

土地利用モデルによるマイクロシミュレーションにおける並列処理の枠組み

北詰 恵一¹・井ノ口 弘昭²

¹正会員 関西大学 環境都市工学部 (〒564-8680 大阪府吹田市山手町3-3-35)

E-mail: kitazume@kansai-u.ac.jp

²正会員 関西大学 環境都市工学部 (〒564-8680 大阪府吹田市山手町3-3-35)

E-mail: hiroaki@inokuchi.jp

土地利用モデルにおけるマイクロシミュレーションでは、膨大な計算量を比較的短時間で処理しなければならない。しかし、市民参加等に用いられる土地利用モデルが、ハイスペックコンピュータの利用可能な状況でシミュレーションできる場合は限られる。このため、パーソナルコンピュータベースでの計算スペックが求められることになるが、この場合、並列処理計算が有効である。本研究では、マイクロシミュレーションを実施する土地利用モデルにおいて、どのような計算アルゴリズムが発生し、その条件下でどのような並列処理の枠組みが望ましいかを検討した。多く主体が過去の履歴を持ちながら時系列的に進むタイプに対して並列関係でCPUが分担することから、CPU分担マネージャの必要性が示された。

Key Words : *land use model, micro-simulation, parallel computing*

1. はじめに

マイクロシミュレーション型土地利用モデルでは扱うデータ量や計算量が圧倒的に多くなり、それと同時に計算時間も長くなってしまふ。土地利用モデルが、都市政策実施による都市の将来像を市民に示し、行政とともに政策検討を進めていくツールと考えれば、必ずしも高性能のコンピュータを用意できない場面での利用も考えられ、計算時間の短いシミュレーションの仕組みが必要となる。しかし、CPU 開発による計算速度の向上は限界を迎えるといわれていおり、PC による並列計算の仕組みの構築が求められる。並列処理ではコンピュータにおいて、同時に複数の演算処理を実行することによって一定時間に処理できるデータ量を増やすことができ、計算時間の短縮を行うことが可能になる。

本研究は、一般に普及しているパーソナルコンピュータを想定して、それらを並列処理した場合のマイクロシミュレーション型土地利用モデルの枠組みを検討したい。我々研究グループは、これまで仮想的な都市における並列処理のプログラムを開発し、そのパフォーマンスを確認している。しかし、エージェント数が多くなったときの具体的な住宅選択部分の複雑化に伴うシミュレーションのパフォーマンスについては検討していない。

また、並列処理に関する研究は、情報処理学の分野等で進められており、さまざまな工夫による処理時間の短縮成果を得ている。例えば、木村ら³⁾は、命令レベル以外の並列性を抽出し、プロセッサチップ内で利用するアーキテクチャの開発を目指した研究を行い、機能向上結果を得た。また、吉田ら⁴⁾は、階層的に粗粒度並列処理を行う階層型マクロデータフロー処理におけるデータローライゼーション手法について検討し、処理時間の短縮を図った。石坂ら⁵⁾は、共有メモリマルチプロセッサ上でキャッシュ最適化を進め、後に示す粗粒度タスクの並列処理によって性能向上を果たしている。小林ら⁶⁾は、例えば有限要素法などへの適用の場面で、**Enriched Free Mesh Method(EFMN)**における並列処理効率の向上手法を検討し、その成果を得ている。

このように、プログラムそのもの、プロセッサに関する開発、および適用対象に応じた手法開発などを通じ、計算時間の短時間化が進んでいるが、土地利用モデルによるマイクロシミュレーションに対する並列処理について研究をしているものはない。本研究は、マイクロシミュレーション型土地利用モデルの特徴を、並列計算の観点から捉え直し、その特徴に応じた並列処理の枠組みを検討することを目的とする。

2. 並列処理

(1) 並列処理の効率化要件

並列処理をより効率的に行うための要件は、一般に、次のように整理できる。

- ① プログラム全体の中で並列処理ができない逐次処理部分の割合を少なくする。
- ② 各 CPU の負荷を均等に分散する。並列計算後は、それぞれの結果を集約するので、すべての CPU の処理が終わらなければ、集約ステップに進めない。
- ③ プロセッサ間の通信に要する時間を短くする。
CPU 間の通信速度は一般に遅いので、プロセッサ間の通信量を減らすことが望まれる。

