

# ハードウェア・シミュレータを用いた 最適径路計算システム

古 村 文 伸

## 1. はじめに

径路誘導システムでは各車を誘導する径路として、刻々変化する道路網内の混雑状況、通行規制状況に従って目的地までの最適径路すなわち最短時間径路をリアルタイムに計算することが必要となる。この最短径路問題は従来ラベリング法等のアルゴリズムを用いた計算機プログラムで解かれていたが、ネットワークのノード数すなわち道路網の地点数が多くなると処理時間が急速に大きくなるという難点があった。

そこでこれを解決するためわれわれは道路網を模擬したデジタル回路網を作り、その中を伝播する電気信号で車の走行を模擬することによって、最短径路を高速に探索するハードウェア・シミュレータを開発した。

本文ではこのハードウェア・シミュレータの原理と構成について述べる。さらに通産省工業技術院の大型プロジェクト「自動車総合管制技術の研究開発」(自総管)で実施した径路誘導パイロット実験への適用結果を示す。

## 2. 最短径路問題とその従来解法

### 2.1 最短径路問題

最短径路問題は道路交通制御のほか、輸送計画、日程計画等

の数理計画問題にしばしば現われる重要な問題である[1]。この問題はつぎのように定式化できる。

図1に示すような  $N$  個のノード  $\{i\}$  ( $i=1, \dots, N$ ) とノードを結ぶ有向アークから成るネットワークを考える。アーク  $(i, j)$  には距離あるいは時間等に相当する正のアークコスト  $c(i, j)$  が与えられているとする。このとき問題は与えられた出発ノード  $s$  と目的ノード  $e$  に対し、径路に沿ってのアークコストの和

$$J = \sum_{i=1}^{m-1} c(k_i, k_{i+1})$$

が最小となるノード列  $\{k_i\}$  ( $i=1, 2, \dots, m$ ) を求めることである。ただし  $k_1=s, k_m=e$  とする。このとき解として得られたノード列が最短径路を、また  $J$  が径路のコストを与える。

### 2.2 ソフトウェアによる解法

最短径路問題は従来LP, DP, ラベリング法等のアルゴリズムを用いて、計算機ソフトウェアにより解く方法が多数開発されている。このうち

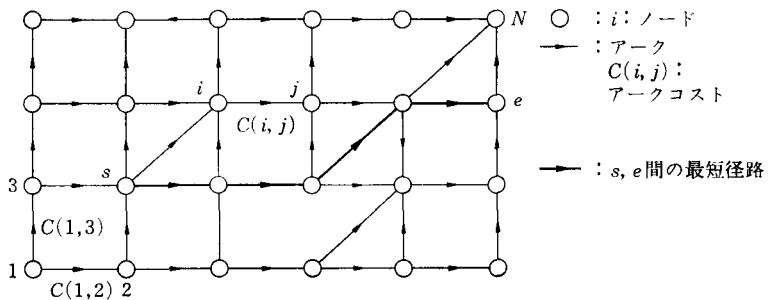


図1 ネットワークと最短径路

こむら ふみのぶ 株式会社製作所

最も一般的なものはラベリング法で、中でも基本形はつぎの古典的ラベリング法[2]である。

すべてのノード  $i$  にラベル  $(d_i, T_i)$  を与える。また  $i$  に入るアークの始ノードの集合を  $P_i$  とする。このときつぎのステップにより  $s$  から他の全ノードへ到る最短径路が求まる。

ステップ1: 各ノード  $i$  のラベルの初期値を  

$$\begin{cases} (s, \infty) & i \neq s \text{ のとき} \\ (s, 0) & i = s \text{ のとき} \end{cases}$$
 とする。

ステップ2: すべてのノード  $i$  について  

$$\min_{j \in P_i} \{T_j + c(j, i)\}$$
 なる  $j$  を求め、ラベルを  $(j, T_j + c(j, i))$  でおきかえる。

ステップ3: すべてのノードについて第2ラベルの値が変化しなくなるまでステップ2を繰り返す。

こうして得た各ノード  $i$  の第1ラベル  $d_i$  は最短径路上で  $i$  の直前にあるノードを、また第2ラベル  $T_i$  は  $s$  から  $i$  までの最短径路のコストを与える。

