用神经网络方法求解优化调度问题[1]

杨圣祥　　　　　汪定伟

(东北大学　信息科学与工程学院　系统工程系　沈阳　110006)

【摘要】 本文概述了神经网络在求解优化调度问题上的应用，首先介绍了神经网络与优化调度问题的结合背景，接着对优化调度问题进行了描述，然后概述了求解优化调度问题的各种神经网络和主要模型，并进行了简要的比较，最后给出了神经网络求解优化调度问题的结论。

【关键词】 优化问题，调度问题，神经网络，线性规划

1. 引 言

优化调度问题就是解决如何合理分配资源来完成不同的任务，以达到最佳效益的问题。优化调度问题是运筹学的一个分枝，基本上属于优化问题类。传统的优化理论与方法大体上分为三大类：有效的最优化方法、枚举最优化方法和启发式算法[2]。传统的优化方法特点是确定性方法，采用单点串行运算，从而存在一个最主要的问题就是求解时间随问题规模呈指数上涨，运算效率低。为适应求解大规模优化调度问题的快速性要求，出现了许多智能化优化新方法，如遗传算法、禁忌搜索、模拟退火和人工神经网络优化算法等。

Hopfield 首次采用神经网络求解了 TSP 问题[4]，从而开辟了神经网络优化计算新的研究领域。神经网络是“由多个简单元件及其层次组织构成的大规模并行互连网络，能以生物神经系统相同的方式与现实世界中的对象进行相互作用”[13]。神经网络是由大量的单个处理单元（神经元），依一定结构互连而成，用以完成不同智能信息处理任务的一个大规模非线性动力系统，它通过大量的神经元并行处理单元来实现较高的计算速率。

一般优化神经网络系统具有有限个渐进稳定平衡点，与系统能量函数的局部极小点对应。如果把神经网络动力系统的稳定吸引子考虑为适当能量函数（或广义目标函数）的极小点，则系统从一个初始点开始，其运动轨道在相空间中总是朝着能量函数减小的方向移动，最终到达系统的平衡点——即能量函数的极小点，从而找到目标函数的一个极小点。系统趋向稳态的时间就是其计算时间，这就是神经网络优化计算的基本原理[25]。

2. 问题描述

2.1 一般优化问题描述

一般优化问题可简述为在有约束或无约束限制条件下，确定优化变量的值，使某个目标函数（或成本函数）取得极大值或极小值。由于极大化函数 f(X)相当于极小化-f(X),因此优化问题数学模型一般以极小化目标函数形式给出,如下所示:

$$\min\{f(x)=C^T X \mid g_i(X)\ge 0, h_j(X)=0, 1\le i\le m, 1\le j\le p\} \tag{1}$$

其中， $X=(x_1,x_2,\cdots,x_n)^T\in R^n$，当 f,g_i,h_j 全为变量 x 的线性函数时,称为线性优化问题，如果 f,g_i,h_j 中至少有一个不是 x 的线性函数时，称为非线性优化问题。

2.2 生产调度问题描述

调度问题就是按时间分配资源来完成任务的问题[1]。描述生产调度问题时，常用工件

[1] 该项研究由国家自然科学基金(No.69684005)和国家 863 计划 CIMS 主题(No.863-511-9609-003)共同资助

代替任务，用机器代替资源，假定有 N 个工件，要经过 M 台机器加工，一个工件在一台机器上的加工称为一道“工序”，工件的“加工路线”即工件的加工工艺过程事先给定。用“加工顺序”表示各台机器上工件加工的先后顺序。生产调度所要解决的问题就是确定每台机器的加工顺序，以及每个工件的所有工序的起始加工时间，以优化某个性能指标，如最小流水时间、最大拖期时间、最少拖期数等[24]。

