

# マルチエージェントシミュレーションのための 余剰計算力に基づく動的負荷分散の検討

玉垣宏行, 村田忠彦, 蟻川 浩



文部科学省私立大学社会連携研究推進拠点  
関西大学政策グリッドコンピューティング実験センター

Policy Grid Computing Laboratory,  
Kansai University  
Suita, Osaka 564-8680 Japan  
URL : <http://www.pglab.kansai-u.ac.jp/>  
e-mail : [pglab@jm.kansai-u.ac.jp](mailto:pglab@jm.kansai-u.ac.jp)  
tel. 06-6368-1228  
fax. 06-6330-3304

## **関西大学政策グリッドコンピューティング実験センターからのお願い**

本ディスカッションペーパーシリーズを転載、引用、参照されたい場合には、ご面倒ですが、弊センター（[pglab@jm.kansai-u.ac.jp](mailto:pglab@jm.kansai-u.ac.jp)）宛にご連絡いただきますようお願い申し上げます。

## **Attention from Policy Grid Computing Laboratory, Kansai University**

Please reprint, cite or quote WITH consulting Kansai University Policy Grid Computing Laboratory ([pglab@jm.kansai-u.ac.jp](mailto:pglab@jm.kansai-u.ac.jp)).

# マルチエージェントシミュレーションのための 余剰計算力に基づく動的負荷分散の検討

玉垣 宏行<sup>1</sup>, 村田 忠彦<sup>1,2</sup>, 蟻川 浩<sup>2</sup>

## Discussion of Dynamic Load Balancing with Idle Processing Power for Multi-Agent Simulation

Hiroyuki Tamagaki<sup>1</sup>, Tadahiko Murata<sup>1,2</sup>, Hiroshi Arikawa<sup>2</sup>

### 概要

本研究では、マルチエージェントシミュレーションを処理対象とし、デスクトップグリッドのように計算力が時々刻々と変化する環境を想定している。並列計算に余剰計算力に基づく動的負荷分散を適用し、各マシンにおけるステップごとの計算終了時間を平滑化することで、ステップごとの計算終了待ち時間を短縮することを目的としている。また、デスクトップグリッドでの動的負荷並列計算の有効な環境を検討するため、処理量や通信量、通信速度を変化させた実験を行い、一台のマシンでの計算と比較を行う。

### Abstract

In this paper, we assume that multi-agent simulation is processed on the environment where the calculation power hourly changes like a desktop grid. Our objective is to shorten the calculation end waiting time of each step by applying a dynamic load balancing based on the surplus calculation power to the parallel computation by smoothing the calculation end time of each step in each machine. We examine the amount of processing, the amount of the communication, and the transmission rate to find an effective environment for multi-agent simulation on a desktop grid. We compare it with the simulation on a computer.

キーワード：マルチエージェントシミュレーション，デスクトップグリッド，動的負荷分散  
Keyword: Multi-Agent Simulation, Desktop Grid, Dynamic Load Balancing

---

1 関西大学 総合情報学部

2 関西大学 関西大学政策グリッドコンピューティング実験センター

## 1. はじめに

マルチエージェントシミュレーションに基づく社会シミュレーションの様に、より現実に近いシミュレーションを実施する場合、エージェント数や処理回数が膨大になり、高性能なコンピュータが必要となる。様々な高性能コンピュータがある中で、対費用効果の高い設備投資として、デスクトップグリッドが注目されている。デスクトップグリッドは、ネットワークにつながった不特定多数のコンピュータ(マシン)の利用されていないCPUパワーを余剰計算力として使い並列計算する。デスクトップグリッドを利用することで、一台のマシンでは処理に膨大な時間がかかる問題であっても、余剰計算力を適切に利用することで高速化を図ることができる。松本[1]らや森下[2]らの研究において、デスクトップグリッドを想定したマルチエージェントシミュレーションの並列計算の有効性を示している。しかし、デスクトップグリッドを利用して各マシンに処理を分散させる時、利用できるマシンの性能や余剰計算力は不均一である。この不均一性により、処理能力に適した処理が割り当てられない場合が起こる。マルチエージェントシミュレーションにおいて、各マシンの処理能力に適した処理が割り当てられないことにより、ステップごとの計算終了時間にばらつきが出て、計算終了時間の早いマシンが遅いマシンを待つ時間が長くなり、高速化が図れないという問題が生じる。そのため、これらの問題について高速化するために各マシンにおけるステップごとの計算終了時間の平滑化、いわゆる負荷分散が必要となる。

