

神经网络与图论*

许 进

(华中科技大学系统科学研究所, 武汉 430074)

保 铮

(西安电子科技大学电子工程研究所, 西安 710071)

摘要 较系统地论述了人工神经网络与图论之间的相互关系: (i) 综述了人工神经网络在图论的许多困难问题研究中的应用, 特别是在一些 NP-完全问题上的应用; (ii) 综述了图论作为一种很有用的工具在人工神经网络中的应用, 诸如应用于离散型前向神经网络模式分类的研究, 以及反馈型人工神经网络稳定性分析的研究等。

关键词 人工神经网络 图论

神经网络可分为两类: 一类是生物神经网络, 另一类是人工神经网络。生物神经网络是自然界中的一种客观存在的、由生物神经系统中神经细胞按照一定的连接方式连接而形成的网络。如人脑神经系统是目前为止所发现的最具有智慧的生物神经网络; 人工神经网络是科学工作者利用电子技术、光学技术等模拟生物神经网络的某些结构、特征以及功能而人为地研究制造的具有智能化的网络。

图论是一门古老的学科, 它的诞生可追溯到 1736 年。然而, 图论的真正兴起并形成一门独立的学科则是 20 世纪四五十年代的事情。图论具有直观、清晰、解决问题简捷等优点。更重要的是, 许许多多的实际问题可以等价地转化为图论问题来处理。正因为如此, 图论已广泛地应用于诸如计算机科学、数学、系统工程、通讯网络、控制工程、心理学和管理科学等。可以说, 图论是一门既古老又年轻的学科, 具有强大的生命力, 同神经网络一样, 对人类科学的发展起巨大的促进作用。

人工神经网络与图论之间有密切的联系: 图论作为一种很有用的工具可广泛地应用于神经网络许多方面的研究中, 诸如神经网络的结构算法, 神经网络的模型设计, 神经网络的稳定性理论以及前向人工神经网络模式分类问题等; 另一方面, 人工神经网络作为一种有力的工具可应用于图论中许多困难问题的研究, 特别是一些困难的 NP-完全问题、网络优化问题等的求解, 也可借助于神经网络来解决图论中某些尚待解决的猜想。基于上述原因, 本文较详细地论述了人工神经网络与图论之间的相互关系: 一是人工神经网络在图论中的应用, 二是图论在人工神经网络中的应用。

本文所言之图皆指有限、允许有自环、但无多重边的有向图。分别用 $V(G)$ 和 $E(G)$ 表示图 G 的顶点集和边集。用 $\omega(G)$ 表示图 G 的连通分支数。本文所用有关神经网络方面的基本概念与术语均与文献[1]和[2]相同; 所用图论中的有关概念均与文献[3]相同。

2001-03-29 收稿

* 国家自然科学基金资助项目(批准号: 69971018)

1 人工神经网络在图与组合优化中的应用

自从 Hopfield 和 Tank^[4]在 1985 年首次成功地将 Hopfield 神经网络模型应用于解决困难的流动推销员问题(travelling salesman problem, TSP)以来,神经网络在图论的许多领域都得到了良好的应用,诸如图的着色问题,图的同构算法,图的顶点覆盖问题,图的最大团与最大独立集,平面测试算法,图的划分问题,各种类型的 TSP 问题,中国邮路问题以及 0-1 背包问题等.关于这方面的研究内容已经很丰富,我们将在这一节里,对其主要内容给予讨论.

用神经网络解决图与组合优化问题的基本原则是:将图与组合优化问题映射到神经网络模型上,使得问题的最优解对应所选定的神经网络模型的最小值点,即能量的最小点.因此,如何选定神经元与神经网络,进而如何构造能量函数是问题的关键,限于篇幅,我们打算在此对用神经网络技术解决优化问题的一般方法与步骤给出详细讨论,只对其中神经元的设置、能量函数的构造以及网络的运行方程给予讨论,其较为详细的讨论可参见文中所指出的有关文献.

1.1 基于神经网络技术的图的着色算法

图的着色问题具有良好的应用背景,在诸如工序问题、排课表问题和存储问题等均有直接应用^[3].然而,无论是求一个给定图 G 的色数 $\chi(G)$,还是对一个任意图 G 进行正常的顶点着色、边着色和全着色,都是困难的 NP-完全问题^[5].因此,图的着色问题受到众多学者的关注与兴趣.图 G 的一个正常 k -顶点着色,简称为图的 k -顶点着色,是指用 k 种颜色对 G 的每个顶点的一种分配,使得相邻的顶点分配(或称为“着”)到不同的颜色.

用 Hopfield 神经网络求解图的着色算法首先由 Dahl^[6]进行了研究.设 G 是一个阶数为 p 的简单图, G 的邻接矩阵为 $A = (a_{ij})_{p \times p}$,神经元的个数为 $p \times (\Delta + 1)$,其中 Δ 表示 G 的最大度数.神经元记为 (i, l) , $1 \leq i \leq p$, $1 \leq l \leq \Delta + 1$,它表示图 G 的顶点 i 与颜色 l 的关系.当网络处于稳定状态时,神经元 (i, l) 的输出为 v_{il} , $v_{il} \in \{0, 1\}$,若 $v_{il} = 1$,则表示图顶点 i 着颜色 l ;若 $v_{il} = 0$,则表示顶点 i 没有着颜色 l .由此所构造的能量函数为

$$E = \frac{A}{2} \sum_{i=1}^p \sum_{l=1}^{\Delta+1} \sum_{m=1, m \neq l}^{\Delta+1} v_{il} v_{im} + \frac{B}{2} \left(\sum_{i=1}^p \sum_{l=1}^{\Delta+1} v_{il} - p \right)^2 + C \sum_{i=1}^p \sum_{j=1, j \neq i}^p \sum_{l=1}^{\Delta+1} a_{ij} v_{il} v_{jl}.$$

文献[7]应用 Hopfield 神经网络对图的顶点着色、边着色以及全着色算法进行了系统的理论分析,理清了图的色换位矩阵与图的 k -着色之间的一一对应关系.改进了文献[6]中的不足,使能量函数得到简化,从而加快了网络的收敛速度,减少了局部极小值点等.其中为图的顶点着色所设计的能量函数为

$$E = \frac{A}{2} \sum_{i=1}^p \left(\sum_{l=1}^k v_{il} - 1 \right)^2 + B \sum_{i=1}^p \sum_{j=1, j \neq i}^p \sum_{l=1}^p a_{ij} v_{il} v_{jl}.$$

由此导出的网络运行方程为

$$\begin{aligned} \frac{du_{il}}{dt} &= -A \left(\sum_{l=1}^k v_{il} - 1 \right) - B \sum_{j=1, j \neq i}^p a_{ij} v_{jl}, \\ v_{il} &= \frac{1}{2} \left(1 + \tanh \frac{u_{il}}{\beta} \right). \end{aligned}$$

网络的权值 $w_{il, jm}$ 和外输入偏置电流 I_{il} 为

$$\begin{cases} w_{il,jm} = -A\delta_{ij} - Ba_{ij}\delta_{lm}(1 - \delta_{ij}), \\ I_{il} = A, \end{cases}$$

其中 k 表示对一个图 G 进行 k 着色的颜色数. 文献[7]中对于图的边着色和全着色算法也进行了研究, 其主要方法是将一个图的边着色、全着色算法分别转化为该图的线图、全图的顶点着色算法.

1.2 基于神经网络技术的图的同构算法

图的同构问题一直受到数学界与工程技术界关注, 特别是应用于系统建模与图的各种类型的构造算法. 应用 Hopfield 神经网络给出图的同构算法的有文献[8~10]等, 陈国良等人^[9,10]. 许进等在文献[8~10]的基础上应用 Hopfield 神经网络求解对图的同构问题进行了新的改进^[11], 通过建立相应的数学模型, 改进并简化了前人所提出的能量函数, 从而简化了网络的复杂性, 提高了网络的收敛速度. 现简介如下:

设 G 与 G' 表示两个同构图, $V(G) = \{x_1, x_2, \dots, x_p\}$, $V(G') = \{y_1, y_2, \dots, y_p\}$. 设神经元为 (x_i, y_j) , 它表示从 G 到 G' 之间的同构映射 σ 把图 G 中的顶点 x_i 映射到图 G' 中的顶点 y_j 的情况. 显然, 由此而构成的神经网络共有 $p \times p$ 个神经元. 当网络稳定时, 用 v_{ij} 表示神经元 (x_i, y_j) 的输出, 当 $\sigma(x_i) = y_j$ 时, $v_{ij} = 1$; 否则, $v_{ij} = 0$. 由此而构造的能量函数为^[11]

$$E = \frac{A}{2} \sum_{i=1}^p \left(\sum_{l=1}^p v_{il} - 1 \right)^2 + \frac{B}{2} \sum_{l=1}^p \left(\sum_{i=1}^p v_{il} - 1 \right)^2 + C \sum_{i=1}^p \sum_{l=1}^p \sum_{j=1}^p \sum_{m=1}^p |a_{ij} - a'_{lm}| v_{il} v_{jm},$$

其中 $A = (a_{ij})_{p \times p}$ 和 $A' = (a'_{lm})_{p \times p}$ 分别表示 p 阶图 G 和 G' 的邻接矩阵. 神经元 (x_i, y_j) 的运行方程为

$$\begin{aligned} \frac{du_{il}}{dt} &= -A \left(\sum_{l=1}^p v_{il} - 1 \right) - B \left(\sum_{i=1}^p v_{il} - 1 \right) - C \left(\sum_{l=1}^p \sum_{m=1}^p |a_{ij} - a'_{lm}| v_{lm} \right), \\ v_{il} &= \frac{1}{2} \left(1 + \tanh \frac{u_{il}}{\beta} \right), \end{aligned}$$

其中 $u_{il}(t)$ 表示神经元 (x_i, y_l) 在 t 时刻所处的状态, A, B, C 以及 β 表示实验参数. 网络的权值 $w_{il,jm}$ 和偏置电流 I_{il} 为

$$\begin{cases} w_{il,jm} = -A\delta_{ij} - B\delta_{lm} - C |a_{ij} - a'_{lm}|, \\ I_{il} = -(A + B), \end{cases} \quad i, j, l, m = 1, 2, \dots, p.$$

1.3 基于神经网络技术的图的最大团与最大独立集算法

设 S 是图 G 的一个顶点子集. 如果 S 中每对顶点在 G 中均不相邻, 则称 S 是图 G 的一个独立集; 如果 S 中每对顶点在 G 中均相邻, 则称 S 是图 G 的一个团. 设 S 是图 G 的一个团, 如果对于 G 中其他的任意一个团 S' , 均有 $|S'| \leq |S|$, 则称 S 是图 G 的一个最大团; 类似地可以定义最大独立集. 图的最大团问题与图的最大独立集问题都是困难的 NP-完全问题, 这两个问题实际上是等价的. 其原因是: S 是图 G 的独立集当且仅当 S 是 \bar{G} 的团 (其中 \bar{G} 是 G 的补图). 图的最大独立集问题不但在工程技术上有直接或者间接的应用, 而且在数学理论本身具有良好的应用, 如著名的 Ramsey 数问题^[12]. 因此, 给出具有快速准确的图的最大团或者最大独立集算法意义重大. 关于这方面的研究结果很多, 可见文献[13~19]. 陈国良等在文献[9]中用神经网络技术给出了求解图的最大团与最大独立集算法, 在此基础上, 文献[20]给出了基于 Hopfield 神经网络的图的最大团与最大独立集问题的一种新的算法, 此算法