当每个工件都有其独特的加工路线时，属于作业车间（Job-Shop）调度问题；当所有工件的加工路线都一致时，要确定工件的加工顺序，属于流水车间（Flow-Shop）调度问题；当所有工件都没有指定的加工路线时，属于开放车间（Open-Shop）调度问题；当只有一台机器时，属于单机车间（Single Machine Shop）调度问题；此外，还有其他种类的生产调度问题[2]。

综上所述，生产调度问题可用 Conway 等人提出的方法简单地表示[3]，该方法用四个参数 n/m/A/B 来表示大多数调度问题：n 表示工件数；m 表示机器数；A 表示车间类型，若 A 为空白，表示单机调度问题；B 表示目标函数，如 $C_{\max}, \bar{F}$ 等。例如，4/2/J/ $C_{\max}$ 表示用 2 台机器加工 4 个工件，使最长完工时间为最小的作业车间调度问题。

3. 基于神经网络的优化调度问题解法

3.1 求解一般优化问题

用来求解一般优化问题的神经网络主要有两种类型[6]:

- Hopfield 类型的神经网络，属于非自组织型（Non Self-Organizing）神经网络;
- Kohonen 类型的神经网络，属于自组织型(Self-Organizing) 神经网络。

3.1.1 Hopfield 类型的神经网络求解优化问题简介：

Hopfield 网络是一个完全互连反馈式网络，有离散型和连续型两种类型[4,5]。Hopfield 离散型网络中神经元是阈值逻辑单元，神经元以随机异步方式改变状态；Hopfield 连续型网络中神经元取值在[0,1]上，以同步、连续和确定的方式改变状态。网络的能量函数如下所示：

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}V_iV_j - \sum_{i=1}^{N} V_iI_i + \sum_{i=1}^{N}\frac{1}{R_i}\int_0^{V_i} g_i^{-1}(V)dV \tag{2}$$

其中，V_i，I_i 和 R_i 分别表示神经元 i 的状态，输入偏差电流和输入电阻，T_{ij} 表示连接权值。对于离散型网络，E 中第三项为零，神经元 i 总的输入为：$\sum_{j\neq i} T_{ij}V_j + I_i$，$V_i$ 取值如(3)式所示；对于连续型网络，动态方程及V_i 取值如(4)式所示，$g_i(U_i)$ 通常是 sigmoid 型转换函数。

$$V_i = \{0, if \quad \sum_{j\neq i} T_{ij}V_j + I_i < U_i; 1, if \quad \sum_{j\neq i} T_{ij}V_j + I_i > U_i\} \tag{3}$$

$$\frac{dU_i}{dt} = -U_i + \sum_{j\neq i} T_{ij}V_j + I_i \quad , \quad V_j = g_j(U_j) = \frac{1}{2}(1+\tanh(U_j)) \tag{4}$$

Hopfield 类型的神经网络求解优化问题的一般步骤为[7]：

1> 选择一个合适的表示问题方法，使神经元的输出与问题的解彼此对应;

2> 构造能量函数，使其最小值对应于问题的最优解;

3> 由能量函数导出连接权值和输入偏差;

4> 设置神经元输入初始值，确定动态方程，求解。

用 Hopfield 网络表达优化问题时，约束冲突以显式方式进入到能量函数中，典型的能量函数形式为:

$$E=\sum_i A_i(\text{"约束冲突项 } i\text{"})+B(\text{"成本"}) \quad (5)$$

其中，参数 A_i, B>0，“成本”是独立于约束冲突的优化成本（或目标）函数，要想成功地使用这样的能量函数需要适当选择 A_i 和 B 值。

3.1.2 Hopfield 网络求解 TSP 问题[4]:

TSP（旅行商）问题就是在 N 个城市中，找出一条最短且经过每个城市各一次并回到起点的路径问题。Hopfield 使用一个 N×N 阶换位矩阵来表达 TSP 问题的解空间。神经元的输出用 V_{xi} 来表示，下标 x 表示城市，下标 i 表示访问次序，V_{xi} 为“1”表示城市 x 被第 i 次访问。对应换位矩阵中每行每列有且只有一个元素为 1 的路径，为一条有效路径。网络能量函数的构造，必须使其极小点趋向于对应有效路径且路径长度最短的网络稳定状态，Hopfield 网络求解 TSP 问题的能量函数如下所示：