これに対して前田ら[3]の研究では、二次元熱拡散問題と三次元分子動力学シミュレーションに動的負荷分散を適用することにより処理終了時間を短縮した。空間の中の処理をマシンの計算力に適応して動的に分割することで全体の処理の高速化している。しかし、前田らの研究においては、分散させるマシンの計算力は一定であり、デスクトップグリッドのような各マシンの計算力が変動する場合は考慮されていない。また、渡邊[4]の研究では、性能の異なる計算機で構成される並列計算環境における処理の最適な割当量を理論的な法則に基づいて導いている。この論文は一般的な内容であり、アプリケーションに即したものではない。そのため、デスクトップグリッドのような利用できる計算力が変動する環境において、負荷変動を考慮した動的負荷分散を適用する有効性を検討する必要があると考えられる。

また、マルチエージェントシミュレーションの並列処理の実現において、各マシン間で環境情報の交換のための通信が頻繁に発生する。並列計算することでステップごとの計算終了時間は短縮されるが、データの分散が適切でないと通信時間が長くなる。並列計算を利用しても処理時間と通信時間の割合によっては、台数に応じた並列処理効果が得られない。そのため、並列計算を適用することが有効な環境を検討する必要がある。

本研究では、各マシンの余剰計算力が変動するデスクトップグリッドを仮想的に構築し、マルチエージェントシミュレーションを対象として実験を行う。第2章では、各マシンにお

けるステップごとの計算終了時間のばらつきを変動する余剰計算力に合わせて、各マシンが担当する処理領域を動的に調整することで平滑化を目的とした、余剰計算力に基づく動的負荷分散の手法を示す。第3章では、提案手法の有効性を実験により示す。第4章では、処理量、通信量、通信速度を変えて、並列計算と一台のマシンでの計算(単体計算)における全試行の処理終了時間を比較することで、並列計算が有効な環境を検討する。

## 2. 提案手法

本研究では、SugarScapeモデルの様に二次元空間情報を有するマルチエージェントシミュレーションの分散処理を対象とする。二次元空間には資源とエージェントが多数存在する。資源は位置によって量が異なる。エージェントは周りのエージェントの位置を認識するための視野を持ち、エージェントの位置から上下左右に視野分の菱形の視界を持つ。視界内の資源との距離と量を考慮して移動する。

今回の環境では、複数のマシンを用いた分散処理を対象としており、各マシンに分配する環境情報およびエージェント情報の分配方法を考えなければならない。本研究では、図1に示すようにマシン台数分の短冊を作成し、各短冊をマシンに分配する。このとき短冊の大きさは、各マシンの余剰計算力に基づいて計算する。具体的な手法を以下で述べる。全エージェントが1行動することを1ステップとする。マシンの余剰計算力はステップごとに変動し、マシンの余剰計算力の変動に合わせて各マシンが処理するのに適したエージェント数が存在するように領域幅を調整する。ただし、変動する段階でのマシンの余剰計算力は不明であるため、前ステップ段階での余剰計算力に基づいて領域幅を調整する。領域幅は各マシンの処理できるエージェント数を基に調整される。マシン*i* ( $i=1\sim n$ : $n$ はマシン台数)の処理するエージェント数:  $processAgents(i)$ , 全エージェント数:  $allAgents$ , 前ステップ段階の余剰計算力:  $previousPower(i)$ として、式(1)を用いて算出する。

$$processAgents(i) = allAgents \times \frac{previousPower(i)}{\sum_{i=1}^n previousPower(i)} \quad (1)$$

式(1)より、マシンの余剰計算力に応じてエージェント数が決定するが、その数に基づき図1のように二次元空間を垂直に分割すると、式(1)で決定したエージェント数通りには分割できない。式(1)で決定したエージェント数分だけ正確に二次元空間を分割することも可能だが、分割面に凹凸ができるため、通信を考えた場合に非常に複雑な通信が必要となる。そのため、式(1)で決定した数からいくつかの誤差を許容して分割を行っている。

