神经网络和启发式算法混合策略解 Job-shop 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： 本文提出一种有效的自适应神经网络和启发式算法混合策略来求解 Job-shop 调度问题。自适应神经网络具有在网络运行过程中神经元的偏置和连接权值自适应取值的特性，被用来求得调度问题的可行解，而启发式算法被分别用来加速神经网络的运行和保证神经网络收敛到可行解、获得确定排序下最优解和提高可行解的质量。仿真表明本文提出的混合策略是快速有效的。

关键词： Job-shop 调度，神经网络，启发式算法，混合策略

分类号：

1 前　言

生产调度问题就是按时间分配资源来完成任务的问题[1]，Job-shop 调度是一类复杂且极具代表性的生产调度问题：给定 m 台机器加工 n 个工件，每个工件有其特定的加工顺序或路线，调度目标就是确定每台机器上各工件工序的加工顺序及开工时间，使某个性能指标最优（如制造周期最小）[2]。Foo S.Y.最早提出用神经网络求解 Job-shop 调度问题[3]，其后又有一些人用神经网络对此问题进行了研究[4,5,6]，但这些网络基本上都是非自适应的，在网络运行前必须先确定神经元自身的偏置和神经元之间的连接权值。本文提出了一种有效的基于约束满足的自适应神经网络（CSANN），结合启发式算法混合策略来求解 Job-shop 调度问题。CSANN 具有神经元的偏置和连接权值在网络运行过程中自适应取值的特性，用来求得调度问题的可行解，而启发式算法被分别用来增强神经网络的性能、获得确定排序下最优解和提高可行解的质量。仿真实验表明本文提出的求解 Job-shop 调度问题的混合策略是快速有效的。

2 Job-shop 调度问题的数学描述

设 $N=\{1,\cdots,n\}$，$M=\{1,\cdots,m\}$，n_i 是工件 i 的工序数，用 O_{ikq} 表示工件 i 第 k 道在第 q 台机器上加工的工序，其开始加工时间和加工时间（工序加工时间已知）记为 S_{ikq} 和 T_{ikq}，S_{ie_iq} 和 T_{ie_iq} 分别是工件 i 最后一道工序的开始加工时间和加工时间，r_i 和 d_i 分别是工件 i 的投料时间和交货期限，P_i 是工件 i 的有序工序对 $[O_{ikp},O_{ilq}]$ 集合，其中 O_{ikp} 优先于 O_{ilq}，R_q 是使用机器 q 的所有工序 O_{ikq} 的集合，以最小化制造周期为性能指标，则问题的纯整数数学模型如下：

$$\min \quad E=\max_{i\in N}(S_{ie_iq}+T_{ie_iq})$$

$s.t.$

$$S_{ilq}-S_{ikp}\geq T_{ikp} \qquad [O_{ikp},O_{ilq}]\in P_i,\ k,l\in\{1,\cdots,n_i\},i\in N \qquad (1)$$

$$S_{jlq}-S_{ikq}\geq T_{ikq}\ or\ S_{ikq}-S_{jlq}\geq T_{jlq} \qquad O_{ikq},O_{jlq}\in R_q,\ i,j\in N,\ q\in M \qquad (2)$$

$$r_i\leq S_{ijq}\leq d_i-T_{ijq} \qquad i\in N,\ j\in\{1,\cdots,n_i\},q\in M \qquad (3)$$

其中，（1）式表示同一工件的不同工序不能同时加工，即工序顺序约束；（2）式表示每台机器同一时刻只能加工一个工件，即资源约束；（3）式表示投料时间和交货期限约束；

[1] 国家自然科学基金(No.69684005)和国家 863 计划 CIMS 主题(No.863-511-9609-003)共同资助课题

目标函数 E 取所有工件最后一道工序完工时间的最大值，最小化 E 即最小化制造周期。

3　自适应神经网络模型

CSANN 是由在通用神经元（见图 1）基础上，分别定义的表示工序开工时间的 ST 类神经元、工序顺序约束是否满足的 SC 类神经元和资源约束是否满足的 RC 类神经元组成，这三类神经元通过相互连接组成工序顺序约束－SC 模块和资源约束－RC 模块。SC 模块的每个单元由一个 SC 类和两个 ST 类神经元组成（见图 2），用来判断同一工件的工序顺序约束是否得到满足，RC 模块的每个单元由一个 RC 类和两个 ST 类神经元组成（见图 3），用来判断资源约束是否得到满足，当工序顺序约束或资源约束条件不满足时，通过适当的反馈调节来消除约束冲突。CSANN 从功能上看由 SC 模块和 RC 模块组成，从整体系统结构上看分为两层：底层由 ST 类神经元组成，顶层由 SC 类神经元和 RC 类神经元共同组成。