通过理论分析, 简化了已有的能量函数, 其中所构造的能量函数为

$$E = -A \sum_{i=1}^p v_i + B \sum_{i=1}^p \sum_{j=1, j \neq i}^p v_i v_j (1 - a_{ij}).$$

网络的运行方程为

$$\frac{du_i}{dt} = -A + B \sum_{j=1, j \neq i}^p (1 - a_{ij}) v_j, \quad i = 1, 2, \dots, p.$$

进而可得连接权值 w_{ij} 和偏置电流 I_i 为

$$\begin{cases} w_{ij} = B(1 - a_{ij})(1 - \delta_{ij}), \\ I_i = -A, \end{cases} \quad i, j = 1, 2, \dots, p.$$

1.4 图的顶点、边覆盖问题的神经网络算法

若图 G 中的一个顶点和一条边相互关联, 则称它们相互覆盖. 覆盖图 G 的所有边的一个顶点子集称为图 G 的一个顶点覆盖. 类似地, 覆盖图 G 的所有顶点的一个边子集称为图 G 的一个边覆盖. G 的所有顶点覆盖中顶点最少的数目称为图 G 的顶点覆盖数, 或者简称为点覆盖数, 记为 $\alpha_0(G)$, 或简记为 α_0 .

类似地, 我们可以定义一个图 G 的边覆盖、最大边覆盖以及边覆盖数. 并用 α_1 来表示一个图 G 的边覆盖数. 我们分别用 β_0 和 β_1 来表示图 G 的独立数和匹配数. 业已得知, 关于一个阶数为 p 的非平凡的连通图的覆盖数与独立数具有如下关系:

$$\alpha_0 + \beta_0 = \alpha_1 + \beta_1 = p.$$

由此结果容易看出, 求一个图 G 的最小覆盖数等价于求这个图的最大独立集. 进而可知: 图的最小覆盖问题、图的最大团问题以及图的最大独立集问题两两等价. 因此, 求一个图的最小覆盖问题完全可以转化为求这个图的最大独立集问题, 或者最大团问题. 但由于图的最小覆盖问题的特殊性, 人们还是独立地应用神经网络方法对图的最小覆盖问题进行了研究. 下面, 我们来介绍应用 Hopfield 神经网络求解图的最小覆盖问题的算法^[9].

设 $V(G) = \{x_1, x_2, \dots, x_p\}$, 神经元为 x_i , 它表示图 G 的顶点 x_i 是否在 G 的某个覆盖集 S 之中. 文献[9]中所构造的能量函数为

$$E = \frac{A}{2} \sum_{i=1}^p \sum_{j=1, j \neq i}^p x_i x_j + \frac{B}{2} \sum_{i=1}^p \sum_{j=1, j \neq i}^p a_{ij} (1 - x_i)(1 - x_j),$$

其中第 1 项为优化项, 它表示 $(\sum_{i \in S} x_i)^2$, 即覆盖集大小的平方; 第 2 项是约束项, 只要 x_i 或 x_j 中的一个属于 S , 则相应的 $1 - x_i$ 或 $1 - x_j$ 为零, 该项取最小值.

网络的运行方程为

$$\frac{du_i}{dt} = -\frac{u_i}{\tau} - A \sum_{j=1}^p x_j + B \sum_{j=1}^p a_{ij} (1 - x_j).$$

1.5 基于神经网络技术的平面测试算法

平面测试问题(又称为可平面图嵌入问题)具有良好的应用背景, 例如, 在印刷电路的布线中, 人们感兴趣的是要求知道一个特定的电网络是否可嵌入平面^[21]; 另外, 平面测试问题在计算机科学的许多领域内都是一种非常有用的工具^[22]. 然而, 图的平面测试问题是一个困难的 NP-完全问题^[23]. 平面测试问题较早的研究工作有 Demucron, Malgranye 和 Pertuiset^[5], 他们是从图的一个圈入手, 然后找出关于这个圈的所有的桥, 再将每个桥逐一嵌

入到平面上. Hopcroft 和 Tarjan 通过附加路的方法, 首先建立了线性时间平面测试算法^[22], 他们的方法是在 Lempel 等的结果^[23]基础上产生的. 1976 年, Booth 和 Lucker 改进了文献^[22]中的结果, 其运行时间是线性的, 他们使用了一种称为 PQ-树的方法. 但上述两种方法均是非常复杂的, 并且每次迭代时构造的局部可平面子图需要不断修改才能获得最终的可平面图嵌入. 于是, 在 1993 年, Shih 和 Hsu 发展了一种简单的线性时间内的测试算法. 这种算法的基本思想仅仅基于深度优先搜索树(depth-first search tree)方法, 这种方法与前面两种方法相比的最大优点是每次迭代时不再需要修正^[24].

上述几种方法均属常规图论方法. Yoshiyasu^[25]给出求解平面测试问题的 Hopfield 神经网络算法. 我们近期在这方面作了更进一步的工作^[26], 现分别介绍如下:

设 G 是一个可平面图, 则下列解等价: 将一个给定的可平面图的顶点安排在一条直线上, 节点之间的连线分成两类——上连接与下连接, 上连接为节点间通过该直线的上方连接, 下连接为节点间通过该直线下方连接(如图 1 所示).

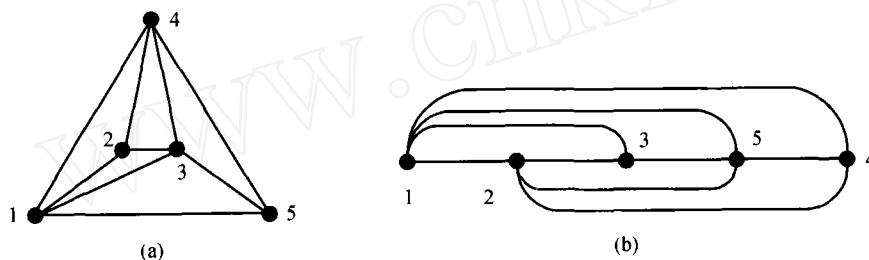


图 1 一个可平面图与它的直线型嵌入

设 G 是一个 p -阶可平面图. $V(G) = \{1, 2, \dots, p\}$. $A(G) = (a_{ij})$ 表示 G 的邻接矩阵. 我们现在来设计 $p \times p$ 个神经元 (i, j) , 它表示图 G 中的顶点 i 与 j 的连接与否. 网络稳定时神经元 (i, j) 的输出 x_{ij} 为: 若图的顶点 i 到顶点 j 有边连接, 则 $x_{ij} = 1$; 否则, $x_{ij} = 0$.

设 $V(G) = \{1, 2, \dots, p\}$, $A = (a_{ij})$ 表示 G 的邻接矩阵, 即

$$a_{ij} = \begin{cases} 1, & \text{顶点 } i \text{ 与顶点 } j \text{ 相邻,} \\ 0, & \text{否则.} \end{cases}$$

为了刻画一个可平面图的直线型嵌入时不允许相交的情况, 我们对 G 的两条边 $\{i, j\}$ 和 $\{l, m\}$ 引入一种为相交函数 $f(l, m, i, j)$: 若两条边 $\{i, j\}$ 与 $\{l, m\}$ 在直线型嵌入中相交, 则定义 $f(l, m, i, j) = 1$, 否则定义 $f(l, m, i, j) = 0$. 更确切地讲,

$$f(l, m, i, j) = \begin{cases} 1, & x_{ij} = x_{lm} = 1, l < i < m < j, \\ 1, & x_{ij} = x_{lm} = 1, i < l < j < m, \\ 1, & x_{ij} = x_{lm} = 1, m < j < l < i, \\ 1, & x_{ij} = x_{lm} = 1, j < m < i < l, \\ 0, & \text{否则.} \end{cases}$$

关于函数 $f(l, m, i, j)$ 定义的图示说明如图 2 所示.

文献^[25]中给出的能量函数为

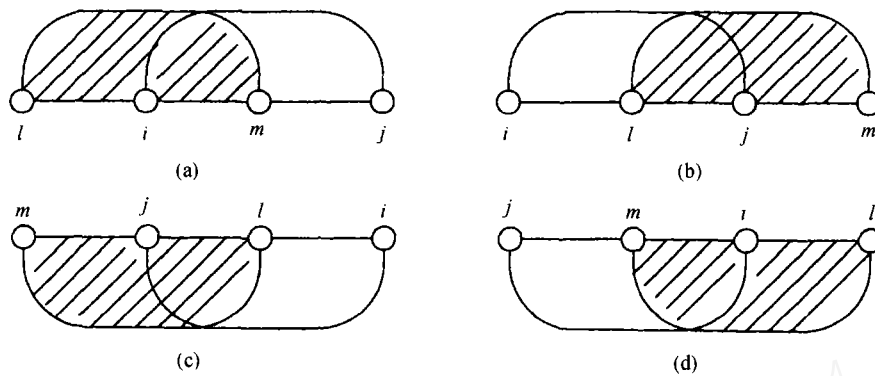


图 2 相交函数定义 4 种情况的图示

$$E = \frac{A}{2} \sum_i \sum_j (x_{ij} + x_{ji} - a_{ij})^2 + \frac{B}{2} \sum_i \sum_j \sum_l \sum_m f(l, m, i, j) x_{ij} x_{lm},$$

神经网络的运行方程为

$$\frac{du_{ij}}{dt} = -\frac{u_{ij}}{\tau} - A(x_{ij} + x_{ji} - a_{ij}) - B \sum_l \sum_m f(l, m, i, j) x_{ij} x_{lm}.$$

文献[26]在文献[25]的基础上给出的改进后的能量函数为

$$E = \frac{A}{2} \sum_i \sum_j (x_{ij} + x_{ji} - a_{ij})^2 + \frac{B}{2} \sum_i \sum_j \sum_l \sum_m \frac{f(l, m, i, j)(x_{ij} x_{lm} + x_{ji} x_{ml})}{1 - \alpha f(l, m, i, j)},$$

神经网络的运行方程为

$$\frac{du_{ij}}{dt} = -A(x_{ij} + x_{ji} - a_{ij}) - B \sum_l \sum_m \frac{f(l, m, i, j)(x_{ij} x_{lm} + x_{ji} x_{ml})}{1 - \alpha f(l, m, i, j)},$$

其中 x_{ij} 满足

$$x_{ij} = \frac{1}{2} \left(1 + \tanh \frac{u_{ij}}{\beta} \right).$$

1.6 基于神经网络技术的 TSP 问题的各种算法

TSP 问题可以描述为: 一个流动推销员推销他的产品到 p 个城市, 他本人出发前正处在这 p 个城市中的某一个, 要求到达其余每个城市必须一次且只有一次, 最后再回到起点城市, 并使得旅行路程最短. 这是一个众所周知的典型的 NP-完全问题, 有很多不同类型的解法.