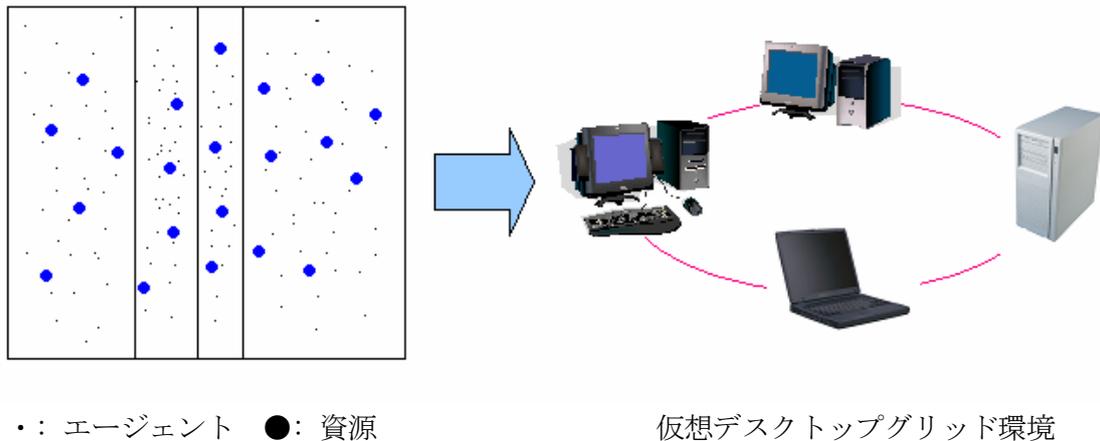


図1 二次元空間の分割

許容するエージェント数誤差:  $E(i)$ , 実際に処理するエージェント数:  $actualAgents(i)$  として, 分割した領域内のエージェント数が式(2)の範囲内にあれば, その値を基に領域を調整する. 本研究では  $E(i)$  を20エージェントとする.

$$\begin{aligned} processAgents(i) - E(i) &\leq actualAgents(i) \\ actualAgents(i) &\leq processAgents(i) + E(i) \end{aligned} \quad (2)$$

変動する余剰計算力に基づき分割領域を調整することで, 均等に分割した場合よりもステップごとの計算終了時間の平滑化が可能になる. その結果, 各マシン間の環境情報を交換するための待ち時間が減らすことができる. ステップごとの計算時間を短縮することで, 全体の処理終了時間も短縮できる.

### 3. 動的負荷分散の有効性の検討

本章では, デスクトップグリッドを想定した環境において, 余剰計算力に基づいて分割する手法(動的分割)と二次元空間をマシン台数分だけ均等に分割する手法(均等分割)のステップごとの計算終了待ち時間と全試行の平均処理終了時間の比較を行い, 動的分割の有効性を示す.

#### 3.1. 実験環境

マルチエージェントシミュレーションを表1のパラメータによって行い, その処理を表2の仮想デスクトップグリッドに分散させる.

マシンの性能の単位を  $W$ , 通信速度の単位を  $S$  とする. 1 エージェントあたりの処理量は, マシンが処理するのにかかる量として単位は  $P$  とする.

表1 シミュレーションの実験パラメータ

1試行のステップ数	100
二次元空間	1000×1000
エージェント数	10000
1エージェントの処理量(P)	10
エージェントの視野	20
エージェントの移動	上下左右 1
資源数	1000
資源量の上限	20

表2 仮想デスクトップグリッドのパラメータ

マシン台数	4
マシンの性能(W)	100~300
マシンの使用率	10~90 %
マシンの使用率増減	-5~5%
マシンの使用発生率	50 %
1エージェント, 資源の通信量(D)	200
通信速度(S)	100

通信量は1エージェントまたは1ヶ所の資源あたりの通信にかかるデータ量として単位はDとする。時間の単位をHとし、処理量Pをマシンの性能Wで除算すると処理時間Hが算出され、通信量Dを通信速度Sで除算すると通信時間Hが算出される。

エージェント数と資源数は固定とする。エージェントは資源を獲得するために移動する。エージェントはステップごとに獲得しようとする目標資源を探索し、エージェントと資源の距離を基に目標資源を決定する。資源を $t$ 、エージェント $i$ と資源 $t$ の距離を $D_{it}$ 、資源量を $R_t$ として、式(3)を基に最大となる $t^*$ を計算し、その資源を目標資源とする。

$$t^* = \text{Max}\{\omega_1 R_t - \omega_2 D_{it}\} \quad (3)$$

ここで、 $\omega_1$ および $\omega_2$ は非負の重みである。式(3)により、より距離が近く、資源量が多いほど目標資源 $t^*$ が選ばれやすくなる。エージェントが目標資源を獲得するとその資源は無くなり、他の位置に新しい資源が発生する。エージェントの視野内に資源がない場合、エージェントはランダムに移動する。

