'93 한국자동제어학술회의논문집 (1993. 10. 20~22)

FMS 스케쥴링 신경회로

°장석호, 남부희
강원대학교 공과대학 제어계측공학과

Linear Programming Neural Networks for Job-Shop Scheduling

°Seok Ho Chang and Boo Hee Nam
Department of Control and Instrumentation Engineering
Kangwon National University

ABSTRACT

This paper presents linear programming neural networks for job-shop scheduling. The starting times of tasks and constraints are formulated as the linear programming problem. A modified Hopfield neural network is proposed for solving job-shop scheduling.

I. 서론

공정계획 신경회로(job-shop scheduling neural network : JSNN)는 FMS에서 주어진 공정순서에 맞는 공정계획을 찾는 것을 목적으로 한다. JSNN이 해결해야 하는 문제는 선행관계를 만족시키는 것이다. 선행관계는 두 가지가 있는데 첫째는 동일 작업안에서의 선행관계(inter-task condition)이다. 이 조건은 라우팅(routing)할 때 정해지는 것으로 고정된 것이다. 두번째는 서로 다른 작업의 공정간의 선행관계(inter-job condition)이다. 유한한 자원을 사용함으로 야기되는 조건이다. 이것은 정해지지 않은 것으로서, 공정계획 신경회로가 해를 구하여 만족시켜야 한다.

Tank와 Hopfield[1]는 선형계획 신경회로(linear programming neural network:LPNN)를 제안했다. 그 뒤, 이 연구결과는 Kennedy와 Chua[2], Maa Shanblatt[3]에 의하여 더욱 발전되었다. Foo와 Takefuji[5]는 공정계획을 구하는데에 선형 신경회로를 적용하였다.

우리는 표준형 신경회로를 이용하여 공정계획을 구하고 그 결과를 보였다.

II. 공정계획문제에 대한 선형계획 공식화

공정계획 문제는 $\sum_{i=1}^{N}\sum_{j=1}^{L} v_{i,j,k}$를 최소화하며 다음의 제약조건을 갖는다.

$$v_{i,1,k} \geq 0 \tag{1}$$

$$\begin{gathered} v_{i,j,k} - v_{i,j-1,h} \geq t_{i,j-1,h} \\ 1 \leq i \leq N,\ 1 \leq j \leq L,\ 1 \leq k,\ h \leq M,\ k \neq h \end{gathered} \tag{2}$$

$$\begin{gathered} v_{u,v,k} - v_{i,j,k} + H(1 - y_{i,u,k}) \geq t_{i,j,k} \\ v_{i,j,k} - v_{u,v,k} + Hy_{i,u,k} \geq t_{u,v,k} \\ 1 \leq u,\ i \leq N,\ 1 \leq v,\ j \leq L,\ 1 \leq k \leq M \end{gathered} \tag{3}$$

$$y_{i,u,k} = \begin{cases} 1 & \text{if } v_{i,j,k} < v_{u,v,k} \\ 0 & \text{otherwise} \end{cases}$$

여기서 N 은 job의 개수, M 은 기계의 갯수, L 은 task의 개수이다. $v_{i,j,k}$ 를 기계 k에서 job i 의 task j 의 시작시간이라고 하자. $t_{i,j,k}$ 를 기계 k, job i 의 task j의 공정시간이라고 하자. zero-one 변수 $y_{i,u,k}$ 는 job i 와 u가 동일 기계 k를 사용하는 충돌을 피하기 위하여 specifies 사용된다. 식 (1)은 모든 공정의 첫 시작시간은 비음수이어야 하는 조건식이며, (2)는 동일 job내에서 task의 우선순위 공정에 관한 제한조건이고, (3)은 동일기계에 대한 여러 job의 충돌회피를 위한 우선순위 조건식이다.

