'93 KACC (1993.10.20~22)

Line Balancing Using a Hopfield Network

Yasunori Hashimoto, Ikuko Nishikawa, Tohru Watanabe, Hidekatu Tokumaru
Ritsumeikan University
Department of Computer Science and Systems Engineering
Kyoto 603-77, JAPAN

Abstract

A new approach using a Hopfield type neural network to solve line balancing problems for manufacturing planning is proposed. The energy function of the network to evaluate solutions is composed of three terms; (a) an operation should be processed at one and only one workstation, (b) the precedence-relationship between two operations should be satisfied, and (c) the cycle-time of operations should be minimized. It is shown that the network can solve the line balancing problems but not always because of the difficulty to keep the precedence-relationship. Therefore, a method to keep the precedence-relationship by software logic is proposed and it is verified that the line-balancing problems can be solved with high probability.

1. Introduction

Line balancing problems are to solve the optimal allocation of operations to multiple workstations placed in a line under the constraint of given precedence-relations between operations. The optimal solution attains the minimum cycle-time by balancing the processing times on workstation.[1],[2] The number of feasible solutions increases exponentially with the problem size, and the computational time to obtain the optimal solution increases with the same order by the conventional approaches. In this paper, a new method to solve the line balancing problems by using a Hopfield type neural network is proposed, and its validity is examined by computer simulation.

2. Modeling of Line Balancing Problems

An example of line balancing problem is shown in Fig.1. Each operation is represented by a circle, labeled by an integer number inside the circle. These circles are connected by arrows indicating the precedence-relationship between operations. Real numbers above circles show processing times of the operations.

Group of operations enclosed by dotted lines in Fig. 1 show the optimal solution. Operations 1, 2, 3 are processed at workstation (WS) 1, operations 4 and 7 at WS 2, operation 6 at WS 3 and operations 5 and 8 at WS 4. Real numbers attached to the groups mean the summation of operation times processed at the workstations. The maximum operation time 3.6 min. at WS 2 is adopted as the optimum cycle-time (tact-time) of the manufacturing system. Therefore, idle times at WSs 1, 2, 3 and 4 become 0.4, 0.0, 0.2 and 0.1, respectively. The conformation rate [(total operation times) / (cycle-time × workstation

number)] to evaluate the efficiency of the solution becomes 95%.

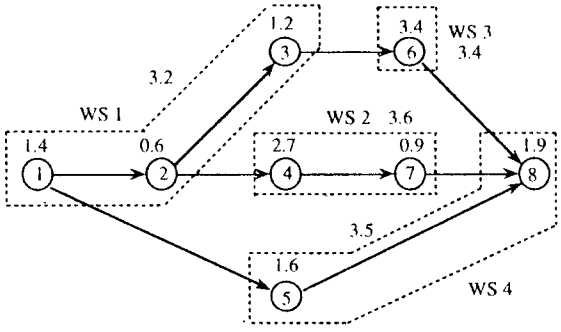

Fig. 1 An example of a line balancing problem.

3. A Hopfield Network

A Hopfield network is composed of neurons connected mutually via weighted synapses. The neuron in the proposed model[3],[4] has two indices; i and m stand for operation number and workstation number, respectively.

Then an input value $u_{im}(k)$ inside (i, m)-th neuron at discrete time k is given by;

$$u_{im}(k)=\sum_{j}^{N}\sum_{n}^{M} w_{im,jn}X_{jn}(k-1)+h_{im} \tag{1}$$

where M and N are the total numbers of workstations and operations, respectively. $w_{im,jn}$ is the synaptic weight from (j, n)-th neuron to (i, m)-th neuron. h_{im} is the threshold value of (i, m)-th neuron.

The output value $X_{im}(k)$ of the neuron is calculated as,

$$X_{im}(k)=\frac{1}{2}\left(1+\tanh\frac{u_{im}(k)}{\mu_o}\right) \tag{2}$$

μ_0 : constant

When the Hopfield network has symmetric weights as $w_{im,jn}= w_{jn,im}$, an energy function

$$E(k)=-\frac{1}{2}\sum_{i,j}^{N}\sum_{m,n}^{M} w_{im,jn}X_{im}X_{jn}-\sum_{i}^{N}\sum_{m}^{M} h_{im}X_{im} \tag{3}$$

is defined as a Lyapunov function, and it decreases monotonically according to the dynamics given by Eqs. (1) and (2). Therefore, the outputs of the network converge to their respective values representing a solution of the problem.

