

大規模道路ネットワークシミュレータDEBNetSの改良と検証*¹

*Improvement and Verification of Dynamic Traffic Flow Simulator "DEBNetS"*¹*

菊池輝*², 加藤義昭*³, 馬淵透*³, 藤井聡*⁴, 北村隆一*⁵

By Akira KIKUCHI*², Yoshiaki KATO*³, Toru MABUCHI*³, Satoshi FUJII*² and Ryuichi KITAMURA*⁴

1. はじめに

近年のコンピュータ演算処理能力の著しい向上により、シミュレーションによる交通需要分析も実用の範疇に入り、時間軸上における個人の生活行動の再現や交通流の再現を図る研究が盛んに行われている¹⁾。交通需要解析におけるシミュレーションの利点は、時間軸を明示的に組み込むことが可能であるほか、交通行動の背後にある複雑な意思決定過程を考慮できる等の点で、旧来の集計的需要予測手法の限界・課題を克服する手法と言える²⁾。また近年では、シミュレーション手法により従来の集計的あるいは非集計的予測手法を代替することの可能性が議論されるとともに、現在ではシミュレーションでしか評価しえない政策や評価指標が増えつつあることも指摘されている³⁾。

これまでの著者らによる研究で、交通の発生・分布・分担を担う生活行動（交通行動）シミュレーションと、配分を担う交通流シミュレーションが開発されてきた。これら2つのシミュレーションモデルが取り扱う交通現象は、相互に影響を及ぼしあうため、近年注目を浴びているTDM等の政策を適切に評価するためには、この相互作用を考慮したモデルシステムが必要となる。

DEBNetS (Dynamic Event-Based Network Simulator) は、上述のような、生活行動シミュレータに基づいた総合的な需要解析のためには道路網上での自動車トリップの干渉現象に伴う移動時間の増加（いわゆる混雑現象）を考慮することが不可欠であるとの認識のもとで開発された交通流シミュレータである⁴⁾。これまでに、内外で様々な交通流シミュレーションが開発されてきたが、それらは、道路ネットワーク上の車両一台一台の挙動を逐一再現するミクロシミュレーションと、複数の車両から構成される車両群の挙動を再現するマクロシミュレーションに大別される。

ミクロシミュレーションは、主に小・中規模程度の道路ネットワークを対象とし、交差点形状の改良や信号制御の改良といった道路施設の評価に用いられることが多く、車両挙動の再現方法として、追従理論を適用したモデル⁵⁾や、交通密度と速度の関係式等を用いた方法⁶⁾がある。一方マクロシミュレーションは、主に大規模な道路ネットワークを対象とし、渋滞の影響の面的広がり等を評価するために用いられることが多い。マクロシミュレーションにおける車両挙動の再現方法は、交通流を流体近似として扱う流体モデル⁷⁾と、複数の車両から構成される車両群を一つの塊（パケット）として扱う離散モデル⁸⁾に分類できる。さらに交通流シミュレーションは、シミュレーション内の時間進行の方式により、ピリオディックスキャンニング方式とイベントスキャンニング方式に分類可能である。ピリオディックスキャンニング方式は、一定の単位時間ずつ時間を進行させる単純な方法であり、イベントスキャンニング方式は、車両が異なる道路区間に移動するといった何らかの事象が発生する毎に時間を進行させる方式である。イベントスキャンニング方式はピリオディックスキャンニング方式に比べ演算時間が短くて済む場合が多い。

筆者らが開発したマクロシミュレーションDEBNetSの特徴は、大都市圏の一般街路を含めた大規模なネットワークを計算対象とすることが可能な点であり、これまでに京都市⁹⁾や大阪市¹⁰⁾を対象とした評価計算に適用されている。大阪市の事例では、3000リンク以上のネットワークを対象としている。DEBNetSのモデル上の特徴は、イベントスキャンニング法を採用することで時間経過を連続的に表現している点である。それ故、生活行動シミュレータで計算される一人一人の生活行動における自動車トリップの出発時刻をデータとして読み込むことで、一人一人の到着時刻を連続時間軸上で計算することが可能となる。