$$\begin{aligned}E=&\frac{A}{2}\sum_x\sum_i\sum_{j\neq i}V_{xi}V_{yj}+\frac{B}{2}\sum_i\sum_x\sum_{y\neq x}V_{xi}V_{yi}+\frac{C}{2}\Big(\sum_x\sum_i V_{xi}-N\Big)^2\\&+\frac{D}{2}\sum_x\sum_{y\neq x}\sum_i d_{xy}V_{xi}(V_{y,i+1}+V_{y,i-1})\end{aligned} \quad (6)$$

极小化能量函数 E 意味着：每个城市至多访问一次，即 E 中第一项为零；路径中每个位置至多被分配给一个城市，即 E 中第二项为零；矩阵中恰有 N 个元素为 1，即 E 中第三项为零；路径长度被极小化，即 E 中第四项达到极小。

(6)式前三项特征化一般的 TSP 问题，保证路径有效性，第四项与具体的 TSP 问题相关。

3.1.3 求解线性规划与非线性规划问题

1984 年，Hopfield 和 Tank 使用 Hopfield 网络求解 30 城市的 TSP 问题，开创了神经计算的新途径[4]。许多后来的研究人员在 Hopfield 网络的基础上进行改善和推广应用到众多的优化问题的求解：Kennydy and Chua 模型[14]，Rodriquez-Vazquez et al. 模型[16]，Stanislaw et al. 模型[17], ShengWei Zhang 的拉格朗日规划[18]，YouShen Xia 的对偶规划模型[15]等。

A．Kennydy and Chua 模型[14]: 要求解的线性规划与非线性规划问题模型可简化为如下的标准形式：

$$\min\left\{f(X)=C^TX \mid g(X)=AX-b\geq 0\right\} \quad (7)$$

Kennydy and Chua 提出了动态通用非线性规划电路，其中 $g_i(X)$，$C^T=[c_1,c_2,\cdots,c_n]$，and X 定义如（7）式所示，$C_1,C_2,\cdots,C_n$ 是神经元单元的电容值，每个单元中电流 i_j 定义为：$i_j=\phi_j(g_j(X))$，电路的计算能量函数和动态方程分别为：

$$E=E(X(t))=C^TX+\sum_{j=1}^{P}\int_0^{g_j(X)}\phi_j(s)ds \quad \text{和} \quad \frac{dx_k}{dt}=-\frac{1}{C_k}\Big(c_k+\sum_{j=1}^{P}i_j*\frac{\partial g_j(X)}{\partial x_k}\Big)$$

B．Rodriquez-Vazquez et al. 模型[16]：对于(7)式所示的问题，Rodriquez-Vazquez 等人提出的神经网络模型中含有“开关”函数 $u(X)$，网络的能量函数和动态方程分别为：

$$E=C^TX+\frac{s}{2}\sum_{j=1}^{P}(g_j^-(X))^2 \quad \text{和} \quad \dot{X}=-u(X)C-s(1-u(X))\nabla g(X)v(X)$$

其中，$u(X)=\{1,g_j(X)\geq 0\ \forall j;0,else\}$，$v_j(X)=\{1,g_j(X)\geq 0;0,else\}$

在该模型的动态过程中，在区域 $\{X|g_j(X)\geq 0,j=1,\cdots,P\}$ 中，轨迹沿着- c 方向运动，而在其他区域，轨迹沿着冲突的约束项负梯度和可能的目标函数的负梯度方向的组合方向运动。因此，从任何一个可行域以外的点出发，网络轨迹将被驱动到可行域，且一旦达到可行域，轨迹将沿着最小化目标函数的方向移动。