このアルゴリズムの計算量を見積ってみる。各ノードに入るアークの数を平均  $M$  とすれば、ステップ2で各ノード当り加算と比較の演算はそれぞれ  $M$  回行なわれる。ノード数  $N$  のとき1回のステップ2で  $N \times M$  回となる。またステップ2全体の反復回数は高々、出発ノードからの最短径路上に含まれるノード数の最大値に等しい。ネットワーク内のノード分布が一樣であればこの最大値は  $N^{0.5}$  のオーダーとなる。したがって1出発ノードから他の全ノードへの最短径路を探索するに要する計算量は  $N \times M \times N^{0.5} = M \times N^{1.5}$  に比例する。

最短径路計算は対象ネットワーク形状、プログラム作成方法により同一アルゴリズムでも処理時間が変化する。ここでは1つの目安として上記アルゴリズムを FORTRAN プログラム化し、自総管パイロット実験の対象である東京の道路網

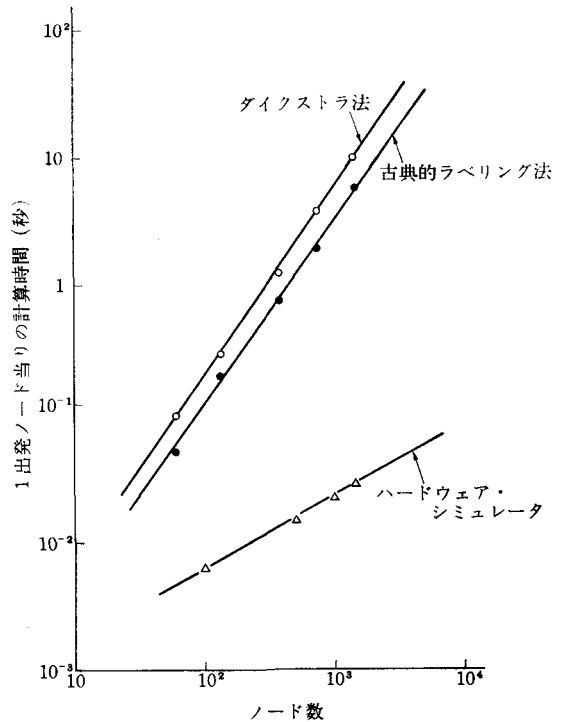


図2 最短径路計算時間の比較

( $N=1465$ ) およびその一部を用いて、汎用計算機 HITAC-M160II でランした結果を図2に示す。一方径路誘導システムで許容される処理時間を考えてみる。上記パイロット実験システムでは、272の出発ノードから全目的ノードへの最短径路を、通行規制の異なる8つのケース(4車種×高速道路利用可・否の2ケース)について5分以内に求める必要がある。すなわち1出発ノードから1回当りの許容処理時間は0.14秒であり、これは上記ラベリング法による処理時間の1/40程度に相当する。古典的ラベリング法の改良としてダイクストラ法[2]がある。これはノード当りのアーク数  $M$  が  $N$  に近い密なネットワークでは処理時間短縮に有効である。しかし道路網では  $M$  は平均3~4で  $N$  に比べはるかに小さく、ラベルのソートに要する処理のためかえって処理時間が増える。

### 3. ハードウェア・シミュレータ

#### 3.1 ハードウェア・シミュレータの原理

ノード数  $N$  が増えるにつれラベリング法の処

理時間が急速に長くなる原因は、上記ステップ2で本来独立した各ノードについての第2ラベルの加算と比較の演算をノードごとに逐次実行することにある。そこでわれわれは発想を転換し、対象ネットワークに相似な（ノードの接続関係が相等的）回路網を作り、その中で電気信号で交通流を模擬することにより最短経路を高速に求めるアプローチをとった。

この考え方にもとづいて新しく開発されたものが、ハードウェア・シミュレータ[3]である。

ハードウェア・シミュレータは図3に示すよう

に、ネットワークのノードに対応したノードモジュールと、アークに対応した信号線から成る。ノードモジュールは、ゲートとアークごとのカウンタをもつデジタル回路で、1つのLSI (Large Scale Integrated circuit) に封入される。このハードウェア・シミュレータを用いてつぎのようにして最短経路が求められる。