このDEBNetSと生活行動シミュレータの統合システムを用いた交通需要予測の適用事例の結果より、DEBNetSに関する以下の課題点が明らかとなった。

- ① 渋滞の延伸現象を再現する必要がある。
- ② 計算時間を短縮する必要がある。

*1 キーワード：ネットワーク交通流，配分交通，シミュレーション

*2 正員，工博，京都大学工学研究科土木システム工学専攻

*3 非会員，オムロン株式会社交通ソリューション事業部

*4 正員，工博，東京工業大学理工学研究所土木工学専攻

*5 正員，PhD，京都大学工学研究科土木システム工学専攻
(京都市左京区吉田本町，TEL075-753-5136, FAX075-753-5916)

前者に関して、DEBNetSはリンク上に存在可能な車両台数についての制約条件を考慮しておらず、渋滞が上流に延伸する現象を再現することができない。ネットワーク上の交通流を忠実に再現するにあたり、渋滞延伸現象の再現はDEBNetSの大きな課題である。また後者の計算時間に関して、京阪神都市圏を計算対象とした場合、24時間のシミュレーションの実行にPC^[注1]で20時間弱を要している。アルゴリズムの改良による計算コストの削減も課題として挙げられる。

本研究では、上述の課題点への対応を図るとともに、交通流シミュレーションモデルとしての検証(verificationおよびvalidation)を行う。

2. DEBNetSの改良

(1) 改良前のDEBNetSの概要

DEBNetS はもともと、非集計モデルによる行動モデルと組み合わせて、各個人のOD所要時間を動的な交通流を考慮した上で出力する計算ツールを念頭に開発されたネットワークシミュレータで、種々の施策による交通状況の変化を再現することが可能である。

DEBNetS は、道路網を構成するリンクを複数の道路セグメントに分割した上で、各車両あるいは車両群(以下、これをパケットと呼称する)の道路網上の挙動を、道路網に関する情報と自動車交通需要に関するデータに基づいて、イベントスキニング法により、シミュレーション内で時間を更新させつつ推定することにより、道路網上の交通状態の変化を再現している。

ここでイベントとして定義しているのは、以下の3つである。

A. 出発イベント

車両が走行を開始するイベント。このイベントが生起した車両は、出発地から目的地までの最短経路を検出し、走行を開始するリンクを決定する。

B. セグメント間移動イベント

当該車両が存在している道路セグメントの下流側道路セグメントへ移動するイベント。もし走行中の道路セグメントが、リンクの最下流セグメントである場合は、最短経路を探索し、最短経路上の下流側リンク最上流セグメントへ移動する。

C. 到着イベント

車両が目的地へ到着し、走行を終了するイベント。

DEBNetS におけるこれらのイベントの処理の手順を図-1に示す。ここで図中の「算定可能なイベント」とは、各車両が次に起こすイベントのことを意味し、各車両の状況に応じて、常に唯一決定される。そ

して、算定された各イベントの生起時刻に基づいて、最も早い時刻の未生起イベントを生起させる。すなわち、シミュレーション時刻をそのイベント生起時刻に更新すると共に、各道路セグメント内の車両台数を更新する。そして、そのイベントに関連する車両が次に起こすイベントの生起時刻を算定する。以上を、全てのイベントが終了するまで繰り返す。

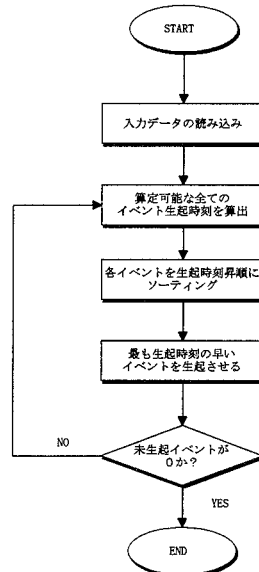


図-1 改良前 DEBNetS の計算手順

なお、各パケットの道路セグメント内での挙動は、道路セグメント内の交通状態が一様であると見なした上で、KV 曲線^[注2]を用いてその所要時間を決定することにより再現している。

