

車種別交通量配分モデルの並列計算効率に関する研究*

Parallelization Efficiency of a Multi-Class Traffic Assignment Model*

井ノ口 弘昭**・土井 孝浩

By Hiroaki INOKUCHI**・Takahiro DOI

1. はじめに

交通量配分モデルは、従来では分割配分法が主に用いられていたが、実務においても均衡配分法が用いられつつある。一方、交通量配分モデルは、道路網計画のほか、交通運用対策や環境影響の評価等にも用いられている。このような場合、大型車と普通車とを分けて交通量を予測することが必要である。通常の分割配分法では、車種別の交通量は理論的に正しく推定できなると言われながらも、車種別の計算が行われることがあった。均衡配分法においても、通常は確定的利用者均衡配分モデルでは、リンク交通量は解の唯一性が保証されるが、経路交通量は保証されないという性質¹⁾から、これを車種別に拡張しても車種別交通量の解の唯一性は保証されない。そこで我々は、経路交通量の解の唯一性が保証される確率的利用者均衡配分に注目し、これを車種別に拡張した車種別確率的利用者均衡配分の提案を行った^{2,3)}。

我々が提案した車種別確率的利用者均衡配分は、アルゴリズムの特徴から、単車種の配分の計算時間のおよそ車種数倍の計算時間で計算が可能である。コンピュータの性能が格段に向上したとはいえ、配分対象道路網が拡大すること、実務においては条件を変えながら何度も繰り返して計算する必要があることから、計算時間を短縮することは意義が大きい。そこで本研究では、車種別確率的利用者均衡配分の計算において PC クラスタを用いた並列計算を行った場合の、計算時間の短縮効果を検討することを目的とする。

2. 本研究で用いる PC クラスタについて

本研究では、4 台のパーソナルコンピュータをネットワーク接続した PC クラスタを用いた。システムの概要を表-1 に示す。PC クラスタシステムは高性能なワークステーション等と比べて安価であり、汎用性が高いのが特長である。近年では、各種の大規模な科学技術計算に用いられている手法である。

* キーワード：交通網計画，配分交通，計画情報

** 正会員，博士(工学)，関西大学 工学部 都市環境工学科

(大阪府吹田市山手町 3 - 3 - 35, TEL/FAX 06-6368-0964,

E-mail hiroaki@inokuchi.jp)

表-1 PC クラスタシステムの概要

CPU	Intel Xeon 3.2GHz×4 台
メモリ	4GB×4 台
OS	Fedora Core 3×4 台 (Windows XP x64 Edition 1 台)
ソフトウェア	SCore Cluster System ⁴⁾ (PC クラスタコンソーシアム) (Windows XP では、Microsoft Visual Studio 2005)
ネットワーク	1Gbit Ethernet (スイッチング Hub 経由)

3. 交通量配分の計算条件

本研究では、都市圏を対象とした比較的大規模な道路ネットワークを想定して、計算効率性の検討を行った。リンク数などは表-2 の通りである。なお、有料道路への料金の考慮は、取扱いが容易な料金抵抗法を用いて、それぞれの車種で時間価値を設定した。

表-2 交通量配分の計算条件

リンク数	13,753
ノード数	9,679
セントロイド数	1,156
車種数	3 車種

4. 車種別交通量配分モデルの並列化の方法

(1) 各ステップの計算時間

まず、車種別確率的利用者均衡配分のアルゴリズムで、繰り返し 1 回当たりの各ステップの計算時間について調べた結果を表-3 に示す。なお、本計算は Windows XP のパーソナルコンピュータ 1 台のみで行ない、Step.1 からは 10 回の繰り返しの平均値である。この結果から、Step.2 の降下方向ベクトルの計算のうち Dial のアルゴリズムによる計算、Step.3 の一次元探索の計算が計算時間の大多数を占めていることが分かる。Step.0 については、Step.2 の Dial のアルゴリズムを用いている。従って、Step.2(Step.0)と Step.3 の部分について、並列化を検討する。

表-3 繰り返し1回当たりの各ステップの計算時間

Step.0 初期許容解の計算	2502 秒	—
Step.1 リンクコスト改訂	0	0%
Step.2 降下方向ベクトルの計算	2462	76%
うち, Dial による計算	2461	76%
Step.3 一次元探索	790	24%
Step.4 解の改訂	1	0%
Step.5 収束判定	3	0%

