

一种求解 TSP 问题的分层免疫算法

吴建辉^{1,2} 章 兢¹ 张小刚¹ 刘朝华¹

(湖南大学电气与信息工程学院 长沙 410082)¹ (湖南大学计算机与通信学院 长沙 410082)²

摘要 为提高人工免疫算法求解旅行商问题的效率,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分层局部最优免疫优势克隆选择算法(HLOICSA)。通过对多个子种群进行低层免疫操作——局部最优免疫优势、克隆选择、基于信息熵的抗体多样性改善和高层遗传操作——选择、交叉、变异,增强优秀抗体实现亲和力成熟的机会,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 的实验结果表明,该算法具有可靠的全局收敛性及较快的收敛速度。

关键词 人工免疫算法,旅行商问题,分层,局部最优免疫优势,克隆选择

中图分类号 TP183 **文献标识码** A

Novel Hierarchical Immune Algorithm for TSP Solution

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

(College of Electrical and Information Engineering, Hunan University, Changsha 410082, China)¹

(College of Computer and Communication, Hunan University, Changsha 410082, China)²

Abstract In order to solve traveling salesman problem more efficiently using artificial immune algorithm, a two-floor model based on multiple sub-populations immune evolution as well as hierarchical local optimization immunodominance clonal selection algorithm(HLOICSA) was put forward. To quickly obtain the global optimum, multiple sub-populations were operated by bottom floor immune operators; local optimization immunodominance, clonal selection, antibody diversity amelioration based on locus information entropy, multiple sub-populations were also operated by top floor genetic operators; selection, crossover, mutation. Through those operators, diversity of antibody sub-population distribution and excellent antibody affinity maturation was enhanced, the balance between in the depth and breadth of the search-optimizing was acquired. Experimental results indicate that the algorithm has a remarkable quality of the global convergence reliability and convergence velocity.

Keywords Artificial immune algorithm, Traveling salesman problem, Hierarchical, Local optimization immunodominance, Clonal selection

TSP 问题可以简单描述为:已知 N 个城市 $V = \{v_1, v_2, \dots, v_N\}$ 以及任意两城市之间的距离 $d(v_i, v_j)$, 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。TSP 问题是著名的 NP 难问题,在电路板布局、VLSI 芯片设计、车辆调度等组合优化问题中有着广泛的应用^[1]。遗传算法(Genetic Algorithm, GA)是模拟自然界的物种进化和自然选择遗传机制形成的一种并行搜索优化智能启发式算法,较以往传统的搜索算法具有简单、鲁棒性强、便于并行处理等特点,因而广泛应用于解决搜索和优化的问题^[2]。但基本遗传算法不能很好地维持解种群中个体的多样性,因此存在易趋于“早熟”收敛而陷入局部最优解、收敛速度慢等缺陷。免疫克隆选择算法(Clonal Selection Algorithm, CSA)^[3]作为人工免疫算法^[4]

的核心算法之一,在组合优化、网络安全以及控制等领域得到了广泛的应用。CSA 算法在运算中后期种群亲和度较高时具有快速随机的全局搜索能力,既能保证收敛速度,又能够维持抗体的多样性;但在运算初期对于系统中的反馈信息利用不足,往往做大量无谓的冗余迭代,求解效率较低。王磊等人在遗传算法中引入免疫算子,构造了一类免疫遗传算法(IGA),并用于求解 TSP 问题^[5]。文献[6]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。

为进一步提高人工免疫算法求解旅行商问题的效率,本文利用上述遗传算法与人工免疫算法各自的特性以及二者的共性,即相似的算法结构和相同的并行性,构造了一种基于多子种群免疫进化的分层框架模型,在此模型的基础上提出了分层局部最优免疫优势克隆选择算法(HLOICSA)。通过对多

到稿日期:2009-07-09 返修日期:2009-09-21 本文受国家自然科学基金重点项目(60634020),国家自然科学基金项目(60874096)资助。
吴建辉(1970—),男,博士生,讲师,主要研究方向为自然计算、人工免疫系统, E-mail: wujianhui123@tom.com; 章 兢(1957—),男,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、自然计算等; 张小刚(1972—),男,博士,教授,主要研究方向为智能控制、数据挖掘; 刘朝华(1983—),男,博士生,主要研究方向为嵌入式系统、人工免疫系统。

个子种群进行低层免疫操作和高层遗传操作,能增强优秀抗体实现亲和力成熟的机会,加快收敛速度,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 的实验结果表明,该算法具有可靠的全局收敛性及较快的收敛速度。

1 HLOICSA 算法

1.1 基本概念描述

1.1.1 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,将当前抗体群中的路径长度最短的抗体看作抗原。

1.1.2 亲和度函数定义

抗体与抗原之间的亲和度反映抗体与抗原的匹配程度。常见的亲和度定义方法是取抗体所对应的路径长度的倒数,其缺点是由于抗体对应的路径长度通常为较大的正数,致使亲和度通常变得很小,且密集分布于一个狭窄区间内,不利于体现出抗体的优劣,因此定义亲和度函数为

