①
36-40

多目标决策问题的神经网络方法

O225

司 昕[①]
(中国科学院自动化研究所)

【摘要】利用 Hopfield 神经网络对多目标决策问题进行研究. 给出有限个方案和无限个方案下的多目标决策问题的网络映射关系和优化算法,同时给出这两种情况下的计算步骤及实例.

关键词:多目标决策,Hopfield 神经网络,非线性优化算法

0 引 言

多目标决策问题大量存在于现实生活中,自从50年代以来,人们从不同的侧面对这类问题进行了广泛而深入的研究. 总的来说,可以根据决策情况将多目标决策问题分为两类,一类是有限个方案的多目标决策问题,另一类是无限个方案的多目标决策问题. 对于前者,研究方法主要有 TOPSIS 法、ELECTRE 法和 LINMAP 法等,后者的研究方法主要有目的规划法、调和解和移动理想点法、STEM 法和 SEMOPS 法,不同的方法因其侧重点不同而各有其特点[1,2].

从理论上讲,多目标决策问题属于数学规划的范畴,利用 Hopfield 神经网络求解数学规划问题,国外已有一些研究[3,4]. Hopfield 神经网络对于一些需要进行实时分析决策的场合,如机器人在线优化、卫星导航等领域,因其计算的并行性而具有很大实用价值. 在国内,也有一些学者将 Hopfield 神经网络应用于多目标决策领域,但往往只注重算法的研究而忽略其网络结构或者算法过于复杂而使神经网络无法实现,实际上 Hopfield 神经网络优化算法非常类似于数学规划中的罚函数法,若不从硬件上加以实现,其在计算方面的并行性就得不到充分利用,因而实用价值也就不大.

本文将以 Hopfield 神经网络为基础,针对有限个方案和无限个方案的多目标决策问题进行研究,并给出了这两种情况下的网络映射关系和优化算法,所构造的网络意义明显,易于实现.

1 有限个方案的多目标决策问题的神经网络方法

对于有限个方案的多目标决策问题,上面谈到的几个方法均要求知道决策矩阵(即由各个方案所对应的目标值所构成的矩阵),在信息不全时,决策矩阵并不很容易能得到,这种情况下上述几中方法均无法使用. 假如决策矩阵不知道,但各方案对每个目标的优先次序以及各个目标的相对重要性(以权表示)已知,这样所需的信息要少一些,这时可根据已知信息构造一个权矩阵,并结合 Hopfield 神经网络来解决这类问题.

1.1 问题描述

对于有 n 个方案、m 个目标的决策问题,各目标的相对重要性可量化为各目标的权. 设 m 个目标的权为 $w^T=[w_1,w_2,\cdots,w_m]$, $\sum_{i=1}^{n} w_i=1$, 对每一个目标各个方案同样有一个排队. 据此构造如下的权矩

---

① 司 昕:博士生,研究方向:人工智能在 DSS、CIMS 中的应用、DEDS 以及混杂动态系统等. 通讯地址:北京 2728 信箱三博公司,邮编:100080.

阵 $W = [w_{ij}]_{n\times n}$，其中 w_{ij} 表示第 i 个方案在所有 m 个目标中排在第 j 的目标权之和，例如，若方案 1 对于目标 1 和目标 2 是最好的(即对于第 1、2 个目标排在第 1 位)，则 $w_{11} = w_1 + w_2$，方案 1 对于目标 3 排第 2 位，则 $w_{12} = w_3$，在所有目标中方案 1 没有排在第 3 位的，则 $w_{13} = 0$，余类推.

得到权矩阵后，令 $J = \{1,2,\cdots,n\}$，计算使和 $w_{1i_1} + w_{2i_2} + \cdots + w_{ni_n}$ 为最大的一个序列 $i_1,i_2,\cdots,i_n$，在得到 $i_1,i_2,\cdots,i_n$ 的值之后，就得到所有方案的一个最佳排队.

1.2　优化算法

上述问题类似于 TSP 问题，利用 Hopfield 神经网络来进行求解. 定义矩阵 $P = [p_{ij}]_{n\times n}$，其元素 p_{ij} 为 0 或 1，当 $p_{ij} = 1$ 时表示方案 i 被排在第 j 位，矩阵 P 有如下特性：(1) 每一行只有一个 1，其余为 0；(2) 每一列只有一个 1，其余为 0；(3) 全部元素之和为 n. 定义了矩阵 P 后，上述决策问题转化为在满足特性 (1)、(2)，(3) 的条件下求 $\min\sum_{i=1}^{n}\sum_{j=1}^{n}(-w_{ij}p_{ij})$ 的问题.

