高速通信ネットワーク制御へのORの適用

河東　晴子

通信ネットワーク制御へのOR適用の歴史は古く，待ち行列理論による電話網のトラヒック制御は百年近い歴史を持つ．その他にもグラフ・ネットワーク理論，最適化理論等さまざまなOR手法が用いられてきた．

最近の情報通信の世界は，マルチメディア化，インターネットの急速な普及等，伝送路の高速化やLSI化技術の発達と相まってめざましく変化している．急激な変化や未知の要素の登場により，過去のデータ蓄積にもとづくモデル化，解析等は困難となってきている．一方，実績がなく経験では予測がつかないものは，理論によって予測せざるをえない．現代のように実績・経験による予測が困難な時代では，新たな意味で，OR的な手法による予測が重要となってくるといえる．

企業で受け入れられやすい手法は，単純(説明すれば理解できる程度)，速い（モデル化・解析・シミュレーション等に時間がかかりすぎない)，危険側/安全側等，結果の妥当性が明らかなものである．またこれからの不確定の時代にはアダプティブな方法が好まれるであろう．以下に，オーソドックスな手法を用いた事例1，新しい手法による事例2の2個の事例を紹介する．

1．事例1：待ち行列モデルによるATM速度変換バッファ容量の検討

1.1　概　論

ATM(Asynchronous Transfer Mode)は，転送情報をセルと呼ばれる48バイトのかたまりに分割し，それぞれにあて先等の5バイトの制御情報を付加して転送する方式である．ATMでは，転送可能なセル速度の上限の範囲内で，多くの情報を高速で送りたいときはセルを次々転送し，送るべき情報が少しの場合はセル間隔をあけて転送することにより，速度の異なる情報を同一サイズのセルで転送することができる．150 Mbps（1秒間に150×10^6bitを伝送可能）のATM回線で1.5Mbps，6.3Mbps，45Mbps等の低速回線を伝送する場合，高速150Mbps回線と低速回線のインタフェース部では速度の変換を行うためのバッファが必要となる．特に高速から低速への変換の場合には，高密度で送られてきたセルを一度バッファに蓄積して，間隔をあけて低速に変換するため，バッファ溢れを起こしやすい．このバッファの容量の見積もりを行うのに当たって，待ち行列のM/D/1(B)モデルと，セル遅延変動（Cell Delay Variation: CDV）依存モデルの2種を使用した．M/D/1(B)モデルは入力がランダムで保留時間一定，バッファ容量B，出力回線1本の一般的なモデルである．CDV依存モデルは，高速回線上に散在する低速回線のセルがCDVによって互いに連続して到着した場合を模擬するモデルである．

1.2　M/D/1(B)モデル

ランダム入力，保留時間一定，バッファ容量B，出力回線1本の一般的なM/D/1(B)モデルの場合，バッファにおけるセル廃棄率を一定値以下にするために必要なバッファの容量を求めた．パラメータを**表1**に示す．

セル廃棄率$F(B)$の算出式を以下に示す［1］．

表1　M/D/1(B)モデルのパラメータ

パラメータ	説明
n	セル長
v	速度
h	保留時間: $h=n/v$
λ	生起率
a	呼量: $a=h\lambda$, 回線使用率
B	バッファ容量
$p(j)$	保留時間中にポアソン呼がj個到着する確率
$P(j)$	平衡状態確率
$\Pi^*(j)$	平衡状態において呼終了直後の系内呼数がjとなる確率
$C(j)$	$C(j)=\Pi^*(j)/\Pi^*(0)$
$F(B)$	セル廃棄率

かわひがし　はるこ　三菱電機㈱　情報技術総合研究所
〒247 鎌倉市大船5-1-1

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

$$p(j) = \frac{e^{-a}}{j!} a^j \quad (j=0,1,2,\cdots\cdots)$$

$$C(0) = 1$$

$$C(j+1) = \{C(j) - p(j) - \sum_{k=1}^{j} p(j-k-1)\,C(k)\}/p(0)$$

$$(j=0,1\cdots\cdots,\ B-1)$$

$$S(B) = \sum_{j=0}^{B} C(j)$$

$$P(j) = \frac{C(j)}{1+aS(B)}$$

$$F(B) = P(B+1) = 1 - \frac{S(B)}{1+aS(B)}$$

回線使用率aを固定した場合のセル廃棄率Fとバッファ量Bの関係を**図1**に示す．回線使用率aは低速回線速度と高速回線速度との割合と等しいと考えると，150Mbpsの場合$a=0.01$，0.04，0.3，0.6はそれぞれ1.5Mbps×1，6.3Mbps×1，45Mbps×1，45Mbps×2に対応する．セル廃棄率10^{-13}の場合は1.5Mbpsで5，6.3Mbpsで6，45Mbpsで14のバッファ容量が必要となる．