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

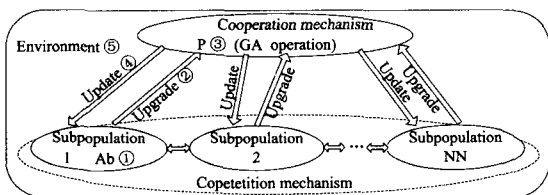
式中, $dist(a_i)$ 为抗体子种群中第 i 个抗体的路径长度, S_0 为相应 TSP 实例的已知最短路径值。

1.2 HLOICSA 算法

1.2.1 TMMPIE 模型

借鉴分层、多种群的概念构造了基于多种群免疫进化的两层框架模型(Two-floor Model based on Multiple Populations Immune Evolution, TMMPIE)。

定义 1(TMMPIE 模型) TMMPIE 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。 $A = \{Ab_1, Ab_2, \dots, Ab_{NN}\}$, A 中有 NN 个抗体子种群,每个抗体子种群有 m 个抗体。这 NN 个子种群最好在设置特性上有较大的差异,在种群规模不扩大的情况下,为将来的高层遗传算法产生更多种类的优良模式。把群体分割成若干个子种群能有效维持群体中个体的多样性,这样可使由于出现不适当个体而产生早熟的现象局部化,从而达到抑制早熟现象的目的。提升操作是通过轮盘赌选择策略将低层 NN 个子种群提升至高层,组成高层抗体子种群 P 。高层操作采用基于协同合作的遗传算法。更新操作是用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高层子操作群 P ④更新操作 ⑤环境

图 1 基于多种群免疫进化的两层框架模型

1.2.2 HLOICSA 算法流程

在 TMMPIE 模型的基础上提出了 HLOICSA 算法,其流程如下:

Step1 初始化 NN 个抗体子种群,计算 NN 个抗体子种群中各

抗体对应的路径长度,获得抗体群的最短路径长度值;

Step2

While(算法终止条件不满足)

For(每个抗体子种群 Ab) //低层 LOICSA 操作

1)选择 Ab 中 K 个高亲和度抗体进行局部最优免疫优势操作;

2)重新计算子种群中抗体亲和度,再选择子种群中 K 个高亲和度抗体进行克隆扩增;

3)动态高频变异;

4)免疫选择,并计算子种群中抗体对应的路径长度,获得子种群的最短路径长度值;

5)基于信息熵的抗体多样性改善;

End For

If 迭代代数 I_{th} 代的整数倍 //基于协同合作的高层遗传操作

1)根据轮盘赌选择策略选择低层子种群进入高层;

2)根据单点交叉方法进行子种群交叉;

3)根据基本位变异方法进行子种群变异;

4)转到 Step2,并更新相应的低层子种群;

Else

转到 Step2;

End

End While

Step3 输出种群最优抗体及其路径长度值。

1.3 HLOICSA 算法低层操作

低层操作是对每个子种群采用局部最优免疫优势克隆选择算法(Local Optimization Immunodominance Clonal Selection Algorithm, LOICSA),各操作算子如下。

1.3.1 局部最优免疫优势(Local Optimization Immunodominance, LOD)算子

免疫学认为,虽然一个抗原分子上可以有多个表位,但在诱导宿主免疫应答时只有一种或一个表位起主要作用,使宿主产生以该特异性为主的免疫应答,这种现象称为免疫优势(Immunodominance)或免疫显性,起关键作用的表位称为显性表位。用于 AIS 的免疫优势概念,包括抗原免疫优势和抗体免疫优势^[9]。对于 TSP 这样的非函数优化问题,抗原免疫优势需对目标函数进行变换,但这是很困难的。而且对于复杂问题,往往很难获得有益的先验知识,所以针对 TSP 问题采用抗体免疫优势。由于确定抗体免疫优势位的代价太大,因此分析每次迭代中的最优抗体(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。

(1)针对 TSP 问题的抗体 LOI 算子原理

定义 2(抗体 LOI 算子) 定义 $ID = \{(\Delta d_i, v_i, v_j, v_{i+1})\}$ 为抗体 LOI 算子,其中 $\Delta d = \min(d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1}))$, $v_i, v_{i+1} \in e_1, v_j \in e_2$ 。

定理 1 设抗体 a_p (a_p 的节点数为 N), $a_p \in Ab_k$ 。设有 n ($2 < n < N$) 个节点(城市号)的集合 $e_1 = (v_1, v_2, \dots, v_i, v_{i+1}, \dots, v_n) \subset a_p$, 设待插入节点 $v_j \in e_2 \subset a_p$, e_1, e_2 是抗体 a_p 的两个子集, $e_1 \cap e_2 = \Phi$, Φ 为空集。若有 v_j 插入到 v_i 和 v_{i+1} 之间,使路径长度 $dist_i = d(v_1, v_2) + \dots + d(v_{i-1}, v_i) + d(v_i, v_j) + d(v_j, v_{i+1}) + \dots + d(v_{n-1}, v_n)$ 最短,那么就等价于 $\Delta d_i = d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1})$ 最短。

证明:将 v_j 插入到 v_i ($i \leq n$) 之后,路径为 $\cdot v_1 - \cdot v_2 \dots \cdot v_i - \cdot v_j - \cdot v_{i+1} \dots \cdot v_k - \cdot v_{k+1} \dots \cdot v_{n-1} - \cdot v_n$, 若路径长度 $dist_i = d(v_1, v_2) + \dots + d(v_i, v_j) + d(v_j, v_{i+1}) + \dots + d(v_k,$

