

基于蚁群算法的基因联接学习遗传算法

龚道雄 阮晓钢

(北京工业大学电子信息与控制工程学院, 北京 100022)

E-mail: gongdx@bjut.edu.cn

摘要 论文提出了一种基于蚁群算法的基因联接学习遗传算法。在该算法中遗传算法的种群对应于蚁群,遗传算法的染色体同时是蚁群算法的一只蚂蚁。在每一次进行交叉或突变操作时,算法首先根据蚁群算法的信息素矩阵计算父代个体的基因间联接强度,然后根据该联接强度选择交叉和突变位点。这样可以避免积木块过多地被遗传操作所破坏,减少遗传算法的搜索空间,并指引寻优的方向。联接学习在该算法中是并行进行的,而在 Harik 的算法中是串行进行的;该算法的编码长度不会随着等位基因数量的增加而成倍地增加。文章通过有界难度问题和 TSP 问题的实验研究验证了算法的有效性。

关键词 遗传算法 蚁群算法 联接学习

文章编号 1002-8331-(2004)26-0010-06 文献标识码 A 中图分类号 TP18

An ACO-based Linkage Learning Genetic Algorithm

Gong Daoxiong Ruan Xiaogang

(School of Electronic Information & Control Engineering, Beijing

University of Technology, Beijing 100022)

Abstract: This paper proposes an Ant Colony Optimization(ACO)based Linkage Learning Genetic Algorithm(LLGA).In this approach, every individual(chromosome)of GA is at the same time an ant of ACO. Whenever GA performs the operation of crossover and mutation, the linkage strength of parent chromosome(s) is computed according to the pheromone matrix of ACO and is used to guide the selection of crossover or mutation point(s). The stronger linkage strength of two neighboring genes is, the less probably these genes will be separated by genetic operations. Different from Harik's LLGA, this approach learns the linkage parallel and the chromosome length will not multiple as the number of allele increase. Experiments have validated the scheme.

Keywords: Genetic Algorithm, ant colony optimization, linkage learning

1 引言

基因联接在生物学上表现为同一个染色体上的两个或多个基因之间的关联关系,这些基因离得越近,它们在 DNA 修复或复制过程中被分开的可能性就越小,因而它们共同被遗传给下一代的可能性就越大。遗传算法(Genetic Algorithm, GA)的研究也越来越注意基因联接的重要性。人们在研究中发现,染色体的编码方案对于遗传算法的性能具有很大的影响,一个编码的指导原则就是尽可能地将问题中相关的特征编码在一起^[1]。这是遗传算法应用中的一个难点,因为在大多数的应用中人们缺乏足够的先验知识来指导编码。Holland 提出了紧密联接基因(tightly linked genes)和松散连接基因(loosely linked genes)的概念,并提出了让遗传算法在通过进化得出最优方案的同时通过进化得出最优的基因编码次序的思想^[2]。Harik 于 1997 年首次提出了基因联接学习遗传算法(Linkage Learning Genetic Algorithm, LLGA)^[3-4]。随后有许多的研究者对 Harik 的基因联接学习遗传算法作出了改进和完善^[5-7],比如 Lobo, Deb 等于 1998 年提出了压缩内含子(Compressed Introns)的编码方法以解决 LLGA 的内含子数量随着问题的规模指数增长的问题;

Chen 和 Goldberg 于 2002 年为 LLGA 引入起始表达基因(Start Expression Genes)和改进的交叉算子,提出了 LLGA(se)以改善算法求解均匀尺度问题(Uniformly scaled problems)的性能。尽管这些算法都分别在不同的程度上取得了成功,该领域仍然有待于作进一步的研究和探索。比如,LLGA 对于积木块的学习是串行进行的,当积木块的数量较多的时候必然导致算法的计算效率降低;LLGA 的编码方式要求每一位编码的所有等位基因至少出现一次,并且要求有大量的内含子存在,极大地加大了编码串的长度,尤其是在等位基因数量较多的时候问题将变得更加突出,从而成倍地扩大了遗传算法的搜索空间,增大了遗传算法的寻优难度。

蚁群算法(Ant Colony Optimization, ACO)自 1991 年由 Dorigo 等提出以来在多方面获得了成功的应用^[8-12]。蚁群算法具有与遗传算法相似的迭代搜索过程。在每一次迭代(相当于遗传算法的进化代)中,每一只人工蚂蚁均产生一个关于所优化问题的解,该解的形成过程被描述为在一个决策图中寻找一条路径的过程。所有蚂蚁在解搜索过程中所获得的信息全部存储在信息素矩阵中,信息素矩阵中元素 τ_{ij} 的大小指示了选择

基金项目:国家自然科学基金重点项目资助(编号:60234020)

作者简介:龚道雄(1968-),男,博士,讲师,感兴趣的方向为智能优化计算、进化计算及其在生物信息学中的应用。阮晓钢(1958-),男,教授,博士生导师,研究领域包括:自动控制、模式识别、神经计算学、人工智能、智能信息处理。