C．Stanislaw et al. 模型[17]：引入松弛变量把不等式约束变成等式约束，其问题模型为：

$$\min\left\{ f(X)=C^T X \mid g(X)=AX-b=0,\ X\geq 0\right\}$$

在网络模型中，把等式约束以 p 阶范数形式引入能量函数中，如下所示：

$$E=C^T X+\rho\left\|AX-b\right\|_p+\gamma\sum_{i=1}^{n}x_i^- \quad \text{和} \quad \dot{X}=-\nabla E=-C-\rho\nabla\left\|AX-b\right\|_p-\gamma\sum_{i=1}^{n}x_i^-$$

其中， $\rho>0,\gamma>0,1\leq p\leq\infty$， $x_i^-=\{-x_i,x_i>0;0,x_i\leq 0\}$

D． 拉格朗日规划模型[18]：问题数学模型以下式给出： $\min\{f(x)|h(x)=0\}$

引入拉格朗日函数： $L(x,\lambda)=f(x)+\lambda^T h(x)$

构造能量函数： $E(x,\lambda)=\dfrac{1}{2}\left|\nabla_x L(x,\lambda)\right|^2+\dfrac{1}{2}\left|h(x)\right|^2$

定义变量神经元： $\dfrac{dx}{dt}=-\nabla_x L(x,\lambda)=-(\nabla f(x)+\lambda^T\nabla h(x))$

定义拉格朗日神经元： $\dfrac{d\lambda}{dt}=\nabla_\lambda L(x,\lambda)=h(x)$

拉格朗日神经元把动态轨迹拉向可行域(h(x) = 0), 而变量神经元寻求 f(x)的最小点。

3.1.4 Kohonen 类型的神经网络求解优化问题

1984 年, Kohonen 提出自组织特征映射(Self-Organizing Feature Map)网络, 属于自适应神经网络，其权值可进行无导师学习，该网络被先后应用到 TSP 问题的求解上[8,19,20,21,22],基本思想是：用圆周点到城市点的拓扑保序映射, 采用"竞争"学习机制, 使最终闭合路径上点的分布概率与城市点的分布一致。以二维空间上 N 个城市 $C_i(C_{xi},C_{yi})$，$i=1,\cdots,N$ 的 TSP 问题为例, 其网络有两个输入神经元, N 个输出神经元, 输出神经元以与两个输入神经元的连接权值为坐标, 在城市平面上的映射点 $W_i(W_{xi},W_{yi})$，$i=1,\cdots,N$ 构成一个闭合环。

在求解 TSP 问题时，先初始化权值, 随机地选一个城市, 提供给输入神经元, 输出神经元通过"竞争"找到与之最近的“竞胜”神经元 i, 称 i“对应”该城市, 竞争机制为:

$$\left\|W_i-C_k\right\|=\min_j\left\{\left\|W_j-C_k\right\|\right\}$$

然后,对该节点及其邻域 $N(i,d)$ 内节点的权值进行修改, 使其靠近其对应城市, 修权公式为:

$$W_j^{new}=W_j^{old}+\alpha(C_k-W_j^{old}) \quad j\in N(i,d),\ \min\left\{\left|i-j\right|+1,N-\left|i-j\right|-1\right\}\leq d$$

遍选所有城市为一个循环, 经过若干次循环训练后, 所有相邻节点就映射并接近相邻城市, 节点环就遍历或充分靠近所有城市， 从而得到 TSP 的一个解。

3.1.5 两种类型的神经网络求解优化问题的比较

用 Hopfield 类型的神经网络求解优化问题存在的缺陷: 易陷入局部最小；有时产生不可行解；收敛速度慢；存在复杂的参数选择问题。

用 Kohonen 网络求解 TSP 问题时，避免了使用 Hopfield 网络求解 TSP 问题所带来的参数选择问题和不可行解问题，但是也存在竞争时一个节点对应多个城市，即所谓的“Duplicate”问题，因此需要对其进行节点“分离”。针对节点分离问题，最简单的办法是随机排序同一节点所对应的那些城市[19]。Burke and Damany 在其建立的"*guilty net*”网络中，使用了“良心机制”，来惩罚“贪婪”节点，奖励“死”节点——在竞争中没有获胜的节点，以使获胜节点分布更加均匀[20]；Burke 还在其建立的“*vigilant net*”网络中，使用“警戒参数”*p* 来控制节点分离，如果竞争中，某个节点竞胜次数超过 *p* 值，该节点则被禁止竞争[21]; Marco 则提出用"活性平面图"中的平均活性，来选择获胜的神经元[22]。