シミュレータが大規模なネットワークを対象とする場合には、外生的にOD別に走行経路を与えることが困難である。そのためシミュレータ内で経路選択を再現することが必要となる。DEBNetSでは、簡便に経路選択行動を再現するために、各運転者はリンクから流出するたびに、経路選択を行い、その時点での目的地までの最短経路に流入するものとした。また、各ドライバーが所有するCognitive Map(各リンクの知覚旅行時間の情報が記載された道路網地図)は、全てのドライバーについて共通であるとして計算コストの軽減を図った。ただし、Cognitive Mapについては、一定時間間隔毎にその時点での交通状態から算定することとした。また最短経路探索に用いる所要時間には、各リンクの所要時間と高速道路の料金抵抗(時間価値換算)を考慮した一般化費用を用い、平面道路と高速道路の混在ネットワークへの適用を可能としている。

(2) 渋滞の延伸現象の再現

従来の DEBNetS では、当該車両が走行しているリンクの最下流セグメントに到着したときに、KV 曲線からそのセグメントを流出する時刻を算定し、その時刻になると即時に流出、すなわちリンク下流端に到着した車両はすぐに次のリンクへ流入していくことになる。そのため、信号交差点における信号制御の影響を考慮していないばかりか、下流側リンクの道路状況に関係なく下流側リンクへ流入してしまう、すなわちリンク容量が無限大となり、交通流の再現性に大きく影響を及ぼすことになる。

そこで、各リンク内の最下流セグメントの更に下流側に長さ 0 の「流出セグメント」を新たに付加し、このセグメント内で信号制御および渋滞列形成を処理することにする。具体的には、以下に挙げる3つの条件判定を、最下流セグメントより流出セグメントへ流入してきた車両に対して行う。（図-2参照）

① 渋滞判定

まず渋滞が発生していないかどうか、すなわち流出セグメントに1台も車両が存在しないかどうかの判定。この条件が満たされない場合（すなわちすでに渋滞が発生している場合）は、渋滞列最後尾に加わる。

② 受入判定

当該車両の経路上にある下流側リンクが受入可能であるかどうか、すなわち先詰まりがないかどうかの判定。この条件が満たされない場合（すなわち先詰まりが発生している場合）は、新たに渋滞列を形成する^[注3]。

③ 流出判定

当該リンクの流出容量を超過していないかどうかの判定。ここでは予め当該リンクの断面容量とリンク下流端ノードにおける青時間比等から1信号周期内に流出できる台数を決定しておき、それを超過していないかどうかを判定する。この条件が満たされない場合、新たに渋滞列を形成する^[注3]。

ここで、新たに以下のイベントを定義する。

D. 交差点流入イベント

リンクの最下流セグメントから、流出セグメントへ移動するイベント。ここでは渋滞判定、受入判定および流出判定を行う。

E. 渋滞イベント

判定条件が満たされないため、下流側リンクへ移動することができず、渋滞列を形成する、あるいは渋滞列に加わるイベント。またはすでに渋滞列中にある車両が、下流側リンクへ移動できるかど

うかの判定を行うイベント。このイベントを生起するのは、流出セグメントに存在する車両のみであり、受入判定および流出判定を行う。

F. 流出イベント

流出セグメントから経路上の下流側リンクへ移動するイベント。

さらに、2.(1)のセグメント間移動イベントは、次のように定義しなおす。

B'. セグメント間移動イベント

当該車両が存在している道路セグメントの下流側道路セグメントへ移動するイベント。

また信号制御に関しては、上述の流出判定における影響のほか、流出イベントを行う全ての車両に平均待ち時間を付加することで表現する。

なお、この方法で形成される渋滞列は Horizontal Queue ではなく、Vertical Queue であり、流出セグメント以外の道路セグメントにおける KV 曲線は自由流、すなわち渋滞を考慮しないものを用いており、自由流領域での密度による速度変化を考慮している。

