

# 遗传算法在矩阵置换对称性求解中的应用

董继扬, 保 铮

(西安电子科技大学雷达信号处理国家重点实验室, 陕西西安 710071)

**摘 要:** 对称性是减少问题的自由度的一个强有力的工具. 但在实际的应用中, 系统变换操作的总数目将随系统维数的增加而急剧上升, 这给高维系统对称性的计算带来了极大的不便, 从而使得对称性方法的应用受到了很大的限制. 本文以全互连结构的神经网络为例, 提出一种基于遗传算法的搜索方法, 在对称群  $S_n$  中寻找网络的对称置换操作, 给出了计算机上的模拟结果, 并与传统的遍历搜索方法作比较, 分析了各自的优缺点. 结果表明, 这种基于遗传算法的搜索方法能够在极短的时间内找到网络的大部分对称置换操作. 这使得对称性方法在高维神经网络研究及设计中的应用成为可能.

**关键词:** 神经网络; 对称置换; 遗传算法; 权重矩阵

**中图分类号:** TP183

**文献标识码:** A

**文章编号:** 0372-2112 (2003) 03-0321-04

## Application of Genetic Algorithms in Searching for Matrix's Permutation Symmetries

DONG Ji-yang, BAO Zheng

(National Key Lab for Radar Signal Processing, Xidian Univ, Xi'an, Shaanxi 710071, China)

**Abstract:** Symmetry is a powerful tool to reduce the number of degrees of freedom of a problem. But calculating the symmetry of a high-dimensional system would be very difficult since the total number of transformations increases dramatically with the dimension of the system, which places many restrictions on the application of the symmetry method. A novel approach based on genetic algorithms is proposed to search for the permutation symmetries of the weight matrix of full-connected neural networks within the symmetry group  $S_n$ . Searching results for several different dimensional matrixes are given and compared to that of the ergodic searching method. It turns out that the searching method based on genetic algorithms can find the majority of symmetric permutations of the matrix within a short time, which makes it possible to study and design of the high-dimensional neural networks by the symmetry tool.

**Key words:** neural networks; symmetric permutation; genetic algorithms; weight matrix

### 1 引言

一个物理系统的一切对称变换之集合构成群, 称为该系统的对称变换群. 如, 对于全同粒子体系, 任意交换体系中的粒子都不会改变系统的哈密顿量, 即系统具有置换对称性, 相应对称操作的集合称为置换群 (也称对称群)<sup>[1]</sup>.  $n$  个全同粒子体系的对称群表示为  $S_n$ . 在神经网络中, 若每个神经元都具有相同的激活函数, 则网络可以看作是由一些全同神经元组成的系统, 此时, 网络也具有了置换对称性研究的物理基础.

实际上, 人工神经网络领域中的对称性研究已取得了可喜的成果. 如, Baldi 证明了网络的全局性质可以从特定的存储模式集的对称性获得<sup>[2]</sup>; Lautrup 用对称群子群下的不变性将自旋玻璃中的复制对称破缺的平均场理论解进行分类<sup>[3]</sup>; Reimann 更是直接把对称性的分析方法应用于关联记忆网络的研究和设计中<sup>[4,5]</sup>, 并说明了网络的动力学行为在很大程

度上是受网络结构对称性的制约, 等.

当然, 在网络对称性的研究中不可避免地要涉及对称性的计算. 一般地, 当网络的对称群的生成元未知时, 我们可以采用遍历的搜索方法在对称群  $S_n$  中寻找网络的对称置换操作. 但实际上, 由于  $S_n$  的阶随网络神经元个数  $n$  的增加而急剧上升, 即,  $|S_n| = n!$ , 当网络达到一定规模时, 遍历法搜索法就不再适用了. 例如,  $32! \approx 2.6 \times 10^{35}$ , 这使得网络对称性的研究及应用只能局限在小规模网络的范围内进行. 如何快速找出高维网络的对称置换操作就成为急待解决的问题.