여기서 우리는 slack variable를 사용하여 부등식 조건을 등식조건으로 변환한다:

$$v_{i,1,k} - s_{i,1} = 0,\ 1 \leq i \leq N,\ 1 \leq k \leq M \tag{4}$$

$$\begin{gathered} v_{i,j,k} - v_{i,j-1,h} - s_{i,j} = t_{i,j-1,h} \\ 1 \leq i \leq N,\ 2 \leq j \leq L,\ 1 \leq h,\ k \leq M \end{gathered} \tag{5}$$

$$\begin{gathered} v_{u,v,k} - v_{i,j,k} + x_{u,i,k} = t_{i,j,k} \\ v_{i,j,k} - v_{u,v,k} + x_{i,u,k} = t_{u,v,k} \\ 1 \leq u,\ i \leq N,\ 1 \leq v,\ j \leq L,\ 1 \leq k \leq M,\ u \neq i \end{gathered} \tag{6}$$

여기서 $s_{i,j} \geq 0$이고 자유변수인 $x_{u,i,k}$, $x_{i,u,k}$는 다음의 조건을 만족한다.

$$x_{u,i,k}\, x_{i,u,k} \leq 0$$
$$(v_{u,v,k} - v_{i,j,k})\, x_{u,i,k} \leq 0 \qquad (7)$$
$$(v_{i,j,k} - v_{u,v,k})\, x_{i,u,k} \leq 0$$

III. 공정계획 신경회로(JSNN)

우리는 편의상 다음과 같이 변수를 정의한다. 이는 컴퓨터 모의실험을 위하여 필요하다.

$$v_{i,j,k} = v_{L(i-1)+j}, \; 1 \leq i \leq N, \; 1 \leq j \leq L, \; 1 \leq k \leq M$$

$$s_{i,j,k} = s_{L(i-1)+j}, \; 1 \leq i \leq N, \; 1 \leq j \leq L$$

$$x_{i,u,k} = \begin{cases} x_{I(i,k)+u}, & i > u \\ x_{I(i,k)+u-1}, & i < u \end{cases} \qquad (8)$$
$$1 \leq i, u \leq N, \; 1 \leq k \leq M, \; i \neq u$$

여기서 $I(i, k) = (N-1)[(i-1) + N(k-1)]$.

각 신경의 입출력 특성은 다음과 같다.

$$v_i = h(u_i), \; i = 1, \cdots, p$$
$$s_j = h(u_i), \; i = p+1, \cdots, p+q, \; j = 1, \cdots, q$$
$$x_j = h(u_i), \; i = p+q+1, \cdots p+q+r, \; j = 1, \cdots, r \qquad (9)$$

여기서 $h(\tau) = k\tau$, $k > 0$이고, $u^T = [u_1, \cdots, u_n]$, $n = p+q+r$이라 하자.
다음과 같이 제약조건식의 오차를 정의하면

$$e_j(u) = d_j^T u - b_j$$
$$= \sum_{i=1}^{n} d_{ji} u_i - b_j, \; j = 1, \cdots, m \qquad (10)$$

여기서 m 은 제한조건식의 갯수이고 n 은 변수의 갯수이다. 식(6)의 좌변은 $Yv + x$ 와 같이 행렬 형태로 표현되고 y_j 는 Y 의 요소이다.

$$z_j = y_j^T v$$
$$= \sum_{i=1}^{p} y_{ji} v_i, \; j = 1, \cdots, r \qquad (11)$$

제안된 신경회로망의 동특성 방정식은 다음과 같다:

$$C\frac{du_i}{dt} = -\frac{u_i}{R} - L(a_i) - \sum_{j=1}^{m} d_{ji} g(e_j(u)), \; i = 1, \cdots, p$$

$$C\frac{du_i}{dt} = -\frac{u_i}{R} - \sum_{j=1}^{m} d_{ji} g(e_j(u)) - f(u_i), \; i = p+1, \cdots, p+q \qquad (12)$$

$$C\frac{du_i}{dt} = -\frac{u_i}{R} - \sum_{j=1}^{m} d_{ji} g(e_j(u)) - w(u_i; z_{i-p-q}),$$
$$i = p+q+1, \cdots, p+q+r$$

평형점 근방에서의 진동을 줄이기 위하여 다음 선형함수가 이용되었다.

$$L(\tau) = \frac{\tau}{l}, \quad l > 1$$

이 함수는 Aourid et al.[5]이 제안한 함수와 유사하다. 함수 f, g는 제약조건식을 만족시키기위한 페널티 함수로서 제약조건식이 위반될 때 페널티를 가한다.