交差点付近における処理の手順を図-2に示す。図-2中の、渋滞列に加わった車両が当該リンク最下流端から流出する予定時刻を算定する式、すなわち、流出予定時刻算出式は、その車両の状態によって以下のように異なる。

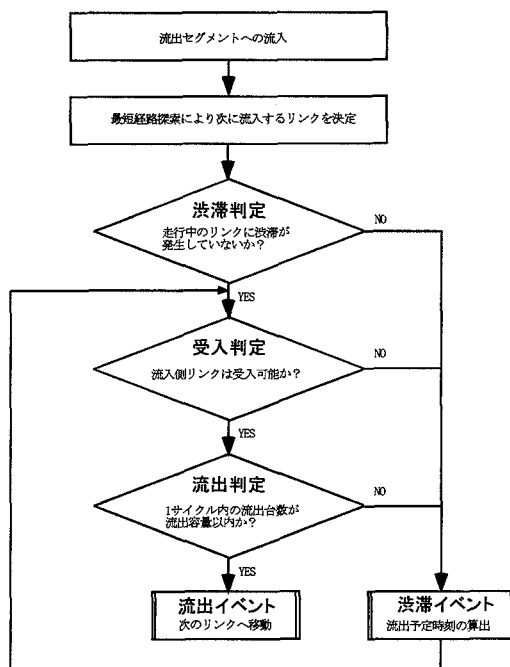


図-2 交差点付近における処理の手順

① 渋滞判定により渋滞列を形成する場合：

$$\text{流出予定時刻} = \text{現在時刻} + \text{渋滞長} / \text{流出容量}$$

すでに渋滞列が発生しているため、その渋滞列が捌けるであろう最早時刻を設定する。

② 受入判定により渋滞列を形成する場合：

$$\text{流出予定時刻} = \text{現在時刻} + \text{一定時間}$$

下流側リンクが受入可能となる時刻は予測困難であるため、一定時間経過後の時刻を設定する。

③ 流出判定により渋滞列を形成する場合：

$$\begin{aligned} \text{流出予定時刻} = & \text{次の信号サイクル開始時刻} \\ & + \text{渋滞長} / \text{流出容量} \end{aligned}$$

当該リンクの流出容量超過が流出不可能である原因であるが、この流出容量は断面容量と青時間比で決定されるため、次の信号周期後に渋滞列が捌けるであろう最早時刻を設定する。

以上で決定される流出予定時刻は、実際に車両が当該リンクを流出する時刻ではなく、流出の機会を与える時刻、すなわち再度、条件判定を行う時刻である。このような意味では、実際の流出時刻よりも早ければアルゴリズム上の問題は生じないが、計算の効率化を図るために、最も早く流出できるであろう時刻を設定している。

また、②で用いている一定時間に関しては、今後、下流側の交通状態を加味してこれを予測する等の改良が必要であろう。

(3) 計算時間の短縮のためのアルゴリズム改善

シミュレータ内の時間の取り扱い方法には、時間を離散化して各々の時間帯ごとの交通状態を再現するピリオディックスキャニング法と、道路網上で生じる各車両のリンク流出や流入等をイベントと捉え、そのイベントが生じる度にシミュレータ内の交通状態とシミュレーション時間を更新していくイベントスキャニング法とがある。一般に、ピリオディックスキャニング法でミクロシミュレーションを実行する場合、個々の車両の位置、速度等を各時間帯で算定する方法が取られる。したがって、時間帯数と道路上の車両台数の積に相当する回数だけ、位置や速度等を計算することが必要となり、KV曲線を用いた場合でさえ、計算コストが大きなものとなる。一方イベントスキャニング法を用いた場合には、計算が必要とされる回数はイベントの回数に等しい。DEBNetsのように道路をセグメントに分割する場合、各車両が起こすイベントの回数は、その車両の走行経路に含まれるセグメント数に1を加えたもの等しく、したがって、ピリオディック