本文提出了一种基于遗传算法的搜索方法, 试验表明, 这种方法能快速而准确地异常巨大的解空间中找到大量符合要求的对称置换操作. 文中给出了几种不同规模网络的计算机模拟结果, 并与遍历法作了比较, 分析了各自的优缺点. 由于这种方法的搜索时间随网络维数的变化不太明显, 使得对称性的研究及其应用向高维网络的拓展成为可能.

收稿日期: 2001-11-05; 修回日期: 2002-01-24

基金项目: 国家自然科学基金 (No. 60071026)

## 2 神经网络中的置换作用

全互连网络的结构可用神经元之间的连接矩阵来表示<sup>[8]</sup>,网络的对称性就是其连接矩阵的对称性,网络的很多有趣的性质都可以从该网络的连接矩阵对称性得出.关于这方面的研究已有很多的文献<sup>[2~7]</sup>,由于这不是本文的研究范围,就不在此详述了.下面给出连接矩阵置换对称性的定义.

**定义 1** 置换群  $S_n$  在连接矩阵  $w \in R^{n \times n}$  上的作用定义为:对于  $S_n$  中任一元素  $s, (s \cdot w)_{ij} = w_{s(i)s(j)}, \forall s \in S_n$ , 即,同时置换矩阵的行和列的下标.

**定义 2** 对于  $\exists s \in S_n$ , 若满足  $s \cdot w = w, w \in R^{n \times n}$ , 则称置换  $s$  为矩阵  $w$  的对称置换. 矩阵  $w$  的所有对称置换操作的集合,  $S_w = \{s | s \cdot w = w, s \in S_n\}$ , 构成群, 称为矩阵  $w$  的置换对称群.

给定一个矩阵  $w$ , 寻找其置换对称群的问题等价于在对称群  $S_n$  中寻找  $w$  的对称置换的集合. 如,

**例 1** 网络连接矩阵为

$$w = \begin{pmatrix} 1 & 1/3 & -1/3 & -1 & -1/3 & 1/3 \\ 1/3 & 1 & 1/3 & -1/3 & -1 & -1/3 \\ -1/3 & 1/3 & 1 & 1/3 & -1/3 & -1 \\ -1 & -1/3 & 1/3 & 1 & 1/3 & -1/3 \\ -1/3 & -1 & -1/3 & 1/3 & 1 & 1/3 \\ 1/3 & -1/3 & -1 & -1/3 & 1/3 & 1 \end{pmatrix}$$

利用遍历法,可以在  $S_6$  中找到网络所有的对称置换操作的集合,

$S_w = \{(1\ 6)(2\ 5)(3\ 4), (1\ 5)(2\ 4), (1\ 4)(2\ 3)(5\ 6), (1\ 3)(4\ 6), (1\ 2\ 3\ 4\ 5\ 6), (1\ 2)(3\ 6)(4\ 5), (1\ 3\ 5)(2\ 4\ 6), (2\ 6)(3\ 5), (1\ 4)(2\ 5)(3\ 6), (1\ 5\ 3)(2\ 6\ 4), (6\ 5\ 4\ 3\ 2\ 1), e\}$ .

在例 1 中,由于置换群  $S_6$  只有 720 个元素,对计算机来说是一个很小的数,因此用遍历法很快就可以求出矩阵  $w$  的置换对称性.但由于  $S_n$  元素的总个数为  $n!$ ,当维数增加时,将出现组合爆炸现象,是一个典型的 NP-难问题,对于维数较大的矩阵,遍历搜索法就会因为搜索时间太长而实际上是不可行的.下面我们提出一种基于用遗传算法的搜索方法.

## 3 算法描述

遗传算法<sup>[9]</sup>是一种基于自然选择和群体遗传机理的搜索算法.它模拟了自然选择和遗传过程中发生的繁殖、交配和突变等现象.它将所求问题每个可能的解看作是群体中的一个个体,并将每个个体编码成染色体的形式,然后根据预定的目标函数对每个个体进行评价,给出一个适应度值.根据个体的适应度值,对群体实施几个遗传算子(复制、杂交、突变)的作用,得到一个新的群体.新群体中的个体继承了上一代的一些优良性状,而优于上一代,通过这样的反复循环使全体朝更优化的方向进化.

