

## IV-226 交通シミュレーションシステム-AVENUE-における交通流計算手法の改良

熊谷組エンジニアリング部 正会員 堀口 良太  
 東京大学生産技術研究所 正会員 桑原 雅夫  
 東京都立大学工学部 正会員 片倉 正彦

## 1. はじめに

本稿では街路網の交通シミュレーションシステムAVENUE<sup>[1]</sup>の交通流モデルであるハイブリッドブロック密度法において、可変長ブロックを導入し、より少ない計算量で過飽和流を再現できるアルゴリズム、マルチスキャンハイブリッドブロック密度法について述べる。また計算機実験をおこない、発進波・停止波を含む渋滞流の上流への延伸などの密度の伝播や、飽和交通流率、交差点での遅れ時間などの基本的な項目の再現性について確認をおこなっている。

従来のハイブリッドブロック密度法は、リンクを一定距離で区切ったブロックを計算単位とし、単位スキャン時間毎に現在のブロック内車両密度と交通量-密度関数から、各ブロックの流出可能量と流入可能量を計算し、つぎにリンク下流端から順に上流側ブロックとの間の流量を、自身の流入可能量と相手の流出可能量の最小値として求め、密度の改訂をおこなうものである（図1）。通常は単位スキャン時間は1秒で、ブロックの長さは1秒間に自由旅行速度で進む距離に設定される。

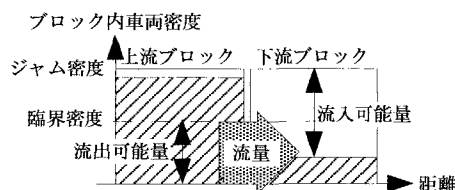


図1：ブロック間の流量の計算

## 2. マルチスキャンハイブリッドブロック密度法による計算量の軽減

前述のアルゴリズムではブロックの長さは固定されているため、単位スキャン毎の計算量はブロック数、すなわちリンクの総延長に比例する。単位スキャン時間を大きくとれば計算量は軽減されるが、信号による交通流の制御などを考えると精度に問題がでてくる。

そこでブロック毎にスキャン時間を可変にし、より長い区間のブロックを混在させることである程度の精度を保ちつつ、ブロック数が減少するマルチスキャンハイブリッドブロック密度法を提案する。具体的にはリンク下流端でのブロックのスキャン時間を最小の1秒に設定し、上流に向かうにつれスキャン時間を同じか整数倍にしたブロックで分割していくものである。ブロックのスキャン時間には精度の面から上限を設けてある。

ブロックの密度の改訂のアルゴリズムは右枠内に示される。リンク上のブロックには下流から番号*i*がつけられ、それぞれスキャン時間 $dt_i$ と長さ $dL_i$ をもっている。第*i*番目のブロックは時刻*t*が $dt_i$ の倍数となる時、ブロック内車両密度 $K_i$ 、臨界密度 $K_c$ 、ジャム密度 $K_j$ をもとに、交通量-密度の関係から、流出可能量 $A_i^{out}$ 、流入可能量 $A_i^{in}$ および上流ブロック*i*+1からの流量 $Q_{i+1,i}$ を計算し、密度 $K_i$ と $K_{i+1}$ の改訂をおこなう。さらに交通量を移動させる際に、上流ブロックの流出可能量 $A_{i+1}^{out}$ から移動させた $Q_{i+1,i}$ の分だけ差し引く。

このようなアルゴリズムにより、長い区間のブロックでは密度分布が平滑化されるものの、設定された交通量-密度関数にもとづいた密度管理が可能になる。計算量については一定距離で分割した場合と比べてブロック数が減少するだけでなく、毎スキャンにおいて必ずしも全てのブロックを計算対象とする必要がないため、大幅に軽減される。この手法をノード数31、リンク数193、リンク総延長73kmの実ネットワークに適用したところ、固定スキャン時間の場合と比べてブロック

## ブロック密度改訂のアルゴリズム

```

Loop1: 各時刻tについて
  Loop2: 各ブロックiについて
    if tが $dt_i$ の倍数 then
       $A_i^{out} = \min(K_c, K_i) dL_i / dt_i$ 
    End of Loop2
  Loop3: 各ブロックiについて
    if tが $dt_i$ の倍数 and  $K_i < K_c$  then
       $A_i^{in} = \min(K_c, K_{i+1} - K_i) dL_i / dt_i$ 
    else if tが $dt_i$ の倍数 and  $K_i < K_j$  then
       $A_i^{in} = K_c (K_{i+1} - K_i) / (K_j - K_c) \cdot dL_i / dt_i$ 
    End of Loop3
  Loop4: 下流より各ブロックiについて
    if tが $dt_i$ の倍数 then
       $Q_{i+1,i} = \min(A_{i+1}^{out}, A_i^{in})$ 
       $K_i dL_i = K_i dL_i + Q_{i+1,i} dt_i$ 
       $K_{i+1} dL_{i+1} = K_{i+1} dL_{i+1} - Q_{i+1,i} dt_i$ 
       $A_{i+1}^{out} = A_{i+1}^{out} - Q_{i+1,i}$ 
    End of Loop4
  End of Loop1
  