通用神经元由对输入信号的线性加权求和函数与非线性函数 $f(\bullet)$ 级联而成，见图 1。图 1 中，$O_i = f(I_i) = f(\sum_{j=1}^{n}(W_{ij} * O_j) + B_i)$ 。

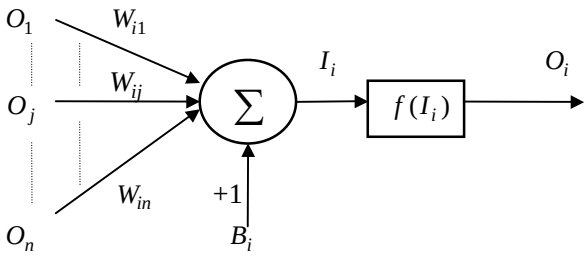

图 1　通用神经元模型

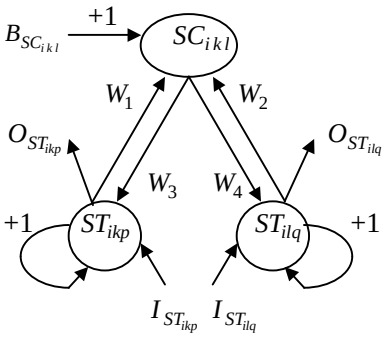

图 2　SC 模块单元

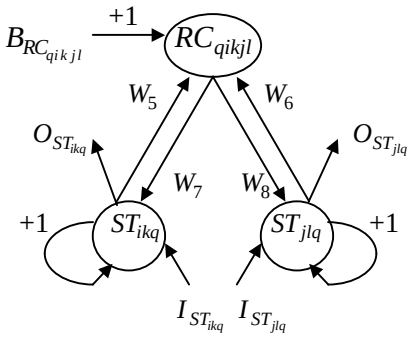

图 3　RC 模块单元

ST 类神经元用于约束各工件的工序加工活动开始时间，ST 类神经元 ST_i 的定义如下：

$$I_{ST_i}(t+1) = \sum_j (W_{ij} * O_{SC\,j}(t)) + \sum_k (W_{ik} * O_{RC\,k}(t)) + O_{ST_i}(t) \qquad (4)$$

$$O_{ST_i}(t+1) = \begin{cases} r_i, & I_{ST_i}(t+1) \le r_i \\ I_{ST_i}(t+1), & r_i \le I_{ST_i}(t+1) \le d_i - T_{ST_i} \\ d_i - T_{ST_i}, & I_{ST_i}(t+1) \ge d_i - T_{ST_i} \end{cases} \qquad (5)$$

其中，（4）式右边的第一和第二项分别是相关的 SC 类和 RC 类神经元的反馈调节作用，第三项是 ST 类神经元 ST_i 上一时刻的输出，（5）式中 T_{ST_i} 是 ST_i 所对应工序的加工时间。

SC 类、RC 类神经元采集 ST 类神经元的输出，分别用来判断（1）式表示的工序顺序约束、（2）式表示的机器资源约束是否满足，定义形式相同，如下所示：

$$I_{C_i}(t)=\sum_j (W_{ij}*O_{ST_j}(t))+B_{C_i} \tag{6}$$

$$O_{C_i}(t)=\begin{cases}0 & I_{C_i}(t)\geq 0\\ -I_{C_i}(t) & I_{C_i}(t)\leq 0\end{cases} \tag{7}$$

其中，$C_i=SC_i$ 或 RC_i，B_{C_i} 是神经元 SC_i 或 RC_i 的偏置设定。

图 2 和图 3 分别给出一个具体的 SC 模块单元和 RC 模块单元，其中，$I_{ST_{ikp}}$ 是随机给定的各工序初始开工时间，作为网络运行时相应的 ST 神经元 ST_{ikp} 的初始输出。图 2 中，$[O_{ikp},O_{ilq}]\in P_i$，SC_{ikl} 是表示第 i 个工件的第 k 道工序和第 l 道工序之间顺序约束是否满足的 SC 类神经元，图 2 中的各权值与偏置取值如（8）式所示。图 3 中，O_{ikq} 和 $O_{jlq}\in R_q$，RC_{qikjl} 是表示在第 q 台机器上加工的第 i 个工件的第 k 道工序和第 j 个工件的第 l 道工序之间的资源约束是否满足的 RC 类神经元，图 3 中的各权值与偏置取值视网络运行 t 时刻具体情况自适应确定，若 $S_{ikq}(t)\leq S_{jlq}(t)$，由（9）式确定，若 $S_{ikq}(t)\geq S_{jlq}(t)$，由（10）式确定。

