

リアルタイム・ネットワークのための

4N-6

ニューラル・スケジューラ

新田健一, 相原玲二, 山下雅史, 阿江忠

(広島大学 工学部)

1. はじめに

マルチメディア・ネットワークでは、画像、音声、テキスト等、多種のメッセージが共通の通信チャネルを使って転送される。そのため、共有資源である通信チャネルへのアクセス競合がメッセージ間で生じ、各メッセージの転送は互いに影響を及ぼしあう。つまり、各メッセージの転送時間はメッセージ相互の関係によって決まる。

本稿では、メッセージの転送時間を、転送要求が発生してから転送が完了するまでの時間とする。音声や画像等では、そのメディアの性質から、メッセージの転送時間に厳しい制約が要求される。このようにメッセージの転送時間に制約があるネットワークを、ここではリアルタイム・ネットワークと呼ぶ。ネットワークの物理的な通信チャネルは複数、すなわち、マルチチャネルを対象とする。

リアルタイム・ネットワークにおける転送時間に関して以下の二つの問題が考えられる。

- (1) 各メッセージの転送時間が制約を満足するかどうかを調べる [1], [2]。
- (2) 各メッセージの転送時間が制約を満足するように、メッセージをチャネルに割当てるスケジューリング問題。

前者は解析的に考えられ、後者は合成的に考えられる。本稿では後者の問題を対象とし、それを限られた時間内に解くために、ニューラル・ネットワークを用いたハードウェア・スケジューラを提案する。ここで扱うスケジューリング問題は、オペレーティング・システムにおいて、プロセスへ資源を割当てる問題と本質的には同じである。ただし、ネットワークの場合には、外部の状況が変化するため(例えば、送信したメッセージにエラーが生じ、相手に正しく到達せず、再送が必要になる)、それに対応して動的なスケジューリングが要求される。

2. リアルタイム・ネットワークにおける

スケジューリング問題

リアルタイム・ネットワークにおけるスケジューリングを、以下の例を用いて説明する。

物理的な通信チャネルを二つ備えたネットワークにおいて、時刻0におけるメッセージの送信要求が以下のように与えられたとしよう。

メッセージ	メッセージ長	時間制約
M1	3	4
M2	2	5
M3	3	5
M4	1	3

表1 スケジューリング問題の例

ここで、1単位長のメッセージは1単位時間で送信できるとする。また、メッセージの転送時間制約の最大値を、その問題のウインド・サイズと呼ぶ(この例では5)。

上の例に対する一つの解を図1に示す。各行はメッセージが転送される論理チャネルを表わし、各列は単位時間を表わしている。

次に、M4送信直後にエラーが発生し、再送が必要になった場合を考える。この場合、新たな状況(メッセージ長、転送時間制約)に対する再スケジューリングが必要である。再スケジューリングは時刻1までに行わなければならない。解の一つを図2に示す。

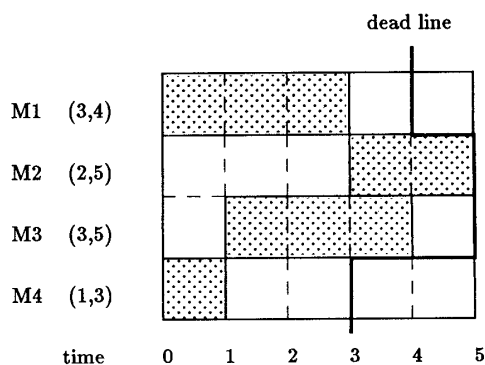


図1 スケジューリング結果

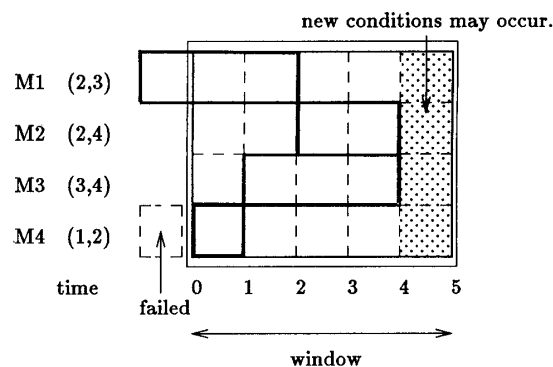


図2 再スケジューリング結果

現在の 64kbps 系の ISDN 等を考えても、単位時間は 10msec 程度となる。そのため、スケジューリングを実現するには、ソフトウェアによる方法では不可能と考えられる。一方、メッセージ長およびメッセージ転送時間に対する制約は各スケジューリング時によって異なるため、プログラマブルでなければならない。そこで、本稿では、スケジューリングのためのプログラマブルな専用ハードウェアを提案する。

