

セル・オートマトン交通シュミレータの クラスターコンピュータ上での実装について

清水 光輝[†] 玉城 龍洋^{††} 高村 健太郎[†]
安江 里佳[†] 北 栄輔[†]

本研究室では、これまでセル・オートマトン法と確率速度モデルに基づく交通シュミレータについて研究を行ってきた。このモデルは計算コストが比較的高いので、本研究ではクラスターコンピュータで並列処理を行うことについて述べる。解析領域を複数の部分領域に分けて、各部分領域を異なるPCで処理する。解析結果より、比較的良好な並列化効率を得られることがわかった。

IMPLEMENTATION OF CELLULAR AUTOMATA TRAFFIC SIMULATOR ON CLUSTER COMPUTER

HIKARU SHIMIZU,[†] TATSUHIRO TAMAKI,^{††} KENTARO TAKAMURA,[†]
SATOKA YASUE[†] and EISUKE KITA[†]

A traffic simulator has been developed in our laboratory. The simulator is based on the cellular automata and stochastic velocity model. Since its computational cost is expensive, the parallel computation of the model on the cluster computer system is described in this paper. A whole object domain is divided into sub-domains and then, PCs are assigned on different sub-domains. Numerical results indicate that the algorithm has relatively good efficiency for parallelization.

1. はじめに

自動車社会と呼ばれるようになって久しい現在、様々な道路交通問題が発生している。とりわけ交通渋滞は、排気ガスによる空気汚染、二酸化炭素による地球温暖化を招き、また、物資を輸送する際の輸送コストや、渋滞に巻き込まれたことによる時間的損失等、大きな問題として取り上げられている。それらの問題に対し、交通渋滞の原因を明らかにし、それらの問題点を排除したよりよい都市計画を作り出すために、現在、多種多様な交通シュミレータが開発されている [1-5]。

交通シュミレータは、マクロモデルとミクロモデルに大別できる。マクロモデルでは交通流を連続体として扱うのに対して、ミクロモデルでは車両1台1台をコンピュータ上で表現し、その創発現象として交通

流をモデル化する。ところで、交通流の正確なシュミレーションにおいて個々の車両挙動を表現することが必要と考えられており、個々の車両挙動をモデル化できるミクロモデルの利用が検討されている。そこで、本研究室でもセル・オートマトン法と確率速度モデルに基づく交通シュミレーションモデルを開発した [6, 7]。しかし、このモデルはミクロモデルなので、大規模な領域をシュミレーションのためには、計算コストが大きくなる欠点がある。そこで、本研究では、交通シュミレーションモデルをPCクラスターコンピュータ上で実行することについての基礎的研究を行う。解析領域を複数領域に分割し、部分領域を異なるPCに割り当てて並列実行する。計算環境はPCとMPIから構成されている。対象な街区を1PC, 2PC, 4PCで計算し、その結果を比較する。

本論文の構成は以下のようになっている。第2章では、シュミレーションに用いたモデルと、並列化手法について簡単に述べる。第3章では、解析結果を示し、第4章はまとめとする。

[†] 名古屋大学大学院 情報科学研究科 複雑系科学専攻
Graduate School of Information Science, Nagoya University

^{††} 宇部工業高等専門学校 経営情報学科
Department of Business Administration, Ube National College of Technology

2. 提案手法とその評価

2.1 本研究で用いる交通シミュレータ

2.1.1 セル・オートマトン

本研究で用いるシミュレーションモデルはセル・オートマトンに基づいている。解析対象領域をセルと呼ばれる多数の区分領域に分割し、各セル上にはある種の離散的状態量を定義する。時間を微小タイムステップで定義し、離散時間ステップごとに、状態量を近傍のセルとの相互作用によりローカル・ルールに従って変化させることで、現象をシミュレートする。

セル・オートマトンモデルを交通流シミュレーションに用いる時、道路は連続するセルの集まりで表現される。状態量は車両の存在、車両種類、道路種類をとり、ローカルルールでは、注目車両の行動を決定するように定義される。

2.1.2 確率速度法

交通流のモデル化では、車両の速度と移動の表現が重要である。速度は1ステップあたりの移動距離によって定義されるが、都市交通のシミュレーションにおいては1ステップあたりの移動距離の定義が車両行動のルール作成に影響する。高速道路と異なり、都市では道路上に多数の交差点や信号がある。これまでの方法では、高速で移動する車両ほど移動距離は大きくなるので、車両の進入している交差点や赤信号などを飛び越えてしまうことが起こりえる。そこで、本研究では確率速度法 [6] を用いている。この方法では、乱数を用いることで、1タイムステップに進む車両の移動量を最大1とする。これにより、上記のような問題を解決している。

2.2 並列化の手法

並列化の前提条件として、基となる交通シミュレータへの改変を最小限に抑えることが求められていた。これは今後、交通シミュレータが拡張された時に、その並列化のためのコストを小さくするためである。