各カウンタには対応するアークのアークコストの値が初期設定されている。1つの出発ノードに対応するノードモジュールのゲートからパルス信号（旅行信号とよぶ）を発する。パルス信号がカ

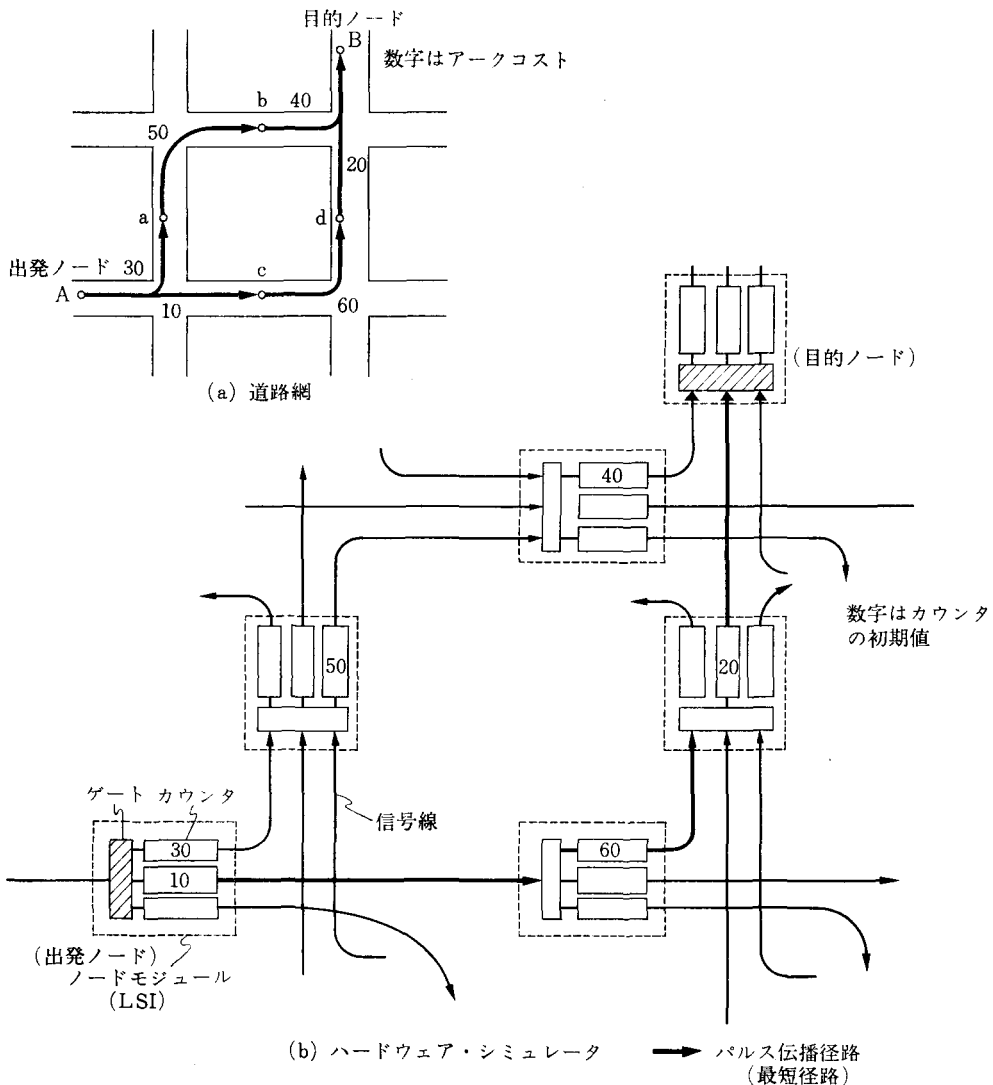


図3 ハードウェア・シミュレータの構成

ウンタに伝わると、モジュールの外部から与えられるクロック信号によってカウンタの内容が1ずつ減少させられる。そしてカウンタの内容が0になった時点で旅行信号が、下流に接続されたモジュールへ信号線を通して伝えられる。信号を受けたモジュールは、上と同様に各アークのカウンタの内容の減算を開始する。それと同時にゲートを閉め、その後他の上流側モジュールからの旅行信号を受け付けないようにする。そしてカウンタの内容が0になると旅行信号がさらに下流側のモジュールに伝えられる。