另外，Kohonen 网络到目前为止主要被用来求解 TSP 问题，对其他优化问题还用的不多。

3.2　生产调度问题的求解

在生产调度问题中，Job-shop 调度问题是最一般的问题，也是最复杂的问题。用神经网络求解 Job-shop 调度问题最早是由 Foo S.Y.于 1988 年提出的[23]，其后又有一些人对此问题进行了研究[9,10,11,12]。下面以清华大学的张长水提出的模型为例说明如何用神经网络求解 Job-shop 调度问题：假设有 N 个工件在 M 台机器上加工，s_{ik} 和 t_{ik} 分别表示工件的第 k 个操作起始时间和加工时间。花费函数是所有工件最后一个操作的起始时间之和 $\sum_{i=1}^{N} s_{ik_i}$，k_i 是工件的最后一个操作，所有变量 s_{ik} 的数值满足下列不等式：

$$s_{ik} - s_{ik+1} + t_{ik} \le 0, i = 1,\cdots,N, k = 1,\cdots,k_i - 1; \tag{8}$$

$$s_{ik} \ge 0, \quad i = 1,\cdots,N; \tag{9}$$

$$s_{ik} - s_{ip} + t_{ik} \le 0 \text{或} s_{ip} - s_{ik} + t_{ip} \le 0 \tag{10}$$

其中(8) 表示任何一个工件只能在加工完前一道工序以后，才能加工后一道工序；

(9) 表示所有工件的第一道工序起始时间要大于 0；

(10) 表示在同一台机器上加工的两个操作 A 和 B，要么先加工完 A 再加工 B，要么先加工完 B 再加工 A。

构造神经网络能量函数为：

$$E = \sum_i s_{ik_i} + \sum_i \sum_k H_1 * F_1(s_{ik} - s_{ik+1} + t_{ik}) + \sum_i H_2 * F_1(-s_{i1}) \\ + \sum_i \sum_k H_3 * \min(F_1(s_{ik} - s_{ip} + t_{ik}), F_1(s_{ip} - s_{ik} + t_{ip}))$$

其中，$H_{1,}$ H_2 和 H_3 是参数，s_{ik} 是神经元输出，$F_1(x) = \{e^x, x > 0; 0, else\}$。

神经元的演变按如下方式进行：　$s_{ik}(t+1) = s_{ik}(t) - \dfrac{\partial E}{\partial s_{ik}}$

4. 结　论

由于神经网络具有大规模分布式并行处理的特点，用来求解优化调度问题时，运行速度快，计算效率高，尤其是对于大规模的问题，比传统的优化方法具有较大的优势。目前研究神经网络求解优化调度问题，一方面是要开发新的网络结构和模型，如建立 Hopfield 类型和 Kohonen 类型网络的混合网络来求解优化调度问题；另一方面要注意与其他智能化方法相结合，如与专家系统、模糊控制等方法结合来求解优化调度问题。

参 考 文 献

[1] Baker. K. R., Introduction to Sequence and Scheduling, John Wiley & Sons, New York,1974

[2] Maccrathy. B.L. and Liu. J.Y., Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling, Int. J. Prod. Res.,31(1993),59-79

[3] Conway. R. W. et al., Theory of Scheduling, Reading Mass: Addison-Wesley ,1967

[4] J. J. Hopfield and D. W. Tank, Neural computation of decisions in optimization problems, Biol. Cybernet., 52(1985),141-152