$$g(\tau) = K\tau \quad \text{for all } \tau \qquad (13)$$

$$f(u_i) = \begin{cases} Ku_i & \text{if } u_i < 0 \\ 0 & \text{otherwise} \end{cases} \qquad (14)$$

동일기계상에서 여러 job의 공정이 충돌하는 것을 방지하기 위하여 다음함수를 도입한다:

if $z_j \geq 0$,
$$w(u_i; z_j) = \begin{cases} Ku_i, & \text{if } u_i > 0 \\ 0, & \text{otherwise} \end{cases} \qquad (15\text{-a})$$

if $z_j < 0$,
$$w(u_i; z_j) = \begin{cases} Ku_i, & \text{if } u_i < 0 \\ 0, & \text{otherwise} \end{cases} \qquad (15\text{-b})$$

페널티 함수 g, f, w 는 수동성(passivity)을 갖으므로, 페널티 평면의 1사분면과 3사분면에서 동작한다.

For all τ and u_i,

$$\tau\, g(\tau) \geq 0$$
$$\tau\, f(\tau) \geq 0$$
$$u_i\, w(u_i; z_j) \geq 0.$$

IV. 시뮬레이션 결과

모의실험은 386MATLAB으로 프로그래밍되어, IBM PC486 상에서 수행되었다. JSNN의 회로 미분방정식을 적분하기 위해 룬게-쿠타 4차 방법이 사용되었다. 적분 구간은 10^{-6}초이고 모든 증폭기는 매 단계마다 동기화된다고 가정한다. 입력

저항과 커패시턴스는 각각 $R = 10^5$ ohms, $C = 10^{-4}$ farads이다.

2 jobs / 3 tasks / 3 machines 할당문제를 고려하여 본다. 표준형 선형계획 문제 공식화를 하면 다음의 제한조건식을 갖는다.

$$
\begin{aligned}
u_1 - s_1 &= 0 & \qquad u_1 - u_5 + x_1 &= t_5 \\
u_2 - u_1 - s_2 &= t_1 & \qquad u_5 - u_1 + x_2 &= t_1 \\
u_3 - u_2 - s_3 &= t_2 & \qquad u_2 - u_6 + x_3 &= t_6 \\
u_4 - s_4 &= 0 & \qquad u_6 - u_2 + x_4 &= t_2 \\
u_5 - u_4 - s_5 &= t_4 & \qquad u_3 - u_4 + x_5 &= t_4 \\
u_6 - u_5 - s_6 &= t_5 & \qquad u_4 - u_3 + x_6 &= t_3
\end{aligned}
$$

JSNN은 한 평형점인 다음의 해로 수렴한다.

$$
\begin{aligned}
u^T = [&-0.0136\ 4.9760\ 12.9725\ -0.0066\ 6.9899\ 12.9725 \\
&-0.0068\ -0.0052\ -0.0018\ -0.0033\ -0.0017\ 2.9825 \\
&10.0035\ -2.0035\ 16.9964\ 0.0018\ -5.9791\ 14.9790]
\end{aligned}
$$

표 1. 2 jobs / 3 tasks / 3 machines 공정계획 라우팅 (괄호안의 수치는 공정시간을 나타냄)

	TASK1	TASK2	TASK3
JOB1	M1(5)	M2(8)	M3(2)
JOB2	M3(7)	M1(3)	M2(2)