决策图中的一条边 l_i 的优劣程度。蚂蚁在搜索过程中采用了一种概率决策机制,而每一个可能解被采纳的概率又与信息素矩阵中对应于决策图中相应边的信息素浓度相关。这样,信息素矩阵就反过来又成为指导以后蚂蚁进行解空间搜索的依据,从而使得算法在一种正反馈过程中逐步逼近全局最优解。

针对前述 LLGA 所存在的两个缺点,论文提出了一种新的基于蚁群算法^[12,13]的基因联接学习遗传算法(ACO-LLGA)。在该算法中遗传算法的种群对应于蚁群算法中的蚁群,遗传算法的每一个个体(染色体)同时是蚁群算法中的一只蚂蚁。每当进行交叉或突变操作时,遗传算法首先根据蚁群算法的信息素矩阵计算父代个体基因间的联接强度,然后根据该联接强度选择交叉和突变位点。基因间的连接强度越大,它们作为整体遗传给子代的概率就越大,相应的这些基因之间的信息素将得到增强,也就是说基因联接将变得更紧密。论文的各部分组织如下,在第二部分简单回顾了 Harik 的基因联接学习遗传算法方法,第三部分详细地介绍了基于蚁群算法的基因联接学习遗传算法,第四部分以有界难度问题和 TSP 问题为例通过实验结果证实了该算法的可行性,最后在文章的第五部分粗略地对该算法和 Harik 的基因联接学习遗传算法进行了比较,讨论了它们之间的异同。

2 LLGA 回顾及其不足

在 LLGA 中,染色体每一位基因编码为一个<位置,等位基因>对。每一个基因的所有等位基因均全部出现在染色体的编码中,其中一个等位基因根据概率方案(Probabilistic Scheme)被表达并参与适应度值的计算。每一位基因编码上互补等位基因的存在确保了群体编码的多样性。在染色体编码中还存在一些非编码基因(内含子),它们不对染色体的适应度值做出贡献,但它们的有利于积木块的分隔和基因联接的形成。LLGA 的染色体呈环状编码,每一个染色体有一个翻译起始点。从该点开始,沿着染色体环的顺时针方向对每一个基因首次出现的编码进行表达。以图 1 为例,基因 1、2 和 3 为编码基因,4、5 为非编码基因,则该染色体所表示的实际编码序列为 (3,1)(2,0)(1,1)。由于有互补的等位基因的存在,通过改变翻译起始点,同一个染色体可以表达为多个不同的实际编码序列。这种机制称为概率表达(Probabilistic Expression, PE)。Harik 将 PE 推广为扩展概率表达机制(Extended Probabilistic Expression)EPE- n ,即对于每一个基因都具有一个表达的等位基因,和至多 n 个未被表达的等位基因。

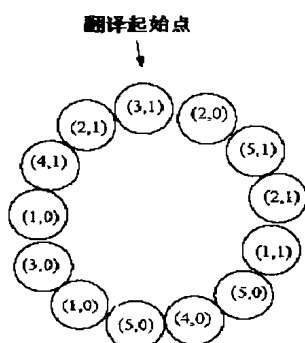


图1 LLGA 的染色体编码例

基因 1、2 和 3 为编码基因,4、5 为非编码基因。从翻译起始点出发,沿着顺时针方向,该染色体所表达的实际编码序列

为(3,1)(2,0)(1,1)。图中的(3,0)等为互补的等位基因。

在 LLGA 中交叉操作的步骤如下:首先,随机地从群体中选择两个个体作为父代个体,其中一个为供体而另一个为受体。随机地从供体中截取一个染色体片断并将其随机地插入到受体中。然后,受体染色体沿着顺时针方向删除多余的基因以便其子代染色体仍为一个合法的 EPE- n 个体。另一个交叉操作的子代个体可以通过交换两个父代个体的供体和受体角色而得到。

无论是对于具有指数尺度(Exponentially scaled)积木块(Building Blocks)的问题,还是具有均匀尺度(Uniformly scaled)积木块的问题,经过改进之后的基因联接学习遗传算法都可以对染色体中的基因自动进行重组,从而使得相互关联的基因相互靠近,完成基因联接学习和积木块分离的功能。但是 LLGA 仍然存在以下的弱点:(1)LLGA 对于积木块的学习是串行进行的,当积木块的数量较多时必然导致算法的计算效率降低。LLGA 的这种串行学习机制显然不如并行基因联接学习机制的效率,因此有必要研究基因联接学习遗传算法的并行基因联接学习机制;(2)LLGA 的编码方式要求每一位编码的所有等位基因至少出现一次,并且要求有大量的内含子存在,这极大地加大了编码串的长度,极大地扩大了遗传算法的搜索空间,增大了遗传算法的寻优难度。这个问题在等位基因数量较多时将会变得更加突出。

3 基于蚁群算法的基因联接学习遗传算法

3.1 算法的基本思想

遗传算法模拟生物的遗传和进化机制,通过适者生存的原则使得个体(问题的解的编码)一代比一代具有更优的适应度,从而达到全局寻优的目的。蚁群算法模拟蚂蚁所表现出来的群体智能,一群只具有有限记忆的无智能的蚂蚁个体,可以通过信息素来达到交流信息的目的,并最终在这个群体中突现出一种智能的行为,比如找到一条从巢穴到食物的最短路径。虽然这两种优化算法之间具有较为明显的区别,它们仍然具有许多共同的地方:这两者都对于一个解的群体进行操作,并且它们都是一种迭代寻优的过程。这些共同点正是将这两种算法结合在一起的结合点之所在。