$v_{k+1}) + \dots + d(v_{n-1}, v_n)$ 最短,将 v_j 插入到任一节点 $v_k (k \leq n)$ 之后路径为 $\cdot v_1 - \cdot v_2 \dots \cdot v_i - \cdot v_{i+1} \dots \cdot v_k - \cdot v_j - \cdot v_{k+1} \dots \cdot v_{n-1} - \cdot v_n$, 路径长度 $dist_k = d(v_1, v_2) + \dots + d(v_i, v_{i+1}) + \dots + d(v_k, v_j) + d(v_j, v_{k+1}) + \dots + d(v_{n-1}, v_n)$, 那么有 $dist_i < dist_k$, 即 $dist_i - dist_k < 0$, 那么 $(d(v_1, v_2) + \dots + d(v_i, v_j) + d(v_j, v_{i+1}) + \dots + d(v_k, v_{k+1}) + \dots + d(v_{n-1}, v_n)) - (d(v_1, v_2) + \dots + d(v_i, v_{i+1}) + \dots + d(v_k, v_j) + d(v_j, v_{k+1}) + \dots + d(v_{n-1}, v_n)) < 0$ 可得 $\Delta d_i = (d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1})) < \Delta d_k = (d(v_k, v_j) + d(v_j, v_{k+1}) - d(v_k, v_{k+1}))$ 。反之, 如果有 $\Delta d_i = (d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1})) < \Delta d_k = (d(v_k, v_j) + d(v_j, v_{k+1}) - d(v_k, v_{k+1}))$, 则有 $dist_i < dist_k$ 。

$\Delta d = \min(d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1}))$ 对应的 v_i, v_j, v_{i+1} 就是抗体中的免疫优势位。本算法能保证待插入节点 v_j 插入的位置是 e_1 中最优的位置, 使得路径长度 $dist_i$ 最短。这样, 只通过计算 v_i, v_j, v_{i+1} 之间的路径长度 $\Delta d_i = d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1})$, 就可以避免计算复杂的 $e_1 = (v_1, v_2, \dots, v_i, v_j, v_{i+1}, \dots, v_n)$ 的路径长度 $dist_i$, 大大减少了计算量。

(2) LOI 算子构造

Begin:

while($e_2 \neq \Phi$) do

{ 求抗体免疫优势 ID = $\{(\Delta d, v_i, v_j, v_{i+1})\}$, 其中 $\Delta d = \min(d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1}))$, $v_i, v_{i+1} \in e_1, v_j \in e_2$;

将 v_j 插入 v_i 之后;

从 e_2 中移出 v_j ;

}

LOI 算子的另一个特点是, 即使是两个完全相同的最优抗体, 经过 LOI 操作后, 仍可以得到新的抗体。显然, 针对 TSP 问题的抗体 LOI 算子, 是针对特定抗体的一个动态调整局部最优过程, 其操作是通过两个抗体的相互作用, 并依据抗体免疫优势确定节点相对位置, 以提高这两个抗体的亲密度。

1.3.2 改进的克隆扩增算子 β

在每一次克隆扩增之前, 将抗体子种群中的抗体依据抗体与抗原间的亲密度大小按照降序来排序, 并根据排序结果将抗体子种群分为两部分: $Ab_H = [a_1', a_2', \dots, a_H']$, $Ab_L = [a_{H+1}', a_{H+2}', \dots, a_{H+L}']$ 。 $H = L = m/2$, 其中 a_1' 的亲密度最大, a_{H+L}' 的亲密度最小。为了减少计算量, 只选择 Ab_H 中抗体进行克隆, 按照下式计算克隆扩增算子 β :

$$\beta = \text{round}(\alpha \cdot \frac{\text{affinity}(a_i)}{i} + b) \quad (2)$$

式中, α 为大于 1 的比例系数, $\text{round}(\cdot)$ 为取整, b 为大于 1 的常量, i 为抗体 a_i 所对应的位置。当 a_i 亲密度最高时, $i = 1$ 。从式(2)可知, 抗体的克隆倍数与亲密度大小成正比, 与降序排序的位置成反比。同时, 为了保证每个抗体都有一定克隆数量, 加上了常量 b 。

1.3.3 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:

$$p = p \times (1 - \frac{It}{It_m} \times 0.01) \quad (3)$$

式中, p 为变异概率, It 为当前进化代数, It_m 为截止代数。

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^[1] 优于传统的交叉算子, 如 PMX, OX, CX 等。

1.3.4 免疫选择

$S_i = T_i(a_i) = \max\{\text{affinity}(a_i')\}$, a_i' 表示抗体 a_i 经克隆扩增、变异后得到的抗体, 然后 $S_i \cup a_i \rightarrow a_i^{\text{new}}$ 。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。