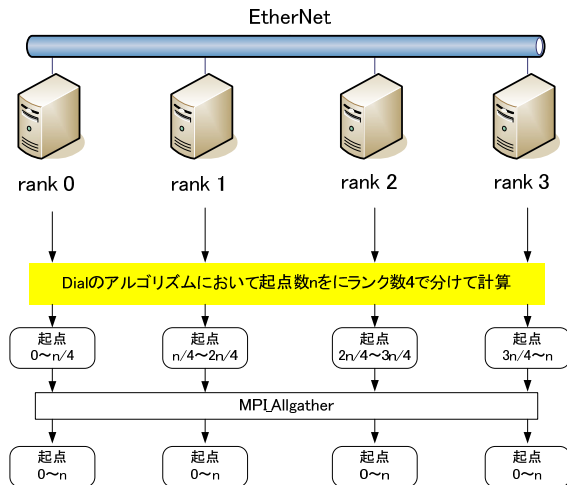


図-1 並列化の方法①

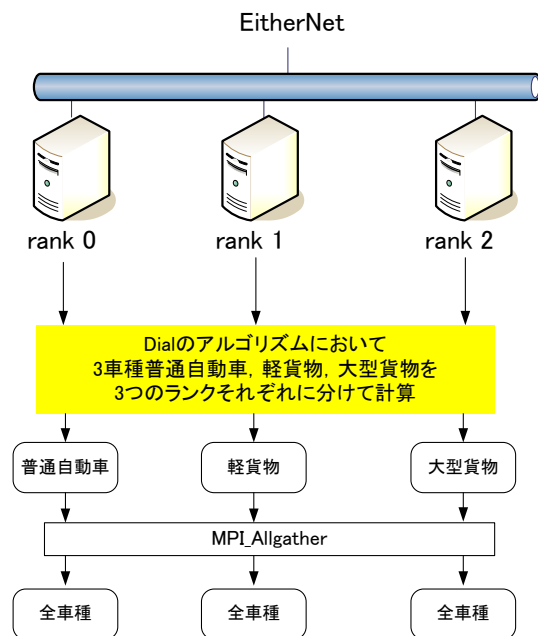


図-2 並列化の方法②

(2) Dial のアルゴリズムの並列化

Dial のアルゴリズムによる計算は、起点ごとに独立であるため、並列化の方法として、①1 起点から全てのノードまでの最小交通費用を求める最短経路探索において、

この各起点での繰り返し作業を n 個のプロセスに分けて計算させる(図-1)、②最短経路探索において、車種別に起点で繰り返す作業を1プロセスあたり1車種に分けて計算させる(図-2)の2つについて検討を行った。

(3) 一次元探索の並列化

一次元探索においては、目的関数の計算が主となるため、これの並列化を考える。車種別確率的利用者均衡配分の目的関数は次式の通りである。

$$\min .Z(x) = \sum_{ij} \int_0^{x_{ij}} t_{ij}(\omega) d\omega - \frac{1}{\theta} \sum_c \sum_r \{HL(x^{c,r}) - HN(x^{c,r})\}$$

$$\text{s.t. } x_{ij} = \sum_c \sum_r E_c x_{ij}^{c,r}, \quad x_{ij}^{c,r} \geq 0$$

ここで、 c は車種、 r は起点、 E_c は車種 c の乗用車換算係数、 HL, HN は以下で示されるエントロピー関数である。

$$HN(x^{c,r}) = -\sum_j \left(\sum_i E_c \cdot x_{ij}^{c,r} \right) \ln \left(\sum_i E_c \cdot x_{ij}^{c,r} \right),$$

$$HL(x^{c,r}) = -\sum_{ij} E_c \cdot x_{ij}^{c,r} \ln(E_c \cdot x_{ij}^{c,r})$$

目的関数の右辺第1項は、リンクコスト関数の積分項である。この項は、起点数・車種数には関係なく、リンクに関する繰り返しであるため、リンクの繰り返し作業を n 個のプロセスに分けて計算させる。 HL, HN の計算項は、車種および起点での繰り返し処理を行っている。従って、Dial のアルゴリズムの並列化と同様に、起点を n 個に分ける方法と、車種別に分ける方法が考えられる。この2つの方法について検討を行った。

5. 並列化の計算効率性の検討

今回の計算では、全体の計算時間の76%を占めるDialのアルゴリズムと24%を占める一次元探索について並列化を行ったため、理想的にはCPU数が n 個であれば、計算時間はほぼ $1/n$ になる。しかしながら、一般的にはCPU間でデータの受け渡しが必要であるため、理想値にはならない。

PC1台のみを用いた場合と2~4台のPCクラスタを用いた場合の繰り返し1回当たりの各ステップの計算時間を表-4,5に示す。表-4は、Dialのアルゴリズム、一次元探索の並列化を①の起点別に行ったもので、表-5は②