スキャニング法よりも計算コストは小さなものとなる。

しかしながら、このイベントスキャニング法にも計算効率に影響を及ぼしかねない問題点がある。車両データは常にイベント生起時刻の昇順にソートされている必要があるという点である。すなわち何らかのイベント生起後に次のイベント生起時刻を算出し、それをもとに全車両を生起時刻順にソーティングしなくてはならない。全体の車両数が少ない場合はあまり問題とならないが、車両数が多い場合、このソーティングの実行時間は不安定で、シミュレーション全体に大きな影響を及ぼすと思われる。

そこで、イベントスキャニング方式にスキャニング・インターバル（離散化した時間帯）を導入し、このスキャニング・インターバルごとに車両データを管理することにする。そしてこの管理された車両のイベントを順次生起させ、生起後にはその車両の次のイベント生起時刻を算定し、スキャニング・インターバルとの管理関係を更新する。

例えば、図-3で示すように、シミュレーション時刻が、

- ・ S1～S2 の間にイベントが発生する車両は 3 台 (T1,T2,T3)
- ・ S2～S3 の間には 1 台 (T4)
- ・ S3～S4 の間には 2 台 (T5,T6)
- ・ S4～(S5)の間にイベントが発生する車両は 2 台 (T7,T8)

となっているときに、T1, T2, T3 の車両のイベントを順次生起させ、各車両の次のイベント生起時刻を算定し、その生起時刻を含むスキャニング・インターバルを求める。たとえば T1 の車両の次のイベント生起時刻が、S3～S4 に含まれるとすると、その車両データは、S1 のカテゴリーから解放され、S3 のカテゴリーに管理、すなわち T6 車両の次に位置づけられる。

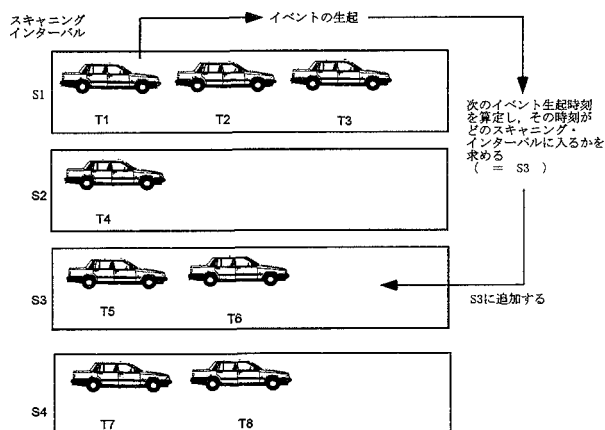


図-3 改良後の車両データの管理方法

このようにして、ソーティングの時間負荷を軽減することで、通常のイベントスキニング方式、および、通常のピリオディックスキニング方式よりも、計算効率の良い処理が可能となる。

(4) 改良効果の確認

a) 渋滞の延伸現象の再現

交通流シミュレーションモデルの開発において、モデルが備えている交通状況の再現能力を評価するために必要とされる、標準的な検証手順について一般に公開されている「交通流シミュレーションモデルの標準検証プロセス¹¹⁾」に基づいて、渋滞の延伸が適切に表現できているか否かを確認した。図-4のような、容量2,200[台/時]の5本のリンクが直列につながった単路状のネットワークにおいて、最下流のリンクがボトルネックになるよう設定し、図中に示されるような時間ごとに変化する需要を与えたときの各リンクの流入交通量を観測した。ボトルネック容量は800[台/時]に設定している。表-1にネットワークの設定を示す。

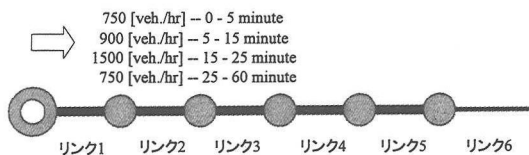


図-4 渋滞延伸現象の再現確認のためのネットワーク

表-1 ネットワークの設定

	リンク長	リンク容量
リンク1～5	500m	2,200[台/時]
リンク6		800[台/時]