用 Hopfield 神经网络求解 TSP 问题构造的神经元有两种: 第 1 种神经元, 记作 (x_i, l) , 它表示流动推销员在某一例旅行中, 第 l 次访问到城市 x_i , 若网络处于稳定状态时, 神经元 (x_i, l) 的输出记作 v_{il} , 定义为

$$v_{il} = \begin{cases} 1, & \text{若流动推销员第 } l \text{ 次访问城市 } x_i, \\ 0, & \text{否则,} \end{cases}$$

其中 $l = 1, 2, \dots, p$, 城市集为 $C = \{x_1, x_2, \dots, x_p\}$.

第 2 种神经元, 记作 (x_i, x_j) , 它表示流动推销员在某一例旅行中, 访问城市 x_i 之后访问城

市 x_j (记作 $x_i \rightarrow x_j$). 当网络处于稳定状态时, 神经元 (x_i, x_j) 的输出记作 v_{ij} , 定义为

$$v_{ij} = \begin{cases} 1, & \text{若 } x_i \rightarrow x_j, \\ 0, & \text{否则.} \end{cases}$$

第1种神经元是 Hopfield 等^[4]首先提出来的, 他们由神经元 (x_i, l) 出发, 构造的能量函数为

$$E = \frac{A}{2} \sum_{i=1}^p \sum_{l=1}^p \sum_{m=1, m \neq l}^p v_{il} v_{im} + \frac{B}{2} \sum_{i=1}^p \sum_{j=1, j \neq i}^p \sum_{l=1}^p v_{il} v_{jl} + \frac{C}{2} \left(\sum_{i=1}^p \sum_{l=1}^p v_{il} - p \right)^2 \\ + \frac{D}{2} \sum_{i=1}^p \sum_{j=1}^p \sum_{l=1}^p d_{ij} v_{il} (v_{j, l+1} + v_{j, l-1}),$$

其中前3项为约束项, 第4项为优化目标项. Hopfield 等在文献[4]中指出: 在20例模拟实验中, 有16例收敛于最优解或接近最优解. 然而, Wilson 在文献[27]中用同样的方法求解 TSP, 并未达到文献[4]中所给出的结果. 在100例重复实验中, 只有15例能在1000次迭代计算以内收敛到有意义的巡回路径上, 45例被“冻结”在无意义的旅行路线上, 其他40例在1000次迭代计算中不收敛.

Aiyer^[28]通过对网络特征值进行详细分析, 从超空间的角度解释了 Hopfield 网络求 TSP 问题时经常陷入无效解的原因, 并将能量函数修改为

$$E = \frac{A}{2} \sum_{i=1}^p \sum_{l=1}^p v_{il}^2 \sum_{m=1, m \neq l}^p v_{il} v_{im} + \frac{B}{2} \sum_{i=1}^p \sum_{j=1, j \neq i}^p \sum_{l=1}^p v_{il} v_{jl} + A_1 \sum_{i=1}^p \sum_{l=1}^p v_{il}^2 + \frac{C}{2} \left(\sum_{i=1}^p \sum_{l=1}^p v_{il} - p \right)^2 \\ - \left(\frac{A_p - A + A_1}{p} \right) \sum_{i=1}^p \sum_{l=1}^p \sum_{j=1}^p \sum_{m=1}^p v_{il} v_{jm} + \frac{D}{2} \sum_{i=1}^p \sum_{j=1}^p \sum_{l=1}^p d_{ij} v_{il} (v_{j, l+1} + v_{j, l-1}).$$

显然, 此能量函数的复杂程度足以使人“望而生畏”. 由此导出的网络运行方程必定复杂, 以致使网络运算量大, 收敛速度慢等.

在上述结果的基础上, 孙守宇等^[29]对能量函数作了进一步的简化, 他们提出的能量函数为

$$E = \frac{A}{2} \sum_{i=1}^p \left(\sum_{l=1}^p v_{il} - 1 \right)^2 + \frac{B}{2} \sum_{l=1}^p \left(\sum_{i=1}^p v_{il} - 1 \right)^2 + \frac{D}{2} \sum_{i=1}^p \sum_{j=1}^p \sum_{l=1}^p d_{ij} v_{il} v_{j, l+1}.$$

第2种神经元是由 Xu 和 Tsai 等提出来的, 他们视神经元 (x_i, x_j) 与神经元 (x_j, x_i) 的输出是等效的. 用第2种神经元处理 TSP 问题需要的神经元数目为 $\frac{1}{2}p(p-1)$ 个 (其中 p 为城市数), 并且他们认为用 $p(p-1)$ 个神经元更为有效. 由第2种神经元, Xu 和 Tsai 给出如下能量函数:

$$E = \frac{A}{2} \sum_{i=1}^p \left(\sum_{j=1, j \neq i}^p v_{ij} - 2 \right)^2 + \frac{B}{2} \sum_{j=1}^p \left(\sum_{i=1, i \neq j}^p v_{ij} - 2 \right)^2 + C \sum_{i=1}^p \sum_{j=i+1}^p v_{ij} d_{ij},$$

其中 d_{ij} 表示城市 x_i 与 x_j 之间的距离. 第1项保证最多有两个城市之间相连接; 第2项保证最多有两个城市与城市 j 相连接; 第3项是优化目标项. 式中 A, B 和 C 为大于零的常数. 由上式导出神经元 (x_i, x_j) 的动态方程为

$$\frac{du_{ij}(t)}{dt} = -u_{ij}(t) - A \sum_{j=1, j \neq i}^p v_{ij} - B \sum_{i=1, i \neq j}^p v_{ij} - Cd_{ij} + 2A + 2B.$$

遗憾的是,按上述能量函数所构成的网络计算 TSP 问题时常常会陷入产生子回路的困境.

在考虑神经元 (x_i, x_j) 与 (x_j, x_i) 不等效时,定义的能量函数为

$$E = \frac{A}{2} \sum_{j=1}^p \sum_{i=1, i \neq j}^p \left(\sum_{l=1, l \neq i, j}^p v_{il} - \frac{a}{d_{ij}} \right)^2 + \frac{B}{2} \sum_{i=1}^p \sum_{j=1, j \neq i}^p \left(\sum_{m=1, m \neq i, j}^p v_{mj} - \frac{a}{d_{ij}} \right)^2,$$

其中 a 为神经元的外加激励电压,取值应满足 $a/d_{ij} < 1$. 第 1 项限定了只有一个城市与城市 i 相连,且路径最短;第 2 项限定了最多只有一个城市与 j 相连且路径最短. 由上式导出的神经元动态方程为

$$\frac{du_{ij}(t)}{dt} = -\frac{u_{ij}(t)}{\tau} - A \sum_{l=1, l \neq i, j}^p \left(v_{il} - \frac{a}{d_{ij}} \right)^2 - B \left(\sum_{m=1, m \neq i, j}^p v_{mj} - \frac{a}{d_{ij}} \right).$$

令 $A = B, a_1 = 2a$ 有

$$\frac{du_{ij}(t)}{dt} = A \left(\frac{a_1}{d_{ij}} - \frac{u_{ij}}{A\tau} - \sum_{l=1, l \neq i, j}^p v_{il} - \sum_{m=1, m \neq i, j}^p v_{mj} \right).$$

在 TSP 问题的研究中,无论是采用第 1 种神经元还是第 2 种神经元,我国学者都做了一些富有意义的创新结果. 诸如孙守宇等^[29]对基于第 1 种神经元的能量函数的简化改进;张忆军、叶中行^[30]对文献[4]中所获 Hopfield 神经网络权值进行了改进;余道衡及他的研究小组也做了许多改进与创新^[31];靳蕃及他的研究小组提出满意度的概念,以求 TSP 的满意解^[32];樊社民等¹⁾在采用第 1 种神经元研究的同时,也对第 2 种神经元中如何消除子回路问题进行了探索.

1.7 图的划分问题

图的划分问题是指将一个图的顶点集划分成若干个互不相交的顶点子集,使其端点在这些不同的顶点子集的所有的边的数目最少. 图的平衡划分问题(balanced partitioning problem),是指图的划分顶点子集中,最大的顶点子集的元素个数与最小的顶点子集的元素个数之差至多为 1. 当图的划分的顶点子集的数目为 2 时,通常称为图的对分问题. 更详细地讲,图 $G = (V, E)$ 的一个 0/1 平衡对分是指对 V 的一个剖分,即 V 可分为两个子集 V_1, V_2 , 使得:

- (i) $V_1 \neq \emptyset, V_2 \neq \emptyset$, 且 $V_1 \cup V_2 = V$;
- (ii) $V_1 \cap V_2 = \emptyset$;
- (iii) 当 $|V|$ 为偶数时, $||V_1| - |V_2|| = 0$, 否则为 1.

本文用 $f(V_1, V_2)$ 来表示图 G 的割 (V_1, V_2) 中边的数目.

图的划分问题产生于计算机科学的许多领域,诸如并行计算、稀疏矩阵因子化、网络划分,以及 VLSI 电路设置等. 特别,划分问题可建立数据的分布于具有分布式记忆的多处理系统计算机之中的模型,其中计算工作量必须平衡分配于不同的处理器之中,且通讯总量必须最小化^{[33~37]2)}.

图的对分问题是一个基本问题,已得到广泛的研究^[38~40]. 这个问题对一般的图而言是一个 NP-困难问题,对偶图而言,也是一个 NP-困难问题^[41],甚至对一般图或者可平面图,寻

1) 樊社民. 神经网络与组合优化. 博士学位论文, 西安交通大学, 1993

2) Williams R. Unification of spectral and intertial bisection. Technical Report, California Inst of Technology, 1994, <http://www.csrcr.caltech.edu/roy/paper/>

找一个较好的近似解也是困难的 NP-完全问题^[42]. 近似算法对偶划分问题(即图的两个子集元素的数目不要求相等的划分问题)是有效的,但到目前为止还未发现偶划分问题有多大的实际意义^[43].

用贪心试探法求解图的划分问题的不同方面在文献[35,44~47]中已经给予较为详细的讨论;应用模拟退火方法对图的划分问题的研究已经有不少的结果,可参见文献[48~53]等. 文献[54]在 Bui 等^[55]文章的基础上,对图的对分问题前面的研究工作和试探法给出了详细的讨论,对它的解提出了一种所谓的技术-状态遗传算法,并称为 BFS-GBA 方法,并与模拟退火算法以及经典的 Kernighan-Lin(KL)算法^[56]等进行了广泛的比较. KL 算法的不同的实现方面的讨论可参见文献[57~59]. 文献[55]对 KL, SA 和 GA 试探法在图的划分问题上给出一个较为详细的讨论. Tabu 搜索算法是由 Glover 提出来的^[60],这种算法在图的划分问题上的应用见文献[54,61,62]. 特别,文献[63,64]提出应用一般偶划分的遗传算法来解决 0-1 偶划分问题的算法,但文献[62]认为是不适应的,原因是它是由随机化的贪心过程^[46]所证明的. 多级技术在图的划分问题上的应用见文献[58,65].