本文参照了 $(\mu + )$ 的个体选择规则<sup>[10]</sup>,即,群体中共有  $\mu$  个个体,经杂交和变异之后产生  $\mu$  个后代,然后从这 $(\mu + )$ 个父体和后代中选出前  $\mu$  个适应值较好的个体作为下一代种群,进行演化和迭代,并引入了更新频率这一参数  $P_u$ ,以

此避免算法陷入局部最优点.算法的具体步骤如下:

- (1) 参数设置:设置种群和后代的大小  $\mu$  和  $\lambda$ ,最大演化代数  $T$ ,杂交率  $P_c$ ,变异率  $P_m$  和更新频率  $P_u$ ;
  - (2) 初始化:随机产生由  $\mu$  个个体组成初始种群,令此时演化代数为  $t = 0$ ;
  - (3) 杂交操作:根据预设的杂交率对个体进行杂交,产生  $\mu \times P_c$  个后代;
  - (4) 变异操作:根据预设的变异率对个体进行变异,产生  $\mu \times P_m$  个后代;
  - (5) 评估:计算每个个体的适应值  $f(i), i = 1, 2, \dots, \mu + \lambda$ ;
  - (6) 选择:从  $\mu + \lambda$  个父体和后代中选出适应值较好的  $\mu$  个个体作为下一代种群;
  - (7) 保存对称变换:从  $\mu + \lambda$  个父体和后代中选出满足要求的对称变换,将它保存起来,然后用一个随机产生的个体代替该个体;
  - (8) 更新:如果演化代数  $t \bmod P_u = 0$  (且  $t > 0$ ), 则用一组随机产生的群体代替子代群体进行下一代的演化.
  - (9) 终止:令  $t = t + 1$ , 如果  $t > T$  则终止,否则转向(3).
- 下面将对上述步骤进行具体的描述.

### 3.1 编码问题

如何将问题的解转换为编码表达的染色体是遗传算法的关键问题.传统的编码方法是二进制串表达,但对于许多遗传算法的应用,这种简单的编码方法很难直接描述出问题的性质.本文根据置换的特点,将置换  $s = \begin{pmatrix} 1 & 2 & \dots & n \\ i_1 & i_2 & \dots & i_n \end{pmatrix}$  简单地用  $i_1, i_2, \dots, i_n$  表示.编码结果使得任一个染色体都是自然数 1 到  $n$  的一个排列,即一个染色体中不会出现有相同的基因.对于长度为  $n$  的染色体,这种编码方式所能得到的不同的染色体的个数为  $n!$ ,与对称置换的个数相同,且不会产生非法个体,因此是一个较好的编码方式.

### 3.2 适应度的估计

适应度的估计是本文研究的问题中最难的一个环节,主要体现在适应度函数的选择上,因为在数学上,置换操作只能是“是”或者“不是”矩阵  $w$  的对称置换,而不能区别那一个置换操作较好或者较差,因此,所选择的适应度函数带有一定的盲目性,还有待于进一步的探讨.本文所取的适应度函数为,

$$f(s) = - \sum_{i,j} |w_{s(i)s(j)} - w_{ij}|$$

根据这个适应度函数,当一个置换是矩阵的对称置换时,则其适应度为 0,否则就是一个小于 0 的数.总之,适应度越大的置换保留到下一代的几率越大.

### 3.3 杂交操作

下面用一个例子来说明染色体的杂交操作.

设选出如下两个染色体做交换,

染色体 1: [1 2 3 4 5 6 7 8 9 10 11 12]

染色体 2: [2 5 1 9 6 7 3 4 10 8 12 11]

交换阴影(加边框)部分的基因,并把冲突的基因用  $x$  代替,分别按顺序提取两个染色体的冲突部分的基因,颠倒次序后

部分子代 1: [x 2 **1 9 6 7 3 4** x 10 11 12]

部分子代 2: [2 x **3 4 5 6 7 8** 10 x 12 11]

再进行交换,

染色体 1 的冲突基因 [1 9] 与 染色体 2 的冲突基因 [5 8] 进行交叉

按上述的杂交方法,填充冲突位,

子代 1: [8 2 **1 9 6 7 3 4** 5 10 11 12]