図-5に各リンクの流入累積交通量を、図-6に各時間帯（0-5,5-15,15-25,25-60min）に発生した先頭車両の位置を示す。これらの図では、ボトルネックのリンク容量を超えた車両群がボトルネックへ到着すると、下流側リンクへ流入することができず渋滞が発生し、渋滞が上流側リンクに延伸していく挙動が再現されている。

b) 計算時間の短縮

従来の DEBNetS の事例研究として報告されている大阪市への適用事例¹⁰⁾（リンク数：2994、ノード数：1050、セントロイド数：292、車両データ：990,575 トリップ）では計算時間が17時間31分^{註4)}であったが、これと同一の条件で、上記の改良を施した後の計算時間は2時間25分^{註5)}となり、約7分の1に

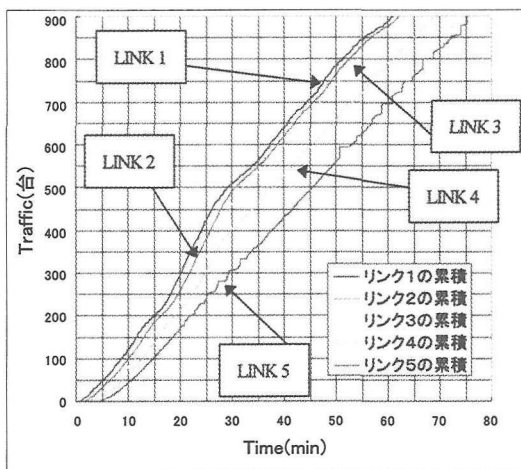


図-5 各リンクの流入累積交通量

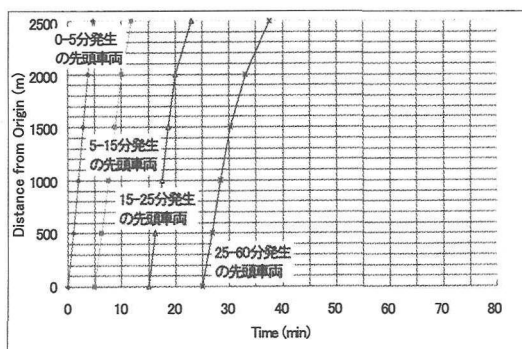


図-6 先頭車両の軌跡

短縮し改善効果が確認できた。

3. 実ネットワークを用いた検証

上述の交通流シミュレーションモデルの標準検証プロセスにおいて、実データを用いたシミュレーションモデルの検証（Validation）のために街路ネットワーク用として公開されている吉祥寺BMデータ¹¹⁾を用いて、DEBNetSを実行し、各リンクの交通量を実データと比較した。その結果、実データの観測交通量とDEBNetSによる計算交通量の相関係数が0.95となり、概ね現況を再現できていることが確認された（図-7）。さらに10分間交通量に基づいた時間帯別の相関係数について表-2にまとめた。リンク交通量がない状態から計算を開始したために7時50分台を集計に含んだ場合相関は低いが、それ以降では約0.83と良好であった（8:00-10:00の集計結果を図-8に示す）。

交通量の再現結果は良好であったが、各リンクの走行速度の相関係数（8:00-10:00）は約0.32と高い値ではなかった。（図-9）。今後のDEBNetSの改良にあたり、走行速度の再現性向上の検討は大きな課題