こうしてすべてのモジュールに共通のクロック信号を用いて旅行信号の遅延と伝播を進めれば、各モジュールへは可能な全径路のうち最もコストの小さい径路をとった旅行信号だけが受け付けられる。こうしてすべてのモジュールに旅行信号が到着したならば動作を停止する。このとき各モジュールに記憶された先着アーク番号をたどれば、出発ノードから他の全ノードへの最短径路が得られる。またモジュールにクロック信号を計数するカウンタを別途設け、旅行信号がそのモジュールに到着した時点の値を保持させれば、これは最短径路のコストを与える。このハードウェアシミュレータの演算過程は、旅行信号を車の1つの集団とみなせば、出発ノードからいっせいに走り始めた車群が、途中の各ノードで各アークに分岐し、可能なすべての径路を走行して各ノードへの先着を競うレースを模擬している。

### 3.2 計算の速度

ハードウェア・シミュレータにより1出発ノードから他の全ノードへの最短径路を求めるに要する時間は、出発ノードから最も遠いノードへ旅行信号が到達するまでの時間に等しい。ノード数が $N$ のとき、ノードの分布、アークコストが一様であればこれは $N^{0.5}$ のオーダーに比例する。したがって処理時間がほぼ $N^{1.5}$ に比例するラベリング法によるソフトウェア解法に比べ、 $N$ が大きくなるほど有効になる。前述の東京の道路網にハー

ドウェア・シミュレータを適用した時の処理時間の推定結果を図2に示す。ラベリング法に比べ10倍ないし100倍程度高速である。パイロット実験( $N=1465$ )の1出発ノード当りの処理時間は要求条件を十分満足している。

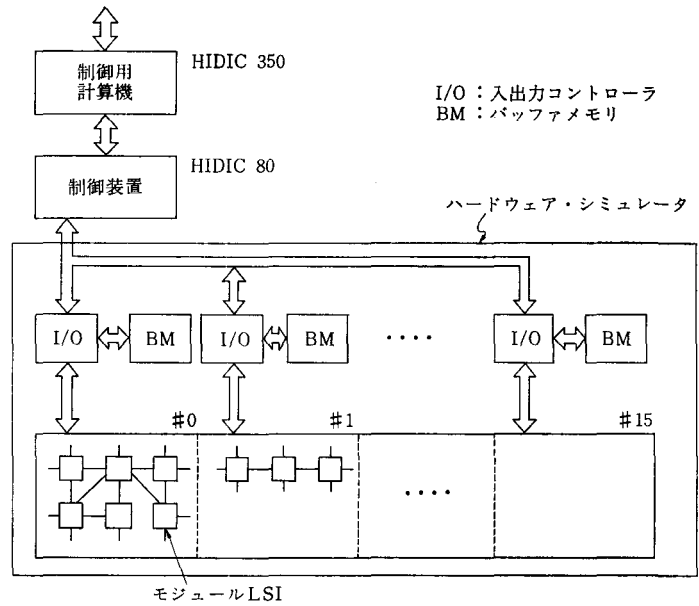
ハードウェア・シミュレータによる解法をラベリング法と比較すると、ラベリング法のステップ2の演算のうち、第2ラベルの加算をハードウェア・シミュレータではカウンタによる旅行信号の遅延で、また第2ラベルの比較による最小値の検出をハードウェア・シミュレータではゲートを閉じる操作で実現している。しかもこれらの動作は出発ノードから波紋のように伝播する旅行信号の前線に当るノードで同時並列に行なわれる。このようにハードウェア・シミュレータはアークコストを所要時間とみなし、電気信号を利用して物理的シミュレーションを行なうことにより、ラベリング法のステップ2の逐次処理による処理時間増加の難点を解決したものである。

### 3.3 専用計算機としての位置づけ

道路網交通流を模擬するハードウェアとして従来、ダイオード回路を用いたアナログ・シミュレータ([4]など)、専用ハードウェアとデジタル計算機を組合せたシミュレータ[5]等がある。前者は道路網内の最適交通流分布をネットワークフロー問題[1]として解くものである。後者は個々の車の動きを再現でき、信号制御効果の事前評価等に利用されている。これに対し本ハードウェア・シミュレータは交通流の再現を目的とするのではなく、最短径路を求めるために車の流れを模擬している。したがって道路網以外のネットワークの最短径路問題にも適用できる。