(2) 並列処理の命令およびデータの流れ

並列計算機の種類によく用いられる Flynn の分類によれば、命令の流れとデータの流れによって4種類に分けられる。このうち、特に注目すべきは SIMD と MIMD である。SIMD は、複数のデータの流を個々に処理する同期化された CPU に対して同時に命令を出す。それぞれの結果は後に集約される。この方式は、オーバーヘッドと呼ばれる CPU 管理が容易である一方で、CPU 間の負荷を均等に分散させることが困難であり、CPU 間のロードバランスが悪くなる可能性が高いことが欠点となる。MIMD は、複数のデータの流を処理する CPU それぞれに命令を出す。このため、CPU ごとに異なった処理を行うことができ、柔軟な対応が可能となるメリットがある。

土地利用モデルによるマイクロシミュレーションでは、各 CPU に何を分担させるにしても、そこでの世帯推移や立地状況は、かなり不安定に変化する。個々の CPU に異なった柔軟な処理を課すのが普通であり、そのロードバランスも一定ではない。当研究グループが、仮想都

市に対して適用した仕組みは簡単なものであったので SIMD で検討したが²⁾、実際の都市圏に対して適用する場合は、MIMDを採用することが望まれる。

3. マイクロシミュレーションモデル

(1) マイクロシミュレーションモデルの課題と改善

マイクロシミュレーションモデルの概略構成は、図-2 に示す通りである。

これまでの研究で、最も計算時間のかかるステップは逐次処理部分の離家モデルであり、より詳細に言えば、エージェントとしての世帯の消滅と誕生後、新しいエージェントリストを作成する段階ということになる。消滅した世帯を空欄として放置し誕生した世帯を後に加える方式であればリスト作成が容易であるが、その後無駄なデータを持つリストをもとにシミュレーションを繰り返すことになる。一方で、消滅した世帯に誕生した世帯を当てはめる形式であればデータ効率は高くなるが、そのたびごとに消滅世帯のサーチを繰り返す必要があり、計算時間を長くする大きな要因となる。これは、実際に世帯がどれぐらい活発に消滅・誕生するかに依存すると考えられ、それほど多くは消滅・誕生しない世帯属性あるいは住宅タイプや土地条件があれば、それをを用いて適切な方法を選ぶことができる。

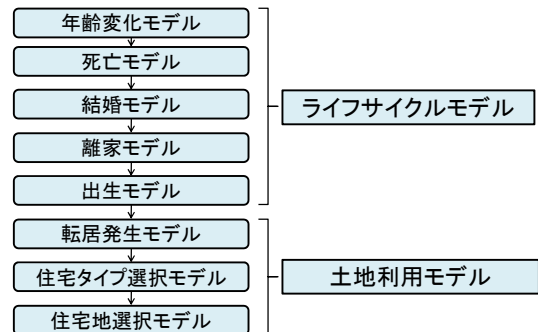


図-2 モデルの構成

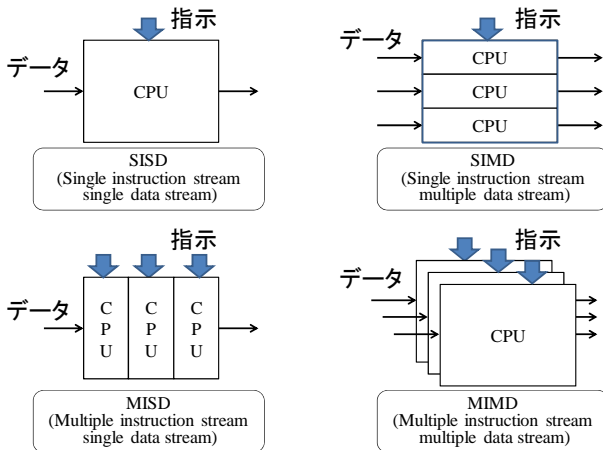


図-1 Flynnの分類による並列計算機の種類

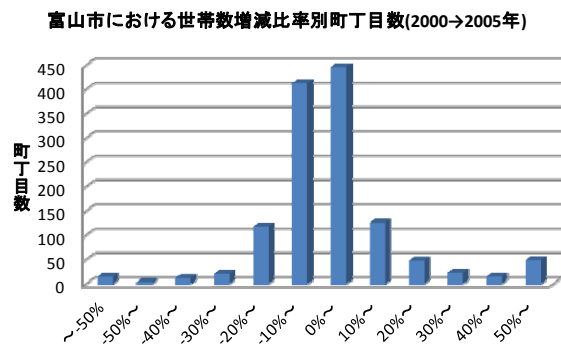


図-3 富山県における世帯数増減比率別町丁目数

そこで、実際適用を考えている富山市の世帯について検討した。富山市においてマイクロシミュレーションを実施することを想定して、世帯変化を概観する。総世帯数は、いずれも国勢調査ベースで2005年には、2000年からの5年間で5,906世帯増加し151,727世帯となり、4.1%増となっている。なお、2005年から2010年には4.9%の増加である。さらに細かく見ると、図-3に示す通り、空間分析単位に近い町丁目単位で、世帯数の増減比率別にその数を整理したものである。-10~10%で多くの町丁目の増減率が収まっているものの、-10%以下、あるいは10%以上の町丁目も100以上見られる。