表-4 並列化の方法①の繰り返し1回当たりの各ステップの計算時間

CPU数	1	2	3	4
Step.0 初期許容解の計算	2435	1248 (51%)	856 (35%)	643 (26%)
Step.1 リンクコスト改訂	0	0 (100%)	0 (100%)	0 (100%)
Step.2 降下方向ベクトルの計算 うち、Dialによる計算	2380 2381	1236 (52%) 1238 (52%)	860 (36%) 862 (36%)	649 (27%) 650 (27%)
Step.3 一次元探索	952	508 (53%)	327 (34%)	238 (25%)
Step.4 解の改訂	3	3 (100%)	3 (100%)	3 (100%)
Step.5 収束判定	0	0 (100%)	0 (100%)	0 (100%)

単位：秒。()内は、CPU数1の場合との計算時間の比率である。

表-5 並列化の方法②の繰り返し1回当たりの各ステップの計算時間

CPU数	1	3
Step.0 初期許容解の計算	2435	980 (40%)
Step.1 リンクコスト改訂	0	0 (100%)
Step.2 降下方向ベクトルの計算 うち、Dialによる計算	2380 2381	974 (41%) 976 (41%)
Step.3 一次元探索	952	326 (34%)
Step.4 解の改訂	3	3 (100%)
Step.5 収束判定	0	0 (100%)

単位：秒。()内は、CPU数1の場合との計算時間の比率である。

の車種別に行ったものである。起点別に並列化を行ったものは、Step.2のDialの計算の並列化、Step.3の一次元探索の並列化とも、理想値に近い並列化効率が見られることが分かる。また、車種別に並列化を行ったものは、Step.3の一次元探索では起点別と同様の並列化効率が見られているが、Step.2のDialの計算では、起点別と比べて若干、効率が落ちている。今回用いたODデータでは、車種2,3のOD交通量で0の箇所が半分近くあった。従って、車種別に並列化を行った場合は、車種2,3を計算するCPUは車種1に比べて早く終了して待ち状態が出来ていたと考えられる。それに比べて起点別に並列化を行った場合は、このような待ち状態はほとんど発生しないで、ほぼ同様の負荷で計算出来ていたと考えられる。起点別に計算する場合の1CPUが受け持つOD交通量の個数に大幅なばらつきがない限り、①の起点別に並列化を行った方が良いと言える。

なお、1台のみで計算した時の計算結果とPCクラスタを用いて計算した時の計算結果を比較した結果、同様の結果が得られたことを確認した。

6. おわりに

配分対象道路網の拡大等による計算時間の増大に対応するため、PCクラスタシステムを使用して交通量配分の並列計算の計算効率について検討した。その結果、車種

ごとにODデータの個数が大幅に異なる場合は、起点を対象に並列化を行うのが効率的であることが分かった。また、計算効率は理想値に近い値であり、車種別確率的利用者均衡配分は、並列化を行うと効率よく計算出来ることが分かった。

今回の計算では、実用性を重視することおよび予算の制約から、市場に出回っている汎用的なコンピュータを用いた。ネットワークカード、HUBについても、コンピュータに付属されている標準的な1Gbit Ethernet Card、安価で一般的な1Gbitスイッチングハブを用いた。この程度の機器を使用しても並列化の効果は十分得られることが実証された。また、PC数は変更することが可能であるため、計算の規模、計算時間の許容値、予算に応じて柔軟に対応できる。

今回は、単純に起点数をPC数で割って割り当てを行ったが、負荷の平滑化を考えた場合、計算するOD交通量の個数も指標に入れることが考えられる。また、車種別確率的利用者均衡配分の計算アルゴリズムは、高速化のためのアルゴリズム改良の余地が残っているため、今後、改良を進めていく必要がある。

謝辞：本研究は文部科学省の科学研究費(課題番号：17760433)の助成を受けた研究成果の一部である。

参考文献

- 1) 交通ネットワークの均衡分析—最新の理論と解法—, 土木学会, 1998.
- 2) 井ノ口 弘昭, 岡田 良之: 車種別確率的利用者均衡配分の実用化に向けての検討, 土木学会第 57 回年次学術講演会, IV-32, 2002.
- 3) 荻田 美幸, 河上 省吾, 井ノ口 弘昭: 有料道路を含む道路網への車種別確率的利用者均衡配分の適用法に関する研究, 土木計画学研究・講演集, Vol.30, No.56, 2004.
- 4) PC Cluster Consortium : <http://www.pccluster.org/>
- 5) 道路交通需要予測の理論と適用 第 I 編 利用者均衡配分の適用に向けて, 土木学会, 2003.
- 6) 金森 亮, 河上 省吾: 車種を考慮した確率的利用者均衡配分モデルに関する研究, 土木学会年次学術講演会講演概要集第 4 部, Vol.56, pp.702-703, 2001.
- 7) 井ノ口 弘昭, 岡田 良之: 車種別確率的利用者均衡配分の計算効率性の検討, 第 27 回土木計画学研究講演集 (CD-ROM), 2003.