由此给出 Hopfield 神经网络的能量函数如下

$$E = \frac{A}{2}\sum_{x}\sum_{i}\sum_{j\neq i}p_{xi}p_{xj} + \frac{B}{2}\sum_{i}\sum_{x}\sum_{x\neq y}p_{xi}p_{yi} + \frac{C}{2}(\sum_{x}\sum_{i}p_{xi} - n)^2 + D\sum_{x}\sum_{i}(-w_{xi}p_{xi})$$

网络运动方程

$$\frac{du_{xi}}{dt} = -\frac{u_{xi}}{\tau} - A\sum_{j\neq i}p_{xj} - B\sum_{y\neq x}p_{yi} - C(\sum_{x}\sum_{j}p_{xj} - n) + Dw_{xi}$$

$$p_{xi} = \frac{1}{2}[1 + \tanh(\frac{u_{xi}}{u_0})]$$

这就是此类决策问题的 Hopfield 神经网络优化表达式. 为求得结果，需要求解 $n\times n$ 个非线性一阶联立微分方程式，以得到矩阵 P 中 $n\times n$ 个元素的全部状态.

1.3　映射关系

此类决策问题映射为一个具有 $n\times n$ 个神经元的 Hopfield 神经网络，其网络参数

$$I_{xi} = \text{Constant\_of}(\frac{\partial E}{\partial p_{xi}}) = C\cdot n$$

$$T_{xi,yj} = \text{Constant\_of}(\frac{\partial^2 E}{\partial p_{xi}\partial p_{yj}}) = -A\delta_{xy}(1-\delta_{ij}) - B\delta_{ij}(1-\delta_{xy}) - C$$

其中 Constant_of(·) 表示多项式的常数项，$\delta_{ij} = (i = j?1:0)$. 网络稳定时，位于 (i,j) 神经元的输出 $p_{ij} = 1$ 表示方案 i 排在第 j 位.

1.4　计算实例

某一有 4 个目标 f_1,f_2,f_3,f_4 的工程项目有 5 种方案 x_1,x_2,x_3,x_4,x_5 可供选择，并且目标 f_1,f_2 重要于目标 f_3,f_4，由此取权值 $w = [0.3,0.3,0.2,0.2]$. 方案按目标的排序如下表所示：

	f_1	f_2	f_3	f_4
1	x_1	x_1	x_5	x_4
2	x_3	x_2	x_4	x_1
3	x_4	x_5	x_4	x_3
4	x_2	x_1	x_3	x_2
5	x_5	x_4	x_1	x_5

根据这些信息，构造出系统的权矩阵　$W = \begin{bmatrix} 0.3 & 0.2 & 0 & 0.3 & 0.2 \\ 0 & 0.5 & 0 & 0.5 & 0 \\ 0.3 & 0.3 & 0.2 & 0.2 & 0 \\ 0.2 & 0 & 0.5 & 0 & 0.3 \\ 0.2 & 0 & 0.3 & 0 & 0.5 \end{bmatrix}$

利用 Turbo C 语言对这个问题进行了仿真，取参数 $A = B = 0.5$，$C = 0.2$，$D = 1$，$u_0 = 0.02$，步长

$h=0.001$，按上述方法计算矩阵 P，得 $p_{31}=p_{22}=p_{43}=p_{14}=p_{55}=1$、其余为 0，由此可得方案的最优排序为：$x_3$，$x_2$，$x_4$，$x_1$，$x_5$.

2 无限个方案的多目标决策问题的神经网络方法

2.1 问题描述

对于无限个方案的多目标决策问题，其方案集由一集约束条件所限制，这类问题的一般的数学表达形式是：N 个决策变量 $x=[x_1,\cdots,x_N]^T$ 对应于 $n(n\geqslant 2)$ 个目标函数 $f_1(x)$、$\cdots$、$f_n(x)$，在某种约束条件的限制下，不失一般性，约束条件可记为：$x\in X$，$X=\{x\mid g_j(x)\geqslant 0, j=1,\cdots,m\}$，要求在可行域 X 中选择一解使每个目标函数达到最优. 对于等式约束 $h_j(x)=0$，可以通过 $h_j(x)\geqslant 0$ 和 $-h_j(x)\geqslant 0$ 将其化为不等式约束.

由于约束条件的限制，在 x 的可行域中往往找不到一个使所有目标均达到最优的解，这种情况下可以寻找一个满意解（非劣解）来代替最优解，满意解应使目标函数距目的 $\hat{f}_1,\cdots,\hat{f}_n$ 的组合偏差为最小.