そこで、解析領域を複数の部分領域に分割し、それらを異なるPCに割り当てる方法を用いる。部分領域の間には通信路を配置した。通信路とは、それぞれのPCの処理エリア同士の架け橋となっている区間であり、隣接するエリアを担当するPCで共有している。通信路の長さは、その道路の最高速度と通信を行うインターバルの長さによって決める。定められたインターバル毎にお互いの通信路上の車両情報を交換することで、それ以外の時間はそれぞれのマシンが完全に独立して処理を行う。具体的には、インターバルを20タイムステップと設定した場合、タイムステップ

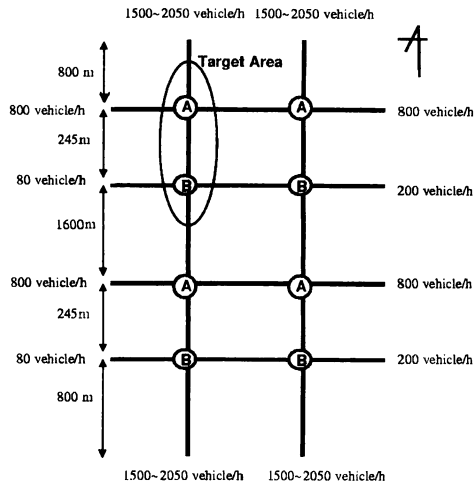


図1 解析対象

Fig.1 Object under consideration

表1 信号の時間設定

Table 1 Signal schedule

信号	主道路			従道路			Offset
	青	右折	赤	青	右折	赤	
(A)	75s	10s	3s	45s	10s	3s	62.3s
(B)	87s	6s	3s	47s	なし	3s	44.6s

が20の倍数になったとき、それぞれの通信路に設定された宛先に、その時点での通信路の道路情報を送信する。その結果、車両は各PC間を移動出来ることになる。

今回の実験では、インターバルを20タイムステップ(実時間の2秒)とした。そのため、最高時速60km/hの道路に通信路を設ける場合は12セル(30m)の通信路を定義することになる。これは、インターバルの2秒の間に高い速度を持った車両が、通信路を完全には通過しきれない距離であり、隣り合ったエリア間の車両の転送に、漏れが生じないようにするためである。

またシミュレータはMicrosoft Visual C++で開発されている。そのため、並列化の際にもVisual C++が使える並列環境が望ましいと考え、並列化の実装には、MPIを用いる。

3. 解析結果

3.1 実験環境

使用した評価環境は以下の通りである。

- PC1 CPU: Athlon XP 2000+ MEM 256MB
- PC2 CPU: Athlon XP 1700+ MEM 256MB
- PC3 CPU: Athlon XP 1700+ MEM 256MB

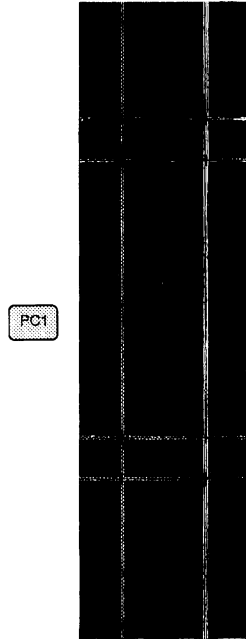


図2 解析領域 1PC

Fig.2 Object under consideration for 1 PC

表2 交差点での車両の右左折比

Table 2 Ratio of turning vehicles at intersections

交差点	左折：直進：右折
(A) 東向き	25 : 50 : 25
(A) 西向き	25 : 50 : 25
(A) 南向き	5 : 90 : 5
(A) 北向き	5 : 90 : 5
(B) 東向き	30 : 40 : 30
(B) 西向き	30 : 40 : 30
(B) 南向き	1 : 98 : 1
(B) 北向き	1 : 98 : 1

• PC4 CPU: Athlon XP 1700+ MEM 256MB
OSはWindows2000であり、MPICH.NT-1.2.5をインストールしている。

図1に解析領域の概略図を、図2、図3、図4に解析領域のマップを示す。信号制御については、図1のように設定する。各交差点において想定している、左折・直進・右折比率を表2に示す。他のシミュレーション条件は表3のようにする。

3.2 結果

シミュレートの結果を図5と図6に示す。表示したデータは、図1の楕円で囲んだ領域に関するものである。2つの解析結果を見比べると、下り方向の平均旅行速度は、その特性がよく一致している。それに比べ