子代 2: [2 9 **3 4 5 6 7 8** 10 1 12 11]

### 3.4 突变操作

根据染色体是 1 到  $n$  的这  $n$  个数的排列这一特点,很自然地,在染色体的突变操作中,我们可以随机地选择交换染色体中的两个基因.如

染色体: [1 2 3 **4** 5 6 7 8 9 **10** 11 12]

突变后的染色体为

子代: [1 2 3 **10** 5 6 7 8 9 **4** 11 12]

## 4 数值实验结果

为了验证上述遗传算法在寻找矩阵置换对称群方面的能力,我们在 DELL 机 (OptiPlex G1, C400/ 128M SDRAM/ Windows 2000 Server) 上用 Visual C++ 6.0 语言编程,并用它来计算维数不同的连接矩阵的置换对称群.在实验中,我们用一些特殊的网络,其连接矩阵的置换对称群均同构于相应的二面体群<sup>[6,7]</sup>,即群元个数为  $2n$ . 由于本文的所提出的算法与矩阵元具体的大小无关,且受篇幅限制,在此不一一列出这些矩阵.

计算结果如表 1 至表 6 所示,其中,表 1 为遍历法的搜索结果,表 2 至表 6 分别为列出了遗传算法在 5 种不同维数的连接矩阵中的搜索结果,每一维数的连接矩阵都分别用 5 种不同的参数进行试验.表中,  $m$  表示搜索到的对称置换操作的个数,  $t$  表示搜索所用的时间,单位为秒,  $\mu$  为种群大小,  $T$  为最大演化代数.算法的杂交率和变异率分别取 0.4 和 0.08,更新率  $P_u$  取  $5/n$  ( $n$  为矩阵的维数).

表 1 遍历法计算矩阵的置换对称性

矩阵维数 ( $n$ )	8	9	10	11	12	13
对称置换数 ( $m$ )	16	18	20	22	24	26
时间 (秒)	<1	4	48	611	8420	108281

表 2 用遗传算法计算矩阵的置换对称性 (矩阵维数  $n=8$ )

$m$	13	16	14	14	14
$t$	<1	3	2	3	3
$\mu$	50	80	70	20	50
$T$	1000	1000	1000	4000	2000

表 3 用遗传算法计算矩阵的置换对称性 (矩阵维数  $n=9$ )

$m$	13	15	14	17	18
$t$	2	3	3	3	5
$\mu$	50	80	80	50	70
$T$	1000	1000	900	2000	2000

表 4 用遗传算法计算矩阵的置换对称性 (矩阵维数  $n=11$ )

$m$	12	15	15	14	20
$T$	3	3	7	5	8
$\mu$	75	80	80	100	80
$T$	1000	1000	2000	1000	2400

表 5 用遗传算法计算矩阵的置换对称性 (矩阵维数  $n=14$ )

$m$	15	18	18	17	19
$t$	10	10	13	11	14
$\mu$	100	80	50	60	70
$T$	1500	2000	4000	3000	3000

表 6 用遗传算法计算矩阵的置换对称性 (矩阵维数  $n=20$ )

$m$	17	14	22	33	23
$t$	71	51	102	190	117
$\mu$	200	150	150	300	200
$T$	3000	3000	6000	5000	5000

## 5 结论与讨论

从模拟过程和表 1~6 的数据可以看出:

(1) 用遍历法计算矩阵的置换对称群时 (见表 1), 矩阵维数每增加一维所用的时间就约增加 12 倍, 即, 其时间复杂度约为  $O(12^n)$ ,  $n=13$  时所用的时间约为一天, 依此计算,  $n=14$  时需要 12 天, 当  $n=20$  时就需要至少  $1.1 \times 10^6$  年才能算完. 可见遍历法的应用范围相当有限.