本ハードウェア・シミュレータの特徴として、1000個以上のノードモジュールLSIから構成されている点がある。LSI化された演算ユニットとして最近高速フーリエ変換や浮動小数点演算を行なうものが出現している。これらは主にマイコンと組合せ数値計算の高速化のために用いることを

図 4 専用シミュレータシステム



目的としている。ハードウェア・シミュレータのように単一の専用 LSI モジュールを多数組合せて特定の問題を解く、大規模な専用計算機は従来なかった。

#### 4. 経路誘導システムへの適用

##### 4.1 専用シミュレータシステム

自総管パイロット実験の経路誘導システム用の最適経路計算装置として「専用シミュレータシステム」[6]を開発した。これは図4に示すように約1200個のモジュール LSI を用いたハードウェア・シミュレータ、その制御装置、制御用計算機とから構成される。

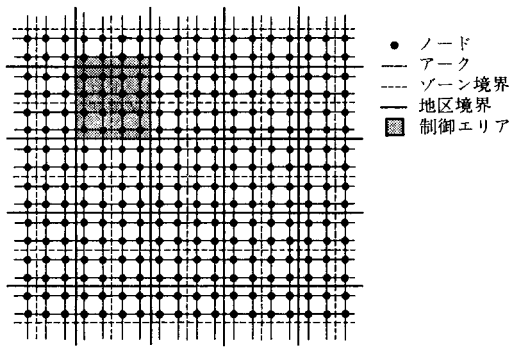
ハードウェア・シミュレータはパイロット実験エリア内の各地点を出発ノードとして、都内の道路網について最短時間経路を探索する。この経路計算の入力として必要な各アークの60分先までの予測アークコストすなわちアークの旅行時間の予測値、および通行規制データは制御装置からセットされる。予測アークコストは15分ごとの未来値が用意され、ハードウェア・シミュレータで1出発ノードから経路計算を開始後15分、30分、45分

に相当する時間だけクロックが進んだ時点で計算をいったん停止し、各アークのカウンタの内容を該当する時点のアークコストに書きかえたのち計算を再開続行する。こうすることにより未来の旅行時間および通行規制の変化に適応した最短経路が得られる。

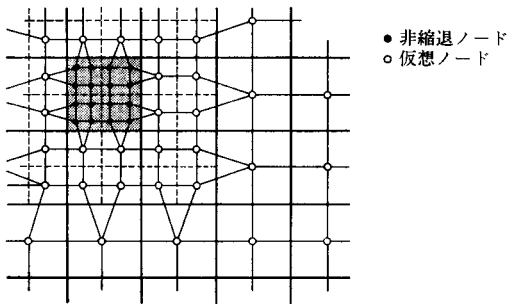
ハードウェア・シミュレータからは計算結果として、各目的ノードへの最短経路について出発ノードからの進路（左折、直進、右折等）が出力される。制御用計算機は、パイロット実験エリア内86個の各交差点についてこの進路データを編集してガイドテーブルを作る。これが各交差点の路上機に送られ車の誘導に使われる。

##### 4.2 ネットワークの縮退モデル

ノード数の多い広域道路網を対象としてハードウェア・シミュレータを作ると、ハードウェアの規模が大きくなり、入出力データの量および処理時間が増大する。そこで元のネットワークからノード数を減らした等価ネットワークモデルを作り、このモデルを用いて最短経路計算を行なうハードウェア・シミュレータを作る必要が生ずる。このようなモデルとしてつぎの透視画法的縮退ネ



(a) 元ネットワーク



(b) 縮退ネットワーク

図5 ネットワークの縮退

ネットワークモデル[7]がある。

対象道路網を誘導制御の対象となるエリアと、その周辺の経路誘導の目的地となるエリアに分ける。自総管パイロット実験では前者はパイロット実験エリアであり、後者は残りの都内の道路網である。図5に示すように制御対象エリアを中心として、周辺エリアの遠くのノードについては、隣接する複数ノードを1つの仮想ノードに縮退させる。このとき中心からの距離が大きくなるにつれ、より多くのノードを同一の仮想ノードに縮退させることにより、透視画法的に粗いネットワークが得られる。