Ramanujam 等首先应用神经网络方法对具有赋权型的图的划分问题进行了研究,此后,有文献对基于神经网络方法的图的划分问题也进行了讨论. 赋权型的图的划分问题可按照 2-划分和 k -划分两类来处理.

图的 2-划分问题: 设 $G = (V, E)$ 是一个具有 p 个顶点, q 条边的简单无向赋权图. 令 $V = \{v_1, v_2, \dots, v_p\}$, 顶点 v_i 有权 $w_i \geq 0$, 边 $v_i v_j$ 有权为 $e_{ij} \geq 0$. 图 G 的 2-划分是指将 G 的顶点划分成两个不相交的子集, 设为 V_1 和 V_2 ($V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$), 使得在这两个顶点子集中所含顶点权和 $\sum_{j \in V_i} w_j$ ($i = 1, 2$) 尽可能相等的情况下, V_1 与 V_2 之间的边权之和最小, 即需求

$$\min \sum_{i \in V_1} \sum_{j \in V_2} e_{ij}$$

之值, 这是一个 NP-完全问题, 容易验证总共有 $2^p - 2$ 种可能的划分方法.

对具有 p 个顶点的图 $G = (V, E)$, 共设计 p 个神经元, 分别记为 $1, 2, \dots, p$, 当神经网络稳定时, 神经元 i 的输出 x_i 为 0, 或者为 1: $x_i = 1$ 表示其对应的顶点 v_i 属于某一个划分; 而 $x_j = 0$ 所对应的顶点 v_j 属于另外一个划分. 在此基础上可构造出如下的能量函数:

$$E = \frac{A}{2} \sum_{i=1}^p \sum_{j=1}^p e_{ij} (x_i + x_j - 2x_i x_j) + \frac{B}{2} \sum_{i=1}^p \sum_{j=1}^p w_i w_j (1 + 2x_i x_j - x_i - x_j),$$

其中 E 的第 1 项为优化目标项, 它最小化了两个划分之间的边权之和; 第 2 项为惩罚项, 只有当划分的顶点权和尽可能地相等时, 该项才取最小值.

基于能量函数 E 所导出的神经网络的动力学方程为

$$\begin{aligned} \frac{du_i}{dt} &= -\frac{u_i}{\tau} - A \sum_{j=1}^p e_{ij} (1 - 2x_j) + B \sum_{j=1}^p w_i w_j (2x_j - 1), \\ x_i &= \frac{1}{2} \left(1 + \tanh \frac{u_i}{\beta} \right). \end{aligned}$$

赋权型图的 k -划分问题: 图 G 的 k -划分问题是指将图 G 的顶点剖分为 $k \geq 2$ 个非空的、彼此不相交的子集 V_1, V_2, \dots, V_k , 在满足 $\sum_{i \in V_1} w_i, \sum_{i \in V_2} w_i, \dots, \sum_{i \in V_k} w_i$ 两两之间尽

可能相等的条件下, 求

$$\min \sum_{i=1}^k \sum_{j=i+1}^k f(V_i, V_j).$$

设 $V(G) = \{v_1, v_2, \dots, v_p\}$, 共设计 $p \times k$ 个神经元, 记为 (v_i, j) , $1 \leq i \leq p$, $1 \leq j \leq k$, 它表示图 G 的顶点 v_i 与划分子集 V_j 之间的关系. 当网络处于稳定状态时, 神经元 (v_i, j) 的输出记为 $x_{ij} \in \{0, 1\}$, $x_{ij} = 1$ 当且仅当图 G 的顶点 $i \in V_j$. 文献[77]中给出的能量函数为

$$E = \frac{A}{2} \sum_{i=1}^p \sum_{l=1}^k \sum_{y=1, y \neq l}^k x_{il} x_{iy} + \frac{B}{2} \left(\sum_{i=1}^p \sum_{l=1}^k x_{il} - p \right)^2 + \frac{C}{2} \sum_{i=1}^p \sum_{j=1}^p \sum_{l=1}^k w_i w_j x_{il} x_{jl} \\ + \frac{D}{2} \sum_{i=1}^p \sum_{j=1}^p \sum_{l=1}^k \sum_{y=1}^k e_{ij} x_{il} x_{jy}.$$

E 的前 3 项为惩罚项, 其中第 1 项为行约束, 当且仅当矩阵的每一行不多于一个“1”时它才为零; 第 2 项为全局约束项, 当且仅当矩阵中所有“1”的个数为 p 时才为零; 第 3 项只有各划分中顶点权和相等时才取最小值, 容易看出, 此项利用了算术平均与几何平均相等时取最小值的原理; 第 4 项为优化目标项, 它用来最小化各划分子集之间的边权和.

基于上述能量函数, 所得到的网络的运行方程为

$$\frac{du_{il}}{dt} = -\frac{u_{il}}{\tau} - A \sum_{y=1, y \neq l}^k x_{iy} - B \sum_{j=1}^p \sum_{l=1}^k x_{jl} - C \sum_{j=1}^p w_i w_j x_{jl} - D \sum_{j=1}^p \sum_{y=1, y \neq l}^k e_{ij} x_{jy}, \\ x_{il} = \frac{1}{2} \left(1 + \tanh \frac{u_{il}}{\beta} \right).$$

1.8 装箱问题

装箱问题是一个 NP- 完全问题, 此问题是在许多实际问题中可以直接应用的一种组合优化模型, 关于这方面研究的结果已经很多, 目前在人工智能中已经有各种搜索算法或近似快速算法^[8]. 我们拟在此小节里仅对神经网络方法给予介绍. 装箱问题可描述如下:

有体积为 V_1, V_2, \dots, V_m 的 m 个货物, 装在体积为 L 的箱子里 ($V_i \leq L$, $i = 1, 2, \dots, m$), 试找出一种装法, 使得所使用的箱子最少.

用神经网络方法求解装箱问题与前面几个用神经网络方法求解的问题有所不同, 因为这里的神经元数目是一个待定的数, 我们不知道最佳情况下用多少个箱子, 但是, 箱子数有一个上界与下界. 当然, 最理想的情况是每个箱子都装满, 这种情况下的箱子数为

$$N_{\min} = \frac{1}{L} \sum_{i=1}^m V_i.$$

由于每个体积为 V_i ($1 \leq i \leq m$) 的货物是整体, 因此, 不同的装法会导致每个箱子或多或少地留有空间, 但无论如何, 箱子数不能超过 $N_{\max} = 2N_{\min} + 1$. 这是因为, 当 $N = N_{\min}$ 时, 平均来讲, 每个箱子中装入的货物体积小于等于 $\frac{1}{2}L$, 因此, 只需要将这两个箱子合并, 便可减少一个. 因此, 所需箱子数, 记作 N , 满足

$$N_{\max} \leq N \leq 2N_{\min} + 1.$$

用 (i, j) 来表示神经元, 它描述了第 i 个箱子与第 j 个货物的状态关系, 其中 $1 \leq i \leq N$, $1 \leq j \leq m$, 因此, 总共有 $N \times m$ 个神经元, 它们构成方阵的状态来表示装箱情况. 当网络处于稳定状态时, 神经元 (i, j) 的状态用 x_{ij} 来表示, 其中 $x_{ij} = 1$ 表示可将第 j 货物装入在第 i

个箱子中, 否则 $x_{ij} = 0$.

基于此, 我们给出此问题的一种能量函数如下^[8]:

$$E = \frac{A}{2} \sum_{i=1}^N \left(\sum_{j=1}^m x_{ij} - 1 \right)^2 + \frac{B}{2} \sum_{j=1}^m x_{ij} f(i),$$

其中第1项表示第 i 种货物必须装在某一箱子内, 第2项表示每一个箱子应尽可能装满,

$$f(i) = \begin{cases} 1, & \sum_k x_{ik} > L, \\ 0, & L > \sum_k x_{ik} > L - V_j, \\ -1, & \sum_k x_{ik} < L - V_j. \end{cases}$$

符合装箱问题的约束条件的动力学方程为

$$\frac{dx_{ij}}{dt} = -\frac{u_{ij}}{\tau} - A \left(\sum_{k=1}^N x_{kj} - 1 \right) - B f(i).$$

因刚开始不知道究竟有几个箱子, 因此, 只能在箱子数的上限与下限之间从小到大都做一遍, 寻求最佳箱子数, 而每给定一个箱子数 N , 则用 $N \times m$ 的网络来寻求最优装法. 当然, 我们在寻求箱子数 N 时, 可应用一些优化技术, 如黄金分割法等.

1.9 图的最大割集问题

设 G 是一个无向图, V 和 E 分别是它的顶点集和边集. 图的最大割集问题是寻找 V 的一个子集 S , 使得 (S, \bar{S}) 之间的权和最大, 即求满足

$$\max \sum_{e \in (S, \bar{S})} w(e)$$

的 S , 其中, $\bar{S} = V - S$, (S, \bar{S}) 表示一个端点在 S 上, 而另一个端点在 \bar{S} 上的 G 的全体边构成的 E 的一个子集, 这个子集显然是图 G 的一个边割集. 求一个边赋权图, 或者非赋权的图的最大割集问题均是 NP-完全问题^[66]. 对于几种特殊的图类(如可平面图^[67]和循环图^[68])已有快速的多项式算法. 最大割集问题已在许多方面有良好的应用, 诸如 VLSI 的设计与统计物理学等^[69,70]. 关于这方面的研究算法可参见文献^[67~74]等. 文献^[75]用二进制神经网络来求解图的最大割集问题. 设置神经元为图的顶点. 因此, 具有 p 个顶点的图所用的神经元共有 p 个, 记为 $\{x_1, x_2, \dots, x_p\}$. 当网络稳定时, 若图 G 的第 i 个顶点 $v_i \in S$, 则 $x_i = 1$; 否则, $x_i = 0$. 其对应的能量函数为

$$E = \frac{A}{2} \sum_{i=1}^p \sum_{j=1}^p e_{ij} ((1 - x_i)(1 - x_j) + x_i x_j).$$

我们在此略去网络的运行方程等.

1.10 其他一些问题

限于篇幅, 我们在这里将简要地提一下其他一些问题及研究概况, 并标出应用神经网络技术解决的文献. 用常规方法求解图的划分问题的一篇最新的研究结果是文献^[56]; 基于神经网络方法的图的最大匹配问题的一种算法见文献^[76]; 应用神经网络方法求解 0-1 背包问题可见文献^[9]; 作业调度问题已有多种方法进行过讨论, 应用神经网络方法的可见文献^[77,78]; 用神经网络求解模拟数字转换问题, 分解-判定问题(decomposition-decision problem)以及线性规划问题见文献^[79]; 各种类型的排序问题, 工序问题见文献^[10]; 基于神经网络

技术的 Steiner 树问题的研究见文献[80]; 文献[78]中对基于神经网络技术的任务分配问题, 以及所谓的八皇后问题(eight queen problem)给予了讨论.