(2) 而对于遗传算法, 当矩阵的维数  $n$  较小时 (如  $n=8$ ) 其效果比用遍历法差 (即所用的时间比较多), 而当矩阵维数较大时, 其优越性就体现出来了, 从表 5 可见, 当  $n=14$  时, 用遗传算法仅用了 14 秒就搜索出 19 个对称置换了 (该矩阵的对称置换总数为 28 个), 即已经算出了大部分的对称置换. 遗传算法的突出优点是其所用的时间随矩阵维数 ( $n$ ) 的增加变化不明显. 如  $n=8$  时用 3 秒种可以搜索出矩阵的大部分甚至所有的置换对称性 (见表 2), 当维数  $n$  增加到 20 时, 搜索出矩阵大部分的置换对称群所用的时间也没超过 200 秒, 可见, 其时间复杂度远远小于遍历法的时间复杂度.

(3) 遗传算法的一个不足是它很难找出所有的对称置换, 而遍历法从理论上讲是可以的. 不过, 这一缺点可以用群本身的一些性质进行一些弥补. 例如, 设用遗传算法求出连接矩阵  $w$  的  $m$  个对称置换  $\{s_i, i=1, 2, \dots, m\}$ , 而连接矩阵  $w$  的置换对称性为  $S$ , 则有  $\{s_i\} \subseteq S$ , 根据群的基本性质, 即群中任意两个元素的乘积也是该群的一个元素, 因此, 任意从集合  $\{s_i\}$  中选取两个置换  $s_i, s_j$ , 设  $s_k = s_i s_j$ , 则有  $s_k \in S$ . 也就是说, 从  $m$  个对称置换  $\{s_i, i=1, 2, \dots, m\}$  中可以构造出连接矩阵  $w$  的更多 (甚至全部) 的对称置换来.

(4) 从表 2 至表 6 可见, 对于本文的算法, 调整群体大小  $\mu$ , 最大代数  $T$  等参数的值可以得到更好的计算结果. 另外, 在本文的所有实验中, 算法的杂交率、变异率和新频率  $P_u$  都为固定值, 其实这些参数都是可调的, 在实际问题中必须适当

地调节它们的取值才会取得好的效果,由于本文只是为了说明遗传算法的可行性和优越性,我们没有对这些参数做精确的调整.这说明本文所提出的方法还有更好的潜力.

总之,利用本文提出的算法计算神经网络连接矩阵的置换对称性,能够在极短的时间内得到满意的结果,这使得对称性方法在高维神经网络研究与设计中的应用成为可能.

#### 参考文献:

- [ 1 ] 韩其智,孙洪洲.群论[M].北京:北京大学出版社,1987.
- [ 2 ] Pierre Baldi. Symmetries and learning in neural network models [J]. Phys Rev Lett,1987,59(17):1976 - 1978.
- [ 3 ] B Lautrup. The Hopfield Model [M]. Singapore: World Scientific, 1982.
- [ 4 ] Stefan Reimann. Symmetry and network structure [J]. Neural Processing Letters,1997,(6):1 - 10.
- [ 5 ] Stefan Reimann. On the design of artificial auto-associative neuronal networks [J]. Neural Networks,1998,(11):611 - 621.
- [ 6 ] 董继扬,徐慎初,吴伯僊.均匀分类 DHNN 结构的对称性[J].厦门大学学报(自然科学版),2000,39(1):46 - 51.
- [ 7 ] 董继扬,徐慎初,吴伯僊.均匀分类 DHNN 的分类对称性[J].厦门大学学报(自然科学版),2000,39(3):329 - 335.
- [ 8 ] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities [A]. Proc Natl Acad Sci USA [C].

Washington DC:National Academy Press,1982. 2554 - 2558.

- [ 9 ] J H Holland. Adaptation in Natural and Artificial Systems [M]. Michigan:University of Michigan Press,1975.
- [ 10 ] Schwefel H - P. Numerical Optimization for Computer Models [M]. Chichester,UK:John Wiley,1981.

#### 作者简介:



董继扬 男,1974 年生于福建安溪,西安电子科技大学博士,研究领域为遗传算法、神经网络及图像处理等.



保 铮 男,1927 年生于江苏南通,中科院院士,教授,博士生导师,主要研究领域为雷达系统与检测.