$$W_1=-1,\quad W_2=1,\quad W_3=-W,\quad W_4=W,\quad B_{SC_{ikl}}=-T_{ikp} \tag{8}$$

$$W_5=-1,\quad W_6=1,\quad W_7=-W,\quad W_8=W,\quad B_{RC_{qikjl}}=-T_{ikq} \tag{9}$$

$$W_5=1,\quad W_6=-1,\quad W_7=W,\quad W_8=-W,\quad B_{RC_{qikjl}}=-T_{jlq} \tag{10}$$

上述各式中，W 是正的可调参数。

4 启发式算法和混合策略描述

整个 Job-shop 混合调度策略是由 CSANN 和启发式算法组成，前者用来得到问题的可行解，后者用于增强 CSANN 的性能、获得确定排序下最优解和提高可行解的质量。

4.1 启发式算法

算法 1（相邻工序互换排序算法）：为了加速神经网络的求解过程，在神经网络运行过程中 t 时刻，对同一工件的相邻工序 $[O_{ikp},O_{ilq}]\in P_i\ (i\in N)$，当 $S_{ikp}(t)\geq S_{ilq}(t)$ 时，互换 O_{ikp} 和 O_{ilq} 的开工时间以互换其加工顺序，算法如下所示：

$$S_{ikp}(t+1)=S_{ilq}(t),\quad S_{ilq}(t+1)=S_{ikp}(t) \tag{11}$$

另一方面，在神经网络运行过程中，由于工序顺序约束冲突的反馈调节和资源约束冲突的反馈调节之间可能存在矛盾，从而造成“死锁”现象，即神经网络陷入死循环而不能收敛到对应问题可行解的稳定状态，此时可通过对同一机器上的相邻工序，满足一定条件时，互换开工时间以互换其加工顺序，从而消除网络求解过程中可能出现的“死锁”现象，以确保得到可行解，算法如下：假设 O_{ikq} 和 $O_{jlq}\in R_q\ (q\in M)$，当 $T_{qikjl}(t)\geq T$ 时，

$$S_{ikq}(t+1)=S_{jlq}(t),\quad S_{jlq}(t+1)=S_{ikq}(t) \tag{12}$$

其中，$T_{qikjl}(t)$ 是网络运行 t 时刻，机器 q 上加工的工序 O_{ikq} 和 O_{jlq} 对已经连续需要调整的次数，T 是设定的常数。

算法 2（非延迟调度算法）：对 CSANN 得到的可行解，在不改变各机器上工序排序的情况下，对所有机器压缩空闲等待同时有工件可加工的时间，得到确定排序下的最优解，以提高可行解的性能或缩短制造周期。按时间从小到大顺序，将遇到的所有工序的开工时间作如下修改：

$$S_{ikp}(t+1)=\begin{cases}S_{ljp}(t)+T_{ljp}, & S_{ljp}(t)+T_{ljp}\geq S_{ik-1q}(t)+T_{ik-1q}\\ S_{ik-1q}(t)+T_{ik-1q}, & S_{ljp}(t)+T_{ljp}<S_{ik-1q}(t)+T_{ik-1q}\end{cases} \tag{13}$$

其中，t 时刻是算法运行前时间，$S_{ikp}(t+1)$ 是算法运行后得到的确定排序下最优解中 O_{ikp} 的

开工时间，O_{ik-1q} 是 O_{ikp} 同一工件 i 的紧前工序，O_{ljp} 是机器 p 上 O_{ikp} 的紧前工序。

在 Job-shop 调度问题的可行调度中，当没有机器在至少有一个工件等待其加工时处于空隙，这样的调度称为非延迟的（non-delay）；而当没有任何一个工序的开工时间在不推迟其他工序的条件下能够有所提前时，这样的调度称为有效的（active）。若调度是有效的，则它是最优的，而有效调度集是非延迟调度集的子集。算法 2 所得到的确定排序下最优调度实际上就是非延迟调度，当其恰好落入有效调度集时，则得到了问题的最优解，这就是算法 2 隐含的理论依据。