1台のコンピュータで仮想的に複数のマシンを構築する。マシンの性能は初期にランダムに決定し、ステップごとに使用率に基づき増減する。余剰計算力:  $Power(i)$  は、性能:  $Performance(i)$ 、使用率:  $UseRate(i)$  から式(4)を用いて算出される。

$$Power(i) = Performance(i) \times (1 - UseRate(i)) \quad (4)$$

通信は、各マシンが隣り合う領域の情報を交換する際に発生する。通信する領域は分割した境界からエージェントの視野が届く位置までの列内である。その領域のエージェントと資源の情報を交換する。通信時間はエージェントと資源の総通信量を通信速度で除算して算出し、ステップごとの計算終了時間に加えられる。総通信量はマシン間の環境情報を交換するための通信領域内に存在するエージェントと資源数に通信量を乗算して算出する。

表3 平均処理終了時間比較

	均等分割	動的分割
平均 (H)	164496.5	102515.5
標準偏差 (H)	12559.2	1214.0

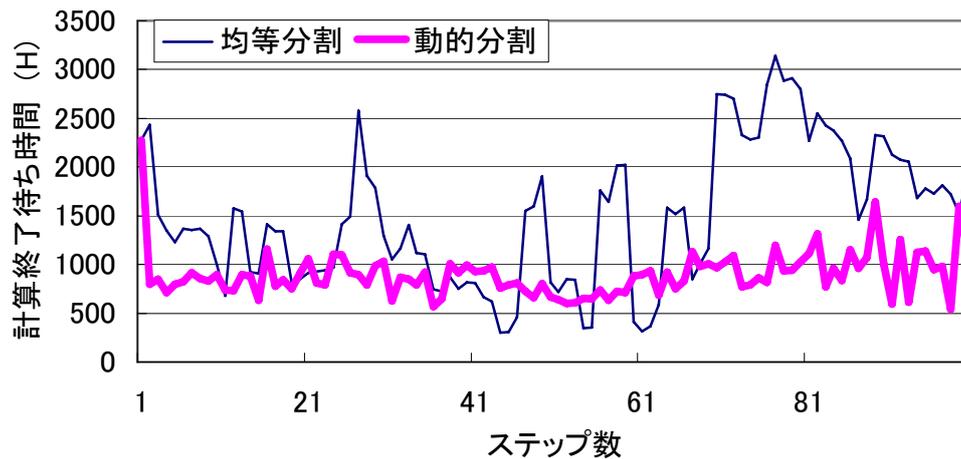


図2 計算終了待ち時間比較

比較対象として、二次元空間を均等に分割して分散処理する均等分割を用いる。均等分割は領域幅が変化することなく、二次元空間を列方向にマシン台数分で分割する。なお、本手法と比較対象の環境(マシンの性能、マシンの使用率の変化、エージェントの移動、資源の発生など)は共通とする。

### 3.2. 実験と考察

エージェントの100ステップを1試行とする。余剰計算力に基づく分割(動的分割)と均等分割の1試行におけるステップごとの計算終了待ち時間の比較を図2に、100試行の終了時間の平均と標準偏差の比較を表3に示す。ステップごとの計算終了待ち時間は、ステップごとに最後にマシンのステップごとの計算が終了した時間と他マシンのステップごとの計算が終了した時間との差の合計とする。均等分割は余剰計算力の変動とエージェントの移動に伴い、ステップごとの計算終了時間に大きなばらつきが生じる。提案手法では、1ステップ目は分割の基準となる前ステップの余剰計算力が存在しないため、均等分割と同じ計算終了待ち時間になる。しかし、それ以降は前ステップの余剰計算力を基準として、動的に領域の幅を調整することでステップごとの計算終了時間のばらつきを平滑化し、各マシンの計算終了待ち時間を減らすことができている(図2)。動的分割法により、全体の終了時間を約60%短縮することができる(表3)。

表4 単体計算環境のパラメータ

単体計算のマシン性能(W)	200
マシンの使用率	10~20%
マシンの使用率増減	-1~1%
マシンの使用発生率	20%

表5 並列計算環境のパラメータ

並列計算のマシン台数	4
並列計算の各マシン性能(W)	100, 150, 250, 300
マシンの使用率	10~90 %
マシンの使用率増減	-5~5 %
マシンの使用発生率	50 %
通信速度(S)	100

