

分散制約充足系への モバイルエージェントの導入と評価

西田 真和* 澁田 厚* 栗原 正仁* 沼澤 政信**

*北海道工業大学 **小樽商科大学

1. はじめに

ネットワーク技術の進展によりコンピュータは単体よりもインターネットなどのネットワークを介して使用するというような環境になり、情報処理や情報サービスにおいて、モバイルエージェントと呼ばれるホスト間を移動するエージェントの技術が有効であると考えられる。

それに伴って、人工知能の分野においても、知的エージェントの実現基盤の一つとして知られている制約充足技術も分散制約充足技術の研究に移行してきている。

本研究では、以上の二つの技術を用いてエージェント間のメッセージ通信だけで行っていた分散制約充足にモバイルエージェントを導入することによってシステムを拡張し、シミュレーションを行うことにより、移動が通信量に与える効果について評価する。

2. モバイルエージェント

モバイルエージェントとは、エージェント自身が利用者の代理人としてネットワーク上を自律的に移動しながら特定のタスクを遂行するものである。

現在、筆者らは“Telepathy”というモバイル・エージェント・システムをJavaにより開発・研究している。

またエージェントは“Telepathy”が提供するクラスライブラリを利用することで実現できる。

3. 分散制約充足

分散制約充足問題とは制約充足問題の変数と制約が複数のエージェントに分散された問題とみなすことができ、制約充足問題では、初期状態において一つのエージェントが全体の問題を見渡すことができるのに対し、分散制約充足問題の特徴は初期状態にはもちろんのこと、任意の状態においてもそれ

Introducing mobile agents to distributed constraint satisfaction systems and its evaluation

Masakazu Nishida, Atsushi Shibuta, Masahito Kurihara

*Hokkaido Institute of Technology

Masanobu Numazawa

**Otaru University of Commerce

ぞれのエージェントは全体の問題を知らない。

分散制約充足問題を解くアルゴリズムとしては横尾、平山[1]によって提案された反復改善型分散制約充足アルゴリズム(分散 breakout)が効果的な状態空間探索法の一つとして注目されている。

本研究ではこの研究を拡張し、モバイルエージェント間の分散制約充足問題とし、その解法アルゴリズムを検討しシミュレーションより実験的に評価する。

4. 通信量による移動アルゴリズム

4.1. 移動アルゴリズム

基本は分散 breakout を用いてメッセージ通信で行い、移動条件になると条件を満たしたエージェントが相手のいる場所に移動する。

4.2. 移動条件

エージェント間のメッセージの累積量が所定の量(エージェントサイズ)を超えたときに、移動条件を満たしたエージェントが個々のメッセージ量が最も多いエージェントのいる場所に移動する。

最大値が同数の場合はその中から非決定的に選択し移動先の Place を決める。

また移動条件は、移動するごとに増加していかねば常に条件を満たしてしまうので、移動後に2倍にしていく。

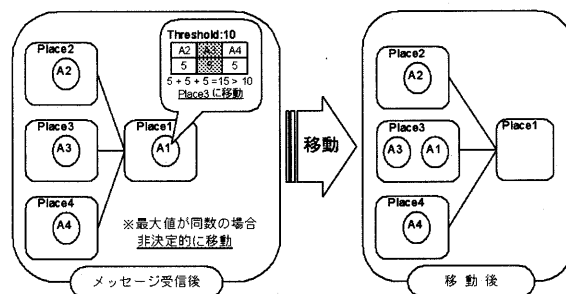


図 4.1 移動アルゴリズム

4.3. シミュレーション実験

以下(表 4.1)の条件で移動に関するシミュレーション実験を行い、移動しなかった場合と比較を行う。

問題	分散グラフ色塗り問題
色数	3色
エージェント数	30, 60, 90, 120, 150
リンク数	エージェント数により異なる
メッセージサイズ	1
エージェントサイズ	10 ~ 90, 100 ~ 1000 ※ エージェント数: 30, 60, 90は500まで, エージェント数: 120は600まで
移動条件の初期しきい値	エージェントサイズ x 1
実験回数	各リンクごとに100回