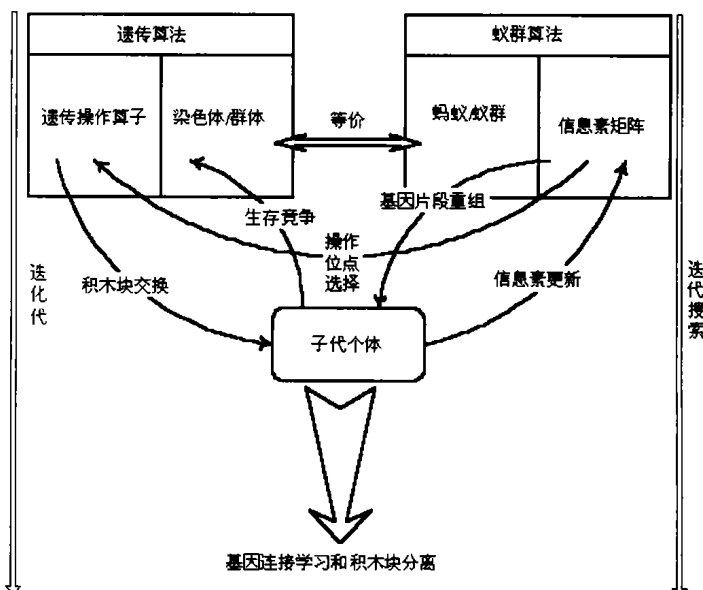


图2 ACO-LLGA 的基本思想

蚁群算法和遗传算法在群体的规模上被结合成一个整体,遗传算法的每一个个体等同于蚁群算法中的一只蚂蚁,遗传算法的每一个进化代等同于蚁群算法的一次迭代搜索。信息素矩阵、遗传操作算子以及新生个体之间的关系如图所示。

ACO-LLGA 算法中各组成部分之间的关系如图 2 所示。在该算法中,蚁群算法和遗传算法在群体的层次上被结合成一个整体,遗传算法的每一个个体等同于蚁群算法中的一只蚂蚁,遗传算法的每一个进化代等同于蚁群算法的一次迭代搜索。该算法仍然以遗传算法为主体,蚁群算法在这里被视为一个增强学习系统,通过增强学习来更新遗传算法中各个基因之间的连接强度。该连接强度通过蚁群算法的信息素浓度来衡量。在 LLGA 中基因联接的发现是通过其独特的染色体编码方式、交叉操作算子以及概率表达机制实现的。在算法中,作者通过采用 ACO 作为单独的基因联接发现过程来显式地获得基因联接信息,并将该基因联接信息存入信息素矩阵。该信息素矩阵被用来发现积木块,指导遗传操作位点的选择以及新个体的形成,这些新生个体的评价价值又被反过来用于信息素矩阵的更新。至此,一个基因联接学习的正反馈回路就形成了,从而本算法得以在其进化计算的过程中高效率地发现基因联接并将不同的积木块分离开来。

3.2 算法的整体结构

基于蚁群算法的基因联接学习遗传算法采用遗传算法的世代模型,其伪代码如下所示:

```

procedure ACO_based LLGA
begin
    t:=0;
    Initialize the pheromone matrix;
    Initialize P(t);
    Local and global pheromone update;
    while not(terminate condition)do begin
        t=t+1;
        while not(offspring pop_size)do begin
            Roulette selection of parents;
            Linkage strength computation;
            Linkage-based crossover operation;
            Linkage-based mutation operation;
            Local pheromone update;
        end while
        Survive competition;
        Global pheromone update;
    end while
end

```

在该算法中,遗传算法的每一个个体都同时是蚁群算法中的一只蚂蚁。在产生遗传算法的初始群体之前,先初始化蚁群算法的信息素矩阵 $\tau(i,j)$, $\forall i,j \in \{1,2,\dots,N\}$ 为一个较小的值,文中取为 $\tau_0=0.00001$,其中, N 为问题的编码长度。在算法的个体编码中不包含等位基因的拷贝,究竟哪一个等位基因会出现在染色体编码中将通过信息素矩阵来确定。每生成一个新个体,算法都会在评价该个体的同时对于信息素矩阵作局部更新:

$$\tau(i,j)=(1-\rho)\cdot\tau(i,j)+\rho\cdot\tau_0 \quad (1)$$

其中, $0<\rho\leq 1$ (根据文献[9],这里取值为 $\rho=0.1$) 为信息素的局部挥发因子,该挥发因子用于实现一种遗忘机制。当遗传算法完成一个进化代的操作之后,算法将考察当前代的最优个