采用加权的 p 范数作为目标函数与目的 $\hat{f}_1,\cdots,\hat{f}_n$ 的偏差测度，p 范数定义如下

$$d_p(f(x),\hat{f})=\left(\sum_{j=1}^{n}w_j|f_j(x)-\hat{f}_j|^p\right)^{\frac{1}{p}}$$

式中 $1\leqslant p<\infty$，w_j 是第 j 个目标的权，$d_p(\cdot,\cdot)$ 表示 $f(x)$ 和 $\hat{f}$ 之间的距离，$\hat{f}_i$，$i=1,\cdots,n$ 可由具体问题而定，或者通过计算在满足可行域 X 的情况下 $f_i(x)$ 的最优值而得到. 据此无限个方案的多目标决策问题可表示如下

$$\min\ \{d_p(f(x),\hat{f})=\left(\sum_{i=1}^{n}w_i|f_i(x)-\hat{f}_i|^p\right)^{\frac{1}{p}}\}$$
$$\text{s.t.}\ \ x\in X$$

理论上讲，一旦给出 $\hat{f}$ 和 $w^T=[w_1,w_2,\cdots,w_m]$，$w_i>0$，$i=1,2\cdots,n$，而且选定某个 p 值，$1\leqslant p<\infty$，就可以采用适当的优化方法求解上述问题的解.

2.2 映射关系与优化算法

利用 Hopfield 神经网络求解上述问题时，定义其能量函数为

$$E=A\cdot\left[\left(\sum_{i=1}^{n}w_i\cdot|f_i(x)-\hat{f}_i|^p\right)^{\frac{1}{p}}\right]+\sum_{j=1}^{m}b_j\cdot F(g_j(x))$$

E 中第 1 项为优化目标项 $E_{目标}$，第 2 项为约束惩罚项 $E_{约束}$，当不满足约束时则惩罚之，因此可设 $F(y)=\min\{0,y\}$.

常规 Hopfield 神经网络模型对具有 N 个决策变量的上述问题采用具有 N 个神经元的神经网络且各神经元之间完全互相连接，它对这类问题并不是最佳结构，主要原因是由于能量函数右端的约束项 $E_{约束}$ 的存在，使权值不好映射到网络上，用硬件难以实现，即使在某些情况下权值能映射到网络上，其意义也不明显，而且不易于理解. 因此增加神经元数目，并采用非全局连接的 Hopfield 神经网络模型来构造此类问题的结构. 增加神经元后的网络具有 $N+m$ 个神经元，其中的 N 个神经元对应着 N 个决策变量，它们之间完全互连，另外 m 个神经元对应着 m 个约束，一般情况下这些约束条件互相独立，因此除了与 N 个决策变量神经元之间存在着连接外，它们本身之间无连接.

网络的映射关系中按如下方式构造：

对于决策变量神经元，其网络参数为 $I_k^1=-\text{Constant\_of}(\frac{\partial E_{目标}}{\partial x_k})$，神经元 k_1 与 k_2 间的连接权值 $T_{k_1k_2}^1=-\text{Constant\_of}(\frac{\partial E_{目标}}{\partial x_{k_1}\cdot\partial x_{k_2}})$，$k,k_1,k_2=1,\cdots,N$；对于不等式约束神经元 j，其阈值为 $I_j^2=-\text{Constant\_of}(g_j)$，与决策变量神经元 k 的连接权值 $T_{jk}^2=-b_j\cdot\text{Constant\_of}(\frac{\partial g_j}{\partial x_k})$，而决策变量神经元 k 与不等式约

束神经元 j 的连接权值为 $T^2_{kj} = \text{Constant of}(\frac{\partial g_j}{\partial x_k})$，$k = 1,\cdots N$，$j = 1,\cdots,m$. 如此设计的神经网络模型，其约束神经元只是完成对 $F(g_j(x))$ 的计算，使网络权值易于映射，而且便于硬件实现，具有实际意义.

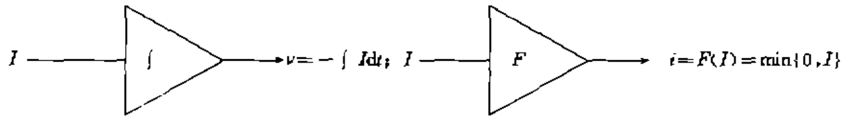
约束神经元的运动方程为

$$\frac{dx_k}{dt} = -A \cdot \frac{1}{p} \cdot \left[\sum_{i=1}^{n} w_i \cdot p \cdot |f_i(x) - \hat{f}_i|^{p-1} \cdot \frac{\partial |f_i(x) - \hat{f}_i|}{\partial x_k} \right]^{\frac{1-p}{p}} - \sum_{j=1}^{m} b_j \cdot \frac{dF}{dg_j} \cdot \frac{\partial g_j}{\partial x_k}$$

不等式约束神经元的运动方程为 $u_{2j} = g_j(x)$

输出　$V_{2j} = F(u_{2j})$，$j = 1,2,\cdots,m$.