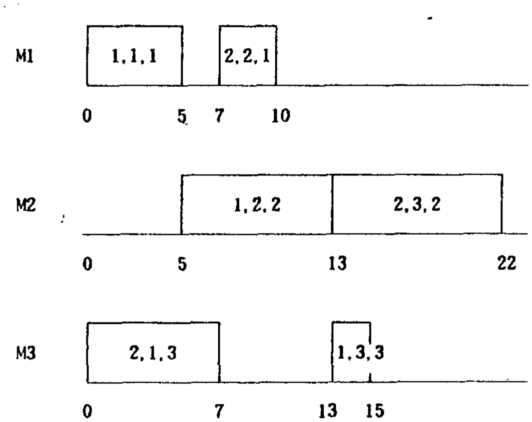

Fig. 1. The obtained solution represented by Gantt chart

V. 결론

본 논문에서, 우리는 공정계획 문제를 해결할 수 있는 신경회로, JSNN(공정계획 신경회로)를 제안했다. 공정계획 신경회로의 제약조건은 한 조의 연립방정식으로 공식화될 수 있다. 우리는 표준형의 제약조건으로 변형하여 제안된 JSNN에 입력하여 해를 구하였다. 수동성 페널티 함수를 갖춘 JSNN은 주어진 문제를 직접 입력하여 안정하게 동작하였다. 이 모델의 스케일링 효과는 많은 분야에 적용될 수 있다.

REFERENCES

[1] D. W. Tank and J. J. Hopfield, "Simple Neural Optimization Networks: An A/D Converter, Signal Decision Circuits, and a Linear Programming Circuit", IEEE Trans. on Circuits and Systems, vol CAS-33, May 1986.

[2] M. P. Kennedy and L. O. Chua, "Neural Networks for Nonlinear Programming", IEEE Trans. on Circuits and Systems, No.5, pp. 554-562, May 1988.

[3] M. Aourid, D. Mukhedkar, B. Kaminska, " Convergence and Stability Study of Hopfield's Neural Network for Linear Programming", IJCNN 92, pp. IV-525-531, 1992.

[4] C. Y. Maa and M. A. Shanblatt, "Linear and Quadratic Programming Neural Network Analysis", IEEE Trans. on Neural Networks, Vol. 3, No. 4, pp. 580-594, July 1992.

[5]Y. P. S. Foo and Y. Takefuji, "Integer Linear Programming Neural Networks for Jop-Shop Scheduling", IJCNN 88, San Diego, CA, July 1988.

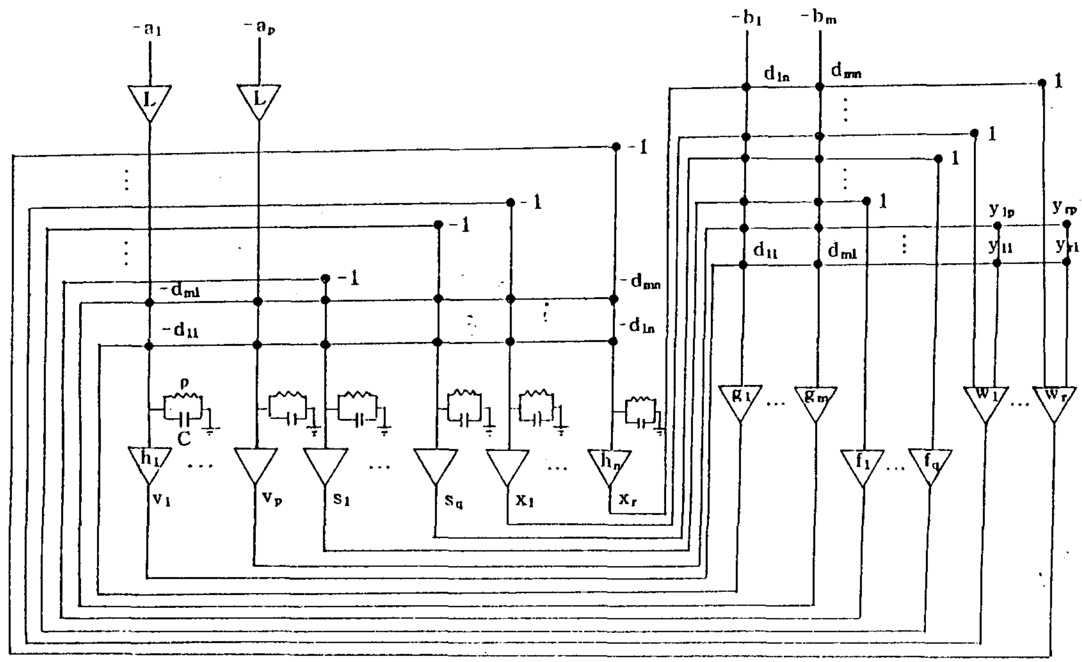

Fig.2.The linear programming neural network for job-shop scheduling