

# 大規模ネットワークにおける並列計算機を用いた動的利用者均衡配分の高速計算

福田 和輝<sup>1</sup>・井料 隆雅<sup>2</sup>・Petprakob Wasuwat<sup>3</sup>・Wijerathne Lalith<sup>4</sup>・  
浦田 淳司<sup>5</sup>・Jr. Peque Genaro<sup>6</sup>

<sup>1</sup>学生会員 神戸大学工学研究科市民工学専攻土木工学科 (〒 657-8501 兵庫県神戸市灘区六甲台町 1-1)  
E-mail: 163t130t@stu.kobe-u.ac.jp

<sup>2</sup>正会員 神戸大学大学院教授 工学研究科市民工学専攻 (〒 657-8501 兵庫県神戸市灘区六甲台町 1-1)  
E-mail: iryo@kobe-u.ac.jp

<sup>3</sup>非会員 東京大学地震研究所 (〒 113-0032 東京都文京区弥生 1-1-1)  
E-mail: wasuwatp18@gmail.com

<sup>4</sup>正会員 東京大学地震研究所 (〒 113-0032 東京都文京区弥生 1-1-1)  
E-mail: lalith@eri.u-tokyo.ac.jp

<sup>5</sup>正会員 神戸大学工学研究科市民工学専攻 (〒 657-8501 兵庫県神戸市灘区六甲台町 1-1)  
E-mail: urata@person.kobe-u.ac.jp

<sup>6</sup>非会員 神戸大学大学院工学研究科 (〒 657-8501 兵庫県神戸市灘区六甲台町 1-1)  
E-mail: gpequejr@panda.kobe-u.ac.jp

動的利用者均衡配分問題の解法にマルコフ連鎖による解法がある。この解法は、交通量配分を何度も繰り返すことで解を得るものである。しかし、その繰り返し計算のため、大規模なネットワークを対象とする場合は計算コストが膨大になってしまう。この計算にあたり、多数の CPU を並列化させて計算できる高性能のコンピュータの活用が不可欠である。本研究では、効率的な並列計算アルゴリズムを提案し、大規模ネットワークにおける計算を行う。交通流の計算には Variational Theory を取り入れる。リンク内での車両の位置を逐次計算する必要がなく計算時間を削減できる。既発表の方法<sup>1)</sup>では、合流比が交通工学的な妥当さをもって、与えられないという問題があり、解決策として妥当な合流比を与えることができる計算アルゴリズムを構築する。また、大規模なネットワークでの並列計算にあたり、最短経路探索のコスト削減、HPC(High Performance Computer)に合わせたメモリの保持方法の導入、CPU 間の通信の効率化を行う。

**Key Words:** *Dynamic Traffic Assignment, Variational theory, Parallel Computing*

## 1. はじめに

道路交通ネットワークの評価のために行われる交通量配分手法の一つに、動的利用者均衡配分 (DUE) がある。DUE は、時間とともに変化する交通状況を考慮するため、より正確な道路交通の評価の実施が期待できる。しかし、DUE の様々な解法の中<sup>2)</sup>で、限定的な状況下で解の算出を保証する手法<sup>3),4)</sup>はあるものの、多くの手法においては解の収束性が担保されておらず、また、DUE の均衡解の唯一性は保証されていない<sup>5)</sup>ため、理論上の重要な課題があるといえる。

解法の一つに、離散マルコフ連鎖を用いた解法がある。ドライバーが日々、前日の交通状況を考慮して最短経路を選択しなおすという調整過程 (Day-to-day ダイナミクス) を離散マルコフ連鎖とみなして計算することで、定常分布を求め、それを均衡解の代わりに解とするものである。<sup>6),7),8)</sup> 具体には、1日に数%の車両をランダムに抽出し、それらの車両を前日の交通状況を基にして最短経路に再配分する。これを繰り返して

うことで、交通状況の定常分布を求める。既存の研究<sup>7)</sup>では、比較的複雑な Sioux Falls Network<sup>9)</sup>に対して数値計算を実施し、動的交通量配分問題の定常分布を算出し、その収束性を確認している。

一般に、動的交通量配分では、待ち行列が交差点を越えて延伸する現象を考慮することにより、混雑をより正確に評価できる。現実の交通において、あるボトルネック地点で生じた渋滞が交差点を越えて延伸することで、ボトルネック地点を通行しない車両も影響を被ることがしばしば見受けられる。福田ら<sup>10)</sup>では、この渋滞の延伸による影響を考慮した交通流モデルを構築し、計算している。また、テストネットワークとして Sioux Falls Network<sup>9)</sup>を用いた試算では、渋滞の延伸の評価の有無によって各リンクの遅れ時間に大きな差が生じることを明らかにしている。しかし、ここでは、リンクに最大滞留可能台数を与えることで渋滞の延伸を表現するような交通流モデルを用いており、交通流理論的には正確な評価をしているとはいえない。より正確な道路交通の評価をするためには、正しい Q-K 特