4. Application of the Hopfield Network to Line Balancing Problems

The energy function for line balancing problems is composed of three terms Φ_α, Φ_β, Φ_γ.

$$E(k)=\Phi_\alpha(k)+\Phi_\beta(k)+\Phi_\gamma(k) \tag{4}$$

Φ_α is defined for satisfying the condition that each operation should be processed at one and only one workstation;

$$\Phi_\alpha(k)=\frac{\alpha}{2}\sum_{i}^{N}\left(\sum_{m}^{M}X_{im}-1\right)^2 \tag{5}$$

α : constant

Φ_α becomes the minimum when each operation is belong to only one workstation.

Φ_β is defined for satisfying the condition that the precedence-relationship between two operations should be satisfied;

$$\Phi_\beta(k)=\frac{\beta}{2}\sum_{m,n}^{M}\sum_{i,j}^{N} a_{ji}q_{mn}X_{im}X_{jn} \tag{6}$$

β: constant

Φ_β takes the minimum value when all precedence relations are satisfied. q_{mn} represents that the order of the workstations along the line is satisfied or not;

$$\begin{aligned} q_{mn} &= 1 \quad \text{for } m<n \\ &= 0 \quad \text{for otherwise} \end{aligned} \tag{7}$$

a_{ij} in Eq. (6) is a binary to represent the precedence relation between two operations. It is introduced from a matrix $P = [p_{ij}]$ to represent only the explicit precedence relationship between two operations as like as shown in Fig. 1. The

matrix $\boldsymbol{P} = [p_{ij}]$ is defined as;

$$\begin{aligned} p_{ij} &= 1 \quad \text{if there is the precedence-relationship for } i \rightarrow j, \\ &= 0 \quad \text{for otherwise} \end{aligned} \tag{8}$$

$N \times N$ binary matrix $\boldsymbol{B}$ is introduced for intermediate processing;

$$\boldsymbol{B} = \boldsymbol{I} + \boldsymbol{P} \tag{9}$$

where $\boldsymbol{I}$ stands for $N \times N$ unit matrix and "+" means Boolean summation.

Matrix $\boldsymbol{A} = [a_{ij}]$ is obtained from the Boolean products of $\boldsymbol{B}$;

$$\boldsymbol{A} = \boldsymbol{B}^N - \boldsymbol{I} \tag{10}$$

Matrix $\boldsymbol{A}$ represents the enlarged precedence relations between all operations. For example, $\boldsymbol{P}$ and $\boldsymbol{A}$ for the problem shown in Fig.1 are calculated as;

$$P=\begin{pmatrix} 0&1&0&0&1&0&0&0\\ 0&0&1&1&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0 \end{pmatrix} \quad A=\begin{pmatrix} 0&1&1&1&1&1&1&1\\ 0&0&1&1&0&1&1&1\\ 0&0&0&0&0&1&0&1\\ 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0 \end{pmatrix}$$

Φ_γ is defined for minimizing the cycle-time of the line-flow;

$$\Phi_\gamma(k) = \frac{\gamma}{2}\sum_m^M \left(\frac{1}{N}\sum_i^N t_i - \sum_i^N t_i X_{im} \right)^2 \tag{11}$$

γ : constant

t_i : a processing time of the i-th operation

Φ_γ becomes the minimum when all operation times at workstations correspond to the mean operation time. The synaptic weight $w_{im,jn}$ and the threshold value h_{im} of Eq. (3) for line balancing problems are introduced by comparing the coefficients of Eq. (3) with those of Eq. (4) substituted by Eqs. (5), (6), (11).

$$w_{im,jn} = -\alpha\,\delta_{ij} - \frac{1}{2}\beta\left(a_{ji}q_{mn} + a_{ij}q_{nm}\right) - \gamma\, t_i t_j \delta_{mn}$$

$$\begin{aligned} \delta_{ij} &= 1 \quad \text{for } i = j \\ &= 0 \quad \text{for otherwise} \end{aligned} \tag{13}$$

$$h_{im} = \alpha + \gamma \frac{1}{N}\sum_j^N t_j \cdot t_i \tag{14}$$

t_i : the operation time of operation i

An additional constant term to Eq. (3) is contained in Eq. (4) as;

$$const. = \frac{\alpha}{2}N + \frac{\gamma}{2}M\left(\frac{1}{N}\sum_i^N t_i\right)^2 \tag{15}$$

The term $a_{ij}q_{nm}$ in second term of Eq. (12) is added so as to satisfy the condition of symmetry.