4.2 混合策略

求解 Job-shop 调度问题的混合策略的基本步骤为：

1） 建立 CSANN 模型，设定参数 T 和 W 的值，制造周期初始期望值和最大运行时间；

2） 随机给定各工序开始加工时间 $I_{ST_{ikp}}$，作为相应的 ST 神经元 ST_{ikp} 的初始输出；

3） 运行 SC 模块各单元 SC_{ikl}，按（6、7 和 8）式计算 $O_{SC_{ikl}}(t)$，若 $O_{SC_{ikl}}(t)\neq 0$，表示（1）式对应的工序顺序约束不满足，则按（14）式或（11）式（若算法 1 条件满足）和（5）式修改 $S_{ikp}(t+1)$ 和 $S_{ilq}(t+1)$；

$$S_{ikp}(t+1)=S_{ikp}(t)+W_3*O_{SC_{ikl}}(t),\quad S_{ilq}(t+1)=S_{ilq}(t)+W_4*O_{SC_{ikl}}(t) \qquad (14)$$

4） 运行 RC 模块各单元 RC_{qikjl}，按（6、7、9 或 10）式计算 $O_{RC_{qikjl}}(t)$，若 $O_{RC_{qikjl}}(t)\neq 0$，表示（2）式对应的资源约束不满足，则按（15）式或（12）式（若算法 1 条件满足）和（5）式修改 $S_{ikq}(t+1)$ 和 $S_{jlq}(t+1)$；

$$S_{ikq}(t+1)=S_{ikq}(t)+W_7*O_{RC_{qikjl}}(t),\quad S_{jlq}(t+1)=S_{jlq}(t)+W_8*O_{RC_{qikjl}}(t) \qquad (15)$$

5） 重复步骤 3 和步骤 4，直至所有的 SC 类神经元和 RC 类神经元的输出都为 0，从而得到可行解；

6） 运用算法 2，得到确定排序下最优解；

7） 若解的制造周期有所改善或连续保持不变的次数小于 X 次，且运行时间小于设定的最大运行时间，则把得到的解的制造周期作为新的制造周期期望值，返回步骤 2；否则，停止运行，并输出最好解。

以上 7 步称为一次运行，实际运行时可以采用多次运行的办法，把获得的最优的最好解作为最终解。

5 仿真结果

表 1 给出一 6 x 6 Job-shop 调度问题的原始数据，（s, m, t）表示工件的第 s 道工序在第 m 台机器上加工，加工时间为 t；“→”表示工序间先后顺序，问题最优值（最小制造周期）是 55。

工件 1	（1,3,1） → （2,1,3） → (3,2,6) → (4,4,7) → (5,6,3) → (6,5,6）
工件 2	（1,2,8） → （2,3,5） → (3,5,10) → (4,6,10) → (5,1,10) → (6,4,4）
工件 3	（1,3,5） → （2,4,4） → (3,6,8) → (4,1,9) → (5,2,1) → (6,5,7）
工件 4	（1,2,5） → （2,1,5） → (3,3,5) → (4,4,3) → (5,5,8) → (6,6,9）
工件 5	（1,3,9） → （2,2,3） → (3,5,5) → (4,6,4) →（5,1,3） → (6,4,1）
工件 6	（1,2,3） → （2,4,3） → (3,6,9) → (4,1,10) →（5,5,4） → (6,3,1）

表 1　6 x 6 的调度问题原始数据

仿真实验在 Pentium 586/133 PC 上和 Visual C++ 5.0 开发环境下进行的，仿真实验中各

参数设置为：$T=5$ ， $X=5$ 和 $W=0.5$。每次用神经网络求解时，各工序初始开工时间取[0，100]之间的随机均匀分布，各工件投料时间设为 0，即在 0 时刻所有工件都已释放或可加工。制造周期的期望值看作是所有工件的交货期约束，制造周期的初始期望值设定为等效于无穷大的数 500。在不同的最大运行时间设定下，分别进行 100 个实验，得到的统计结果见表 2，其中，迭代次数为每个实验中神经网络求得可行解的次数。由表 2 可知，最大运行时间设为 15、30 和 60 秒时，混合策略的达优率分别为 66%、75%和 99%。

最大运行时间设定	制造周期 E 平均/最小/最大	求得最好解的时间（秒） 平均/最小/最大	迭代次数 平均/最小/最大	达优率（%）
15 秒	55.40/55/57	8/3/14	6/3/12	66
30 秒	55.25/55/56	10/3/27	8/3/13	75
60 秒	55.01/55/56	20/3/55	9/3/15	99

表 2　6 x 6 调度问题的实验结果

图 4 是最大运行时间设定为 60 秒时，混合策略一次运行的迭代过程，神经网络经过 8 次迭代(求得 8 个可行解)，制造周期由首次的 76，改进到 68、66、58、57、57、57，最后达到最好值 55（也是最优值），在第 9 次迭代时，由于制造周期期望值定为最优值，混合策略因其运行时间超过 60 秒而中止运行，其得到的最优解甘特图见图 5（图中[*i*, *j*]表示第 *i* 个工件的第 *j* 道工序）。