性を持つショックウェーブを考慮してリンク旅行時間を算出できる交通流モデルが必要であるといえる。

本研究では、交通流理論的に妥当な評価が行えるように交通流モデルの拡張を行う。渋滞列の延伸を評価する方法として、車両をマイクロに追跡することで渋滞列の長さを捉える方法と Variational Theory(VT)を用いてショックウェーブを捉える方法<sup>11)</sup>が考えられる。前者では、車両 1 台 1 台の詳細な挙動（車線変更、追従等）を把握できるという利点があるが、一方でリンク内における車両の位置を逐次計算する必要があるため、計算コストが膨大になりうる。DUE の定常状態の算出には、反復計算が必須であるため、1 回の計算コストはできる限り小さくする必要がある。対して、後者は道路上の 2 地点における累積交通量から前進波（Forward Wave）と後進波（Backward Wave）を計算することで、その 2 地点間のショックウェーブの軌跡を計算する方法である。個別の車両の軌跡を計算することなく、渋滞の延伸の評価ができるため、計算コストを小さくすることができるという利点がある。本研究では、計算コストを削減するため、VT を用いて交通流を記述する。また、交通流の合流及び分流の記述は、Daganzo (1995)<sup>12)</sup>の合分流のモデル化を参考にする。この論文では、CTM (Cell-Transmission-Model) における合分流のモデル化を紹介しているが、本研究で扱う交通流モデルにも同様の考え方を導入し、合分流を記述する。

本研究では、配分計算の中でも、最短経路探索と交通流の計算の 2 か所において計算の並列化を行い、計算時間を削減する。福田ら<sup>10)</sup>では、小規模なネットワークに対する計算において交通状況の収束性を確認したものの、ネットワークを拡張すると計算時間が膨大になるという課題を挙げている。計算対象のネットワークが大きくなるほど、各車両の最短経路探索にかかる計算コストが大きくなり、また、交通流の計算も膨大になる。本研究では、多数の CPU を並列させて高性能を得る HPC(High Performance Computer) への実装に向け、効率的な並列計算アルゴリズムを構築し、大規模なネットワークに対しても現実的な時間で計算できる交通流シミュレータを開発する。また、関東圏の大規模な道路交通ネットワークを対象に数値計算を行い、既存の交通流シミュレータ (SOUND)<sup>13)</sup>で得られた結果との比較を行う。

## 2. 交通流の計算方法

### (1) 各種定式化

以下のように定式化を行う。リンクの集合を  $E$ 、ノードの集合を  $V$ 、リンク  $l \in E$  の上流端ノードを  $v_u(l)$ 、

下流端ノードを  $v_d(l)$  とする。リンク  $l$  の上流端ノード  $v_u(l)$  に接続されるリンク  $m$  の集合を  $I_{v_u(l)}$ 、下流端ノードに接続されるリンク  $n$  の集合を  $\theta_{v_d(l)}$  とする。全車両の集合を  $N$  とし、車両  $i \in N$  には、起点ノード  $o_i$ 、終点ノード  $d_i$ 、出発時刻  $s_i$  を与える。車両  $i$  の経路集合を  $R_i$  とする。経路はリンクの集合であり、車両  $i \in N$  が  $o_i$  から  $d_i$  まで連続するリンク  $l_1, \dots, l_L$  を順番通りに通過するとき、経路  $r_i \in R_i$  を  $r_i = (l_1, \dots, l_L)$  と示す。車両  $i$  について、リンク  $l \in r_i$  の流入時刻  $t_i(l)$ 、流出時刻  $u_i(l)$ 、経路  $r_i$  を利用した時の目的地到着時刻  $g_i(r_i)$ 、経路旅行時間  $T_i(r_i)$  を定める。なお、 $g_i(r_i)$  は経路の最下流リンク  $l_L \in r_i$  からの流出時刻  $u_i(l_L)$  と等しい。

時刻  $t$  における交通流の計算で用いる各諸元を以下にまとめる。

- $a_l^+(t)$ : リンク  $l$  に流入しようとする総交通量（上流側のリンクの流出交通量の総和）
- $a_{l,n}^*(t)$ :  $a_l^+(t)$  の下での流出リンク  $n$  別交通量
- $d_{l,n}(t)$ : リンク  $l$  下流端から流出する方向別の交通量（ $\sum_{m \in I_{v_u(l)}} d_{m,l} = a_l^+(t)$  である。）
- $a_l(t)$ : 交通渋滞を考慮した上で、最終的に計算されたリンク  $l$  の流入交通量

### (2) 交通流の記述方法

本研究では、車両のリンクの流入出を交通容量で制限するボトルネックモデルに VT を導入することで渋滞の延伸を評価できるモデルを構築する。また、合分流においては、Daganzo(1995)<sup>12)</sup>の考え方を取り入れ実装する。

交通流計算はリンク単位で行う。また、計算は大きく 2 つの部分で構成され、具体には、リンク  $l$  の流入側及び流出側にそれぞれ与えられた容量制約を満たすように交通流を計算する部分 (Update Flow) と合流比に基づいて流出側の容量制約を適宜更新する部分 (Update Capacity) である。交通流の計算において、車両を離散的に扱うことにより、個別の車両の経路を特定しながら計算を進めることができる。しかし、合流や分流における車両の流れを考えると、交通流を連続量で扱う方が、車両の流入出の優先権を考えやすい。そのため、本研究では、基本的な交通流の流れは離散化された車両で捉えるが、合分流においては交通流を連続量で扱い、計算する。

Update Flow の計算方法については、図-1 を用いて説明する。ここでは、リンク  $l$  の交通流を計算する。まず、ある時刻  $t$  にリンク  $l$  に流入する予定の交通量  $a_l^+(t)$  を上流側リンク  $m \in I_{v_u(l)}$  の方向別流出交通量  $d_{m,l}(t)$  から求める。本研究では、車両を離散的に扱っており、個

