3A−1

自律分散的に秩序形成を行うロボット群の並列計算機上でのシミュレーション

澤田政宏　田邉慎一　吉田紀彦
(九州大学 工学部 情報工学科)

1. はじめに

我々は，並列／分散人工知能の研究の一環として，ロボットの群が集権的な制御機構なしに自律分散的に全体として1つのシステムを構成するような，いわゆる群ロボットの研究を進めている．特に研究の焦点は，そのようなシステムにおいて必要となる実行主体の構成やそれらの間の相互作用形態の解明にある．

これまでに，比較的能力の高いロボットから構成されるシステムについて主にシミュレーションを通じて実験・分析を進め，その基盤となるような計算モデルを提案してきた[1]．一方で最近になって，人工生物（Artificial Life）や包括アーキテクチャ（subsumption architecture）に代表されるような，多数の単純なロボットから構成されるシステムの研究が各所で活発化している．しかし，我々の計算モデルがそのようなシステムに対しても有効であるかどうかは明確でなく，検証が必要である．

本研究では，多数の微小なロボットから構成される単純なシステムについて高並列計算機上でシミュレーションを行って，必要な通信機構の分析を行った．具体的に取り上げたシステムは，ランダムな初期状態から自律分散的に秩序形成を行う簡単な群ロボットである．さらにその成果もふまえて，群ロボット・システムに要求される通信機構をいかに実現するか，我々の計算モデルがどこまで寄与できるかについても検討を進めている．

本稿では，以下，第2章でここで取り上げた自律的な秩序形成，具体的には円周を構成するアルゴリズムを紹介し，第3章でそのアルゴリズムを高並列計算機上で実現する方法を述べる．次いで，第4章で一般にそのような群ロボット・システムを実現する上での通信機構について検討を加える．

2. 自律的に円周を構成する群ロボットのアルゴリズム

ここで例題として取り上げるのは，無秩序な初期状態からロボットどうしが互いに情報を交換しつつ，中央集権的な制御機構なしに自律分散的に円周という秩序的な形を形成するアルゴリズムであり，ミルウォーキー大の鈴木によって考案された[2]．この概要は次の通りである．

i) (最初，各ロボットは最終的な円の直径を知っている)．
ii) 自分と他のロボットとの距離を求め，誰が最遠ロボットおよび最近ロボットなのかを知る．
iii) 最遠ロボットとの距離が円の直径より長い場合は，最遠ロボットに少し近付く．
iv) 最遠ロボットとの距離が円の直径よりある割合だけ短い場合は，最遠ロボットから少し遠ざかる．
v) 最遠ロボットとの距離が円の直径にほぼ等しい場合には，最近ロボットから少し遠ざかる．
vi) すべてのロボットがii)～v)を繰り返し，自分がv)の状態になったとすべてのロボットが判断したら終了する．

3. 高並列計算機上での実現

ここでは，前章で述べたアルゴリズムを株式会社富士通研究所の高並列計算機AP1000上で実現した．AP1000は多数のプロセッサ・エレメント(セル)を2次元正方格子上で結合したアレイプロセッサ計算機であり，現在の構成ではセルプロセッサはSparc，最大構成は1024台となっている．

AP1000のプログラミングはライブラリ関数を用いて行い，CないしFortranのインタフェイスが用意されている．AP1000上の実際のプログラムは，ホスト計算機上で走るホストプログラムとセルプロセッサ上で走るセルプログラムから構成される．

前章で示した自律的な整列を行う群ロボットのアルゴリズムについて，これをAP1000上で実現するためのホストプログラムとセルプログラムの構造を，以下に示す．

[ホストプログラム]
初期状態（円周の直径と各ロボットの最初の位置）を設定し，各々のセルプロセッサにそれを送信する．
以下を繰り返す．
　各々のセルプロセッサからロボットの位置を受信して，出力する．
終了条件を判定し，満していれば最終結果を出力して終了．

[セルプログラム]
初期状態を受信する．
以下を繰り返す．
　他のすべてのセルプロセッサとの間でロボットの位置を互いに交換する
　(すべてのプロセッサから他のすべてに向けて放送する)
　最遠ロボットとそれへの距離，最近ロボットとそれへの距離を求める．
　(アルゴリズムを述べたように）状況に応じて自分のロボットを動かす．
　適当な繰り返し回数ごとに，ホストに自分のロボットの位置を送信する．
終了条件を判定し，満していれば終了．

Highly-Parallel Simulation of Self-Organizing Autonomous Decentralized Robotic Systems.
Masahiro SAWADA, Shinichi TANABE, and Norihiko YOSHIDA
Kyushu University

このプログラムの実行結果の例を図1に示す．これはロボットを点と見做してLaTeXで出力したものである．なお，この例では円が多少歪んでいるが，終了条件を厳しくすれば正しい円に近づく．