选用如下几种单元来构造网络



当出现高次项时，可选用乘法器来实现.

上述神经元网络模型的运动过程为：决策变量神经元根据其状态方程调整它们的状态，并通过权值将它们的输出传给约束神经元；约束神经元根据决策变量神经元的输出，计算 u_{2j}、V_{2j}，$j = 1,2\cdots,m$，并通过权值将其输出反馈给决策变量神经元，直到决策变量神经元状态不再改变或状态的改变量达到一定的范围为止. 可以证明，上述过程是收敛的(证得 $dE/dt \leqslant 0$ 即可，证明略).

在给出适当的初始参数后，按上述方法进行求解，可得到此类问题的满意解，不同的参数可能会对应着不同的满意解，这种情况下，决策人可根据其偏好从这些解中选择一个他认为最佳的解，从这点上看，这也是一个交互的决策过程.

2.3　算法与实例

利用 Turbo C 语言同样对多目标决策问题的 Hopfield 网络算法进行了仿真，其网络设计及算法步骤如下：

(1) 根据决策变量数和约束条件数目分别选取"$\int$"单元(N 个)和"F"单元(m 个)，并按顺序编号.

(2) 根据 I^1_k，I^2_j，$T^1_{k_1k_2}$，T^2_{kj}，T^2_{jk}，确定各神经元的阈值及连接权值，出现高次项时，用乘法器来实现.

(3) 计算 dx_k/dt 及 $x_k(t+h) = x_k(t) + h \cdot dx_k/dt$，$k = 1,\cdots,N$，得到下一时刻的决策变量，$h$ 为计算步长.

(4) 如果决策变量神经元的变化量均小于规定的值，则迭代结束；否则转入(3).

例如，某多目标决策问题是寻求 $x = [x_1, x_2]$，使

$$\min f_1(x) = (x_1 - 1)^2 + (x_2 - 3)^2$$

$$\min f_2(x) = (x_1 - 3)^2 + (x_2 - 1)^2$$

受约束于：

$$x_1^2 + x_2^2 - 4 \leqslant 0, x_1 \geqslant 0, x_2 \geqslant 0$$

取 $p = 2$，$A = 1$，$b_j = 100$，$j = 1,2,3$，步长 $h = 0.001$，在决策人认为这两个目标同样重要的情况下，可取 $w_1 = w_2 = 0.5$，应用以上算法求得的解为[1.413 5，1.413 5]；若决策人认为第 1 个目标的重要程度是第 2 个目标的 3 倍，则取 $w_1 = 0.75$，$w_2 = 0.25$，此时求得的解为[1.714，1.028]. 这两种情况下该问题的理论解分别[1.414，1.414]和[1.715，1.029].

3 结束语

用神经网络求解优化问题时，构造出对应于具体问题的能量函数和网络的拓扑结构是十分重要的，本文在这方面对多目标决策问题进行了探讨. 在信息相对较少的情况下，给出了有限个多目标决策问题的一个排序算法和网络映射关系，在方案数较大时此算法尤其具有实用价值；对于无限个方案的多目标决策问题，本文重构了 Hopfield 神经网络模型，其意义明显，易于硬件实现，并给出了一般情况下的神经网络优化算法. 仿真实验表明，本文提出的方法是切实可行的，据此构造出的神经元网络对一些需要进行实时分析的决策问题尤其具有实际意义.

参考文献

1 Bernardo J J ,Blin J M. A programming model of consumer choice among multi-attributed brands. Journal of Consumer Research,1977,4:111～121

2 陈珽. 决策分析. 北京：科学出版社，1987

3 Kennydy M P, Chua L O. Neural network for nonlinear programming. IEEE Transactions on Circuits and Systems, 1988,35:554～562

4 Chia-Yiu Maa, Michal A Shanblatt, Linear and Quadratic Programming Neural Network Analysis. IEEE Transactions on Neural Network,1992,3(4):580～594

5 王士同，陈剑夫. 问题求解的人工智能神经网络方法. 北京：气象出版社，1995

The Neural Network Method for Multiple Objective Decision Problems

Si Xin

Institute of Automation, Chinese Academy of Sciences

Abstract In this paper, we study multiple objective decision problems though use of Hopfield Neural Network. The network mapping relations and optimal algorithms are presented for finite and infinite plans' multiobjective decision problems, also the computational procedures and examples are given.

Keywords: multiple objective decision, Hopfield neural network, nonlinear optimal algorithm