```

数が20%に、1スキャンあたりの計算量が15%に軽減された。実際の計算時間も1/4程度まで短縮された。

### 3. 過飽和流の再現の検証実験

このアルゴリズムにおいて、ブロックの最大スキャン時間を2秒として細かく分割した場合と、最大16秒として粗く分割した場合とで、交通流の再現性にどのように影響するか、計算機実験をおこなった。

#### (1) ショックウェーブの伝播

図2に各時刻におけるリンク上のブロックの車両密度分布を色分けしたものを示す。色が濃い区間ほど密度が高い。網掛け線はショックウェーブの理論的な軌跡を示している。粗く分割した場合はショックウェーブの波面を追うことが困難であるが、どちらの場合も発進波が停止波に追いついた時点で密度の高い区間が解消していることから、ショックウェーブが正しい速度で伝播していることがわかる。

#### (2) 渋滞の上流リンクへの延伸

図3に示されるネットワークにおいて、各リンク流入端での流率を交通量累積図と比較する。図中の破線はショックウェーブが理論速度で各リンク流入端に到達する時刻をつないだものである。どちらの場合もリンク2の流出量はボトルネック容量で一定の流率を示しており、設定された飽和交通流率で流出しているのがわかる。また、ショックウェーブが到達する時刻に各リンク流入端での流率がボトルネック容量に低下していることから、渋滞が正しい速度で上流リンクに延伸していることがわかる。

#### (3) 非飽和交差点での遅れ時間

図4に示されるネットワークにおいて、飽和度をかえて1台あたりの平均遅れ時間を調べる。図中の点は平均遅れ時間を10サイクルにわたって観測した結果を示している。上の実線はWebsterの公式の遅れ時間を下の実線は一樣到着の遅れ時間を示している。

細かく分割した場合の結果は、Websterの遅れ時間と一樣到着の遅れ時間の間に分布する。Websterの遅れに一致しないのはシステムのランダム関数の性質や、シミュレーションを十分長い間おこなっていないなどの理由が考えられる。一方、粗く分割した場合は一樣到着の遅れよりも下回る場合も見られる。これはリンク上流端の長いスキャンのブロックでランダム到着がならされてしまい、一樣到着に近くなってしまう影響と、その際に車両の走行時間が短くなってしまうためと考えられる。

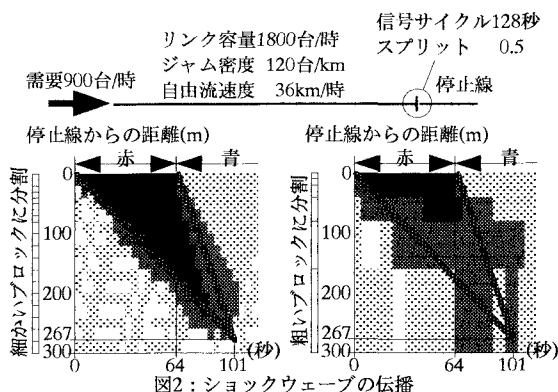


図2：ショックウェーブの伝播

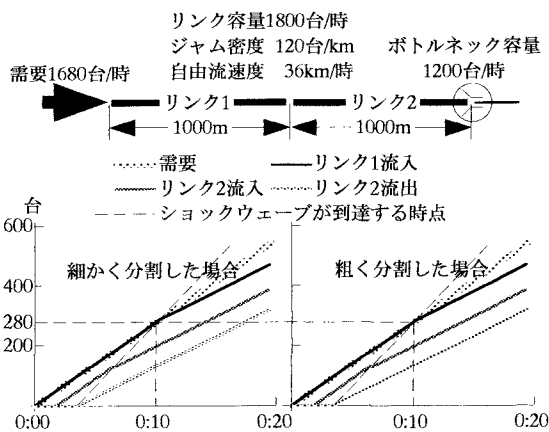


図3：渋滞の上流リンクへの延伸

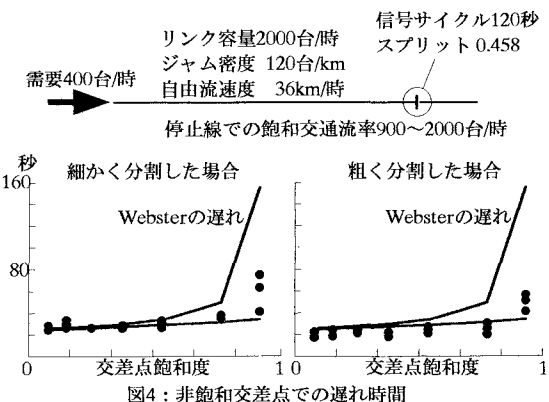


図4：非飽和交差点での遅れ時間

### 【参考文献】

- [1] R. Horiguchi, M. Kuwahara, M. Katakura, "The Model Validation of Traffic Simulation System for Urban Road Networks: AVENUE", The Second World Congress of Intelligent Transport Systems, Vol.4, 1995, pp.1977-1982