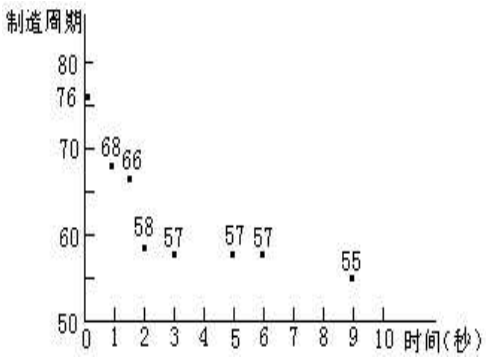

图 4　混合策略一次运行迭代过程

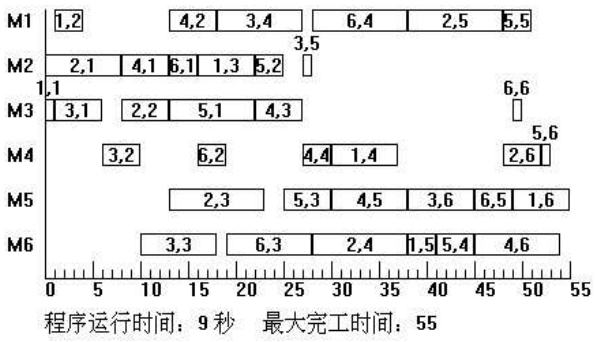

图 5　混合策略求得的最优解甘特图

6　结　束　语

通过文中例子和其它实例的仿真实验表明：针对 Job-shop 调度问题，在不同的最大运行时间设定，经过较少的几次迭代（对于文中例子在最大运行时间设为 15、30 和 60 秒时，混合策略的平均迭代次数为 6、8 和 9 次），混合策略就能以较高的达优率快速得到较好的近优解或最优解。仿真实验表明本文提出的求解 Job-shop 调度问题的混合策略是快速有效的。应用混合策略求解实际的 Job-shop 调度问题时，可以从适当的最大运行时间设定值开始求解，逐渐增大最大运行时间设定值，如果得到的最好解连续几次保持不变，则得到的最好解就可以看作问题的近优解或最优解，从而可以用作实际的调度解。

参　考　文　献

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

[2] Conway, R. W., *Theory of Scheduling*, Reading Mass: Addison-Wesley ,1967

[3] Foo, S. Y. and Takefuji, Y., Integer-linear programming neural networks for job-shop scheduling, *Proc.*

IEEE IJCNN'88, 1988, San Diago, 341-348

[4] Zhou D. N., Charkassky, V., Baldwin, T. R. and Hong, D. W., Scaling neural network for job-shop scheduling, *Proceedings IEEE Int. Joint Conference on Neural Networks*, 1989, New York, 3, 889-894

[5] Willems, T. M. and Brandts, L. E. M. W., Implementing heuristics as an optimization criterion in neural networks for job-shop scheduling, *Journal of Intelligent Manufacturing*, 1995, 6, 377-387

[6] 张长水，阎平凡，解 Job-shop 调度问题的神经网络方法，*自动化学报*, 1995, 21, 706-712

A neural network and heuristics hybrid strategy for job-shop scheduling

Yang Shengxiang　　Wang Dingwei

Department of Systems Engineering, Northeastern University, Shenyang 110006

Abstract: A new efficient neural network and heuristics hybrid strategy for job-shop scheduling is presented. The neural network has the property of adapting its connection weights and biases of neural units while solving feasible solution. Heuristics are used to accelerate the solving process of neural network and guarantee its convergence, and to obtain non-schedule schedule from solved feasible solution by neural network with orders of operations determined and unchanged. Computer simulations have shown that the proposed hybrid strategy is of high speed and excellent efficiency.

Keywords: job-shop scheduling, neural network, heuristics, hybrid strategy

作　者　简　介

杨圣祥　男，1972 年 5 月生。1993 年毕业于东北大学自动控制系工业自动化专业，1996 年于东北大学获工业自动化专业工学硕士学位，现在东北大学信息科学与工程学院系统工程系攻读博士学位。主要研究神经网络、智能优化和智能调度等。

汪定伟　男，1948 年 11 月生。东北大学博士，曾在美国北卡罗来拉州立大学作博士后。现为东北大学教授、博士导师。中国自动化学会管理与系统专业委员会委员、《控制与决策》杂志编委、国家 863 计划 CIMS 主题 09 专题专家。主要研究生产计划与调度理论、建模与决策、智能优化方法。