#### 4. 並列計算が有効な環境の検討

本章では、デスクトップグリッドを想定した環境での並列計算と一台のマシンでの計算環境(単体計算)の処理終了時間の比較を行う。処理対象となるマルチエージェントシミュレーションの環境を変化させ、並列計算が単体計算よりも有効な環境を検討する。

##### 4.1. 実験環境

処理対象となるマルチエージェントシミュレーションのパラメータは表1と同じとするが、1 エージェントあたりの処理量、1 エージェントまたは1 ケ所の資源あたりの通信量、通信速度を変化させる。単体計算環境のパラメータを表4に、並列計算環境のパラメータを表5に示す。

デスクトップグリッドを想定した並列計算の環境では、第3章の動的分割を適用し、利用できる4台のマシンの性能をそれぞれ100W、150W、250W、300Wと設定する。単体計算の環境では、マシンの性能を並列計算の利用できるマシンの性能の平均である200Wとする。単体計算では、マシン間での環境情報の交換による通信が必要ないため、通信時間は発生しない。単体計算では不特定多数のマシンを利用するわけではないので、計算力の変動が少ない設定にする。

##### 4.2. 実験と考察

並列計算、単体計算ともに100試行の平均処理終了時間とする。そして、並列計算は通信量を変化させた(図3)。

本実験では、1 エージェントまたは1 ケ所の資源あたりの通信量を変化させると、1 エージェントあたりの処理量10P、通信速度100Sに対して、通信量170D以降から並列計算のほうが単体計算より処理終了時間が長くなっている。これは並列計算により各マシンの処理時間を分散させても通信時間に長い時間がかかることにより、結果として高速化が図れ

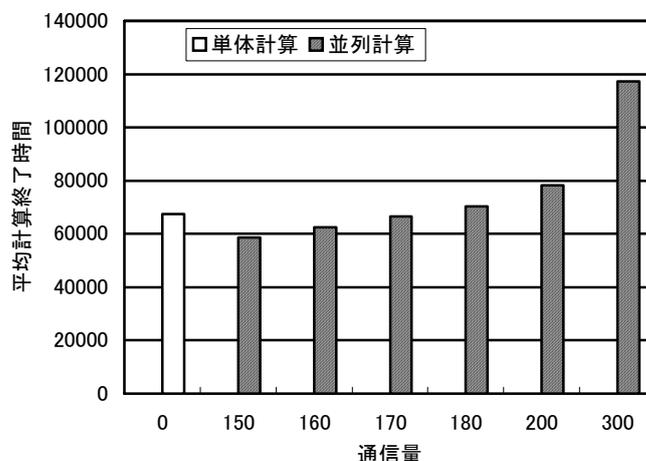


図3 平均終了時間比較(通信量変化)

なくなるためである．並列計算で高速化の効果を得るためには，1 エージェントあたりの処理量をもっと多い場合か通信速度をもっと高い必要がある．

この点を確認するため，通信速度 100S，通信量 170D に設定して，並列計算と単体計算ともに 1 エージェントあたりの処理量を変えて処理終了時間を比較する(図 4)．単体計算では，処理量の増加に比例して処理終了時間が長くなっている．並列計算では，処理量が増加しても処理終了時間にはほとんど影響がない．処理終了までのほとんどの時間を通信に使っていることがわかる．

また，通信量 170D，処理量 10P に設定して通信速度を高速化した場合の処理終了時間の比較をする(図 5)．通信速度を高速化した場合，処理終了時間はかなり短縮される．高速な通信回線を確保することにより，並列計算の有効性を高められることがわかる．

デスクトップグリッドでは，使える余剰計算力や通信環境，対象となる問題などによって並列計算が有効である状況が異なる．本研究の実験から，処理するマルチエージェントシミュレーションの環境は，環境情報交換のための通信量が少なく，エージェントあたりの処理量が多いものほど並列計算は有効である．また，通信速度が高速な分散環境であるほど並列計算を高速利用できることがわかった．