体,如果这个最优个体的评价价值优于上一代的最优个体的评价价值,则对于该最优个体编码所对应的信息素矩阵中的每一个元素进行信息素全局更新:

$$\tau(i,j)=(1-\gamma)\cdot\tau(i,j)+\gamma\cdot\Delta\tau(i,j) \quad (2)$$

其中, $\gamma=0.1$ 为信息素矩阵的全局衰减因子, $\Delta\tau(i,j)$ 用于加强信息素矩阵中对应于当前最优个体编码的元素的值。

$$\Delta\tau(i,j)=\begin{cases} \frac{Q}{F} & \text{如果}(i,j)\text{对应于当前最优个体的相邻位编码} \\ 0 & \text{否则} \end{cases} \quad (3)$$

其中, F 为当前最优个体的评价价值 (假定最优个体为评价价值最小的个体), $Q=100$ 为一个常数。

3.3 基于基因联接强度的遗传操作

在该算法中,交叉和突变操作位点的选择都根据父代个体的基因间连接强度决定。染色体中任意两位相邻基因编码间的基因联接强度等于这两个基因在蚁群算法的信息素矩阵中相应的信息素浓度。这种基于蚁群算法信息素矩阵的基因联接学习机制的引入,有利于保护有用的模式或积木块不至于过多地被遗传操作所破坏。以单点交叉操作为例,对于给定的两个父代个体,算法首先根据蚁群算法的信息素矩阵获得这两个父代个体的基因联接强度;然后算法根据父代个体的基因间连接强度采用赌盘选择方法选择交叉操作位点,父代个体位于交叉操作位点之前的部分编码序列直接保留到子代个体,位于交叉操作位点之后的部分序列则被分割成若干个片断。分割时以当前信息素矩阵的平均值作为阈值,连接强度大于该阈值的连续的基因序列作为一个整体被保留,它们被认为是强连接的,而小于该阈值的基因片段则被分隔成为一个个单个的基因,它们被认为是弱连接的;最后,这些强或弱连接的基因片段根据如下算法加入到子代个体的编码中去:

设第 k 个染色体的当前位基因为 i , 以下确定其下一位基因的编码。如果 i 是上述某个强连接的染色体片断中的一个基因并且不是该片断中的最后一个基因, 则将该染色体片断自 i 以后的部分全部拷贝到子代个体中。否则,根据如下公式选择 i 的后继编码 j :

$$j=\begin{cases} \operatorname{argmax}_{u \in J_k(i)} \{\tau(i,j)\} & \text{if } q \leq q_0 \\ \text{Equation 6} & \text{else} \end{cases} \quad (4)$$

$$p_k(i,j)=\begin{cases} \frac{\tau(i,j)}{\sum_{u \in J_k(i)} \tau(i,u)} & \text{if } j \in J_k(i) \\ 0 & \text{else} \end{cases} \quad (5)$$

其中, $q \in [0,1]$ 为一个服从均匀分布的随机数, q_0 为一个介于 0 到 1 之间的常值参数。 $J_k(i)$ 为第 k 号染色体中基因 i 合法的后继基因编码的集合。上述两式的意义是,算法以概率 q_0 根据信息素矩阵从候选基因集中选择信息素最大的基因作为后继基因,以概率 $1-q_0$ 根据信息素矩阵从候选基因集中按赌盘选择法选择一个基因作为后续基因。

在突变操作中采用移位算子(Shift operator)。算法首先根据蚁群算法的信息素矩阵获得父代个体的基因联接强度,然后根据该基因联接强度采用赌盘选择方法分别选择一段基因编码片断和一个移动目标位点,截取所选中的基因编码片断并将其插入到目标位点。基因联接学习遗传算法的一个主要动机就是要使遗传算法得以识别基因所属的积木块,将属于不同积木块的基因区分开来,使属于同一个积木块的基因紧密地连结在一起,从而便于遗传算法通过交叉操作高效率地交换积木块,达到提高遗传算法寻优效率的目的。基于蚁群算法的基因联接

学习遗传算法以信息素作为衡量基因联接强度的依据,达到了识别积木块及其组成的目的;同时算法还采用了基于基因联接强度的遗传操作位点选择机制和基因片段重组机制,通过这两种机制分离不同积木块,使得属于同一个积木块的基因紧密连接,从而达到高效率地交换积木块、提高遗传算法效率的目的。

4 实验研究

作者采用两类有界难度问题 (Bounded Difficult Problems),即均匀尺度问题(uniformly scaled problem)和指数尺度问题(exponentially scaled problem),来检验算法的性能。这两类问题被认为是许多可分解问题的抽象版本并被广泛应用于 LLGA 的性能测试。这两类问题均由许多子问题构成,在均匀尺度问题中各子问题对于整个函数值具有相同的贡献,而在指数尺度问题中各子问题对于整个函数值的重要性则有着显著的区别。论文所采用的测试函数包含 10 个子函数,每个子函数均为一个具有完全欺骗性的 4 阶陷阱函数:

$$\text{trap}_4(u) = \begin{cases} u & u=k; \\ k-1-u & u=\{1,2,3\} \end{cases} \quad (6)$$