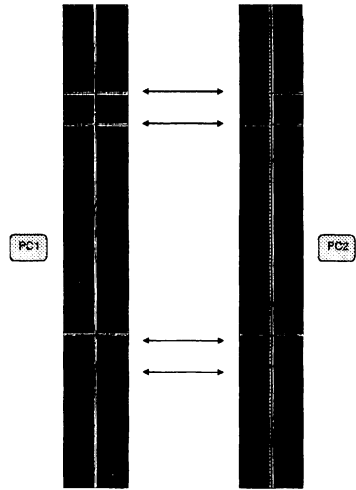


図3 解析領域 2PC

Fig.3 Object under consideration for 2 PC

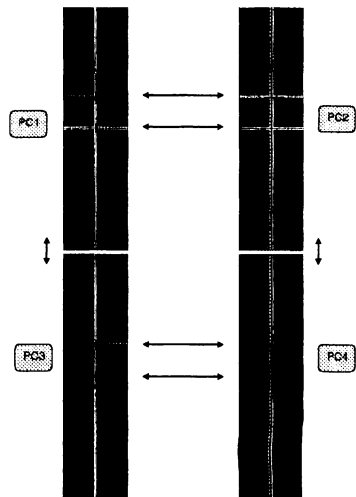


図4 解析領域 4PC

Fig.4 Object under consideration for 4 PC

表3 解析パラメータ

Table 3 Simulation parameters

セルの大きさ	3m × 3m
1タイムステップ	0.1s
境界条件	開放境界条件
車両発生	発生交通量に応じて等間隔に発生
最高速度	50km/h
安全車間距離	$0.0029v^2 + 0.3049v$ (v: 速度)
加速度	0.6 < α < 2.4 (加速時) 1.2 < β < 3.0 (減速時)
タイムステップ数	204400

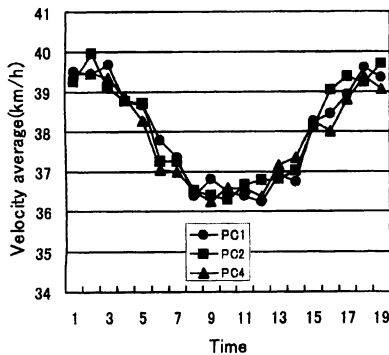


図 5 下り方向平均旅行速度

Fig. 5 Average traveling velocity along down line

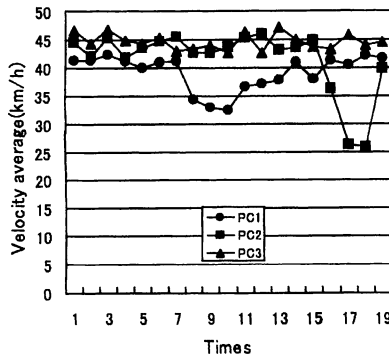


図 6 登り方向平均旅行速度

Fig. 6 Average traveling velocity along up line

ると、登りの平均旅行速度に多少のばらつきがみられる。その原因のひとつとして、通信路の設定状況が考えられる。車両の側から考えると、通信路は直進道路のように認識される。このために、車両によっては不自然に加速して通り抜けるようになり、その影響が解析結果に現れていると思われる。

解析にかかった時間は、PC1 台の場合で 32,153 秒、PC2 台だと 17,057 秒となり、PC4 台で 8,342 秒となった。PC1 台使用時の計算時間を基準に台数効果を求めてみると、PC2 台で 0.94、PC4 台だと 0.96 という結果になった。通信路が 20 タイムステップごとに通信する数は、PC2 台で 192 回、PC4 台だと 336 回となり、使用する PC が多くなるほど、通信回数は増える傾向にある。しかし今回の解析結果では、それ

でも PC4 台使用時の方が高い台数効果を示している。その理由は、実験環境の計算機の性能差だと考えられる。1 台目の計算機は、Athlon XP 2000+ を使用している。PC1 台使用時の計算は、すべてこの CPU で行うことになるので、2 台使用時や 4 台使用時に比べ、幾分か有利に働いていることが予想される。

4. 結 論

本研究では、確率速度モデルによるセル・オートマトン交通シミュレータを MPI を用いて構築された PC クラスター計算環境で実装することについて述べた。

この方法では、解析領域を複数の部分領域に分け、おのおの部分領域を異なる PC に割り当てることで処理を行う。この際に、部分領域の間に通信路と名付けた接合部分を配置した。この部分は、隣接する複数の部分領域で共有するようになっている。

アルゴリズムを簡単な解析対象に適用した。全領域を 1 領域とした場合、2 領域とした場合、4 領域とした場合について解析を行った。解析結果から良好な並列効率を得ることが出来たが、通信路の設定によって、複数領域における解析結果が 1 領域における解析結果と一致しないという問題点が現れた。そこで今後は、この問題点を解決することを考えている。

参 考 文 献

- 1) アイ・トランスポート・ラボ：街路網交通流シミュレーションモデル AVENUE.
- 2) フェニックス・リサーチ：交通流シミュレータ NETSIM 日本語版.
- 3) 豊田中央研究所：NETSTREAM.
- 4) 鹿島建設：ITS 道路インフラ計画支援システム REST.
- 5) (株) 日立製作所・日立研究所：交通流シミュレーションシステム TRAFFICSS.
- 6) 玉城龍洋, 安江里佳, 北英輔：確率速度モデルと CA 法による交通シミュレーション, 情報処理学会論文誌, Vol.45, No.3, pp. 858-869 (2004).
- 7) 玉城龍洋, 安江里佳, 北英輔：確率速度モデルを用いたセル・オートマトンによる都市交通シミュレーション, 情報処理学会論文誌 数理モデル化と問題解決, Vol.48, No.3, pp. 19-22 (2004).