1.3.5 基于信息熵的抗体多样性改善

假设抗体子种群由 m 个抗体组成, 每个抗体有 N 位基因, 采用的编码符号集大小为 $|S|$, $S = \{S_1, S_2, \dots, S_{|S|}\}$, 则对应于基因座 j 的信息熵定义为

$$H_j(m) = -\sum_{i=1}^{|S|} P_{ij} \log_2(P_{ij}) \quad (4)$$

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

$$H = \frac{1}{N} \sum_{j=1}^N H_j(m) \quad (5)$$

子种群平均信息熵越大, 种群的分布性越好。随着进化代数 It 的增加, 子种群亲和度的不断提高, 种群的多样性会逐渐降低。

1) 多样性判断: 定义每代的多样性阈值

$$\phi(It) = A \cdot \log_2(|K|) \cdot e^{-\frac{B \cdot It}{It_m}} \quad (6)$$

式中, A 为调节系数, $A \in (0, 1)$; B 为大于 0 的加速因子, It 为当前进化代数; It_m 为截止代数; K 为常数。为了防止系统陷入局部最优, 在进化过程中, 加入多样性判断。若子种群的多样性过早地低于阈值, 则需要多样性改善, 以提高子种群的多样性。

2) 多样性改善: 在子种群中 $Nr (Nr > 1)$ 个路径长度相同抗体(一般为最优抗体或次优抗体)中选择 $Nr - 1$ 个抗体, 对每一个选择抗体进行低多样性基因位的倒位操作, 直到子种群的多样性超过预定阈值。

1.4 HLOICSA 算法高层操作

高层操作采用基于协同合作的遗传算法。协同合作是通过子种群的抗体交换实现, 各操作算子如下。

1.4.1 选择算子

定义 3(子种群的适应度函数) 子种群的适应度函数定义为

$$\text{Fit}(Ab_k) = (l_1 \cdot \frac{\frac{1}{NN} \sum_{i=1}^{NN} \text{MITL}(Ab_i)}{\text{MITL}(Ab_k)} + (1 - l_1) \cdot \frac{\text{maff}(Ab_k)}{\frac{1}{NN} \sum_{i=1}^{NN} \text{maff}(Ab_i)}) \quad (7)$$

式中, $\text{Fit}(Ab_k)$ 为第 k 个子种群 Ab_k 的适应度函数, $\text{MITL}(Ab_k)$ 为 Ab_k 中最优抗体所对应的最短路径值(MITL, Minimal value of Tour Length), $\text{maff}(Ab_k)$ 为 Ab_k 中所有抗体亲和度的平均值, NN 为子种群个数。 l_1 为权重系数, l_1 愈大, 子种群的 MITL 值的影响愈大; 反之, l_1 愈小, 子种群的抗体亲密度均值的影响愈大。 l_1 的自适应变化公式如下。

$$l_1 = l_1 \times (1 - \frac{It}{It_m} \times 0.01) \quad (8)$$

在进化初期 l_1 取值较大, 突出子种群的 MITL 值的影响。随着进化代数 It 的不断进行, 各个子种群的 MITL 值相

差不多。不断加大子种群亲和度均值的权重,会使亲和度均值比较高的子种群有更多的选择机会。根据子种群适应度函数运用轮盘赌选择算子进行选择,适应度值高的子种群被复制,甚至于多次被复制,而适应度值低的子种群可能会被淘汰。当 NN 个子种群中具有最小 $MITL$ 值的子种群没有被选择时,将这个子种群选中并替换选中的任一子种群。

1.4.2 交叉算子

按照较大的高层交叉概率 P_{ch} 从种群中选择两个子种群。采用单点交叉(简单交叉)方法,在两个子种群中随机设置一个交叉点,然后将两个配对子种群在该点后的部分抗体相互交换,通过子种群的抗体交换达到信息交流、协同进化的目的。

1.4.3 变异算子

依照高层变异概率 P_{m} 确定变异抗体。采用基本位变异方法,将要变异的抗体,用有 N 个城市随机排列的新抗体来代替,从而改善抗体子种群的多样性。高层操作经过交叉和变异后,采用了最优保存策略,它可以防止算法运行中抗体出现退化,减缓收敛速度。

NN 个低层子种群经过一定代数的低层 LOICSA 操作后,均可获得位于个体串上的一些特定位置的优良模式。通过基于协同合作高层遗传算法的操作,新子种群可以获得包含不同种类的优良模式的新抗体,从而为它们提供了更加平等的竞争机会。

2 HLOICSA 算法收敛性分析

文献[7]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。同样, HLOICSA 算法操作过程状态变化均在有限空间中进行,只是种群中抗体是离散实数。 $A(k+1) = T(A(k)) = T_s \circ T_m \circ T_c \circ T_d \circ T_{hs} \circ T_{hcr} \circ T_{hm} \circ \Theta(A(k))$, 其中 $\Theta, T_s, T_m, T_c, T_d, T_{hs}, T_{hcr}, T_{hm}$ 分别代表基于信息熵的抗体多样性改善、免疫选择、变异、克隆、局部最优免疫优势、高层遗传选择、高层遗传交叉、高层遗传变异。 $\Theta, T_s, T_m, T_c, T_d, T_{hs}, T_{hcr}, T_{hm}$ 均与 n 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。上述算法初始种群规模为 $m' = m * NN$, 通过 HLOICSA 算法的相关算子操作后,得到优秀抗体群 m_0' 。可将种群中的全部近似解看成是状态空间 $S := X^{m_0'}$ 中的一个点, $s_i \in S$ 表示 S 中的第 i 个状态。用 s^* 表示由 $f(x)$ ($f(x)$ 是 X 上的适应度函数)在 S 上取最优值的所有抗体组成, $s^* = \{x \in X | f(x) = \max_{x_i \in X} f(x_i)\}$, 则有