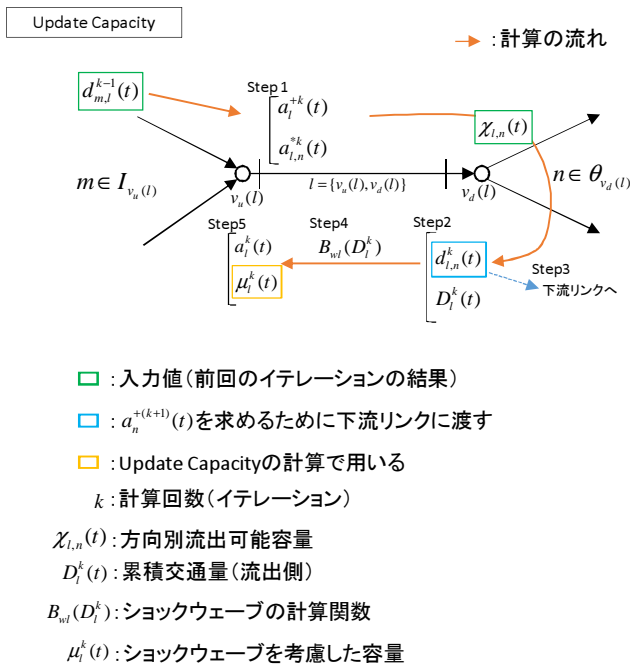


図-1 各リンクの交通流計算

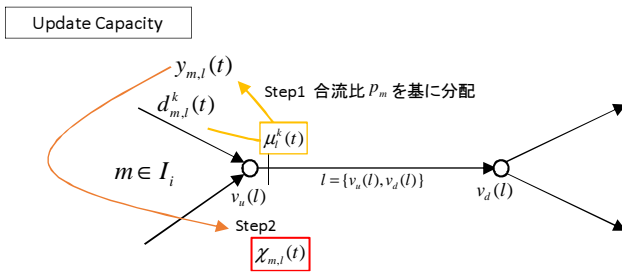
別の車両が次にどのリンク  $n$  へ流出するかを判断できるため、方向別の交通量  $a_{l,n}^*(t)$  も求めることができる (Step1)。次に、下流側の容量制約に基づき、リンク  $l$  の方向別流出交通量  $d_{l,n \in \theta_{v_d(l)}}(t)$  を算出する。リンク  $l$  の下流側にあるリンク  $n \in \theta_{v_d(l)}$  には、それぞれ方向別の流出可能交通容量  $\chi_{l,n}(t)$  が与えられており、リンク  $l$  の方向別流出交通量  $d_{l,n}(t)$  はその値に基づいて計算される。しかし、分岐がある場合は、必ずしも与えられた方向別の流出可能交通容量  $\chi_{l,n}(t)$  で車両が流出できるとは限らない。 $\chi_{l,n}(t)$  は、分岐する各リンク  $n$  で個別に計算されているもので、他の分岐リンクの混雑を考慮していないためである。例えば、リンク  $l$  の下流側に 2 本のリンク  $n_1, n_2$  が分岐して伸びており、それぞれのリンク  $n_1, n_2$  への流出可能交通容量が  $\chi_{l,n_1}(t)$  と  $\chi_{l,n_2}(t)$  で与えられている場合を考える。リンク  $n_1, n_2$  それぞれに向かう方向別交通量  $a_{l,n_1}^*(t), a_{l,n_2}^*(t)$  がどちらも方向別交通容量以下である場合は、全ての交通量が下流側のリンク  $n_1, n_2$  に流出できる。しかし、リンク  $n_2$  に向かう交通量  $a_{l,n_2}^*(t)$  が  $\chi_{l,n_2}(t)$  よりも大きいとき、リンク  $n_2$  への流入地点がボトルネックとなり、交通渋滞がリンク  $l$  で生じる。すると、リンク  $n_1$  へ向かう車両もその渋滞の影響を受けるため、リンク  $n_1$  への実質の方向別交通容量は  $\chi_{l,n_1}(t)$  よりも小さくなる。したがって、リンク  $n_1$  に向かう方向別流出交通量  $d_{l,n_1}(t)$  は、もう一方の分岐リンク  $n_2$  の混雑の影響を考慮した上で、決定される必要がある。各下流リンク  $n$  への実質の方向別流出交通量  $d_{l,n}(t)$  は以下の式 (1) により計

算できる。

$$\frac{a_{l,n_1}^*(t)}{d_{l,n_1}(t)} = \frac{a_{l,n_2}^*(t)}{d_{l,n_2}(t)} = \frac{d\tau_l(t)}{dt} + 1 \quad (1)$$

ここに、 $\tau_l(t)$  はある時刻  $t$  にリンク  $l$  に流入した車両の旅行時間である。Daganzo(1995)<sup>12)</sup> では、交通流の分流現象を考えると、FIFO (First-in-First-out) の原則に従い、同時刻  $t$  にあるリンクへ流入した全車両は同じ旅行時間でそのリンクを流出しなければならないという考え方を採用することが必要だとしている。式 (1) は、この原則を満たすように実質の方向別流出交通量  $d_{l,n}(t)$  を算出する式である。この式で計算された方向別流出交通量  $d_{l,n}(t)$  からリンク  $l$  の流出側の累積交通量  $D_l(t)$  も算出できる (Step2)。ここで得られた下流リンク  $n$  への方向別流出交通量  $d_{l,n}(t)$  は下流側リンク  $n$  の  $k+1$  回目の交通流の計算 (特に、Step1) で使用するため、伝達する (Step3)。本研究では、まず、Step1 から Step3 を繰り返し計算し、交通流を収束させる。収束後、流出側の累積交通量  $D_l(t)$  を用いて、リンク  $l$  のショックウェーブを計算する (Step4)。この計算により、リンク  $l$  に実際に流入できる交通量  $a_l(t)$  及び交通容量  $\mu_l(t)$  が求められる (Step5)。ここで得られた交通容量  $\mu_l(t)$  は、次に説明する Update Capacity で使用する。また、Step4 と Step5 については、一度だけ計算を行う。