これらのことから、町丁目別の世帯数変化のほとんどは、5年間で-20%~20%の幅で収まることがわかる。市域全体での総世帯数変化もわずかであることから、基本的には、誕生世帯をデータリストの後に追加し、消滅世帯のサーチ回数を減らしていくことで、時間短縮が図られるものと期待できる。

図-2に見るように、シミュレーションモデル全体はいくつかのステップにわかれた一連のモデルとなっている。このモデルの並列処理の観点から見た特徴を再整理すると、次のようになる。

- ① 粗粒度単位のプロセスが多く、ループ計算が多い。
ルートディストリビューションによる改善可能性が高い。例えば、すべての世帯データに同じ処理を繰り返すのであれば世帯データを分割させることもできるし、すべてのゾーンに対して同じ処理を繰り返すのであればゾーンデータも分割可能である。
- ② データ従属性が発生する場合、特に条件を設定しなければデータ全体にわたる。逆に言えば、土地利用モデルの特性に応じた条件設定を行うことで効率化が期待できる。
- ③ 一連の計算プロセスが、データ状況を変えて、何度も繰り返される。すなわち、ある1回のシミュレーション状況のデータを蓄積すれば、次回以降のシミュレーションの改善に対する学習情報として活用できるということである。

ここで粒度とは、逐次処理されるプログラムの単位であり、粒度が粗い(大きい)ほど通信コストが小さくなるものの並列度が下がる指標である。

これらを踏まえて、離家モデル部分以外でより大きな問題となる住宅選択モデル部分について、より細かい計算ステップを整理したのが図-4である。大きく世帯データとゾーンデータの処理が2つに分かれており、マッチングモデルを実施するときに、互いにデータのやりとりをするが、ゾーン選択肢形成やゾーンリスト作成ステップ、および対象期待世帯決定や世帯リスト作成ステップは、1回作成してしまえば、それを5年間シミュレーションの間は変えずにすむことができる。

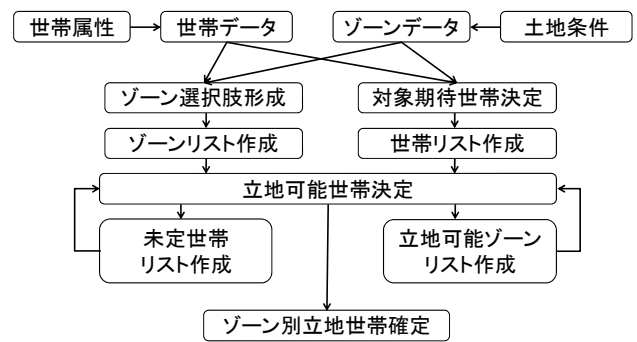


図-4 住宅選択モデル部分の詳細

より工夫が求めら得るのは、立地可能世帯決定から各リストの再作成のループであり、ループの過程でどの世帯とゾーンの組み合わせが決定し、どの組み合わせが決定していないかを、繰り返し計算ごとに把握しながら進めなければならない。

このような状況を踏まえて、枠組みを整理したい。

① ゾーン選択肢の限定化

いわゆる多選択肢問題の改善と同様に、ここでも特定の要件を満たす世帯が選ぶことのないゾーン選択肢を作成せず、数少ない選択肢でリストを構成することによって効率化を図る。特に、ゾーン選択肢形成がある限られた地域に限定されるならば、それをもとにCPU分担可能である。

② 対象期待世帯の限定化

地価を中心とした土地条件によって提供できる住宅の価格帯はある程度限定される。従って、個々のゾーンは一定所得以上の世帯のみを対象期待世帯とすることができ、リストを短くすることができる。さらに、それらが限定されるならば、それをもとにCPU分担可能である。

③ 繰り返し計算の一部スキップ

繰り返し計算部分を細かく分割し、特定の条件のエージェントはその条件に基づいて一部の分割された繰り返し計算部分をスキップする操作も可能である。これにより、計算時間の短縮が期待でき、CPU分担可能である。