1.3 CDV依存モデル

CDV（セル遅延変動）は，セルが本来到着するべき位置からのゆらぎの許容値を表わす．CDV依存モデルは**図2**に示す単純なモデルで，最小セル間隔TpとCDV許容値tpから最大連続セル個数Xpを求め，さらに最大待ちセル個数Lpを求めることができる．このモデルのパラメータを**表2**に示す．

150Mbpsインタフェースで1.5Mbps，6.3Mbps，45Mbpsの各回線を伝送する場合のCDV値と必要バッファ量の関係を，**図3**に示す．6.3Mbpsで2 msのCDV許容値の場合30，45Mbpsで1 msのCDV許容値の場合110程度のバッファ量となる．このモデルではCDVにより多数個のセルが連続する場合を仮定しているのでM/D/1(B)モデルに比べてバッファ量は大きくなる．ゆえにCDVの発生が予想される場合は，CDV依存モデルのバッファ量が安全側の値を与える．

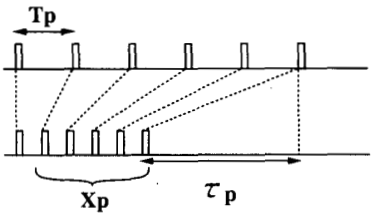


図2　CDV依存モデル

結論として，図3のCDV依存モデルのグラフから，インタフェース速度と予想CDV値に見合うバッファ容量を求めれば良い．

2．事例2：ニューラルネットを応用した最短経路問題の実時間制御への適用

2.1 概　論

WWWの普及，企業内ネットワークのインターネッ

表2　CDV依存モデルのパラメータ

パラメータ	説明
Tp	最小セル間隔 (cell)
τp	CDV許容値 (cell)
Xp	最大連続セル個数: $Xp = \frac{\tau p}{Tp-1}$
Lp	最大待ちセル個数: $Lp = Xp - \lfloor \frac{Xp}{Tp} \rfloor$