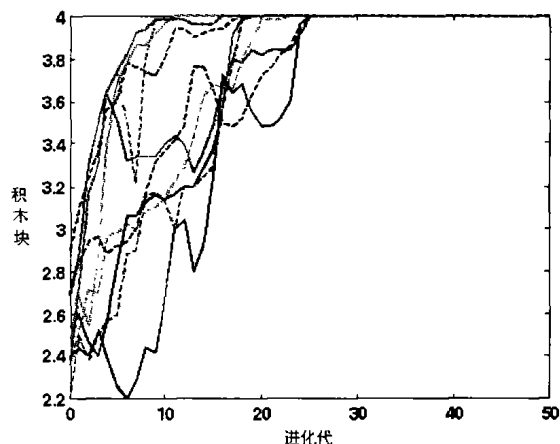
其中, u 为在 4 阶位串中字符 1 出现的次数。

在染色体编码中,各个基因被依次排成一串,每一个基因在一个固定的位置上采用<位置,等位基因> 对进行描述。比如,给定一个具有 4 位基因编码的问题,每位基因编码的取值为 0 或 1,则染色体编码串(3,1)-(4,0)-(1,1)-(2,0)表示这 4 个基因的编码次序为 gene3-gene4-gene1-gene2,它们的取值依次为 1-0-1-0。在算法的基因编码中,同一个基因编码的等位基因不允许同时出现在同一个染色体编码中,问题编码的多样性通过信息素矩阵和基于信息素矩阵的遗传操作算子保证。以试验的测试函数为例,每个基因编码具有两个等位基因 0 和 1,则其信息素矩阵如(7)所示。信息素矩阵对角线上 2×2 零矩阵的意义是,一旦某个基因出现在染色体编码中,则该基因本身及其等位基因都不可能再出现在同一个染色体中。

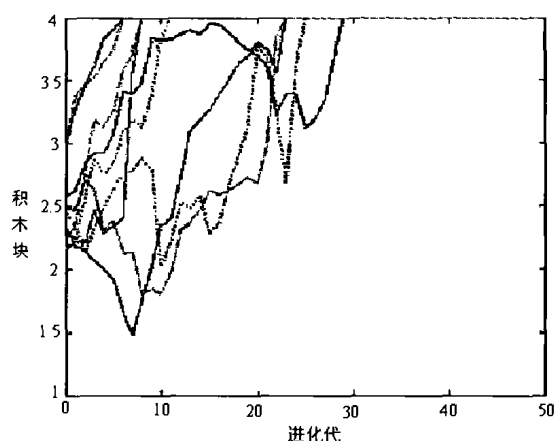
$$\begin{array}{c} \text{gene}_1 \quad \text{gene}_2 \quad \cdots \quad \text{gene}_n \\ a0 \quad a1 \quad a0 \quad a1 \quad \cdots \quad a0 \quad a1 \end{array} \quad \begin{array}{c} \left[\begin{array}{cc|cc|cc|cc} a0 & a1 & & & & & & \\ \hline 0 & 0 & & & & & & \\ \hline a1 & & & & & & & \\ \hline & & 0 & 0 & & & & \\ & & \hline & & 0 & 0 & & & & \\ & & & & \ddots & & & \\ & & & & & & 0 & 0 \\ & & & & & & \hline & & & & & & 0 & 0 \\ & & & & & & & \hline a0 & a1 & & & & & & \end{array} \right] \end{array} \quad (7)$$

在实验中这 10 个积木块的形成过程如图 3 所示。在算法开始的第零代,染色体编码通过遗传物质随机结合而成,因此在群体中没有积木块出现;随着演化的进行,一些积木块逐渐形成并显现出来。由图 3 可见,在算法中基因联接学习的效率是很高的,在算法运行到 10 代左右时就有积木块陆续形成,并且全部 10 个积木块都在 30 代以内被发现并分离出来。从这些积木块的形成过程可见,算法对于基因联接的学习和对于积木块的分离是并行进行的。由于这种并行基因联接学习机制的存在,无论是均匀尺度问题还是指数尺度问题,算法都能在一个很有限的时间之内求解。作为对比,LLGA 对于指数尺度问题

的求解时间随着问题的规模呈线性增长,对于均匀尺度问题的求解时间随着问题的规模呈指数增长^[7,14]。



(a) 均匀尺度函数的实验结果



(b) 指数尺度函数的实验结果

图 3 积木块的发现和形成过程

(图中的每一条线为一个积木块,每一个积木块为一个 4 位的陷阱函数。)

论文还采用 TSP 问题作为测试函数来检验算法的性能。采用 TSP 问题是基于如下的理由:首先,TSP 问题代表了一大类路由和规划等组合优化问题,具有广泛的代表意义,它比前述的有界难度问题更接近真实的问题,同时 TSP 问题也是遗传算法和蚁群算法的典型研究对象之一,采用 TSP 问题有便于算法的叙述和理解。其次,在 TSP 问题中包含了大量的积木块(子路径),是基因联接学习能够发挥其长处地方。再次,在 LLGA 研究中所采用的测试函数都是由一些互不交叠的积木块所组成,测试函数的整体评价价值就是这些单个积木块评价值的加和,在这一点上它们具有与 TSP 问题相同的性质,此外,LLGA 的环状染色体编码也与 TSP 问题的路径回路编码具有极为相似的表达形式。由此可见,在文中采用 TSP 问题作为实算对象是合适的。