定理 2 对于任意的初始分布均有

$$\lim_{k \rightarrow \infty} \sum_{s_i \in S} P\{A_k^i\} = 1 \quad (9)$$

该式表明,当算法迭代到足够多的次数后,群体中包含全局最佳个体的概率接近于 1,则称算法收敛。

证明:记 $I = \{i | s_i \cap X^* \neq \emptyset\}$, 设随机过程 $\{A_k\}$ 的转移概率为 $p_{ij}(k) = p\{\frac{A_{k+1}^i}{A_k^j}\}$, 讨论 $p_{ij}(k)$ 的两种特殊情况:

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,以保证抗体不会变差,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 $p_k^i, p_k = \sum_{i \in I} p_k^i$, 由马尔可夫链性质可知 $p_{k+1} = \sum_{s_i \in S} \sum_{j \in I} p_i(k) p_{ij}(k) = \sum_{i \in I} \sum_{j \in I} p_i(k) p_{ij}(k) + \sum_{i \notin I} \sum_{j \in I} p_i(k) p_{ij}(k)$ 。因为 $\sum_{i \notin I} \sum_{j \in I} p_i(k) p_{ij}(k) + \sum_{i \in I} \sum_{j \in I} p_i(k) p_{ij}(k) = \sum_{i \in I} p_{i(k)} = p_k$, 所以 $\sum_{i \notin I} \sum_{j \in I} p_i(k) p_{ij}(k) = p_k - \sum_{i \in I} \sum_{j \in I} p_i(k) p_{ij}(k)$, 则 $0 \leq p_{k+1} \leq p_k - \sum_{i \in I} \sum_{j \in I} p_i(k) p_{ij}(k) \leq p_k \leq 1$, 于是有 $\lim_{k \rightarrow \infty} p_k = 0$, 则 $1 \geq \lim_{k \rightarrow \infty} \sum_{s_i \in S} p_i(k) \geq \lim_{k \rightarrow \infty} \sum_{i \in I} p_i(k) = 1 - \lim_{k \rightarrow \infty} p_k = 1$ 。因此可证明式(9)以概率 1 收敛。

3 仿真实验

3.1 算法比较

为了更好地说明本算法的有效性,选用国际上通用的 TSPLIB 测试库中的多个实例进行测试。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 NGA, IDIA, LOICSA, HLOICSA 4 种算法针对 Ch130(城市数为 130)进行 10 次测试,具体数据如表 1 所列。截止代数设为城市数的 3 倍, HLOICSA 算法的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_a , 变异初始概率 $P_m = 0.75$, 抗体种群规模为 $m' = 30 * 10$ (子种群有 30 个抗体,共有 10 个子种群), 克隆扩增算子比例系数 $\alpha = 300$, 克隆算子常数 $b = 20$, 高层交叉概率 $P_{ch} = 0.8$, 高层变异概率 $P_{m} = 0.06$ 。LOICSA 算法是只采用一个抗体种群并且无高层遗传算法操作, 抗体数为 130(城市数目), 其他参数同 HLOICSA。NGA 为基于交叉算子效果较好的 OX 法的遗传算法[8], IDIA 为基于优势抗体的免疫算法[7]。图 2 是这 4 种算法针对 Ch130 的收敛过程对比图。从图 2 可知, HLOICSA 的收敛速度及最优解均优于其它 3 种算法,其原因在于 LOI 算子及多子种群的应用。

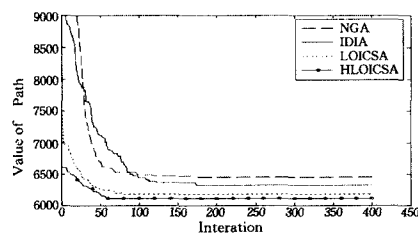


图 2 4 种算法针对 Ch130 的收敛过程对比

表 1 到表 3 为 10 次独立实验的统计结果。表 1 为 NGA, IDIA, LOICSA, HLOICSA 4 种算法针对 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的次数 N^{best} 、最长路径值 MTL (Maximal value of Tour Length)、最短路径值 $MITL$ 、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 $\sigma = \frac{\sum_{i=1}^{10} (S_{T_i} - S_0)}{10S_0} \times 100\%$ (其中 S_{T_i} 是第 i 次的最短路径, S_0 是已知最短路径)。NGA 与 IDIA 的求解数据来自文献[7], 文献中无 MEI 数据, 用“—”表示。表 2 中带 * 号表示 HLOICSA 算法的抗体种群规模为 $m' = 30 * 10$, 不带 * 号表示抗体种群规模为 $m' = 40 * 10$ 。LOICSA 算法的抗体数取相应城市数目, 其他参数同 HLOICSA。