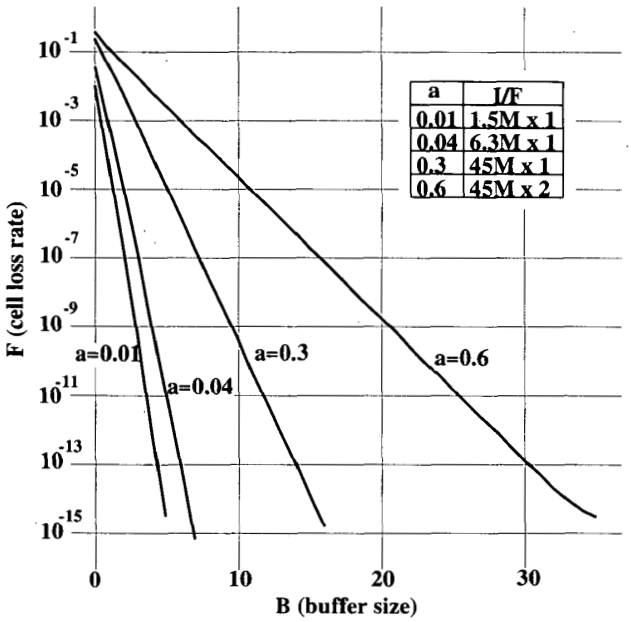


図1　セル廃棄率とバッファ容量の関係

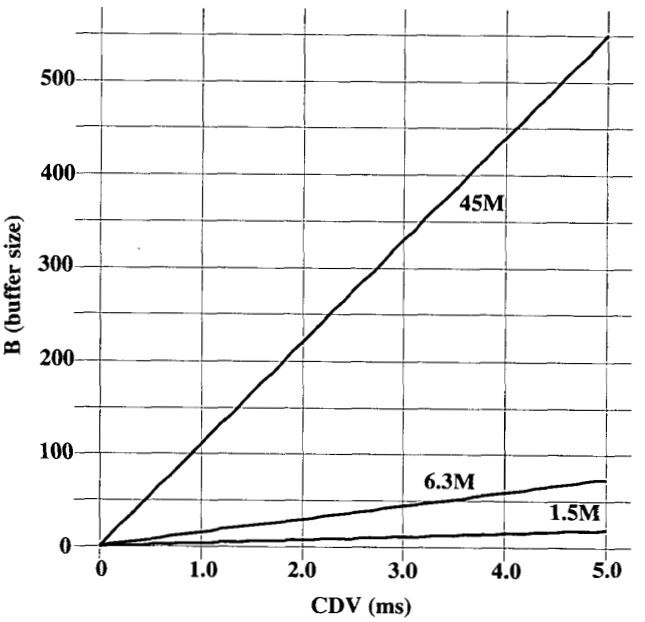
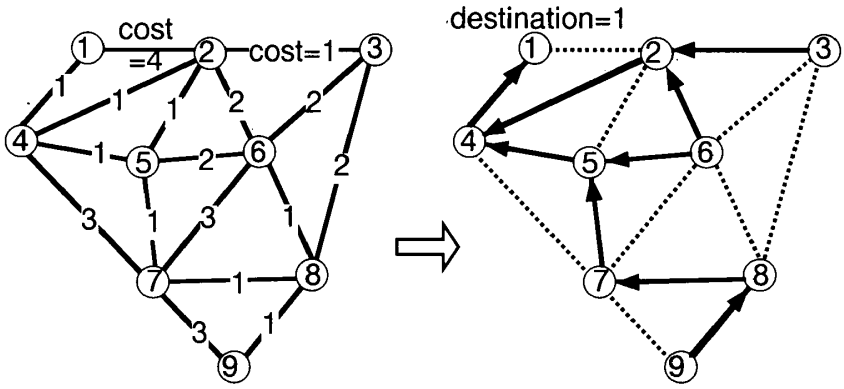


図3　バッファ容量とCDV許容値の関係

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

トによる構築等により，インターネットの利用は急激に伸びている．これまでのインターネットは自律分散的な小規模LANの接続で実現されており，ルーチングも個々のルータが固定的なテーブルをもとに行っていた．しかしこれには輻輳時の信頼性の低下等の種々の問題があった．大規模なインターネット時代になると，ルータの上位から広範囲を見渡す集中管理が必要となる．この集中管理の際に，動的な最適経路設計を行うことができれば，網資源の有効活用を図ることができるので，高速な最適経路設計方式が求められている．



a) 通信ネットワークモデル　　b) 水先案内平面

図4　通信ネットワークモデル

そこでニューラルネットを応用して最短経路問題の動的計画法による解法を実時間制御に適用する方法を考案し，高速最適経路設計を実現しようとしている[2]．

最短経路問題にはダイクストラ法等の多項式時間の解法が存在しているが，これらのアルゴリズムは計算機の使用を前提としていて，高速性が求められる実時間制御への適用は困難であった．ここで紹介する方式は，ニューラルネットの2値性を利用してハードウエアディジタル論理回路で実現できることを特長とし，これにより経路設計時間を従来の方式に比べて4桁～6桁高速化することができる．また，LSI化が容易であるという特徴も有する．

ニューラルネットによる方式の多くは学習機能を特徴としているが，この方式は，ニューラルネットの2値性と，電子回路の2値性の整合性により，ニューラルネットがハードウェアディジタル論理回路で容易に実現されることを利用して，ハードウェア回路による高速計算を実現している．

2.2　通信ネットワークモデル

通信ネットワークモデルとして，n個のノードとノード間のリンクで構成されるネットワークを考える．ノードi,j間のリンク(i,j)には方向性があり，そのコストをa_{ij}とする．コストは距離，遅延，品質等何でも良い．隣接ノード$A(i)$をノードiからリンク(i,j)が出ているノードjの集合と定義する．ここで発信ノードsと宛先ノードd間の総コストが最小となる経路を設計する問題を考える．図4-aに通信ネットワークモデルの例を示す．