作者采用 ulysses16 问题作为计算实例,为了便于算法的说明并与 LLGA 所采用的(位置,等位基因)对的编码方式相一致,作者采用如下基于城市-次序的编码方法:不失一般性,假定每一个路径方案均从第一个城市出发,每一个染色体的编码均为一个从数字 1 到 n 的没有重复数字的序列,其中 n 等于 TSP 问题的城市数。如果在染色体中第 i 个基因的编码为数字 j ,则表示城市 i 为履行商整个旅程中的第 j 站。比如对于 8 个

城市的 TSP 问题,染色体编码{1-3-4-6-5-7-2-8}所代表的实际路径为:{1-->7-->2-->3-->5-->4-->6-->8}。实验中算法的参数设定如表 1 所示。

表 1 算法的参数设定

群体规模	交叉概率	突变概率	最大进化代数	τ_0	γ	ρ	q_0
20	0.85	0.05	100	0.0001	0.1	0.1	0.90

如前所述,在基于蚁群算法的基因联接学习遗传算法中采用了基于信息素浓度的基因联接强度计算方法,并采用该联接强度来指导交叉和突变操作位点的选取,以达到发现和生成积木块的目的。作者观察在 TSP 计算实例 *ulysses16* 中积木块的生长过程。在算法的初始阶段,由于整个信息素矩阵的元素都是相同的而且很微弱,因此没有积木块存在。随着算法的运行,积木块逐渐形成,到第 20 代时出现了 4 个积木块,它们是:1
0.90 1.0 1.0 0.90 1.0 1.0 0.90 0.90 0.70 0.70
 $\leftrightarrow 8 \leftrightarrow 4 \leftrightarrow 2 \leftrightarrow 3$, $9 \leftrightarrow 11 \leftrightarrow 5$, $15 \leftrightarrow 6 \leftrightarrow 7$ 以及 $12 \leftrightarrow 13 \leftrightarrow 14$, 其中整数表示在履行商的实际路径中所途径的城市,箭头上的小数表示归一化的基因间连接强度。到第 50 代时,前两个积木块得到了增长,而后两个积木块被破坏并重组成为一个新的积木块,它们分别为:1
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
 $\leftrightarrow 8 \leftrightarrow 4 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 16$, $10 \leftrightarrow 9 \leftrightarrow 11 \leftrightarrow 5 \leftrightarrow 15$, 以及
1.0 1.0
 $6 \leftrightarrow 7 \leftrightarrow 12$ 。当算法进化到第 80 代时,这三个积木块都得到了进一步的生长和加强:1
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
 $\leftrightarrow 8 \leftrightarrow 4 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 16$, $10 \leftrightarrow 9 \leftrightarrow 11 \leftrightarrow 5 \leftrightarrow 15$, $6 \leftrightarrow 7 \leftrightarrow 12 \leftrightarrow 13$ 。最后,算法在 100 代以内寻优到了问题的最优解,该最优解略好于 TSP 数据库^[15]所给出的迄今已知的最优解,这些解如图 4.5 所示。由算例可见,基于蚁群算法的基因联接学习遗传算法的确可以在遗传算法进化寻优的同时,发现各个基因之间的联接关系,区分属于不同积木块的基因,并使同一个积木块内的基因之间的联接更为紧密,从而指引遗传算法的搜索方向,提高其优化计算的效率。

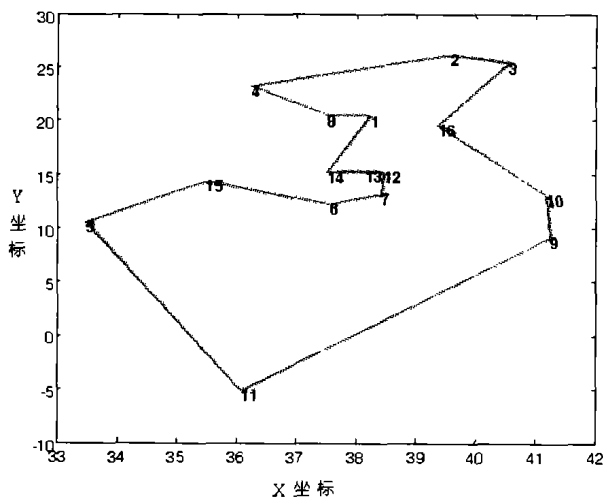
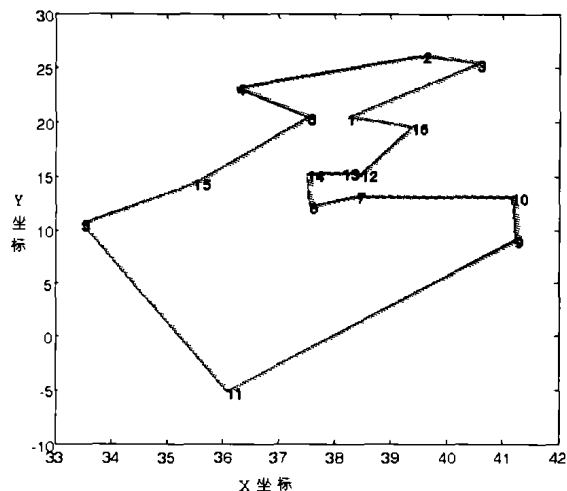