次に、Update Capacity の計算について説明する。ここでは、各リンクに与えられている方向別の流出可能容量 (上流側の各リンク  $m \in I_{v_u(l)}$  からの流入に関する容量制約)  $\chi_{m,l}(t)$  の更新を行う。Update Flow の計算では、与えられた方向別の流出可能容量  $\chi_{m,l}(t)$  に基づいて交通流の計算がされる。しかし、交通流が更新された場合、方向別の流出可能容量  $\chi_{m,l}(t)$  も更新する必要がある。例えば、合流地点や分流地点において、この更新を行わなければ、下流リンクに混雑が生じていないのに、そのリンクに車両が流入しないというような現象も起こりうる。本研究では、Update Flow の計算を行った後、方向別の流出可能容量  $\chi_{m,l}(t)$  を Update Flow の交通流の計算結果を用いて更新する。図-2 に具体的な方法を示す。ここでは、リンク  $l$  の上流部に合流があるときを考え、リンク  $l$  への方向別交通容量  $\chi_{m,l}(t)$  の更新方法を説明する。また、合流する各リンク  $m$  の合流比は  $p_m$  で表す。リンク  $l$  への流入可能容量は、Update Flow で計算した  $\mu_l(t)$  であり、その容量を各合流リンク  $m$  で分担する。まず、合流比  $p_m$  に基づいて交通容量  $\mu_l(t)$  を実際に流入できる台数  $y_m$  に分配する (Step1)。全ての交通容量  $\mu_l(t)$  が分配し終えたとき、その時の  $y_m$  を方向別交通容量  $\chi_{m,l}(t)$  とする (Step2)。ここで、交通量が少ない合流リンクがある場合は、他の合流リン



□ : Forward waveの計算で用いる  
 $y_{m,l}(t)$  : 実際に流入できる台数  
 $p_m$  : 合流比

図-2 方向別交通容量の更新方法

クに残りの容量を割り振る。最終的に求める交通流は、Update Flow と Update Capacity を繰り返し計算することで得る。ただし、繰り返し計算の初期段階では、実際に通すべきリンク交通量と方向別の交通容量  $\chi_{m,l}(t)$  の間に差が生じるが、計算を繰り返すことで、最終的にその差は収束すると考えられる。

### 3. 配分計算の流れ

Day-to-day ダイナミクスを離散マルコフ連鎖で記述し、繰り返し計算を行うことで定常性のある交通状況を算出する。本研究では、石原らの研究<sup>7),8)</sup>と同様に、「1日に複数台の車両が前日の交通状況を基にして同時に最短経路を選択しなおす」という Day-to-day ダイナミクスを定式化し、計算する。計算の具体的な手順を以下に示す。

**Step1 : 初期化**

初期の交通状況を仮定し、 $n = 0$  日目とする。

**Step2 : 最短経路探索**

全車両の数%の車両をランダムに抽出する。抽出した個別の車両  $i$  に対して、 $n$  日目の交通状況を基に最短経路探索を実施する。得られた最短経路  $w_i^n$  に車両  $i$  を再配分する。

**Step3 : 交通流の計算**

Step2 の配分結果を基に、 $n + 1$  日目の交通流を計算する。

**Step4 : 終了判定**

定常分布に到達すれば計算を終了する。到達していなければ、 $n \leftarrow n + 1$  として Step2 に戻る。

### 4. 並列計算アルゴリズムの概要

繰り返しの配分計算にかかる時間を削減するためには、各日の計算時間を削減しなければならない。実現に

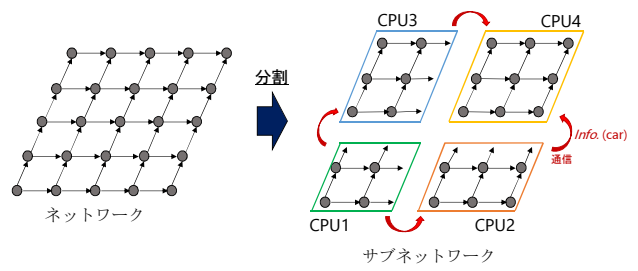


図-3 ネットワークの分割イメージ

は、複数の CPU を組み合わせることで計算を並列して実行できる HPC の利用が必須となる。本章では、HPC への実装に向けて構築した効率的な並列計算アルゴリズムについて説明する。具体的には、配分計算の中でも、最短経路探索と交通流の計算の 2 つの部分について並列化を行い、計算時間の削減を試みる。