2.3　ニューラルネットワーク

各々のノードiにニューロンiを配置し，ニューロンi,j間のシナプス結合荷重w_{ij}を式(1)のように設定する．このニューラルネットのダイナミクスを神経方程式(2)，記憶のメカニズムを記憶方程式(3)に示す．

$$w_{ij} = \begin{cases} a_{ij} & \text{if } j \in A(i) \\ 0 & \text{if } j \notin A(i) \end{cases} \tag{1}$$

$$x_i^k = G[\sum_{j=1}^{n} w_{ij} \cdot x_j^{k-w_{ij}}] \tag{2}$$

$$w_{ij}^* = G[\sum_k \sum_{j=1}^{n} |x_i^k - x_i^{k-1}| \times w_{ij} \cdot x_j^{k-w_{ij}}] \tag{3}$$

ただし$x>0$のとき$G(x)=1$, $x \leqq 0$のとき$G(x)=0$とする．このニューラルネットによる最適経路設計アルゴリズムは以下のとおりである．

1．ノードiにニューロンiを配置し，式(1)に従って結合荷重を設定する．各ニューロンの出力を$x_d^0=1$（活性），$i \neq d$の場合は$x_i^0=0$（不活性）と初期化する．
2．各ニューロンの出力x_i^kを式(2)に従って更新した後，シナプス荷重w_{ij}を式(3)に従って更新し，交互にこれを繰り返す．
3．全ニューロンが活性化されたら更新を停止．そこで宛先dの水先案内平面の結合荷重w_{ij}^*が得られる．
4．発信元sから宛先dへの個々の最適経路は，水先案内平面をsからdへノード毎にたどることによって得られる．
5．異なる宛先dについてはステップ1から3，異なる発信元sについてはステップ4を繰り返す．

水先案内平面（pilot plane）は宛先d毎に求められる方向性のあるネットワークで，各ノードに水先案内人(pilot)がいて次に進むべき方向を示しているイメージを表わす．発信元sから出発してノード毎に水先案内人の指示に従って進めば最適経路を通って宛先dに達することができる．図4-bに水先案内平面の例を示す．

このニューラルネットの直感的な説明は次の通りである．シナプス結合荷重w_{ij}はニューロンiからjへの結合の有無と強さを示す．ダイナミクスでは，まず全ニューロンを不活性化し宛先d対応のニューロンに刺

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

激を与える．この刺激はシナプス結合を通して伝播しニューロンを次々と活性化していく．それぞれの結合のシナプスにおいて伝播は遅延し，遅延量は荷重係数に比例する．刺激の伝播がネットワーク全体に達した時，すべてのニューロンが活性化している．刺激伝播時に各ニューロンは自分が活性化する原因となった一番影響を受けた隣接ニューロンとの結合を強化してこれを記憶する．その結果として水先案内平面を得る．

上記のニューラルネットでは最適化の指標としてリンクコストa_{ij}のみを考えた．しかし結合荷重(1)を変形することによりリンクの要求容量，ノードコストの考慮，発信元毎に水先案内平面を求めることもできる．

2.4 ハードウェア論理回路による実現

前節で示した式(2)(3)は，ニューラルネットの2値性により，下記の論理演算式で表現することができる．

$$x_i^k = (x_{j_1}^{k-w_{ij1}} \vee x_{j_2}^{k-w_{ij2}} \vee \cdots\cdots \vee x_{j_{mi}}^{k-w_{ij_{mi}}}), \quad \text{for } i \neq d,$$

$$x_d^k = 1 \tag{4}$$

$$w_{ij_l}^* = S[(x_i^k \oplus x_i^{k-1}) \wedge x_{j_l}^{k-w_{ij_l}}] \tag{5}$$

ただし$l=1,2,\cdots,m_i$で，$j_1,j_2,\cdots,j_{m_i}$はiの隣接ニューロン，$\vee$は論理和，$\wedge$は論理積，$\oplus$は排他的論理和を示す．また$S[\ ]$はセットを表す関数で，xが一度1になったらリセットするまで1を保持する．上式は排他的OR，ANDゲート，Dフリップフロップ，SRフリップフロップによるディジタル論理回路で実現できる．