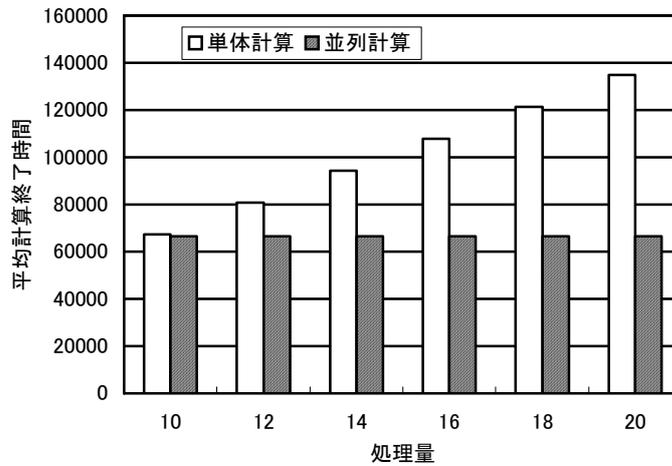


図 4 平均終了時間比較(処理量変化)

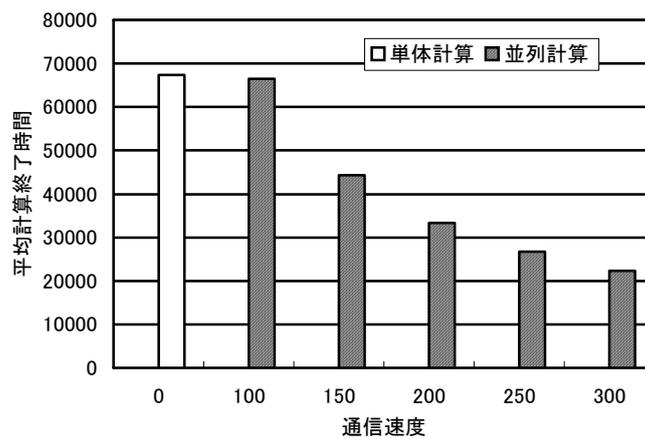


図 5 平均終了時間比較(通信速度変化)

### 4.3. 課題

今回の実験では、デスクトップグリッドの利用を想定した並列計算の有効な環境を検討した。実験結果から並列計算が有効な計算環境やマルチエージェントシミュレーションの環境の指針を示した。しかし、本実験では1台のマシンで仮想的にデスクトップグリッドを構築しているため、そのパラメータには具体性がなく、通信の遅延や利用できるマシン数の変化などは考慮されていない。

今後の課題として、実際に複数の計算機に分散処理させることで、よりデスクトップグリッドに近い環境のパラメータや計算時間を得る必要がある。その値を基に並列計算の最

適な利用をすることで、デスクトップグリッドを利用するにあたって、さらに有効なパラメータ設定や分散設定の指針を示すことができると考えられる。

## 5. おわりに

本研究では、デスクトップグリッドにおいて変動するマシンの余剰計算力に応じて処理する領域を動的に調整する動的余剰計算力分割手法を提案した。マルチエージェントシミュレーションに適用することによりステップごとのエージェントの計算終了待ち時間を減らし、全体の処理終了時間を短縮できた。また、1 エージェントあたりの処理量、1 エージェントまたは1 ケ所の資源あたりの通信量、通信速度を変化させ、デスクトップグリッドの利用を想定した並列計算の有効な環境を検討した。

今回は1 台のマシンで仮想的にデスクトップグリッドを想定してシミュレーションを行った。しかし、実際の並列計算環境に適応した場合には、使えるマシンの性能や通信環境の多様性、マシンの故障、通信遅延などの影響により同程度の効果が得られるかは不明である。今後、実際の並列計算環境における有効性を確認する必要がある。

## 参考文献

- [1] 松本優美, 村田忠彦, 蟻川浩, 山口正聡, 小橋博道, 門岡良昌 : グリッドによるマルチエージェントシミュレーションの分散処理, 計算工学講演会論文集, Vol. 11, pp. 545-548, (2006).
- [2] 森下仙一, 蟻川浩, 村田忠彦 : 環境改善資源調達マルチエージェントシミュレーションのMPIによる並列計算, ファジィ・システム・シンポジウム公演論文集, Vol. 22, pp. 107-112, (2006).
- [3] 前田太陽, 仙田康浩, 田子精男 : 分散メモリ並列計算機のための動的負荷分散ライブラリ, 計算工学講演会論文集, Vol. 7, pp. 187-190, (2002).
- [4] 渡邊誠也 : ヘテロジニアスな並列計算環境における最適な負荷割当, 電子情報通信学会論文誌, D - I, Vol. 88-D- I, No. 11, pp. 1688-1695, (2005).