5. Computer Simulation

Fig. 2 shows an example of the transition of the neuron transition for the problem shown in Fig. 1. Y_{im} is defined as the state to represent the ignition of the (i, m)-th neuron;

$$\begin{aligned} Y_{im} &= 1 \quad \text{for } X_{im} \geqq 0.5 \\ &= 0 \quad \text{for } X_{im} < 0.5 \end{aligned} \tag{16}$$

The initial condition of the network outputs is set by using random numbers. The outputs converge to the optimum state after 236 times learning. The network reaches the optimum condition, in this case.

However, feasible solutions, which satisfy the two constraints, cannot be always given. Especially, it is rather difficult to get solutions to keep througthly the precedence-relationship. Therefore, a practical method is proposed. That is, the solution is proposed by using only the first and third terms of Eq. (4), for keeping the symmetry. When the output X_{im} is given at time k, it is checked that the precedence relations on X_{im} is kept or not. If it is kept, X_{im} does not change, if not, changes to zero as follows,

$$\begin{aligned} X_{im} &= X_{im}; \quad a_{ji}q_{mn}Y_{im}(k)Y_{jn}(k) = 1 \\ &= 0\,; \quad a_{ji}q_{mn}Y_{im}(k)Y_{jn}(k) = 0 \end{aligned}$$

$$\text{for } i = 1, \sim, n,\ i \neq j \qquad n = 1, \sim, m - 1 \tag{17}$$

This method is examined for 3 cases studies of line balancing problems; (M, N) = (3, 8), (5, 10) and (6, 12). Fifty problems are prepared for each case.

The obtained solutions are evaluated by a performance index to represent a formation efficiency defined as;

$$PI=\frac{\text{(total processing time for all operations)}}{\text{(cycle-time)}\times\text{(number of workstations)}}\times 100(\%) \quad (18)$$

			Operation No. i							
			1	2	3	4	5	6	7	8
Ignition Y_{im}	WS No. m	1	1	1	0	1	0	0	1	0
		2	0	1	1	0	1	1	1	0
		3	1	1	0	0	0	1	0	1
		4	0	0	1	0	1	0	1	1

(a) Ignition state at the initial condition

			Operation No. i							
			1	2	3	4	5	6	7	8
Ignition Y_{im}	WS No. m	1	1	1	1	0	0	0	0	0
		2	0	0	0	1	0	0	0	0
		3	0	0	0	0	1	1	1	0
		4	0	0	0	0	0	0	0	1

Cycle-time = 5.9 min.

(b) Ignition state after 100 times learning

			Operation No. i							
			1	2	3	4	5	6	7	8
Ignition Y_{im}	WS No. m	1	1	1	1	0	0	0	0	0
		2	0	0	0	1	0	0	1	0
		3	0	0	0	0	0	1	0	0
		4	0	0	0	0	1	0	0	1

Cycle-time = 3.6 min.

(c) Ignition state after 236 times learning

Fig. 2 Transition of the ignition of the Hopfield network for the problem shown in Fig. 1.

Table 1 Rate of the obtained solution with PI more than 90% and 99%

(M, N)	$PI \geqq 90\%$	$PI \geqq 99\%$
(3, 8)	96.0 (%)	90.0 (%)
(5, 10)	78.0	86.0
(6, 12)	46.0	60.0

Table 1 shows the rate of getting the formation efficiency more than 90% and 99% for the three cases of (M, N), each averaged for 50 examples. Fairly good result is obtained especially for the case of (M, N) = (3, 8). But the performance index gets worse according to the increase of workstations and operations. The existence of local minima becomes more serious for the large scale problem.

6. Conclusion

A method to solve line balancing problems by a Hopfield type neural network is proposed. The energy function for the network is composed of two constraint terms on the occupation of workstations and precedence-relationship between operations, and a cost term which should be minimized by the optimal solution. It is shown that the network can solve the line balancing problems but not always because of the difficulty to keep the precedence-relationship. Therefore, a practical method is proposed to keep the precedence relationship by software logic and it is shown that solutions are given with high probability.

7. References

[1] K. Hitomi, *Manufacturing Systems Engineering*, London. Taylor & Francis, 1993.

[2] K. Hitomi, *Manufacturing Systems Engineering* (in Japanese), Japan. Kyoritsuyuppan, pp. 66-69, 1993.

[3] J. Hopfield, " Neural networks and physical systems with emergent collective computational properties," *Proc. Nat. Acad. Sci.*, USA, vol. 79, pp. 2554-2558, 1982.

[4] J. Hopfield, and D. W.Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141-152, 1985.