上式の$x_i^k(i=1,2,\cdots,n)$の値をDフリップフロップに保持し，これをニューロンiの隣接ノード$j_l,(l=1,2,\cdots,m_i)$毎に，シフトレジスタで順次シフトしてw_{ij_l}タイミング前の値$(x_{j_l}^{k-w_{ij_l}})$まで記憶しておく．ニューロンiの状態数x_i^kは，すべての隣接ノード$j_l,(l=1,2,\cdots,m_i)$のw_{ij_l}タイミング前の値の論理和をとることにより得られる．例として，図4のネットワークのノード2に対応する状態数x_2^kを求める回路を**図5**のeq.(4)に，図4のネットワークのノード2周辺の荷重$w_{21}^*,w_{23}^*,\cdots,w_{26}^*$を求める回路を図5のeq.(5)に示す．

2.5 評　価

上記のディジタル論理回路で水先案内平面一面を求めるのに要する時間は，基準クロック毎にx_i^kとw_{ij}の更新を行うとすると，原理的に，(リンクコストの段階数×全ノード数n×基本クロック時間)を越えない．全ノード数$n=56$のネットワーク水先案内平面一面を求めるのに要する時間を算出したものを**図6**に示す．基準クロックは50MHzとした．図6には，基本クロック50MHzのSun SPARC station 10で実測した，ベルマン・フォードアルゴリズムで同じ水先案内平面を求めるのに要する時間も示してある．

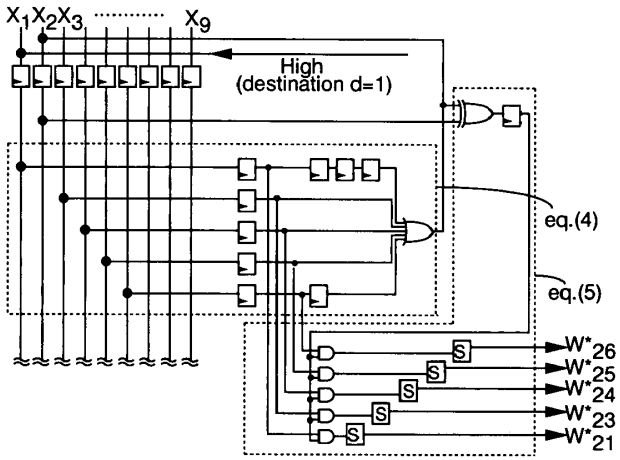


図5　ハードウェアディジタル論理回路

提案方式ではリンクコストの段階数に比例して計算時間は増加するが，ベルマン・フォードアルゴリズムの場合に比べて4桁から6桁小さくなっている．一方，この方式はリンクコストの段階数が数個～数十個の場合は有効であるが，一般の計算機の変数のように精度を上げることは，実質上不可能である．従来の方式が長時間で厳密な最適解を求めるのに適した方式であったのに対して，この方式は高速に粗い最適解を求めるのに適した方式といえる．

参考文献

[1] 秋丸，川島："情報通信トラヒック，"電子通信協会

[2] 河東他，"ATM網ルーチング方式へのニューラルネットの応用，"電子情報通信学会'95秋季大会，B-557，95/9.

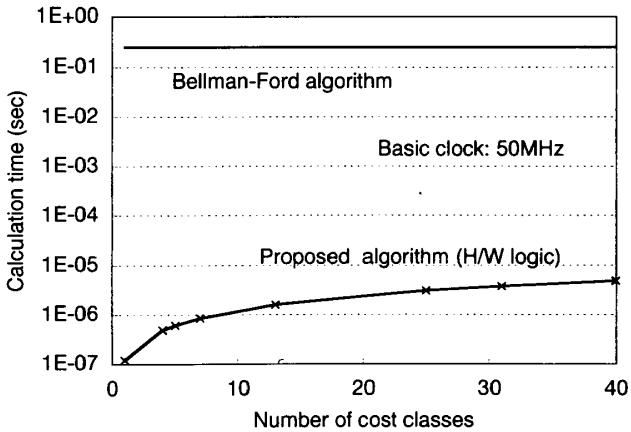


図6　計算時間の評価

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．