となるであろう。特に、非渋滞時におけるリンク内の走行挙動としてKV曲線のパラメータ設定の問題、そして渋滞時におけるアルゴリズムの改善等に取り組む必要がある。

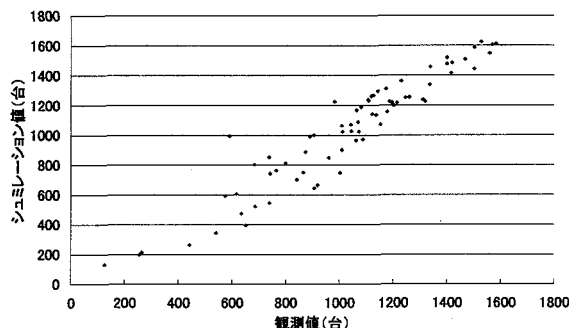


図-7 実ネットワークを用いたリンク交通量の検証

表-2 時間帯別交通量の相関係数

集計時間帯	相関係数
8:00-8:30 (30分間)	0.859
8:30-9:00 (30分間)	0.842
9:00-9:30 (30分間)	0.822
9:30-10:00 (30分間)	0.798
7:50-10:00	0.612
8:00-10:00	0.830

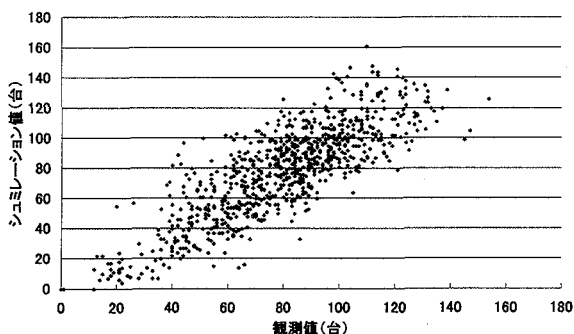


図-8 時間帯別交通量の再現結果 (8:00-10:00)

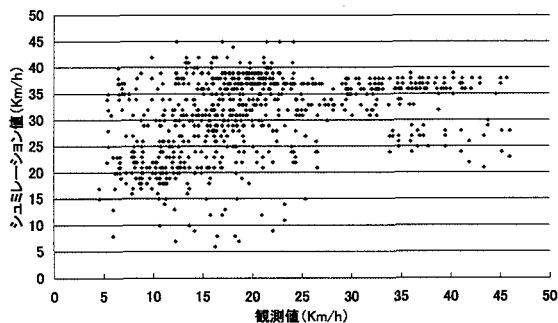


図-9 時間帯別走行速度の再現結果 (8:00-10:00)

4. おわりに

本研究では、交通流シミュレータDEBNetSの課題を整理し、その対処を図った。具体的には、1)渋滞の延伸現象を再現するために流出セグメントを各リンクの最下流端に導入し、このセグメントにて流出条件の判定および渋滞の取り扱いを行った、2)計算時間短縮のためにアルゴリズムを改良した。そしてそれらの効果を確認するために仮想ネットワークおよび実ネットワークにおける検証を行ったところ、リンク走行速度の相関係数は高い値ではなかったが、リンク交通量の再現性は良好であった。また改良を施した、渋滞の延伸現象および計算時間の短縮の効果を確認することができた。今後の課題としては、走行速度の再現性を向上させるほか、経路選択モデルの精緻化が挙げられる。

注

- [1] PentiumII 333MHzのマシン、Linux上で実行。最短経路探索の時間間隔は15分。
- [2] DEBNetSにおけるKV曲線は、例えばDrew型¹²⁾もしくはUnderwood型¹³⁾を採用した上で、そのパラメータを道路種別ごとに回帰分析等により算出する。
- [3] 図-2に示したように、受入判定は渋滞判定にて当該リンクに渋滞列が存在していない場合においてのみ駆動される判定であり、それ故、受入判定が満たされない場合には、その車両は渋滞列を形成し、渋滞列先頭車両となることとなる。同様に、流出判定も、渋滞判定、受入判定の双方にて当該リンクに渋滞列が存在しない場合においてのみ駆動される判定であるため、それが満たされない場合には、渋滞列を新たに形成することとなる。
- [4] PentiumIII 800MHz/2CPUのマシン、SunOS5.8上で実行。最短経路探索の時間間隔は15分に設定。
- [5] 実行環境は注4と同じ。最短経路探索の時間間隔は5分に設定。

参考文献

- 1)例えば、藤井 聡、大塚祐一郎、北村隆一、門間俊幸：時間的空間的制約を考慮した生活行動軌跡を再現するための行動シミュレーションの構築、土木計画学研究・論文集 No.14, pp.643-652, 1997.
- 2)北村隆一：交通需要予測の課題一次世代手法の構築にむけて、土木学会論文集, No. 530/IV-30, pp.17-30, 1996.
- 3)藤井聡：交通計画におけるシミュレーション手法の適用可能性、土木計画学研究・論文集, No.16, pp. 19-34, 1999.