表 4.1 移動条件

4.4. 実験結果

シミュレーション実験の結果として Agent 数 90 のときの通信量と移動回数のグラフを以下に示す。

移動することにより総通信量が削減できていることがわかる。ただし、直観的にも明らかなことだが、エージェントサイズがあまり大きい場合は、移動することによって、その移動に必要な大きな通信量が加算されるので、総通信量は移動しないときよりかえって大きくなる。

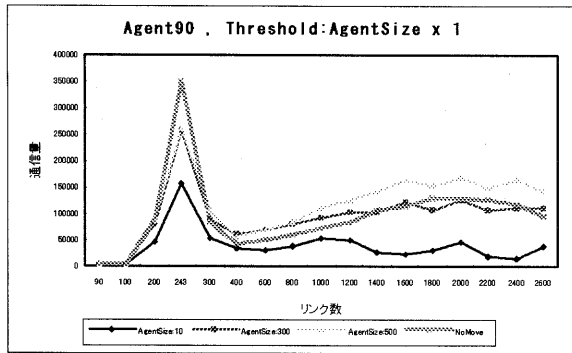


図 4.2 通信量

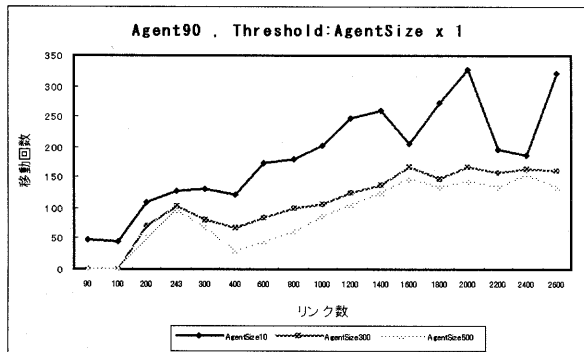


図 4.3 移動回数

4.5. 移動の境界線

前節で述べたように、エージェントサイズがある程度小さければ移動の効果があり、逆にある程度を

越えて大きければ移動は逆効果となる。そこでその境界に相当するものを定義し、実際に求めてみた。境界は、移動なしのアルゴリズムの場合の通信量のグラフとのある種の差(関数値の差の積分値の絶対値)が最小となる場合のエージェントサイズとして定義する。これはエージェント数に依存して図 4.4 のようにほぼ線形になる。

AgentSize	Agent数				
	30	60	90	120	150
70	200	300	400	500	

表 4.2 境界の AgentSize

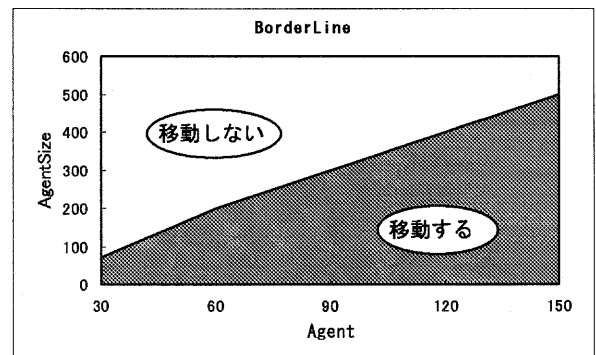


図 4.4 移動の境界線

5. おわりに

今回の実験より、移動の境界と考えたエージェントサイズよりエージェントサイズが小さいとき、メッセージ量が削減され大きいときには全体的に多く通信量がかかるということを得た。また、より難しい問題(通信量の多くかかる問題)が移動により通信量を下げることができるとも得た。

今後の課題として、さらに通信量を減らす移動方法を検討したい。また、1つのホストに多数のエージェントが集中すると、CPUタイムが各エージェントに分配されるので、1エージェント当りの計算スピードが減少する問題があるのでそれもモデルに取り入れたい。

さらに、実際にモバイル・エージェント・システム“Telepathy”上で実装を行い、実環境レベルでの移動に関する実験をしたい。

参考文献

- [1] 横尾真、平山勝敏：“分散 breakout : 反復改善型分散制約充足アルゴリズム”, 情報処理学会論文誌, Vol.14, No.6, pp.1889-1897 (1998)