#### (1) 最短経路探索の並列化

最短経路探索は Dijkstra 法により行うが、再配分の対象となる車両が多くなるほど、またネットワークが大規模になるほど計算コストは増加する。大規模なネットワークに計算を適用するためには最短経路探索の並列化が欠かせない。最短経路探索の並列化の方法としては、以下の 2 通りが考えられる。計算対象ネットワークを図-3 のように分割し、1 台の車両に対して行う Dijkstra 法を分割したサブネットワークごとに実行する方法（ネットワークベース）と各 CPU に前日の交通状況の記録を保持させ、複数の車両の最短経路探索を各 CPU で分担して行う方法（車両ベース）が考えられる。前者において、各 CPU は、割り当てられたサブネットワークの交通状況の情報のみを保持すればよいためメモリを削減できるという利点があるが、Dijkstra 法を終えるためには CPU 間での通信が複数回必要となり、結果として通信によるオーバーヘッドが大きいという欠点がある。また、並列する CPU 数を増加させるほど、このオーバーヘッドは増大する。対して後者では、各 CPU が保持するデータ数は多くなるが、ネットワーク全体の交通状況のデータを一度通信を行い共有すれば、その後は通信の必要がなく計算できる。そのため、並列する CPU 数の増加で得られる効果（スケラビリティ）が前者に比べ大きい。したがって、本研究では車両ベースの並列化を行う。具体的には、抽出された  $I$  台の車両を  $K$  個の CPU で並列的に計算する場合、各 CPU に  $I/K$  台ずつ車両を割り当てる。そして、各 CPU は、割り当てられた個別の車両に対して最短経路探索を行う。

## (2) 交通流計算の並列化

図-3 に示すようにネットワークを複数のサブネットワークに分割し、各 CPU がそれぞれ分担してサブネットワークの交通流を計算する。前述したように、交通流の計算はリンク単位で行うため、このようにネットワークを分割して並列計算することが可能となる。しかし、車両によっては出発地から目的地にかけて複数のサブネットワークを経由して走行する必要がある。そのため、そのような車両の情報は CPU 間で通信し合う必要がある。

ネットワークの分割方法は交通流の計算時間に大きく影響する。車両の情報は、通信を行う 2 つの CPU がそれぞれのサブネットワークの交通流の計算を終えた後にまとめて通信し合う。そのため、例えば、CPU1 の計算が早く終了した場合であっても、通信し合うもう一方の CPU2 の計算が終了していなければ通信を実行できない。この間、CPU1 には無駄な待ち時間が生じてしまい、効率良く並列計算ができていないといえない。この待ち時間を小さくするためには、サブネットワークごとの計算時間を同程度にすることが求められる。本研究では、初めに各リンクを走行する車両の台数を見積もり、サブネットワークごとの計算時間が同程度になるようにネットワークを分割することで、CPU の無駄な待ち時間を削減する。

## (3) 並列計算の流れ

本研究で構築した並列計算アルゴリズムを図-4 に示す。まず、計算の対象となる入力値（ネットワークデータ、OD 交通量データ等）を読み込む。その後、ネットワークを分割し、並列する CPU の数だけサブネットワークを生成する。生成されたサブネットワークは各 CPU に割り当てられる。以上で初期設定が終了し、これ以降配分計算を繰り返す。

最短経路に車両の再配分を行うため、まず、数%の車両をランダムに抽出する。その後、最短経路探索を並列化するため、各 CPU に同じ割合で車両を分配する。各 CPU は、記録された前日の交通状況を利用して最短経路探索を実行する。更新された最短経路の情報は、当日の交通流の計算で用いるため、全ての CPU で共有する。

交通流の計算を行う前に、経路探索を行った車両を新しい経路に再配分する。計算量を削減するため、 $n$  日目の交通流は、各リンクに記録された  $n-1$  日目の交通状況を更新することで得る。そのため、まず、各リンクに記録された  $n-1$  日目の車両の通行情報から経路変更を行った車両の情報を取り除く（アンローディング）。その後、新しい経路の各リンクに車両の通行情報を与える（ローディング）。このとき、車両が起点から初めに流入するリンクの流入時刻は車両の出発時刻を