从表 1 可知, 1) 在 IDIA 和 LOICSA 两个单种群算法中,

基于 LOI 算子的 LOICSA 算法性能 (N^{best} , MTL , $MITL$, $METL$, $\sigma(\%)$) 优于 IDIA 算法, 从而验证了 LOI 算子的有效性。2) 对于表 2 中绝大多数的 TSP 实例, 基于多种群的 HLOICSA 算法性能优于或等于基于单种群的 NGA, IDIA,

LOICSA 算法, 并且 HLOICSA 算法的 MEI 值均小于 LOICSA 的 MEI 值, 从而验证了 HLOICSA 算法的有效性。3) 实验也发现随着城市数目增加, 算法性能会有所下降, 尤其对于 Pcb442 问题算法性能下降较多。

表 1 TSP 实例测试

Problem So	Method	NGA	IDIA	LOICSA	HLOICSA	Problem So	Method	NGA	IDIA	LOICSA	HLOICSA	Problem So	Method	NGA	IDIA	LOICSA	HLOICSA
att48	N^{best}	0	1	6	9	* Eil76	N^{best}	0	2	1	2	* Rd100	N^{best}	0	.1	1	7
	MTL	35594	34076	33701.2618	33588.3434		MTL	589.30	564.61	553.1175	550.7486		MTL	8723.1	8182.0	7944.3219	7959.0996
	MITL	34228	33524	33523.7085	33523.7085		MITL	568.89	544.37	544.3691	544.3691		MITL	8291.2	7910.4	7910.3962	7910.3962
	METL	34584	33666	33562.7882	33530.1720		METL	577.16	552.87	549.7043	547.3938		METL	8449.9	8021.4	7931.0242	7922.0517
	MEI	-	-	144	39.3		MEI	-	-	228	256.2		MEI	-	-	400	165.9
33524	$\sigma(\%)$	3.16	0.42	0.12	0.019	545.32	$\sigma(\%)$	5.84	1.38	0.8	0.415	7910.4	$\sigma(\%)$	6.82	1.40	0.26	0.147
	N^{best}	0	2	1	7		N^{best}	0	0	0	2		N^{best}	0	0	0	0
	MTL	717.6	673.0	654.6519	646.8819		MTL	6724.9	6318.3	6231.7526	6153.3945		MTL	4377.7	4144.2	3997.4192	3938.9447
	MITL	685.0	641.2	642.2416	640.2116		MITL	6364.8	6153.7	6112.8669	6110.7219		MITL	4169.4	3898.4	3925.6913	3878.3564
	METL	695.5	654.1	647.0273	642.4802		METL	6570.1	6218.8	6170.1797	6127.5864		METL	4272.0	4020.5	3954.3721	3915.6573
* Eil101	MEI	-	-	404	306.6	* Ch130	MEI	-	-	390	304.7	3859	MEI	-	-	675.0	515.1
	$\sigma(\%)$	8.28	1.84	0.736	0.107		$\sigma(\%)$	7.51	1.77	0.97	0.274		$\sigma(\%)$	10.70	4.18	2.471	1.468
	N^{best}	0	0	0	0		N^{best}	10	10	10	10		N^{best}	10	10	10	10
	MTL	5763.34	5468.83	5276.6	5213.9856		MTL	16887.78	16790.83	15638.1011	15559.2922		MTL	3463.0	3290.6	3199.6368	3168.9853
	MITL	5657.77	5184.49	5118.0	5115.6345		MITL	16558.14	15739.69	15445.5290	15441.8438		MITL	3321.1	3200.6	3135.6174	3128.6477
642.3	METL	5705.79	5278.40	5193.0	5170.3142	19330.8	METL	16674.70	16146.23	15521.4352	15485.1120	3952.54	METL	3416.7	3275.3	3176.0864	3149.2949
	MEI	-	-	1326	688.1		MEI	-	-	1683	1420		MEI	-	-	1998	1881.6
	$\sigma(\%)$	12.36	3.94	2.26	1.811		$\sigma(\%)$	0	0	0	0		$\sigma(\%)$	0	0	0	0
	N^{best}	0	0	0	0		N^{best}	10	10	10	10		N^{best}	10	10	10	10
	MTL	5763.34	5468.83	5276.6	5213.9856		MTL	16887.78	16790.83	15638.1011	15559.2922		MTL	3463.0	3290.6	3199.6368	3168.9853
Pcb442	MITL	5657.77	5184.49	5118.0	5115.6345	Pa561	MITL	16558.14	15739.69	15445.5290	15441.8438	* Gr666	MITL	3321.1	3200.6	3135.6174	3128.6477
	METL	5705.79	5278.40	5193.0	5170.3142		METL	16674.70	16146.23	15521.4352	15485.1120		METL	3416.7	3275.3	3176.0864	3149.2949
	MEI	-	-	1326	688.1		MEI	-	-	1683	1420		MEI	-	-	1998	1881.6
	$\sigma(\%)$	12.36	3.94	2.26	1.811		$\sigma(\%)$	0	0	0	0		$\sigma(\%)$	0	0	0	0