[5] D. W. Tank and J. J. Hopfield, Simple neural optimization networks: an A/D converter, single decision circuit and a linear programming circuit, IEEE Trans. Circ. Syst., 33(1986),533-541

[6] Chee-Kit Looi, Nueral network methods in combinatorial optimization, Computer Ops. Res.,19(1992),191-208

[7] J. Ramanujam and P. Sadayapppan, Optimization by neural networks, Proc. IEEE Second Int. Neural Network Conf., Vol. II(1988),325-332
[8] J. C. Fort, Solving a combinatorial problem via self-organizing process: an application of the Kohonen algorithm to the TSP, Biol. Cybernet., 59(1988),33-40
[9] D. N. Zhou et al., Scaling neural network for job-shop scheduling, Proc. IJCNN, San Diago, Calif., Vol. III(1990),889-894
[10] Simon Y. Foo et al, Job-shop scheduling based on modified Tank-Hopfield linear programming networks, Engineering Application and Artificial Intelligent, Vol.7, No.3(1994), 321-327
[11] T. M. Willems and L. E. M. W. Brandts, Implementing heuristics as an optimization criterion in neural networks for job-shop scheduling. Journal of Intelligent Manufacturing , 6(1995),377-387
[12] Chang-shui Zhang and Ping-fan Yan, Neural network method of solving job-shop scheduling problem, ACTA Automation Sinica,21(1995),706-712
[13] Kohonen T., An introduction to neural computing, Neural Networks,1(1988),3-16
[14] Kennydy M. P. and Chua L. O., Neural network for nonlinear programming, IEEE Transactions on Circuits and Systems, 35(1988),554-562
[15] Youshen Xia, A new neural network for solving linear programming problems and its application, IEEE Transactions on Neural Networks,Vol.7,No.2(1996),525-529
[16] A. Rodriguez-Vazquez,R. et. al., Nonlinear switched-capacitor “neural” networks for optimization problems, IEEE Trans. Circ. and Syst., Vol.37(1990),384-397
[17] .Stanislaw H. Zak, Viriya Upatising and Stefen Hui, Solving linear programming problems with neural networks: a comparative study, IEEE Transactions on Neural Networks,Vol.6(1995), 94-104
[18] Shengwei Zhang and A. G. Constantinides, Langrange programming neural networks, IEEE Transactions on Circuits and Systems, Vol.39(1992), 441-452
[19] F. Favata and R. Walker, A study of the application of Kohonen-type neural networks to the Traveling Salesman Problem, Biological Cybernetics,64(1991),463-468
[20] Laura I. Burke and P. Damany, The guilty net for the Traveling Salesman Problem, Computers and Operations Research,19(1992),255-265
[21] Laura I. Burke, “Conscientious” neural nets for tour construction in the Traveling Salesman Problem: the vigilant net, Computers and Operations Research,23(1996),121-129
[22] Marco Budinich, A self-organizing neural network for the Traveling Salesman Problem that is competitive with Simulated Annealing, Neural Computation, 8(1996),416-424
[23] Foo S. Y. and Takefuji Y., Integer-linear programming neural networks for job-shop scheduling, Proc. IEEE IJCNN’88(1988), San Diago,341-348
[24] 陈荣秋. 排序的理论与方法. 华中理工大学出版社, 1987
[25] 焦李成. 神经网络计算. 西安电子科技大学出版社, 1995

Solving optimization and scheduling problems with neural network methods

Yang Shengxiang Wang Dingwei

Abstract: This paper briefly reviewed the applications of neural networks in optimization and scheduling problems. The background of combining neural networks with optimization and scheduling problems is first introduced, and the common problem of optimization and scheduling are described briefly. Following that, this paper gives out various neural network models for optimization and scheduling problems and their comparisons, and shows the main models. Finally the conclusion and future research in this field are proposed briefly.

Key words: Optimization, Scheduling, Neural Networks, Linear Programming