chinese)

[3] 戚玉涛, 焦李成, 刘芳. 基于并行人工免疫算法的大规模 TSP 问题求解[J]. 电子学报, 2008, 36(8): 1552 - 1557.

Qi Yutao, Jiao Licheng, Liu Fang. Parallel artificial immune algorithm for large-scale TSP[J]. Acta Electronica Sinica, 2008,

36(8): 1552 - 1557. (in Chinese)

[4] 戚玉涛, 刘芳, 焦李成. 求解大规模 TSP 问题的自适应归约免疫算法[J]. 软件学报, 2008, 19(6): 1265 - 1273.

Qi Yutao, Liu Fang, Jiao Licheng. Immune algorithm with self adaptive reduction for large-scale TSP[J]. Journal of Software, 2008, 19(6): 1265 - 1273. (in Chinese)

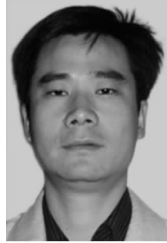
[5] M A Potter, K A De Jong. A cooperative coevolutionary approach to function optimization [A]. Proceedings of the 3rd Conference on Parallel Problem Solving from Nature [C]. Berlin, Heidelberg: Springer-Verlag. 1994. 249 - 257.

[6] 王本年, 高阳, 谢俊元, 等. 基于生态种群捕获竞争模型的进化遗传算法[J]. 计算机应用与软件, 2005, 22(7): 20 - 21.

Wang Bennian, Gao Yang, Xie Junyuan, et al. An evolutionary

- genetic algorithms based on ecological population prey-competition model[J]. Computer application and software, 2005, 22(7):20-21. (in Chinese)
- [7] H D Nguyen, I Yoshihara, K Yamamori, et al. Implementation of an effective hybrid GA for large-scale Traveling Salesman Problems[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2007, 37(1):92-99.
- [8] Z Michalewicz, et al. How to Solve It-Modern Heuristics[M]. Berlin: Springer-Verlag, 2000.
- [9] 杜海峰, 焦李成, 刘若辰. 免疫优势克隆算法[J]. 电子与信息学报, 2004, 26(12):1918-1924.
Du Haifeng, Jiao Licheng, Liu Ruochen. Immunodominance clone algorithms[J]. Journal of electronics & information technology, 2004, 26(12):1918-1924. (in Chinese)
- [10] 吴建辉, 章兢, 张小刚, 等. 一种求解 TSP 问题的分层免疫算法[J]. 计算机科学, 2010, 37(6):256-260.
Wu Jianhui, Zhang Jing, Zhang Xiaogang, et al. Novel Hierarchical immune algorithm for TSP solution[J]. Computer Science, 2010, 37(6):256-260. (in Chinese)
- [11] S Kiranyaz, T Ince, A Yildirim, et al. Fractional particle swarm optimization in multidimensional search space [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2010, 40(2):298-319.
- [12] 黄岚, 王康平, 周春光, 等. 粒子群优化算法求解旅行商问题[J]. 吉林大学学报(理学版), 2003, 41(4):477-480.
Huang Lan, Wang Kangping, Zhou Chunguang, et al. Particle swarm optimization for traveling salesman problems[J]. Journal of Jilin University (Science edition), 2003, 41(4):477-480. (in Chinese)
- [13] 陈贵敏, 贾建援, 韩琪. 粒子群优化算法的惯性权值递减策略研究[J]. 西安交通大学学报, 2006, 40(1):53-56.
Chen Guimin, Jia Jianyuan, Han Qi. Study on the strategy of decreasing inertia weight in particle swarm optimization algorithm[J]. Journal of Xi'an Jiaotong University, 2006, 40(1):53-56. (in Chinese)
- [14] 焦李成, 杜海峰, 刘芳, 等. 免疫优化计算、学习与识别[M]. 北京: 科学出版社, 2006. 93-104, 133-143.

作者简介:



吴建辉 男, 1970 年出生于湖南益阳, 湖南大学信息科学与工程学院讲师, 在职博士生. 主要研究方向为自然计算, 人工免疫系统.
E-mail: wujianhui123@tom.com



章 兢 男, 1957 年出生于湖南湘潭, 湖南大学电气与信息工程学院教授, 博士生导师. 主要研究方向为智能控制, 复杂系统工业控制, 自然计算.

张小刚 男, 1972 年出生于河南郑州, 湖南大学电气与信息工程学院教授, 博士. 主要研究方向为智能控制, 数据挖掘.