3. ニューラル・スケジューラ

2で示したスケジューリング問題は、組合せ最適化問題の一つであり、それをハードウェア的に解くときのモデルの一つとして、ニューラル・ネットワーク・モデルがあるが、従来のモデル[3]では、各素子の入力に付加された係数、しきい値を問題に応じて与える必要がある。その際、問題を写像したニューラル・ネットワーク全体を考える必要があるが（この構成法を直接法と呼ぶ）、問題が複雑になると、全体を構成することが非常に難しくなると考えられる。それに対して、動作の保証されたニューラル・ネットワークを基本単位と考え、それを組み合わせることによって問題を解くシステムを構成しようとする方法（これをビルディング・ブロック法[4]と呼ぶ）が考えられる。本稿では、後者の手法の一つである直交アーキテクチャ[5]を用い、先のスケジューラを実現する。

ビルディング・ブロック法では、以下のようにしてシステムを構成する。

(1) 安定点の集合が判っているネットを基本ブロックとして用意する。

(2) 解に近い安定点を持つように(基本)ブロックを合成する。

(3) (近似)解が得られるまで(2)を繰返す。

この手法では、(2)でブロック合成アルゴリズムが必要であり、その良し悪しが全体の性能を左右する。

ここでは、基本ブロックとして k -out-of- n 回路 (${}_n C_k$)を用いる[6]。 ${}_n C_k$ は以下のように表わされる。

しきい値関数

$$y_i = f(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$$

$$y_i = \begin{cases} 0 & \text{if } (\sum_{j=1}^n a_j x_j) - \theta > 0 \\ 1 & \text{if } (\sum_{j=1}^n a_j x_j) - \theta < 0 \end{cases}$$

ここで、 $a_i = 0$, $a_j = 1$ または 0 ($i \neq j$), $\theta > 0$ である。

このネットワークは、しきい値 $n-k-1 < \theta < n-k$ に対して ${}_n C_k$ 個の安定点を持つことが知られている。インプリメンテーションでは、実現容易さの点から、しきい値素子を等価なブール素子に置き換えて実現する。

直交アーキテクチャ(図3)では組合せ問題を行列形式で入力する。行列は図1, 2をそのまま記述した形式になる。時間制約(デッドライン)がマスクに相当し、しかるのち、制約条件がビットにより表現される。スケジューリング問題を単位時間内に解くために、基本ブロック(図3における一つの行または列)の集合を基本演算とするスキーマを作る必要があるが、グリーディな方法で十分である。

現在、 $n=8$ を基本ブロックとするプロトタイプを製作中である。一般のブール関数を容易に実現できることから、素子としてはROM/RAMを用いている。

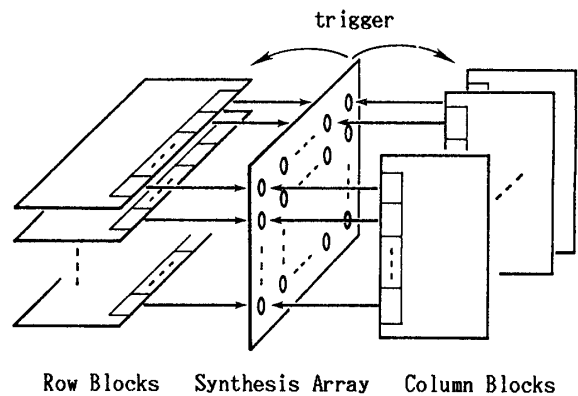


図3 直交アーキテクチャ

なお、本研究の一部は電気通信普及財団の援助によるものである。

参考文献

- [1] 松本,クニア,山下,阿江: "マルチチャネルネットワークにおける通信時間予測の複雑さ", 信学誌, Vol.J70-D, No.8, pp.1462-1468, Aug. 1987.
- [2] Ae, T., Yamashita, M. and Matsumoto, H.: "A response time estimation of real-time networks", Proc., IEEE Real-Time Systems Symposium, pp.198-207, San Jose, 1987.
- [3] Hopfield, J.J. and Tank, D.W.: "'Neural' computation of decisions in optimization problems", Biol. Cybern., 52, pp.141-152 1985.
- [4] 阿江,相原: "ニューロコンピュータの超並列性とその実現について", 信学全大, SD-4-3, Mar. 1988.
- [5] Ae, T. and Aibara, R.: "A neural network for 3-D VLSI accelerator", (to appear at International Workshop on VLSI for Artificial Intelligence), 1988.
- [6] Ae, T., Nagami, H. and Yoshida, N.: "On multi-stable transistor circuits using threshold logic operation", International Journal of Electronics, 36, 6, pp.849-856, 1974.