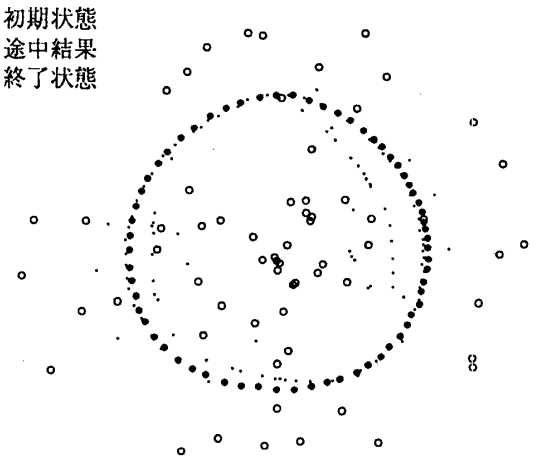

図 1: 円周上に並ぶ点の初期状態，途中結果及び終了状態

4. パターン照合通信の実現

上で述べた簡単なシミュレーション結果でも明らかなように，自律分散的な群ロボット・システムにおいては，通常の並列／分散処理システムにみられるようなメッセージの送受信とは異なる通信機構が必要となる．具体的には，ロボットAからある特定のロボットBに向けてメッセージを送るのではなく，逆にロボットAがメッセージを掲示してそれを必要とするいずれかのロボットがそれを受け取るという形の通信機構である．この通信機構は，受け取り手（の候補者）は必要とするメッセージのパターンを示してメッセージを受け取るのでパターン照合通信とでも呼ぶべきものである．

このパターン照合通信は，一般に並列／分散人工知能システムにおいても黒板システムのような形で利用されており，また群知能ロボットの分野でもこれを採用してシミュレーションを行った例が見られる [3]．冒頭で触れた我々がすでに提案している協調型問題解決のための計算モデルも，この通信機構を基盤の1つに据えている．ＡＰ１０００はこの形の通信機構を用意しておらず，また「掲示板」として利用できる共有資源も持たないため，上のプログラムではこの掲示処理を放送で代用し，受信者の側で受け取ったすべてのメッセージの内から必要なものを取捨選択している．したがって，無駄なメッセージの量が多い．

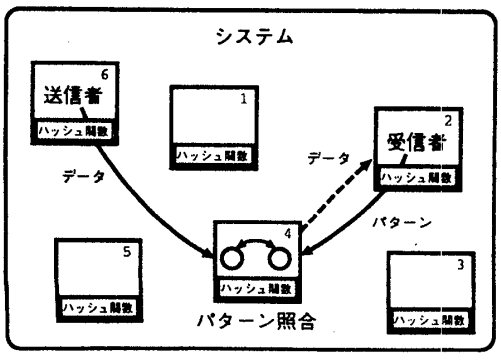

図 2: 自律分散環境におけるパターン照合通信の実現

本研究では上の実験結果なども踏まえて，自律分散的な群ロボット・システムのような共有資源のない状況でパターン照合通信をいかに効率的に実現するかについて検討を行った．

自律分散環境でのパターン照合通信の実現にあたって最も留意すべきことは，負荷の集中の回避と通信量の低減である．したがって，共有資源のサーバを用意する，すべてのメンバに共有資源の同一のコピーを持たせる，などの簡単な方法は適当でない．「掲示板」を何らかの方法で論理的に分割してすべてのメンバ（群ロボットであれば各ロボット，超並列計算機であれば各プロセッサ）に配分する必要がある．

ここでは掲示板を分割するために，すべてのメンバに番号を付与しておいて，メッセージ内容のハッシュ値に基づいてメッセージを配分するという方式を採用した．すなわち，メッセージのデータ，パターンともそのその内容をハッシュして，得られる値に等しい番号のメンバにそれを渡す．そして，データとパターンの照合処理はそれぞれのメンバの上で個別に行う．この概要を図2に示す．この方式では，コピーの存在による一貫性の問題も発生しない．なお，大きなメッセージについては，メッセージそのものではなくその所在だけを示す代理メッセージを渡すことによって，さらに通信量を低減することができる．

5. おわりに

現在は，群ロボットのシミュレーション実験を一段落させ，第4章で述べたパターン照合通信の処理系を実際にＡＰ１０００上に作成しつつある．今後は，まずこの処理系を完成させ，これを用いてさらに群ロボットのシミュレーション実験やより一般的な自律分散システムや分散人工知能システムの研究を進め，ここで提案した方式の有効性を検証していく．なお，本研究は，これまで科学技術計算や画像処理，グラフィクスなどに利用されてきた高並列ないし超並列計算機について，人工知能やロボティクスという新たな応用を提案するものでもある．

参考文献

[1] N. Yoshida, "A Modeling Framework for Group Behaviors and Its Application to Cooperative Problem Solving", Proc. Int'l Symp. on Autonomous Decentralized Systems, (1993) to appear.

[2] 山下雅史, 私信.

[3] J. Wang, "Deadlock Detection and resolution in Distributed Robotic Systems", Proc. Int'l Symp on Distributed Autonomous Robotic Systems (1992) 93-101.