3.2 抗体种群规模对 HLOICSA 算法性能影响

表 2 为抗体种群规模对 HLOICSA 算法性能影响的分析, 其中种群规模 (Population Size, PS) $m' = m * NN$, 这里 m 为每个子种群的抗体个数, NN 为子种群数。T 为 10 次测试结果平均运行时间, 单位是秒。从表 2 可知, 对于表 2 中的

TSP 实例, 无论子种群规模取何值, HLOICSA 的解均优于或等于 NGA, IDIA, LOICSA。总体而言, 随着子种群规模的增加, 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, 子种群规模取 30 或 40 时较优。

表 2 子种群规模对 HLOICSA 性能的影响

Problem So	PS $m' = m_0 * NN$	$m' = 20 * 10$	$m' = 30 * 10$	$m' = 40 * 10$	$m' = 50 * 10$	Problem So	PS $m' = m_0 * NN$	$m' = 20 * 10$	$m' = 30 * 10$	$m' = 40 * 10$	$m' = 50 * 10$
1	N^{best}	8	9	10	CX	9	N^{best}	2	2	4	
	MTL	682.5748	682.5748	677.1096	682.5748		MTL	6174.0083	6153.3945	6198.6332	6160.4380
	MITL	677.1096	677.1096	677.1096	677.1096		MITL	6110.7219	6110.7219	6110.7219	6110.7219
	METL	678.2027	677.6561	677.1096	677.4584		METL	6138.5834	6127.5864	6135.7011	6135.3635
	MEI	53.98	57	41.4	67.5		MEI	315	304.7	211.5	262.65
St70	T(s)	82.4	68.74	61.34	101.96	* Ch130	T(s)	352.38	499.38	441.77	731.6
	$\sigma(\%)$	0.117	0.059	0	0.029		$\sigma(\%)$	0.454	0.274	0.407	0.401
	N^{best}	10	10	10	10		N^{best}	0	0	1	1
	MTL	492.4333	492.3238	493.4012	491.6062		MTL	2651.0134	2641.2155	2662.7500	2677.6915
	MITL	487.9018	488.4269	488.4501	488.4162		MITL	2596.3825	2599.3292	2586.7696	2586.7696
678.5975	METL	490.4043	489.8160	490.1570	489.6828	A280	METL	2618.0728	2615.8590	2609.581	2612.0077
	MEI	487.5	471.5	442.9	418.5		MEI	593.9	627.1	781.2	617.1
	T(s)	835.24	1163.45	1462.72	1787.82		T(s)	1601	3085.66	3856.8	5413.34
	$\sigma(\%)$	0	0	0	0		$\sigma(\%)$	1.209	1.123	0.88	0.975

3.3 克隆扩增算子 β 中比例系数对 HLOICSA 性能的影响

表 3 为克隆扩增算子 β 中比例系数 α 对 HLOICSA 性能影响的分析。对于表 3 中绝大多数的 TSP 实例, 无论克隆扩增算子中比例系数 α 取何值, HLOICSA 的解均优于或等于

NGA, IDIA, LOICSA。总体而言, 随着比例系数 α 的增大, 算法性能有所提高, 但计算时间增加较大。因此, 比例系数 α 取 300 较优。

表 3 比例系数对 HLOICSA 性能的影响

Problem So	α	100	300	500	Problem So	α	100	300	500	Problem So	α	100	300	500
Att48	N^{best}	6	9	9	St70	N^{best}	9	10	9	* Eil76	N^{best}	2	2	4
	MTL	33588.3434	33588.3434	33588.3434		MTL	682.5748	677.1096	682.5748		MTL	552.2753	550.7486	549.9850
	MITL	33523.7085	33523.7085	33523.7085		MITL	677.1096	677.1096	677.1096		MITL	544.3691	544.3691	544.3691
	METL	33549.5625	33530.1720	33530.1720		METL	677.6561	677.1096	677.6561		METL	548.5774	547.3938	546.7881
	MEI	99.6	39.3	57.5		MEI	78.2	41.4	63.1		MEI	237.6	256.2	218.8
33524	T(s)	66.65	38.92	83.52	678.5975	T(s)	67.44	61.34	102.57	545.32	T(s)	142.98	274.44	304.82

(下转第 264 页)

0.812	0.824	0.846	0.8273	15
0.800	0.811	0.851	0.8207	16
0.802	0.789	0.861	0.8173	17
0.798	0.821	0.852	0.8237	18
0.769	0.809	0.847	0.8083	19
0.773	0.820	0.837	0.8100	20

从表7中可以看到,当 $k \in [11, 20]$ 时,实验结果都可以达到80%以上。因此,可以对 k 取该区域内的任意一个随机数来降低程序计算的复杂度,在简化实验过程的同时又不会降低判定准确率。

综合以上实验分析,当 $\delta_1 = 0.7, \delta_2 = 0.3$ 时, k 在闭区间 $[11, 20]$ 内取任意值,文本的情感分类准确率可以达到比较理想的结果,可以很有效地判别文本的情感倾向。

结束语 本文改进了词语相似度计算方法,提出了一种改进的基于语义理解的文本情感分类方法来判定文本的情感倾向性。通过实验分析,证明了该方法的有效性。

然而,由于没有公认统一的语料,本文的语料都是作者从网站中筛选出来的,语料的代表性还需要进一步研究。同时,在文本预处理之前,人工挑选出了情感强度比较大的句子作为实验语料,因此在文本情感值计算中,没有考虑句子的主客观性以及词语的语义消歧,这将在今后的研究中进一步细化。