2 图论在人工神经网络中的应用

本节主要综述图论作为一种很有用的工具在人工神经网络中的应用, 诸如应用于前向神经网络模式分类的研究、反馈人工神经网络稳定性分析等^[81~83].

2.1 图论在前向神经网络中的应用

前向人工神经网络的两个最基本的功能是分类和函数逼近, 因而被广泛地应用于模式识别、信号处理等许多领域. 离散前向人工神经网络用于分类问题的一个主要课题是 Boole 函数的线性可分性问题. 判别一个 Boole 函数是否为线性可分的, 这个问题不但是神经网络的基本问题, 对网络结构的设计有意义, 而且在计算机科学许多问题的研究中具有直接或者间接的应用. 然而, 关于这方面的研究结果是比较少的. 关于线性可分 Boole 函数的计数问题仅解决了维数 $n \leq 8$ 的情形(如表 1 所示)^[1,2].

表 1 维数 ≤ 8 的线性可分的 Boole 函数的数目

n	1	2	3	4	5	6	7	8
$LB(n)$	4	14	104	1882	94572	15028134	8378070864	17561539552946

文献[84,85]中对于线性可分的 Boole 函数的有关基本性质进行了研究, 应用图论的方法给出了判别一个 Boole 函数线性可分的充要条件. 当维数 $n \geq 9$ 时, n 维线性可分的 Boole 函数的计数问题是一个悬而未解的问题. 如何把 n 维线性可分的 Boole 函数构造出来更是一个没有解决的难题. 目前对维数 $n \leq 4$ 的线性可分 Boole 函数的完全问题已在文献[86]中给予解决. 但当维数 $n \geq 5$ 时, 如何把 n 维 Boole 函数全部构造出来是一个困难的尚待解决的问题. 这是因为, 随着 Boole 函数维数 n 的增加, 线性可分 Boole 函数的数目将剧增, 这个事实可以通过表 1 及线性可分 Boole 函数数目的估计式得以证实^[1]:

$$LB(n) \simeq 2 \frac{2^n}{n!},$$

其中 $LB(n)$ 表示 n 维线性可分的 Boole 函数的个数. 为了构造出全部的, 或者我们感兴趣的 n 维线性可分 Boole 函数, 我们首先将它们分为完全同构的等价类. 然后对每个等价类, 构造出一个代表即可.

本文以图论为工具来研究 Boole 函数的线性与非线性可分性. 其基本思想是首先将一个 Boole 函数转化为 n 维超立方体的子图. 然后通过刻画这些子图的基本特征来研究 Boole 函数的线性可分性问题. 为此, 我们先来介绍一下有关的一些概念与记号.

2.1.1 有关概念与定义

设 A, D 表示两个非空有限集, 由 A 到 D 的全体映射构成的集合记为 D^A , 即

$$D^A = \{f; f: A \rightarrow D\},$$

称为 A 到 D 的幂函数. 设 $B = \{0, 1\}$ (有时取 $B = \{-1, 1\}$). n 维 Boole 空间, 记作 B^n , 定义为

$$B^n = \{(x_1, x_2, \dots, x_n); x_i \in B, i = 1, 2, \dots, n\}.$$

n 维 Boole 函数, 记作 f , 是指 B^n 到 B 的一个映射

$$f: (x_1, x_2, \dots, x_n) \rightarrow x,$$

其中, $(x_1, x_2, \dots, x_n) \in B^n, x \in B$. 全体 n 维 Boole 函数构成的集合显然是幂函数 B^{B^n} . 易知

$$|B^n| = 2^n, \quad |B^{B^n}| = 2^{2^n}.$$

n 维超立方体, 我们仍然采用记号 B^n , 它是一个具有良好对称性的 n -正则图, 其顶点集 $V(B^n)$ 是由 B^n 中给出的 Boole 空间中的元素, 并且 B^n 中任意两个元素 $X = (x_1, x_2, \dots, x_n)$ 与 $Y = (y_1, y_2, \dots, y_n)$ 相邻当且仅当 X 与 Y 的 Hamming 距离 (即对应分量不同元素的个数) $d_H(X, Y) = 1$.

设 \mathbb{R}^n 表示 $n (\geq 2)$ 维实空间, $X_1, X_2, \dots, X_m \in \mathbb{R}^n$ 为超立方体的顶点, 可以用列向量表示. 对这 m 个点用红和蓝两种颜色进行着色, 着红色的点 X_i 用 X_{ir} 来表示, 蓝色的点 X_j 用 X_{jb} 来表示. 如果在 \mathbb{R}^n 中存在一个 $(n-1)$ 维的超平面 $WX - \theta = 0$ (其中, $W = (w_1, w_2, \dots, w_n)$, X 是列向量), 使得对于上述所有着红色的 X_{ir} , 均有 $WX_{ir} - \theta \geq 0$; 而对于上述所有着蓝色的点 X_{jb} , 均有 $WX_{jb} - \theta < 0$, 则称 \mathbb{R}^n 中这 m 个点关于红色与蓝色是线性可分的. 否则称为线性不可分的, 或者非线性可分的.

n 维 Boole 函数的定义域为 n 维 Boole 空间 B^n . 类似于 \mathbb{R}^n 中线性可分性问题的定义, 我们对线性可分 Boole 函数给出如下的定义: 设 $f \in B^{B^n}$. 不失一般性, 令

$$f(x_1, x_2, \dots, x_n) = x.$$

如果在 \mathbb{R}^n 中存在一个 $(n-1)$ 维的超平面 $\pi: WX - \theta = 0$, 它能够把等于 1 的所有顶点构成的集合从所有等于 0 的顶点构成的集合中区分出来, 则称 f 是线性可分的. 否则为线性不可分的, 或者非线性可分的.

2.1.2 Boole 函数与超立方体导出子图的对应转换

设 $f \in B^{B^n}$, $X \in B^n$, 如果 $f(X) = 1$, 则给 n 维超立方体 B^n 所对应的顶点 X 着以红色; 如果 $f(X) = 0$, 则给 n 维超立方体 B^n 所对应的顶点 X 着以蓝色. 于是, f 按照上述法则惟一地对对应 n 维超立方体 B^n 的一种 2-着色; 我们把着红色顶点构成的子集记为 V_r ; 着蓝色顶点所构成的子集记为 $V_b = V(B^n) - V_r$. 显然, 当 V_r 确定后, V_b 也随之确定. 因此, 我们可以说, $\forall f \in B^{B^n}$, f 按照上述法则, 惟一地对对应 B^n 中的一个导出子图 $G[V_r]$, 特别, 当 $V_r = \emptyset$ (空集) 时, 即对于 B^n 中每个元素 X , 均有 $f(X) = 0$ 时, f 对应于不含顶点和边的空集子图. 我们把导出子图 $G[V_r]$ 称为由 f 导出的子图, 或简称为 f 的导出子图, 并记作 G_f .

反过来, 对于 B^n 中任一点导出子图 $G[V_r]$, 我们构造一个 n 维 Boole 函数 f 如下: 如果 $G[V_r] = \emptyset$, 则定义

$$f(X) = 0, \quad \forall X \in B^n, \quad (1)$$

如果 $G[V_r] \neq \emptyset$, 则定义

$$f(x_1, x_2, \dots, x_n) = \begin{cases} 1, & (x_1, x_2, \dots, x_n) \in V_r, \\ 0, & \text{否则.} \end{cases} \quad (2)$$

显然,按照上述所构造的 Boole 函数 f 是惟一的. 如果用 $\rho(B^n)$ 表示 n 维超立方体 B^n 中包括空集在内的所有顶点导出子图构造的子集族,则我们容易得到下列结果:

定理 1 B^{B^n} 按照(1)和(2)式所给出的定义式同构于 $\rho(B^n)$.

由此结果,我们可以惟一地将一个 n 维 Boole 函数转化为 B^n 的一个子图.

基于定理 1,今后称 Boole 函数是(非)线性可分的与 G_f 是(非)线性可分的为同义语.

我们现在通过下例对上述将一个 n 维 Boole 函数等价地转换成 B^n 的一个顶点导出子图的方法给予说明.

例 1 一个 3 维 Boole 函数 f 所对应的真值表如下:

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

f 所对应的子超立方体 G_f 由图 3 给出,其中 $V(G_f)$ 上标定的十进制数对应相应的二进制.

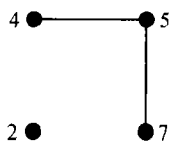


图 3 Boole 函数与图的转化过程说明图

2.1.3 基于图论方法的 Boole 函数线性与非线性可分性的一些基本结果

设 $X_1, X_2, Y_1, Y_2 \in B^n$, $X_1X_2, Y_1Y_2 \in E(B^n)$, 如果 $X_2 - X_1 = Y_2 - Y_1$, 则称 X_1X_2 与 Y_1Y_2 平行; 如果 $X_2 - X_1 = -(Y_2 - Y_1)$, 则称 X_1X_2 与 Y_1Y_2 反向平行.

设 $V_r \subseteq V(B^n)$ 表示红色顶点集, $\bar{V}_r = V_b = V(B^n) - V_r$ 表示蓝色顶点集. 我们用 (V_r, \bar{V}_r) 表示红色顶点集到蓝色顶点集之间的边的集合.

定理 2 设 $f \in B^{B^n}$, 由 f 导出的 B^n 的 2-着色中红色顶点集为 V_r . 如果在 (V_r, \bar{V}_r) 中存在反向平行线, 则 f 是非线性可分的.

定理 3 设 $f \in B^{B^n}$, 由 f 导出的 B^n 的 2-着色中红色顶点集为 V_r . 如果在 (V_r, \bar{V}_r) 无任何平行线, 则 f 是非线性可分的.

定理 4 设 $f \in B^{B^n}$.

(i) 若 G_f 不连通, 则 f 是非线性可分的;

(ii) 如果 G_f 是一个 $m \leq n$ 的子超立方体, 则 f 是线性可分的;

(iii) 如果 $G_f \cong K_{1,m}$, 即 G_f 是一个星图, 则 f 是线性可分的;

(iv) 若 $G_f = B^m \cup K_2 \cup \dots \cup K_2$, 且 $B^m \cap K_2 \cap \dots \cap K_2 = K_1$, 则 f 是线性可分的.

由上述结果, 我们很容易推出, 3 维以及 4 维线性可分的 Boole 函数分别可由图 4 与 5 中所对应的超立方体的子图给出. 由这些子图, 我们可以很清楚地知道 3 维和 4 维线性可分 Boole 函数的结构等特征.