このときどれだけのノードを1仮想ノードに縮退させるべきかは、元のネットワークで求めた最短経路と縮退ネットワークで求めた最短経路との誤差評価により判断できる。たとえば、ある出発ノードから周辺エリアのゾーンZ内のノード*i* ( $i=1, 2, \dots, N$ )へ到る最短経路の進路を*l<sub>i</sub>*、また経路

のコストを  $T_i$  とし、縮退ネットワークでゾーンZへ到る最短経路の進路およびコストを  $l, T$  とする。このときある閾値  $\theta_1, \theta_2$  に対し、

$$\frac{1}{N} \sum_{i=1}^N \frac{|T_i - T|}{T} < \theta_1$$

$$\frac{1}{N} (l_i \neq l \text{ なる } i \text{ の個数}) < \theta_2$$

が成り立つときのみゾーンZを縮退させればよい。このような計算をあらかじめオフライン的に行なうことによってネットワークモデルを作ることができる。パイロット実験に適用した結果、1465ノードの元ネットワークから599ノードの縮退ネットワークが得られた。

### 4.3 処理速度の評価

パイロット実験の結果、ハードウェア・シミュレータによる最短経路計算処理時間は、データの入出力処理を含め1出発ノード当り1回平均0.046秒であった。272出発ノード、8車種の全ケースでは100秒となる。これとアークコストの予測、ガイドテーブル編集の処理を加えた、専用シミュレータシステムの処理時間は8.5分であり、15分周期のガイドテーブル更新には充分余裕があることが確認できた。

## 5. おわりに

ネットワークの最短経路問題を高速に解くためのハードウェア・シミュレータを開発し、交通流の経路誘導システムに適用した。ハードウェア・シミュレータは専用のモジュールLSIを1000個以上用いた専用計算機で、汎用デジタル計算機に比べ10ないし100倍の処理速度をもつ。

今回開発したハードウェア・シミュレータは東京の主要道路網を模擬するもので、制御用ミニコンと同程度の大きさである。パイロット実験と同様の経路誘導システムを都内全域に適用するためにはこの規模のシミュレータを数箇所の地区管制センターに1つずつ設置すればよい。将来超LSI技術の発達により集積度の増した回路素子を使えば、シミュレータを小型化し各交差点の路上機あ

るいは各車の車載装置に組込むことも可能となろう。こうすれば最適経路計算処理を各交差点、各車に分散でき、より迅速できめ細かな道案内、交通流制御の道が開かれよう。

また最短経路計算をサブルーチンとして用いることにより解ける問題(たとえば配分問題[ 8 ]),あるいは類似のネットワーク問題(たとえばネットワークフロー問題)向きの計算機へハードウェア・シミュレータを拡張することも可能である。

本文に述べた最短経路計算用ハードウェア・シミュレータのみならず今後他の分野においても、ますます発達する半導体技術を利用した専用計算機で、大規模な問題を解くアプローチが有効になると思われる。

#### 参 考 文 献

- [ 1 ] 関根泰次：数理計画法，岩波書店，1978.
- [ 2 ] S. E. Dreyfus : An Appraisal of Some Shortest-Path Algorithms, J. of Operations Research, 17, 395-412, 1969.
- [ 3 ] 奈良，萩原：デジタルモジュールによる相同相似形計算機，電気学会誌論文集，96巻8号，1976.
- [ 4 ] J. H. Lewer, J. M. Bullingham : Electronic Simulator Mark II for Quickest Routes in Road Network, Traffic Engineering and Control, 15, 558-559, 1974.
- [ 5 ] 谷口忠勝，高羽禎雄：交通流シミュレーション・システム TRN \* SIM II を用いた車両走行のシミュレーション，昭和51年電気学会全国大会1260.
- [ 6 ] 古村，坪井，萩原，石崎，久保：経路誘導制御用道路網交通流専用シミュレータシステムの開発，第16回 SICE 学術講演会1402，1977.
- [ 7 ] 古村，井原：交通流制御のための道路ネットワークモデルの構成，昭和53年電気学会全国大会1161.
- [ 8 ] オペレーションズリサーチ 1977年12月号：特集 IA 法.