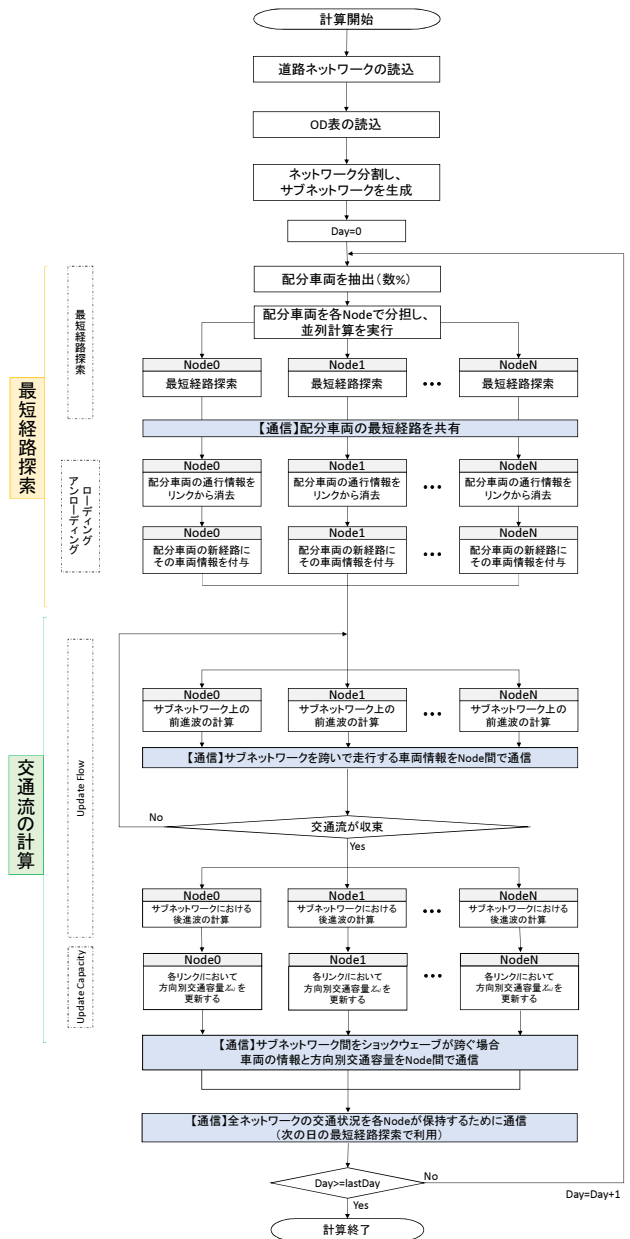


図-4 計算のフローチャート

与え、それ以外のリンクの流入出の時刻は、十分に大きな値で仮に与える。この値は交通流の計算過程で更新される。ここで示したアンローディング及びローディングの処理は、サブネットワークごとに実行できるため、各 CPU で並列して行う。

各リンクの通行情報が更新された後、前述した交通流の計算方法により、 $n$  日目の交通流を計算する。計算は VT に基づき、初めに Update Flow の計算を、続いて Update Capacity の計算を行う。Update Flow の計算において、まず、各 CPU は、サブネットワークの各リンクを走行する車両の流出時刻をそれぞれ計算する（前進波の計算）。次に、計算された車両の流出時刻を下流側のリンクに伝達する。下流側のリンクが異なる

サブネットワークに属する場合は、CPU 間で通信を行い、車両情報を伝達する。この前進波の計算と情報の伝達は、リンク間の車両の流入時刻と流出時刻の間に整合性がとれるまで繰り返し行う。整合性が確認できた後、各リンクにおいてショックウェーブの計算（後進波の計算）を行う。ここでは、車両  $i$  がリンクに流入しようとする時刻（前進波の計算で得られた車両の流入時刻）に、ショックウェーブがそのリンクの流入部まで到達しているかを数値計算により判断する。到達している場合は、その時刻に流入できないため、車両  $i$  の流入時刻を更新する。

次に、各 CPU は、サブネットワークの各リンク  $l$  に対して Update Capacity を実行し、方向別交通容量  $\chi_{m,l}(t)$  を更新する。その後、後進波の計算で流入時刻が変更された車両情報と更新した方向別交通容量  $\chi_{m,l}(t)$  を上流側のリンクに伝達する。このとき、上流側のリンクが異なるサブネットワークに属する場合は、CPU 間で通信を行い、車両情報を伝達する。このようにして得られた  $n$  日目の交通状況は、 $n+1$  日目の最短経路探索で用いるため、全ての CPU で共有する。この配分計算を繰り返し行うことで、交通状況の定常分布を求める。

## 5. 数値計算

これまでに説明したアルゴリズムを用いて、2通りの数値計算を実施した。1つ目は、単純なネットワークに対して行ったもので、構築した交通流モデルで算出された結果が交通流理論と大きく乖離していないかを確認するものである（ベリフィケーション）。2つ目は、大規模な関東ネットワークにおいて計算を行ったものである。

ベリフィケーションは、図-5 に示すように、合分流を含む単純なネットワークで実施した。各リンクの諸元を表-1 に示す。O1 から D1、O2 から D2 へ向かう交通量をそれぞれ 3000 台ずつ、車両の発生間隔はどちらも 3.0 秒としている。O1 から 1 台目の車両が発生した 1.5 秒後に O2 から車両が初めて発生するものとする。最下流のリンク 3 の交通容量が小さいため、上流側のリンクに渋滞が延伸し、先詰まりが発生すると考えられる。このとき、上流側の各リンク 0,1,2 の車両の流出は最下流リンク 3 の影響を受けて制約される。その時の各リンクの旅行時間（先詰まり時）を各リンクの諸元から算出し、シミュレーションで計算した結果と比較する。ベリフィケーションの結果を表-2 に示す。この結果から理論値と計算結果の間に多少の違いはあるものの、大きな乖離がないことが確認できる。ショックウェーブによって流入が制約された交通量を上流側り