(2) CPU分担マネージャーの機能

これらの他、CPU分担マネージャーの導入が必要である。まず、CPU分担ルールテストを行う。これは、世帯別のCPU分担とゾーン別のCPU分担の組み合わせで効率化が図られるとき、どのような条件で、いずれの分担のどのような組み合わせが効率的かをテストしておく、その結果に基づく分担ルールを作成するということである。

また、シミュレーションは、5年ごとの計算の繰り返しであることから、1回の一連の計算の後、CPU別のパフォーマンスを評価し、それをもとに分担ルールを変えていくような動的な並列処理を実施することで、全体

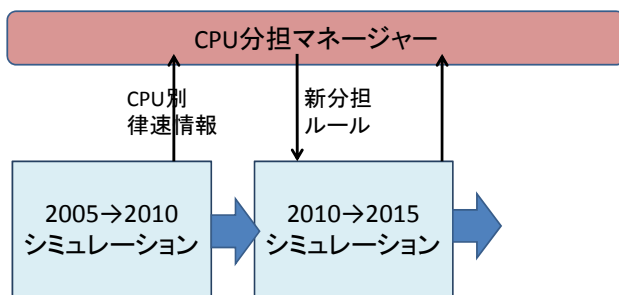


図-5 CPU 分担マネージャーのイメージ

の計算としては、効率化が図られるようにする必要があり。このことは、個々の CPU の能力がばらばらであっても、その均等化が実現できる点でも優れている。

4. まとめ

本研究では、マイクロシミュレーション型の土地利用モデルによって、都市圏レベルの数十万世帯をエージェントとして、20～30年間の住宅分布等を5年逐次シミュレーションする時に、通常市販レベルのパーソナルコンピュータを並列処理計算に用いることを想定した枠組みを整理した。

- ① 実際のシミュレーションにおいては、モデルの特性から、複数の CPU に複数のデータの流れがあり、異なった命令を個々に送る MIMD が望まれる。
- ② 計算時間を要していた世帯消滅および誕生によるエージェントリスト再作成段階では、それほど世帯変更が多くないことから、全体データへのサーチ回数を減らすことで、大きな時間短縮効果が期待できる。
- ③ 複雑な計算ステップが必要になる住宅選択モデル部分では、ゾーン選択枝、世帯選択枝ともに限定化する工夫をほどこす。選択枝形成過程で独立であることがわかれば、それをもとに CPU 分担することで、さらに効率化できる。
- ④ 5年ごとの繰り返し計算が多いことから、CPU 分担マネージャーを置き、逐次パフォーマンスチェックを行いながら、逐次改善していく仕組みを構築す

る必要がある、
などがわかった。

並列化処理は、今後も期待される大きな技術である。情報理工学の立場からの改善もあってよいが、適用対象に応じた計算時間短縮への取り組みも重要である。さらに、今後検討する中で、実際の手順から理論的枠組みに提案できる内容を模索していきたい。

参考文献

- 1) Keiichi KITAZUME, Kazuma YOSHIMURA and Yusuke FUKUNAKA: Development of Microsimulation Model for Land Use Analysis on a Hypothetical City, Proceedings of the Eastern Asia Society for Transportation Studies, Vol.7, CD-ROM, 2009.
- 2) Keiichi KITAZUME, Hiroaki INOKUCHI, Katsumi SATO, and Kazuma YOSHIMURA.: Performance of Parallel Computing for a Hypothetical Land-use Microsimulation Model, Computers in Urban Planning and Urban Management'09, CD-ROM, 2009.
- 3) 木村啓二・加藤孝幸・笠原博徳：近細粒度並列処理用シングルチップマルチプロセッサにおけるプロセッサコアの評価、情報処理学会論文誌、Vol.42, No.4, pp.1234-1245, 2001.
- 4) 吉田明正・越塚健一・岡本雅巳・笠原博徳：階層型粗粒度並列処理における同一階層内ループ間データローカライゼーション手法、情報処理学会論文誌、Vol.40, No.5, pp.2054-2063, 1999.
- 5) 石坂一久・中野啓史・八木哲志・小幡元樹・笠原博徳、共有メモリマルチプロセッサ上でのキャッシュ最適化を考慮した粗粒度タスク並列処理、情報処理学会論文誌、Vol.43, No.4, pp.958-970, 2002.
- 6) 小林陽介・矢川元基、Enriched Free Mesh Method における並列処理効率の向上手法について、日本機械学会論文集 (A 編)、75 巻 750 号, pp.135-141, 2009.

(?????? 受付)