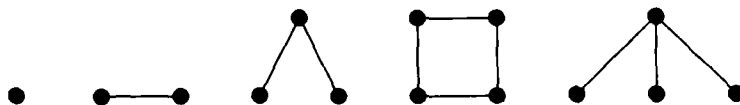


图 4 3 维线性可分的 Boole 函数所对应的子图

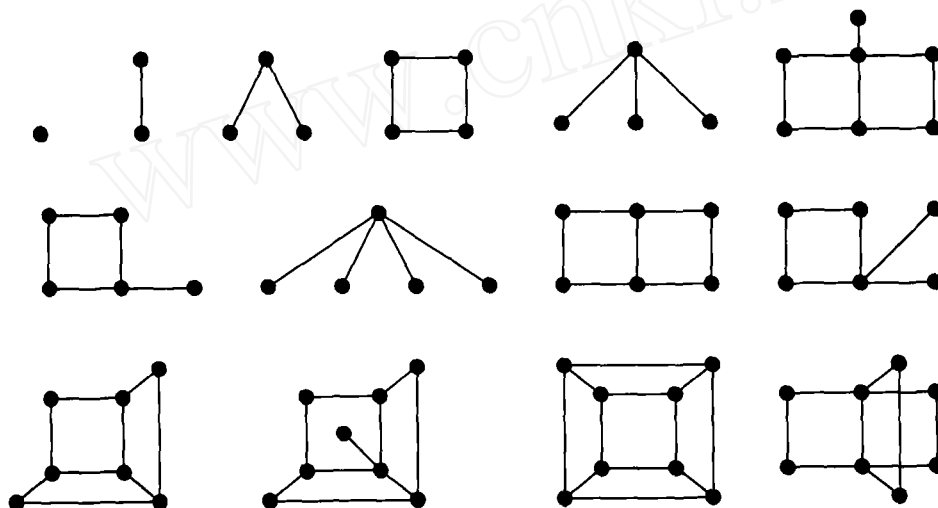


图 5 4 维线性可分的 Boole 函数所对应的子图

我们提出下列问题:

问题 1 B^n 中线性可分的顶点导出子图有多少个? 或者等价地讲, n 维非标定的 Boole 函数有多少个?

通过图 4 与 5, 我们已经知道, 非标定的 3 维线性可分的 Boole 函数是 5 个(不含空集子图, 下同), 4 维非标定的线性可分的 Boole 函数共有 14 个^[85].

非标定线性可分 Boole 函数的计数问题是计数线性可分 Boole 函数的同构类. 相应地, 标定线性可分 Boole 函数的计数问题是指通常所言的线性可分 Boole 函数的计数问题.

问题 2 $LB(n)$ 是多少? 其中 $n \geq 9$ (若 $n \leq 8$ 见表 1); 如何构造出 n 维全体线性可分的 Boole 函数?

2.2 图论在离散反馈型神经网络中的应用

对于任何离散反馈型网络而言, 它们的值域总是在某一个超立方体的顶点之内. 因此,

当一个离散型神经网络处于某种状态,特别是稳定状态或者是极限环稳定状态时,网络的状态应该是一个阶数为 2^n 的有向图.当然,若我们能够获得网络在各种类型下的稳定状态图时,我们可以说对这个网络的稳定性搞清楚了.离散型 Hopfield 神经网络是整个离散反馈型人工神经网络的基础,因此,我们在此主要以 Hopfield 神经网络模型为背景展开讨论.为此,先介绍一下离散型 Hopfield 神经网络的基本模型.

2.2.1 离散 Hopfield 神经网络模型

具有 p 个神经元的离散 Hopfield 神经网络 N (以后常简记为 DHNN),其拓扑结构可以由一个 $p \times p$ 矩阵 $W = (w_{ij})_{p \times p}$ 和一个 p 维列向量 $\theta = (\theta_1, \theta_2, \dots, \theta_p)^T$ 所惟一确定,并记为 $N = (W, \theta)$,其中 w_{ij} 表示神经元 i 与神经元 j 之间的连接强度(或权值), θ_i 表示神经元 i 的阈值($i, j = 1, 2, \dots, p$).若用 $x_i(t)$ 表示神经元 i 在 t 时刻所处的状态,并且只有两种:兴奋(用 $x_i(t) = 1$ 表示)和抑制(用 $x_i(t) = -1$ 或 0 表示,这里用 -1), $t \in \{0, 1, 2, \dots\}$. DHNN 的演化方程(又称运行方程等)为

$$x_i(t+1) = \operatorname{sgn}\left(\sum_{j=1}^p w_{ij} x_j(t) - \theta_i\right), \quad i = 1, 2, \dots, p,$$

其中

$$\operatorname{sgn}(x) = \begin{cases} 1, & x \geq 1, \\ -1(\text{或 } 0), & \text{否则.} \end{cases}$$

若令 $X(t) = (x_1(t), x_2(t), \dots, x_p(t))^T$, $\theta = (\theta_1, \theta_2, \dots, \theta_p)^T$, $W = (w_{ij})_{p \times p}$, 则上式可重新写成

$$X(t+1) = \operatorname{sgn}(WX(t) - \theta).$$

设 $N = (W, \theta)$ 是一 DHNN, N 从初态 $X(t_0)$ 开始,经过一个有限的时刻 t 之后,网络的输出不再发生变化,即

$$X(t+1) = X(t),$$

则称此网络处于稳定状态,并称 $X(t) = X(t+1)$ 是 N 的一个稳态点(或称稳定状态),且称此网络关于初始状态 $X(t_0)$ 收敛(或稳定).易知,若 X 是 DHNN $N = (W, \theta)$ 的一个稳态点,则有

$$X = \operatorname{sgn}(WX - \theta).$$

有时我们把满足此式的 X 称为网络 N 的稳定吸引子或简称为吸引子.若 X 是 DHNN $N = (W, \theta)$ 的一个吸引子, X 的吸引域,记作 $\Gamma(X)$,它表示全体吸引到 X 的初始状态 $X(t_0)$ 构成的集合.

设 $N = (W, \theta)$ 是一 p 阶 DHNN, X_1, X_2, \dots, X_r 是 $r (\geq 2)$ 个二值 $(-1$ 和 $1)$ p 维列向量,如果

$$X_{j+1} = \operatorname{sgn}(WX_j - \theta), \quad j = 1, 2, \dots, r-1, \quad X_1 = \operatorname{sgn}(WX_r - \theta),$$

则称 X_1, X_2, \dots, X_r 为 N 的一个长度 r 的极限环吸引子,记作 (X_1, X_2, \dots, X_r) ,有时简称为环吸引子,或极限环.类似于稳定吸引子的吸引域,环吸引子 (X_1, X_2, \dots, X_r) 的吸引域,记作 $\Gamma(X_1, X_2, \dots, X_r)$,它表示全体吸引到 (X_1, X_2, \dots, X_r) 的所有可能的初始状态构成的集合.对于 p 阶 DHNN,全体可能的状态集 B^p 中,我们取 $B = \{-1, 1\}$. $N = (W, \theta)$ 称为对称的,如果 $W = W^T$ (转置), N 称为反对称的,如果 $W = -W^T$.

2.2.2 网络状态图

以图论为工具引入网络状态图的概念,可清楚地反映该网络的状态结构,以及网络的各种类型的吸引子等稳定状况,是深入研究 DHNN 的一个很好的工具.

先引入网络状态图的概念,并讨论网络状态图的有关基本性质. p 阶 DHNN, $N = (W, \theta)$, 其网络状态图记作 G_N , 是一个 2^p 阶的定向图. G_N 的顶点集为 B^p , 定向弧集为

$$E(G_N) = \{(X_i, X_j) \mid X_i, X_j \in B^p, X_j = \text{sgn}(WX_i - \theta)\},$$

于是 G_N 的汇点对应于 N 的稳定点, G_N 的有向圈对应于网络 N 的环吸引子.

我们通过下面的例子来说明网络状态图的优点: 它可以清楚地反映网络是否收敛. 网络有多少个稳定吸引子, 有多少个环吸引子以及每个吸引子和每个环吸引子的吸引域的大小.

例 2 设 $N = (W, \theta)$ 是一 4 阶 DHNN, 其中 $\theta = 0$.

$$W = \begin{pmatrix} 0 & 1 & 2 & -1 \\ 1 & 0 & 1 & -1 \\ 2 & 1 & 0 & 3 \\ -1 & -1 & 3 & 0 \end{pmatrix}, \quad \text{sgn}(x) = \begin{cases} 1, & x \geq 0, \\ -1, & x < 0. \end{cases}$$

N 共有 $2^4 = 16$ 个状态. 为了方便, 我们用十进制数来代表二进制中对应的向量(对于一般的情况也是如此处理), 于是, 我们有

$$\begin{aligned} 0 &= (-1, -1, -1, -1)^T, & 1 &= (-1, -1, -1, 1)^T, & 2 &= (-1, -1, 1, -1)^T, \\ 3 &= (-1, -1, 1, 1)^T, & 4 &= (-1, 1, -1, -1)^T, & 5 &= (-1, 1, -1, 1)^T, \\ 6 &= (-1, 1, 1, -1)^T, & 7 &= (-1, 1, 1, 1)^T, & 8 &= (1, -1, -1, -1)^T, \\ 9 &= (1, -1, -1, 1)^T, & 10 &= (1, -1, 1, -1)^T, & 11 &= (1, -1, 1, 1)^T, \\ 12 &= (1, 1, -1, -1)^T, & 13 &= (1, 1, -1, 1)^T, & 14 &= (1, 1, 1, -1)^T, \\ 15 &= (1, 1, 1, 1)^T. \end{aligned}$$

容易证明, N 共有 2 个稳定吸引子, 一个是 $0 = (-1, -1, -1, -1)^T$, 另一个是 $15 = (1, 1, 1, 1)^T$. 并且 $\Gamma(0) = \emptyset$, $\Gamma(15) = \{3, 5, 7, 11, 12, 14\}$; N 有两个长度为 2 的环吸引子: $(2, 13)$, $(4, 8)$. 且 $\Gamma(2, 13) = \{1, 9, 6, 10\}$, $\Gamma(4, 8) = \emptyset$. 我们给出 N 的网络状态图如图 6 所示.

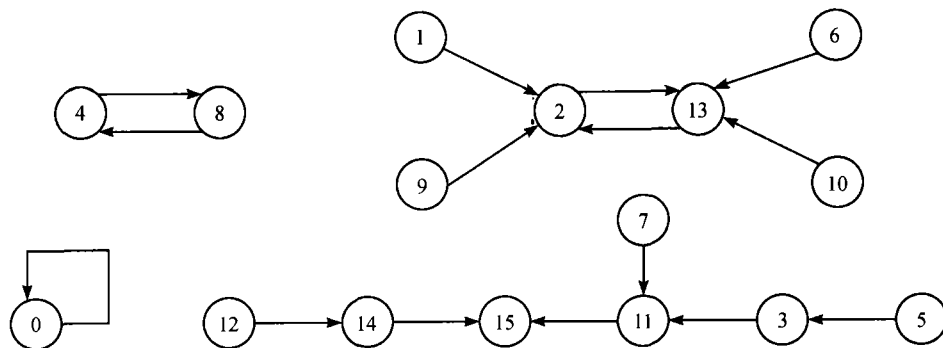


图 6 N 的网络状态图 G_N

定理 5 设 $N = (W, \theta)$ 是一个 p 阶 DHNN, N 在全并行方式下运行. N 共有 m 个稳定吸引子, n 个环吸引子, 则 N 的状态图 G_N 的连通分支数 $\omega(G_N)$ 为

$$\omega(G_N) = m + n.$$

2.2.3 基于图论方法的离散型对称神经网络的稳定性

定理 A^[86] 若 $N = (W, \theta)$ 是一个阶数为 p 的对称 DHNN, 且 N 在全并行方式下工作, 则对 N 的任一初始状态 $X_0 \in B^p$, 网络要么收敛到一个稳定状态, 要么收敛到一个长度为 2 的环吸引子上去.