参考文献

[1] Hatzivassiloglou V, McKeown K R. Predicting the Semantic O-

rientation of Adjectives [A]//Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the ACL[C]. 1997:174-181

[2] Peter T, Michael L. Measuring Praise and Criticism; Inference of Semantic Orientation from Association [J]. ACM Transactions on Information Systems, 2003, 21(4): 315-346

[3] Peter D T. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews [C]// Proceeding of the Association for Computational Linguistics 40th Anniversary Meeting. New Brunswick, N. J., 2002

[4] Pang Bo, Lee Lillian, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques [C]// Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing. 2002: 79-86

[5] 刘群, 李素建. 基于《知网》的词汇语义相似度的计算 [C]// 第三届汉语词汇语义学研讨会. 台北, 2002

[6] 朱嫣岚, 闵锦, 周雅倩, 等. 基于 HowNet 的词汇语义倾向计算 [J]. 中文信息学报, 2006, 20(1): 14-20

[7] 徐林宏, 林鸿飞, 杨志豪. 基于语义理解的文本倾向性识别机制 [J]. 中文信息学报, 2007, 21(1): 96-100

(上接第 260 页)

$\sigma(\%)$	0.077	0.019	0.019	$\sigma(\%)$	0.059	0	0.059	$\sigma(\%)$	0.632	0.415	0.339			
N _{best}	2	2	4	N _{best}	10	10	10	N _{best}	0	0	1			
MTL	6217.5656	6153.3945	6199.6752	MTL	492.5287	492.3238	491.7668	MTL	2651.0134	2641.2155	2662.7500			
*Ch130	MITL	6110.7219	6110.7219	6110.7219	*Gr202	MITL	487.9470	488.4269	488.9267	A280	MITL	2596.3825	2599.3292	2586.7696
6110.9	METL	6142.9789	6127.5864	6134.9483	549.99	METL	490.1223	489.8160	490.1981	2586.8	METL	2618.0728	2615.8590	2609.581
	MEI	291	304.7	235.4	MEI	426.7	471.5	456.8	MEI	593.9	781.2	781.2		
	T(s)	375.05	499.38	584.06	T(s)	703.39	1163.45	1615.38	T(s)	1601	3085.66	3856.8		
$\sigma(\%)$	0.526	0.274	0.395	$\sigma(\%)$	0	0	0	$\sigma(\%)$	1.209	1.123	0.88			

结束语 本文将分层、局部最优免疫优势等概念应用于 TSP 问题,构造了一种基于抗体的 HLOICSA 算法,提高了人工免疫算法求解旅行商问题的效率。本算法在一定子种群规模情况下,通过对多个子种群进行局部最优免疫优势、克隆选择等操作,避免了抗体群被少数亲和力和最高的抗体占满,增强了优秀抗体实现亲和力和成熟的机会。又通过基于信息熵的抗体多样性改善操作,提高抗体群分布的多样性。低层操作作为高层遗传算法产生更多种类的优良模式,通过高层遗传算法的操作,新子种群可以获得包含不同种类的优良模式的新抗体,从而为它们提供了更加平等的竞争机会。在保持优秀个体进化稳定性的同时,避免单个子种群进化过程中出现过收敛现象,在深度搜索和广度寻优之间取得了平衡。

仿真试验表明,本文提出的 HLOICSA 算法在求解城市规模在 700 个城市以下 (MATLAB 在求解大规模 TSP 问题时存在计算时间过长的缺点,所以没有对 700 个以上城市进行验证)的 TSP 问题时非常有效,与 NGA, IDIA, LOICSA 相比,效率更高,全局搜索能力更强,收敛速度更快。通过子种群规模和克隆扩增算子中比例系数对 HLOICSA 性能影响的分析也表明 HLOICSA 具备良好的全局收敛可靠性以及较快的收敛速度。

参考文献

[1] Michalewicz Z, Fogel D B. How to solve It; Modern Heuristics [M]. Berlin Heidelberg: Springer-Verlag, 2000

[2] Merz P, Freisleben B. Genetic local search for the TSP; New results [C]//Proc. of 1997 IEEE Int Conf on Evolutionary Computation. IEEE Neural Network Council, Evolutionary Programming Society, IEEE, 1997: 159-163

[3] de Castro L N, Von Zuben F J. The Clonal Selection Algorithm with Immune Systems and Their Applications [M]. CA: Morgan Kaufman Publishers, 2000

[4] Dasgupta D, Forrest S. Artificial immune systems in industrial applications [C] // Proc. of the Second Int Conf on IPMM '99. Honolulu, 1999, 1: 257-267

[5] 王磊, 潘进, 焦李成. 免疫规划 [J]. 计算机学报, 2000, 23(8): 806-812

[6] 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

[7] 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, 2007: 93-104, 133-143

[8] 梁艳春, 冯大鹏, 周春光. 遗传算法求解旅行商问题时的基因判断保存 [J]. 系统工程理论与实践, 2000, 4: 7-12