- 4)奥嶋政嗣, 藤井聡, 菊池輝, 北村隆一: 個人のOD 所要時間動的算定のための交通流シミュレータの開発, 土木学会論文集 (投稿中)。
- 5)例えば, 斉藤威: 各種交通条件が信号交差点の交通容量に及ぼす影響, 科学警察研究所報告交通編, Vol. 21, No.1, pp. 1-12, 1980.
- 6)例えば, 森津秀夫, 大原竜也, 多田典史, 井上琢弥: 経路誘導による交通ネットワークフローの変化に関する分析, 土木計画学研究・論文集, No. 9, pp.37-44, 1991.
- 7)例えば, 桑原雅夫, 吉井稔雄, 堀口良太: ブロック密度法を用いた交通流の再現方法について, 交通工学, Vol. 32, No. 4, pp. 39-43, 1997.
- 8)例えば, 吉井稔雄, 桑原雅夫, 森田紳之: 都市内高速道路における過飽和ネットワークシミュレーションモデルの開発, 交通工学, No. 30, Vol. 1, pp.33-42, 1995.
- 9)藤井聡, 菊池輝, 北村隆一: マイクロシミュレーションによる CO₂ 排出量削減に向けた交通施策の検討: 京都市の事例, 交通工学, Vol. 35, No. 4, pp. 11-18, 2000.
- 10)菊池輝, 藤井聡, 白水靖郎, 北村隆一: 大規模ネットワークにおける交通流シミュレータ DEBNetS の現況再現について, 第 20 回交通工学研究発表会論文報告集, pp. 49-52, 2000.
- 11)道路交通シミュレーションシステムクリアリングハウス URL:<http://trans1.ce.it-chiba.ac.jp/ClearingHouse/>
- 12)D.R. Drew : Deterministic Aspects of Freeway Operations and Control , Texas Transportation Institute, Research Report 24-4, 1965.
- 13)R.T. Underwood : Speed, Volume and Density Relationships, Quality and Theory of Traffic Flow (New Haven : Yale Bureau of Highway Traffic), pp.141-188, 1961.

大規模道路ネットワークシミュレータ DEBNetS の改良と検証

菊池輝, 加藤義昭, 馬浜透, 藤井聡, 北村隆一

本研究では, 交通流シミュレータ DEBNetS における, 1)渋滞の延伸現象が再現されていない, 2)計算時間を短縮する必要がある, という課題点を克服することを目的とし, DEBNetS に改良を施した。前者の課題に関しては, リンクの表現方法を工夫し, さらにイベントの処理方法を変更することで, 下流リンクからの渋滞の延伸を再現した。また後者に関してはアルゴリズムを変更した。仮想ネットワークおよび実ネットワークにおける検証の結果, これらの課題は克服され, リンク交通量の再現性も高かったが, 走行速度の再現については改良の余地が認められた。

Improvement and Verification of Dynamic Traffic Flow Simulator “DEBNetS”

Akira KIKUCHI, Yoshiaki KATO, Toru MABUCHI, Satoshi FUJII and Ryuichi KITAMURA

In this study, improvements are made to the dynamic traffic flow simulator, “DEBNetS.” First, to reproduce the phenomenon of congestion propagation to upstream, a pseudo segment is added to the series of segments by which each link is represented in the simulation. The method of processing traffic event is also modified. Next, to reduce computational time, the algorithm for sorting events is improved. Verification and validation analyses have indicated substantially reduced computational time and a high correlation between observed and simulated traffic volumes. The results however have indicated that the reproducibility of average traffic speed is yet to be improved.