Bruck 利用图论的方法对 Goles 的工作给出了重新证明^[82]. 其基本思想是将一个给定的全连接网络 N , 即拓扑结构是完全图的网络转化为一个其拓扑结构是偶图的新网络 N' , 但这个新网络的阶数是原网络阶数的 2 倍, 并保持原网络的各种类型的稳定状态不变. 具体方法如下:

设 $N = (W, \theta)$ 是一个 p 阶对称的 DHNN, $N' = (W', \theta')$ 是由 N 按照下列方式构造的新的 $2p$ 阶的对称 DHNN:

$$W' = \begin{pmatrix} 0 & W \\ W & 0 \end{pmatrix}, \quad \theta' = \begin{pmatrix} \theta \\ \theta \end{pmatrix}.$$

可用图 7 说明 Bruck 的思想. 图 7(a) 中,

$$W = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix}, \quad \theta = \begin{pmatrix} 2 \\ 1 \\ 4 \end{pmatrix};$$

图 7(b),

$$W' = \begin{pmatrix} 0 & W \\ W & 0 \end{pmatrix}, \quad \theta' = \begin{pmatrix} \theta \\ \theta \end{pmatrix}.$$

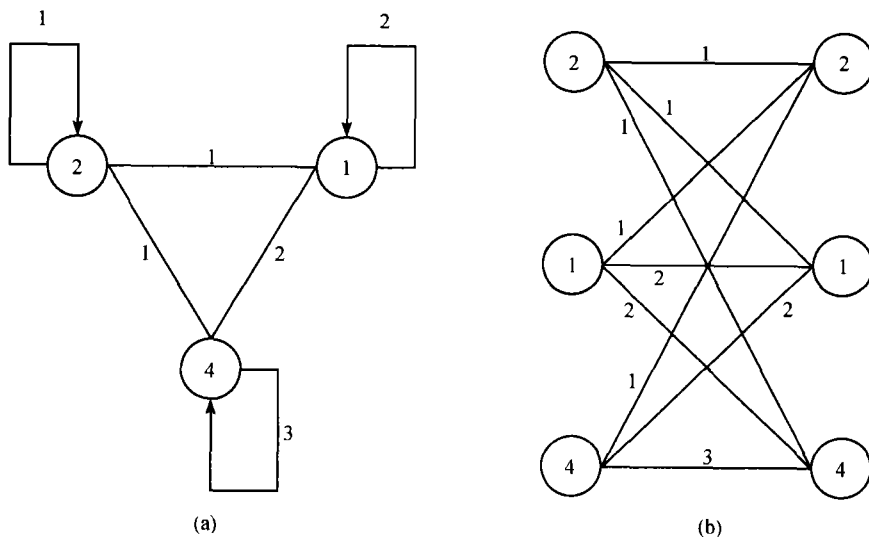


图 7 说明 N 与 N' 拓扑结构的两个图

定理 B^[81] 设 $N = (W, \theta)$ 是一个阶数为 p 的对称 DHNN, $N' = (W', \theta')$ 是由上述方式导出的一个新的网络, 则

(i) 如果 W 对角线上的元素非负, 则当 N 按照串行方式工作时, 等价于网络 N' 按照另一种方式工作;

(ii) 在 N' 中可找到一种串行工作方式, 使得它与 N 的全并行工作方式等价.

基于此结果, Bruck 用图论的方法对 Goles 的结果^[86]给出了重新证明.

2.2.4 基于图论方法的离散型反对称神经网络的稳定性

设 $N = (W, \theta)$ 是一个阶数为 p 的 DHNN. 如果 W 是一个反对称矩阵, 则称 N 为反对称 DHNN; N 称为严格的, 如果 $\forall X \in B^p$, WX 无零分量. Goles 在 1986 年获得了

定理 C 若 $N = (W, \theta)$ 是一个 p 阶的、对角线为零且 $\theta = 0$ 的严格反对称离散 Hopfield 神经网络, 则对任一初始状态 $X_0 \in B^p$, 网络在全并行模式下运行, 总收敛到一个长度为 4 的环吸引子上去.

此后, Bruck 应用图论的方法对这个结果进行了重新“证明”^[82]. 其基本思想是: 将一个给定的全连接网络 N , 即拓扑结构是完全图的网络转化为一个拓扑结构是 2 个完全偶图之并的新的网络 N' , 这个新网络的阶数是原网络阶数的 4 倍, 并保持原网络的各种类型的稳定状态不变. 虽然 Bruck 的思想很好, 但遗憾的是, Bruck 的证明是不正确的. 他所构造的新网络 $N' = (W', \theta')$ 为

$$W' = \begin{pmatrix} 0 & 0 & 0 & -W \\ 0 & 0 & W & 0 \\ 0 & -W & 0 & 0 \\ W & 0 & 0 & 0 \end{pmatrix}, \quad \theta' = 0.$$

对此, 文献[83]给予了纠正. 文献[83]中所构造的新网络 $N' = (W', \theta')$ 为

$$W' = \begin{pmatrix} 0 & -W & 0 & 0 \\ W & 0 & 0 & 0 \\ 0 & 0 & 0 & -W \\ 0 & 0 & W & 0 \end{pmatrix}, \quad \theta' = 0.$$

Goles 的结果仅仅指出了一类收敛于 4-环的反对称 DHNN, 这就是上述所言的严格反对称 DHNN. 文献[83]中列出了一些不是严格的反对称 DHNN, 但它们也是只收敛于 4-环的反对称 DHNN, 这就是所谓的 Δ -型 DHNN. 设 $N = (W, \theta)$ 是一 p 阶 DHNN, 如果 $\theta = 0$, 且权值矩阵 W 是上三角部分全为 1, 对角线全为零, 而下三角部分全为 -1 的反对称矩阵, 显然, 当网络的阶数 p 为奇数时, 它不是严格的, 而当 p 为偶数时是严格的. 自然提出下列几个问题:

问题 3 收敛于 4-环的反对称 DHNN 的基本特征是什么?

问题 4 业已知道, 若 $N = (W, \theta)$ 是一个严格的 p 阶反对称 DHNN, 则 N 收敛于 4-环. 试问, $N = (W, \theta)$ 中存在多少个 4-环?

3 结束语

本文对人工神经网络与图论这两门学科之间的相互关系进行了较为系统的讨论,一方面综述了人工神经网络作为一种工具在图论中许多困难问题研究中的应用,特别是一些困难的 NP-完全问题上的应用;另一方面,文中综述了图论作为一种很重要的工具在人工神经网络几个领域中的应用,这些领域主要有离散型前向人工神经网络的模式分类问题、反馈人工神经网络的稳定性的研究等。

在前向人工神经网络模式分类的研究上,我们已经看到,图论作为离散型模式分类的一种工具,具有直观、清晰等优点。更重要的是,能够把一个模式分类问题转化为一个很简单的图,然后通过该图的特征来分析原模式的分类复杂度等。关于这方面的研究还有许多方面的工作要做;在应用图论研究反馈型神经网络的稳定性问题时,其最大的优点是回避了构造能量函数,实际上,这种方法不仅仅用在离散反馈神经网络稳定性的研究上,对一般反馈型离散系统均可适用。关于这方面的研究我们在文中提出了两个尚待解决的问题。

实际上,网络与图论之间的关系是一种天然的关系,特别是图论与生物神经网络之间的联系,限于篇幅,我们将另文讨论。

参 考 文 献

- 1 Pierre P. An Introduction to the Modeling of Neural Networks. London: Cambridge University Press, 1992
- 2 Gallant S I. Neural Network Learning and Expert Systems. Cambridge: The MIT Press, 1993
- 3 Bondy J A, Murty U S R. Graph Theory With Applications. London: The Macmillan Press LTD, 1976
- 4 Hopfield J J, Tank D W. Neural computation of decisions optimization problems. Biol Cybernetics, 1985, 52: 141 ~ 152
- 5 Gibbons A. Algorithmic Graph Theory. Cambridge: Cambridge University Press, 1985
- 6 Dahl E D. Neural network algorithm for an NP-complete problem: map and graph coloring. In: Proc IEEE First International Conference on Neural Networks, III. New York, 1987. 113 ~ 120
- 7 许进, 张军英, 保铮. 基于 Hopfield 神经网络的图的着色算法. 电子学报, 1996, 24(10): 8 ~ 13
- 8 马颂德. 神经网络与组合优化. 见: 中国神经网络首届大会论文集. 北京, 1990. 22 ~ 28
- 9 陈国良. 神经网络用于求解组合优化问题. 见: 中国神经网络首届大会论文集. 北京, 1990. 122 ~ 129
- 10 梁维发, 等. 用神经网络求解一些困难的图论问题. 中国神经网络首届大会论文集. 北京, 1990. 924 ~ 927
- 11 许进, 张军英, 保铮. 基于 Hopfield 神经网络的图的同构算法. 电子科学学刊, 1996, 18(supp): 116 ~ 121
- 12 许进. 自补图理论及其应用. 西安: 西安电子科技大学出版社, 1999. 第 6 章
- 13 Bron C, Kerbosch J. Finding all cliques of an undirected graph. Commun Ass Mach, 1973, 16: 575 ~ 577
- 14 Gendreau M, Picard J C, Zubieta L. An efficient implicit enumeration algorithm for the maximum clique problem. In: Kurzhaski A, ed. Lecture Notes in Economics and Mathematical Systems, Vol 304. New York: Springer-Verlag, 1988. 79 ~ 91
- 15 Mitten L G. Branch and bound method: general formulation and properties. Ops Res, 1970, 18: 24 ~ 34
- 16 Robson J M. Algorithms for maximum independent sets. J Algorithm, 1986, 7: 425 ~ 440
- 17 Tarjan R E, Trojanowski A E. Finding a maximum independent set. SIAM J Comput, 1977, 6: 537 ~ 546
- 18 Balas E, Yu C S. Finding the maximum clique in an arbitrary graph. SIAM J Comput, 1986, 15: 1054 ~ 1068
- 19 Panos M P, Gregory P R. A branch and bound algorithm for the maximum clique problem. Computers Ops Res, 1992, 19(5): 363 ~ 375
- 20 张军英, 许进, 保铮. 基于 Hopfield 神经网络的图的最大团与最大独立集算法. 电子科学学刊, 1996, 18(supp): 121 ~ 127