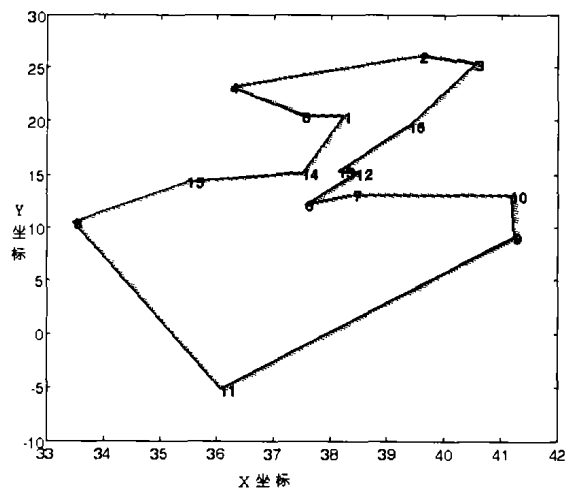
图 4 TSPLIB 中迄今已知关于 Ulysses 16 问题的最优解, 路径长度为 74.1088

5 结束语

论文提出了一种新的基因联接学习遗传算法,该方法使得遗传算法的每一个个体同时又是蚁群算法的一只蚂蚁,采用蚁群算法的信息素浓度来衡量基因间连接的强度,通过信息素矩阵的正反馈更新机制实现基因联接学习,并以此指导遗传操作位点的选择,从而指引了遗传算法的搜索方向,提高遗传算法的寻优效率。文章采用有界难度问题和 TSP 问题的计算实例



(a)路径一,长度 $L=73.9877$



(b)路径二,长度 $L=74.0966$

图 5 由该算法实算获得的关于 Ulysses 16 问题的最优解和次优解,均优于 TSPLIB 所给出的结果

说明了该算法的运行机制,验证了其有效性。

在 Harik 的 LLGA 的概率表达机制中,每一个染色体都包含了任何一个基因全部可能的等位基因,在 EPE- n 机制下每个染色体还要包含更多的等位基因副本,此外还含有大量的作为非编码区而存在的内含子。这种编码方式在问题的维数较大或基因的编码具有较多等位基因(比如 TSP 问题)的时候,就会使染色体的编码非常长,问题的搜索空间非常大,从而降低算法的效率。而在基于蚁群算法的基因联接学习遗传算法中则可以避免这种情况的出现,染色体编码的长度只取决于问题的维数和基因编码的精度,而与每一位基因编码上等位基因的数量无关,也没有非编码区的存在,基因联接的强度通过信息素矩阵来表示。

在 Harik 的 LLGA 中,算法对于积木块的基因联接学习是串行进行的,学习完一个积木块的基因联接之后再学习下一个。从算法的运行机制和所给出的算例可见,在基于蚁群算法的基因联接学习遗传算法中,各个积木块内的基因联接学习是同时进行的。算法的这种并行性是非常重要的,它本身即意味着应用该算法进行基因联接学习的高效性,同时也意味着利用该算法这种并行基因联接学习的特点研究整个算法并行实现的可能,从而可以更进一步提高算法的性能。

虽然论文的中心是介绍基于蚁群算法的基因联接学习遗

传算法而不是 TSP 问题的求解,但从实验研究中所给出的一个简单算例可以看出,论文所提出的算法同时体现出了求解 TSP 问题的潜力,它也许同时为 TSP 问题的求解提供了一种新的思路。(收稿日期:2004 年 4 月)

参考文献

1. Zbigniew Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs[M]. Berlin: Springer-Verlag, 1996
2. Holland J H. Adaptation in natural and artificial systems. Ann Arbor, University of Michigan Press, 1975
3. Harik G R. Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms. Doctoral dissertation, University of Michigan, Ann Arbor Also IlliGAL Report No 97005
4. Harik G R, Goldberg D E. Learning Linkage[C]. In: Belew R K, Vose M D Eds. Foundations of Genetic Algorithm 4, San Francisco, CA: Morgan Kaufmann, 1996: 247~262
5. Harik G R, Goldberg D E. Learning linkage through probabilistic expression. Computer Methods in Applied Mechanics and Engineering 186, 2000: 295~310
6. Chen Y-P, Goldberg D E. Introducing start expression genes to the linkage learning genetic algorithm[R]. IlliGAL Report No. 2002007, Urbana, IL: University of Illinois, IlliGAL 2002
7. Lobo F G, Deb K, Goldberg D E et al. Compressed introns in a linkage learning genetic algorithm[C]. In: Proceeding of the Symposium on