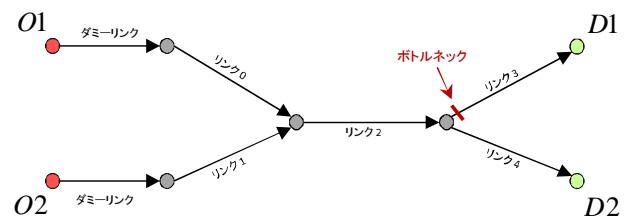


図-5 ベリフィケーション用ネットワーク

リンクに伝える際に、車両を連続量に変換しており、そのときの交通流率の時間間隔を 10 秒とやや大きめに設定しているため、誤差が生じたと考えられる。

次に、大規模な関東ネットワーク（リンク数:428,229、ノード:176,268）に対して計算を行った。関東ネットワークは平成 28 年度版 DRM(Digital Road Map) の基本道路ネットワークに基づき生成されている。発生交通量は、平成 22 年度関東圏道路交通センサスで集計された OD(起終点)交通量（トリップ数:10,760,672）を基に設定する。評価対象時間は、平日の午前 4 時台から午前 7 時台までの 4 時間とする。計算においては、車両 12 台を 1 パッケージとし、トリップ数を圧縮する。52 日分の繰り返し計算を行い、日々の最短経路探索を行う車両の割合は全車両の 0~20 %の間で変化させている。交通流の計算で必要とするリンク  $m \in I_{v_u}(t)$  の初期の方向別交通容量  $\chi(t)$  は、リンク  $l$  の交通容量  $\mu_l(t)$  を等分した値で与える。また、繰り返し計算の中で行う各リンクの方向別交通容量  $\chi(t)$  の更新は、2 章で述べた方法により行うが、その値には下限値を設けており、先詰まりを考慮した容量の 1/4、または流入先リンクそのものの容量の 1/8 のいずれかを下回る値にならないようにしている。使用したプログラミング言語は C++であり、MPI(Message Passing Interface)を用いた分散メモリ型の実装を行った。計算には、スーパーコンピュータ「京」を用い、2304 個のプロセス（288 ノード）で並列計算を行った。

計算結果を示す。まず、関東ネットワークにおけるリンク交通量の分布を図-6 に示す。幹線道路の交通量が多く、また、都心部ほど交通量が多いという一般的な傾向が再現できていることが見てとれる。また、国内で数多くの評価実績を持つ商用交通流シミュレータ SOUND<sup>13)</sup> において、同様のネットワーク・ODを用いて計算した結果（株式会社アイ・トランスポート・ラボより提供）と本シミュレータの計算結果を比較する。図-7 はリンク交通量について比較した結果である。図-7 に示すように、本シミュレータで計算されたリンク交通量の方が、SOUND のそれに比べてやや多い傾向にあるが、2 者の間に大きな差はないといえる。また、本シ

表-1 各リンクの諸元 (ベリフィケーション用)

リンク ID	自由流旅行時間 (0.1s)	交通容量 (veh/h)	自由流速度 (km/h)	渋滞流速度 (km/h)	リンク長 (m)
0	720	2,000	50	20	1,000
1	720	2,000	50	20	1,000
2	720	2,000	50	20	1,000
3	720	250	50	20	1,000
4	720	2,000	50	20	1,000

表-2 ベリフィケーション結果

リンク ID	渋滞時の旅行時間 (s)		
	計算結果 A	理論値 B	比 (A/B)
0	1,836.0	1,893.2	0.97
1	1,911.2	1,893.2	1.01
2	844.2	828.0	1.02

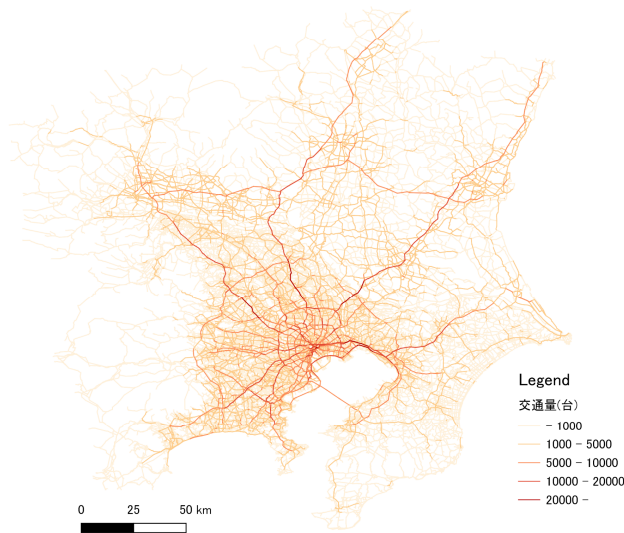


図-6 関東ネットワークの計算結果 (52 日目の交通量)

ミュレータで交通量が多くなった原因の一つとしては、大きな渋滞が発生するリンクが生じ、それを避けるために、迂回する経路をとる車両が多くなったことが考えられる。実際に、総遅れ時間は 6,297,773 時間と過剰に大きな結果が出ているため、交通流の計算方法 (特に Update Capacity) の精査が必要である。また、52 日目の計算では、382,639 パケット (4,591,668 台分) の車両がネットワーク上を走行しており、最短経路探索を含めた 52 日目の計算時間は、2 分 37 秒であった。全体の計算時間は、4 分 56 秒であった。ただし、計算中に複数回のホットスタートを実施しており、この計算

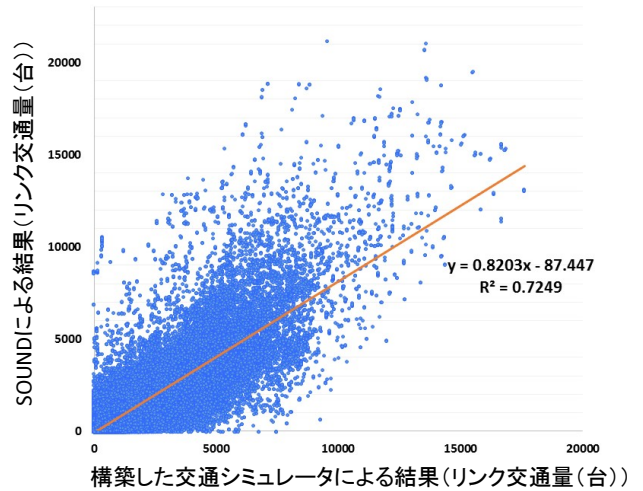


図-7 本ミュレータと SOUNDI の結果の比較 (52 日目のリンク交通量)