- 21 Booth K S, Lucker G S. Testing the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J Comput Syst Sci*, 1976, 13: 335 ~ 379
- 22 Hopcroft J E, Tarjan R E. Efficient planarity testing of graphs. *J Assoc Comput Mach*, 1974, 21: 549 ~ 568
- 23 Lempel A, Even S, Cederbaum I. An algorithm for planarity testing of graphs. In: Rosenstiehl P, ed. *Theory of Graph*. New York: Gordon and Breach, 1967. 215 ~ 232
- 24 Shih W K, Hsu W L. A simple test for planar graphs. 见:第三届中国国际图论会议论文集. 北京, 1993
- 25 Takefuji Y, Lee K C. A near-optimum parallel planarization algorithm. *Science*, 1989, 245: 1221 ~ 1223
- 26 许进, 高琳, 张军英. 可平面图的一种新的神经网络算法. *西安电子科技大学学报*, 2001, (2): 1 ~ 5
- 27 Wilson G V, Pawley G S. On Stability of the traveling salesman problem algorithm of Hopfield and Tank. *Bio Cybern*, 1988, 58: 63 ~ 70
- 28 Aiyer S V B, Niranjani M, Fallside F. A theoretical investigation into the performance of the Hopfield model. *IEEE Trans on Neural Network*, 1990, 1(2): 204 ~ 215
- 29 孙守宇, 郑君里. Hopfield 神经网络求解 TSP 的一种改进算法和理论证明. *电子学报*, 1995, 23(1): 73 ~ 78
- 30 张忆军, 叶中行. 神经网络理论及其应用——94'最新进展. 武汉: 华中理工大学出版社, 1994
- 31 余道衡. 关于用人工神经网络解货郎问题(TSP)的探讨. 见:中国神经网络论文集. 北京, 1990
- 32 靳蕃, 胡飞, 常建武. 用 Hopfield 神经网络求解 C-TSP. 见:中国首届神经网络学术大会. 北京, 1990
- 33 Sanchis L. Multiple-way network partitioning. *IEEE Trans Computers*, 1989, 38: 62 ~ 81
- 34 Fox G C. A review of automatic load balancing a decomposition methods for the hypercube. In: Schultz M, ed. *Numerical Algorithms for Modern Parallel Computer Architectures*. New York: Springer-Verlag, 1988, 63 ~ 76
- 35 Simon H D. Partitioning of unstructured problems for parallel processing. *Computing Systems in Eng*, 1991, 2: 135 ~ 148
- 36 Hendrickson B, Leland R. An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM J Scientific Computing*, 1995, 16(2): 465 ~ 469
- 37 Gilbert J R, Miller G L, Teng S H. Geometric mesh partitioning: Implementation and experiments. In: *Proc Int'l Parallel Processing Symp, (IPPS'95)*. 1995
- 38 Boppana R B. Eigenvalues and graph bisection: an average case Analysis. In: *Proc 28th Symp Foundations of Computer Science*. IEEE Trans on Computer, 1987. 280 ~ 285
- 39 Bui T N, Chaudhuri S, Leighton F T, et al. Graph bisection algorithms with good average case behavior. *Combinatorica*, 1987, 7(2): 171 ~ 191
- 40 Bui T N, Peck A. Partitioning planar graphs. *SIAM J Computing*, 1992, 21(2): 203 ~ 215
- 41 Garey M, Johnson D S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: Freeman, 1979
- 42 Bui T N, Jones C. Finding good approximate vertex and edge partitions is NP-hard. *Information Processing Letters*, 1992, 42: 153 ~ 159
- 43 Leighton F T, Rao S. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In: *Proc 29th Symp Foundations of Computer Science*, 1988. 422 ~ 431
- 44 Farhat C. A simple and efficient automatic FEM domain decomposer. *Computers and Structures*, 1988, 28(5): 579 ~ 602
- 45 Farhat C, Lanteri S, Simon H D. TOP/DOMDEC——A software tool for mesh partitioning and parallel processing. *J Computing Systems in Eng*, 1995, 6(1): 13 ~ 26
- 46 Laguna M, Feo T A, Elrod H C. A greedy randomized adaptive search procedure for the two-partition problem. *Operations Research*, 1994, 42: 677 ~ 687
- 47 Battiti R, Bertossi A A. Differential greedy for the 0-1 equicut problem. In: Du D Z, Pardalos P M, eds. *Network Design: Connectivity and Facilities Location*, 1997. 3 ~ 22
- 48 Johnson D S, Aragon C R, McGeoch L A, et al. Optimization by simulated annealing: an experimental evaluation; part I, graph partitioning. *Operations Research*, 1989, 37: 865 ~ 892
- 49 Kirkpatrick S. Optimization by simulated annealing: quantitative studies. *J Statistical Physics*, 1984, 34: 975 ~ 986
- 50 Kirkpatrick S, Gelatt C D Jr, Vecchi M P. Optimization by simulated annealing. *Science*, 1983, 220(4598): 671 ~ 680

- 51 Rutebar R. Simulated annealing algorithms: an overview. *IEEE Circuit and Devices Magazine*, 1989, 19 ~ 26
- 52 Sechen C, Sangiovanni-Vincetelli A. Timberwolf3. 2: a new standard cell placement and global routing package. In: *Proc 23rd ACM/IEEE Design Automation Conf*, 1986. 432 ~ 339
- 53 Sun W, Sechen C. Efficient and effective placement for very large circuits. *IEEE Trans Computer-Aided Design*, 1995, 14(3): 349 ~ 359
- 54 Battiti R, Bertossi A A. Greedy, prohibition, and reactive heuristics for graph partitioning. *IEEE Trans on Computers*, 1999, 48(4): 361 ~ 385
- 55 Bui T N, Moon B R. Genetic algorithm and graph partitioning. *IEEE Trans Computers*, 1996, 45(7): 841 ~ 855
- 56 Kernighan B, Lin S. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical J*, 1970, 49(2): 291 ~ 307
- 57 Fiduccia C, Mattheyses M. A linear time heuristics for improving network partitions. In: *Proc 19th ACM/IEEE Design Automation Conf. Las Vegas*, 1993. 175 ~ 181
- 58 Hendrickson H, Leland R. A Multilevel Algorithm for Partitioning Graphs. SAND93-1301. SANDIA Nat'l Laboratory, 1993
- 59 Diekmann R, Monien B, Preis R. Using helpful sets to improve graph bisections. In: Hsu D F, Rosenberg A L, Sotteau D, eds. *Interconnection Networks and Mapping and Scheduling Parallel Computations*. *IEEE Tran on Computrer*, 1995. 57 ~ 73
- 60 Glover F. Tabu search——part I. *ORSA J Computing*, 1989, 1(3): 190 ~ 260
- 61 Rolland E, Pirkul H, Glover F. A tabu search for graph partitioning. *Annals of Operations Research, Metaheuristics in Combinatorial Optimization*, 1996, 63: 46 ~ 53
- 62 Dell'Amico M, Maffioli F. A new tabu search approach to the 0-1 equicut problem. In: *Meta-Heuristics 1995: The state of the art*. Amsterdam: Kluwer Academic, 1996. 361 ~ 377
- 63 Pirkul H, Rolland E. New heuristic solution procedures for the uniform graph partitioning problem: extensions and evaluation. *Computers and Operations Research*, 1994, 21(8): 895 ~ 907
- 64 Rolland E, Pirkul H. Heuristic solution procedures for the graph partitioning problem. In: Balci O, ed. *Computer Science and Operations Research: New Developments in Their Interfaces*. Oxford: Pergamon Press, 1992
- 65 Bui T N, Jones C. A heuristic for reducing fill in sparse matrix factorization. In: *Proc Sixth SIAM conf Parallel Processing for Scientific Computing*. *IEEE Tran On Computer*, 1993. 445 ~ 452
- 66 Garey M R, Johnson D S. *Computers and intractability: a Guite to the Theory of NP-Completeness*. San Francisco: New York: Freeman, 1979
- 67 Orlova G I, Dorfman Y G. Finding the maximum cut in a graph. *Eng Cybern*, 1972, 10: 502 ~ 506
- 68 Poljak S, Turzik D. Max-cut in circulant graphs. *Discrete Mathematics*, 1992, 108: 379 ~ 392
- 69 Hsu C P. Minimum——via topological routing. *IEEE Trans CAD/ICAS*, 1983, (CAD)2(4): 235 ~ 246
- 70 Barahona F, Grötschel M, Jünger M, et al. An application of combinatorial optimization to statistical physics and circuit layout design. *Oper Res*, 1988, 36(3): 493 ~ 513
- 71 Delorme C, Poljak S. Laplacian eigenvalues and the maximum cut problem. *Math Program*, 1993, 62: 557 ~ 574
- 72 Delorme C, Poljak S. The performance of an eigenvalue bound on the max-cut problem in some classes of graphs. *Discrete Math*, 1993, 111: 145 ~ 156
- 73 Poljak S, Rendl F. Note and edge relaxations of the max-cut problem. *Computing*, 1994, 52: 123 ~ 137
- 74 Korst J H M, Aarts E H L. Combinatorial optimization on a Boltzmann machine. *J Paral Dist Comput*, 1989, 6: 331 ~ 357
- 75 Funabiki N, Nishiawa S, Tajima S. A binary neural network approach for max cut problems. In: *ICONIP'96-Hong Kong*. HK: The Chinese Univ Press, 1996. 631 ~ 635
- 76 Ramanujam J, Sadayappan P. Optimization by neural networks. In: *Proc IEEE Int' I Conf on Neural Networks, II*. 1988. 325 ~ 332
- 77 Foo Y S, Takefuji Y. Stochastic neural networks for solving job-shop scheduling: Part I : problem presentation. In: *Proc IEEE Int' I Conf on Neural Networks, II*. San Diego, 1988. 283 ~ 290
- 78 陈国良. 神经计算及其在组合优化中的应用. *计算机研究与发展*, 1992, (5): 1 ~ 21
- 79 Hopfield J J, Tank D W. Simple neural optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit. *IEEE Trans on CAS*, 1986, 33(5): 533 ~ 541

- 80 Liou C Y, Chang Q M. Numerical soap film for the Steiner tree problem. In: ICONIP'96-Hong Kong. The Chinese Univ Press, 1996. 642 ~ 648
- 81 Bruck J. On the convergence properties of the Hopfield model. Proceedings of the IEEE, 1990, 78(10): 1579 ~ 1585
- 82 Goles E. Antisymmetrical neural networks. Discrete Applied Mathematics, 1986, 13: 97 ~ 100
- 83 许进, 保铮. 反对称离散 Hopfield 网络的稳定性理论. 电子学报, 1999, 27(1): 103 ~ 107
- 84 许进, 保铮. Boole 函数的线性可分性(I)—— n 维超立方体的基本理论. 电子科学学报, 1996, 图论与系统优化专集: 6 ~ 13
- 85 许进, 保铮. Boole 函数的线性可分性(II)——判别 Boole 函数线性可分性问题的若干准则. 电子科学学报, 1996, 图论与系统优化专集: 14 ~ 21
- 86 Goles E, Fogelman F, Pellegrin D. Decreasing energy functions as a tool for studying threshold networks. Discrete Applied Mathematics, 1985, 12: 261 ~ 277

www.cnki.net