Genetic Algorithms SCA-98, Also available as IlliGAL Report No 97010, 1998

8. Dorigo M, G Di Caro. The Ant Colony Optimization Meta-Heuristic. In D Corne, M Dorigo, F Glover eds. New Ideas in Optimization, McGraw-Hill, 1999: 11~32
9. M Dorigo, L M Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. IEEE Transactions on Evolutionary Computation, 1997; 1(1): 53~66
10. Dorigo M, V Maniezzo, A Colomi. The Ant System: Optimization by a Colony of Cooperating Agents[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 1996; 26: (1) 29~41
11. Bonabeau E, Dorigo M, Theraulaz G. Inspiration for optimization from social insect behavior[J]. Nature, 2000; 406: 39~42
12. Dorigo M, G Di Caro. The Ant Colony Optimization Meta-Heuristic [C]. In: D Corne, M Dorigo, F Glover eds. New Ideas in Optimization, McGraw-Hill, 1999: 11~32
13. M Dorigo, L M Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. IEEE Transactions on Evolutionary Computation, 1997; 1(1): 53~66
14. Ying-ping Chen. On the sequential behavior of the linkage learning genetic algorithm. Available: www.advancedgec.ge.uiuc.edu/termprojects/SequentialBehaviorOfTheLLGA.pdf
15. G Reinelt. TSPLIB-A traveling salesman problem library. ORSA J Comput. Available on-line at <http://softlib.rice.edu/softlib/tsplib>, 1991; 3: 376~384

(上接 3 页)

为了较精确地刻画该加密方法的图像恢复性能,能够定量地表征图像恢复性能,作者引入解密后的恢复图像和待加密的原图像间的均方误差:

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{p}(x, y) - p(x, y)]^2 \quad (15)$$

来表征超混沌广义 Henon 映射 Dead-Beat 同步加密方法的图像恢复性能。其中 x, y 表示图像的空间域坐标, $M \times N$ 表示图像的几何尺寸。被加密的原图像信号为 256 级 $269 \times 279 \times 3$ 的彩色图像 LenaSmall.tif, 经过加密后的图像信号通过 INTERNET 以 E-mail 方式发送和接收,然后再进行图像的解密恢复。

仿真计算结果显示:超混沌广义 Henon 映射 Dead-Beat 同步加密方法的解密图像和原图像之间的均方误差是 $1.9726e-011$ 。由此可见,超混沌广义 Henon 映射 Dead-Beat 同步加密方法对恢复图像的性能影响不大,图像的恢复质量高。

由于超混沌广义 Henon 映射 Dead-Beat 同步加密方法包括了混沌掩盖和混沌调制两种技术,掩盖信号 y' 即被用作混沌同步信号,同时还携带有信息信号,与混沌调制方法一样,信号 y' 也反馈回发送端,因此发送端是很复杂的非自治系统。非法用户在不知道密钥的情况下是无法正确解密并恢复原始信息信号的,因为混沌调制技术已经改变了原始的超混沌广义 Henon 映射的动力学,利用非线性重构技术从接收到的密文信号中无法重构出原始的超混沌广义 Henon 映射的动力学,因此该加密方法具有较高的安全性,如果非法用户不知道准确的密钥值,即使接收端采用非线性观察器技术,也不可能与发送端正确同步,因而不能解密出信息信号。

此外,在文献[7]中对混沌加密系统的概念及其涵义有较深入的阐述,尤其是区别于常见的伪混沌加密系统,并且提出了混沌加密的新方案,其中的第一种级联方案将混沌加密系统与

传统加密系统相结合,利用混沌加密系统的混沌信号不可预测性的优点,及传统加密系统具有成熟的密钥空间设计技术,使整个系统的抗攻击能力明显提高,并作出猜测:对于这种级联加密系统来说,不存在比穷举攻击(唯密文攻击)更加有效的破译方法。

论文提出的超混沌同步加密系统可以作为该级联方案的前级混沌加密器,采用硬件实现时,作为前级的混沌系统密钥信号是不可预测的,故不存在唯一对应的明文密文对,从而使采用传统加密系统的后级不可击破。因此,有可能在一定程度上实现“一次一密密码”的目标。这将是今后继续深入研究的方向。(收稿日期:2004 年 5 月)

参考文献

1. PECORA L M, CARROLL T L. Synchronizing chaotic circuits[J]. IEEE Trans Circuits Syst, 1991; 38(4): 453~456
2. PENG J H, DING E J, DING M et al. Synchronizing hyperchaos with a scalar transmitted signal[J]. Phys Rev Lett, 1996; 76(6): 904~907
3. GRASSI G, MASCOLO S. Nonlinear observer design to synchronize hyperchaotic systems via a scalar signal[J]. IEEE Trans Circuits Syst I, 1997; 44(10): 1011~1014
4. GRASSI G, MASCOLO S. Synchronizing hyperchaotic systems by observer design[J]. IEEE Trans Circuits Syst II, 1999; 46(4): 478~483
5. GRASSI G, MILLER D A. Theory and experimental realization of observer-based discrete-time hyperchaos synchronization[J]. IEEE Trans Circuits Syst I, 2002; 49(3): 373~378
6. BAIER G, KLEIN M. Maximum hyperchaos in generalized Hénon maps[J]. Phys Lett A, 1990; 151(6/7): 281~284
7. 丘水生, 陈艳峰, 吴敏等. 混沌保密通信的若干问题及混沌加密新方案[J]. 华南理工大学学报(自然科学版), 2002; 30(11): 75~80