時間はその際の I/O 関連の時間も含んだ値である。

## 6. 結論と今後の課題

本研究では、大規模なネットワークに対しても現実的な時間で交通流の計算ができる交通シミュレータの開発を行った。交通流理論と整合した渋滞の延伸を表現できる交通流モデルを構築し、また、計算時間の削減のために、配分計算において最短経路探索及び交通流の計算で並列化を行い数値計算を実施した。数値計算の結果、単純なネットワークにおけるベリフィケーションでは、交通流の計算の収束及び交通流の計算結果が交通流理論と大きく乖離していないことが確認できた。また、大規模な関東ネットワークに対しても並列計算を実施することができ、各リンクの交通量については妥当性のある結果を得られた。一方で、過剰な遅れ時間が生じており、その原因の究明が必要である。

今後の課題として、今回の計算において、日々の繰り返し計算の中でリンクの方向別交通容量  $\chi$  がどのように与えられているかを確認するとともに、合流や分流

における交通流の計算方法を精査する必要がある。また、計算の過程でグリッドロックが生じている可能性も考えられるため、確認が必要である。さらに、今回、大規模なネットワークに対して計算を適用することができたが、さらなる計算時間の削減のために CPU の数を増やした際により大きな効果が得られるような並列計算アルゴリズムを構築する。

**謝辞:** 本研究は、「ポスト「京」で重点的に取り組むべき社会的・科学的課題に関するアプリケーション開発・研究開発(文部科学省)」の重点課題(3)「地震・津波による複合災害の統合的予測システムの構築」のサブ課題 B「統合的予測のための社会科学シミュレーションの開発」の一環として実施された。

本論文の結果(の一部)は、理化学研究所のスーパーコンピュータ「京」を利用して得られたものです(課題番号:hp160224) This research used computational resources of the K computer provided by the RIKEN Advanced Institute for Computational Science through the HPCI System Research project (Project ID:hp160224)

#### 参考文献

- 1) 福田和輝, 井料隆雅, ウィジャラトネ ラリット, 浦田淳司: 効率的な並列計算アルゴリズムによる動的交通量配分の高速計算, 第 14 回 ITS シンポジウム 2016, CD-ROM, 2016.
- 2) Iryo, T: Properties of Dynamic User Equilibrium Solution: Existence, Uniqueness, Stability, and Robust Solution Methodology, *Transportmetrica B: Transport Dynamics*, Vol.1, No.1, pp.52-67, 2013.
- 3) Kuwahara, M. and Akamatsu, T.: Dynamic Equilibrium Assignment with Queues for a One-to-Many OD Pattern, in *Transportation and Traffic Theory: Proceedings of the 12th*

- International Symposium on the Theory of Traffic Flow and Transportation*, C. F. Daganzo(Ed.), pp.185-204, 1993.
- 4) 井料隆雅: 車両を離散化した動的交通量配分問題の Nash 均衡解の解法, *土木学会論文集 D3*, Vol.67, No.1, pp.70-83, 2011.
  - 5) Iryo, T.: *Multiple Equilibria in a Dynamic Traffic Network*, *Transportation Research Part B: Methodological*, Vol. 45, pp. 867-879, 2011.
  - 6) 石原雅晃, 井料隆雅: マルコフ連鎖による動的ネットワーク交通量配分, *土木学会論文集 D3*, Vol.71, No.5, pp.503-509, 2015.
  - 7) 石原雅晃, 藤原龍, 井料隆雅: 多数ケースの動的交通量配分の数値計算の高速化, 第 13 回 ITS シンポジウム 2015, CD-ROM, 2015.
  - 8) 石原雅晃, 福田和輝, 井料隆雅: 交通流シミュレーションによる動的利用者均衡配分の高速計算, *情報処理学会 第 78 回全国大会*, 2016.
  - 9) Bar-Gera, H. : *Transportation Test Problems*, <http://www.bgu.ac.il/~bargera/tntp/> (2017 年 4 月 17 日閲覧)
  - 10) 福田和輝, 石原雅晃, 井料隆雅: 渋滞の延伸を考慮したマルコフ連鎖による動的利用者均衡配分, 第 53 回土木計画学研究発表会, CD-ROM, 2016.
  - 11) Newell, G. F.: A Simplified Theory of Kinematic Waves in Highway Traffic, part 1: General Theory, *Transportation Research Part B: Methodological*, Vol. 27, No. 4, pp. 281-287, 1993.
  - 12) Daganzo, C. F.: The Cell Transmission Model, part 2: Network Traffic, *Transportation Research Part B: Methodological*. Vol. 29, No.2, pp. 79-93, 1995.
  - 13) 広域道路網交通シミュレーションシステム SOUND 製品ホームページ, <http://www.itransportlab.jp/products/sound/index.html> (2017 年 3 月 31 日閲覧).

(2017. 4. 28 受付)

## FAST CALCULATION OF DYNAMIC USER EQUILIBRIUM ASSIGNMENT ON A LARGE SCALE NETWORK BY PARALLEL COMPUTER

Kazuki FUKUDA, Takamasa IRYO, Petprakob Wasuwat, Wijerathne Lalith,  
Junji URATA, and Jr. Peque Genaro