

黑板モデルを用いた分散 CLP の協調

1 J-3

黒岩 篤 柴田 和人 岸 義樹

茨城大学 工学部

1.はじめに

近年、並列・分散方式に依拠した協調処理が盛んに研究され現在に至っている。しかし、並列・分散方式によって実際の問題を協調的に解く [1,2,3] に当たっては、各プロセスの管理、問題の分割、通信の方式、同期の取り方、そしてプロセスにより生成される案や解候補の選択、あるいは協調の仕方が整理されておらず、完全なモデルは未だに実現されていないのが現状と考えられる。

そこで本研究では黑板型処理モデルを用いた分散 CLP (*Constraint Logic Programming*) の協調処理 [4] において、これを論理型言語である Prolog によって実現した分散協調処理の一モデルを提案し、実際の覆面算 [5] に対しこのモデルの適用を試みている。

2. Prolog の分散実行環境

本研究は、UNIX の LAN 環境における分散処理を前提にしており、分散して動作するプロセス（エージェントと呼ぶ）と黑板となるプロセス（マネージャ）間との通信には、バークレー版 UNIX (Sun OS など) に実装されているソケットインタフェース [6] を用いる。このプログラムは C 言語で書かれるため、C 言語とのインタフェースを持った Prolog（本研究では K-Prolog）の使用を前提としている。

処理の流れとしては、まず解くべき問題を与えられたマネージャが、問題を解く上で必要なエージェント数を決定する。そして “rsh” コマンドを実行することによって、各マシン上でコマンドファイルを起動させて、自動的にプログラムをロードし、エージェントを生成する。

生成されたエージェントには、マネージャにより分割され割り当てられた問題や、協調された解候補が Prolog に組み込んだソケットインタフェースを介して与えられ、処理を開始する。

3. 分散 CLP

CLP においては、制約を論理で表わした上で、その制約を満たす変数値を決定する。

本研究で用いる分散 CLP は、適当な粒度に分割された制約をそれぞれ 1 つのエージェントに割り当てる。各エージェントに割り当てられた制約中の変数は、1 つのエージェント内でしか束縛されていない。従って、他のエージェント中に存在する（マネージャが識別している）同一の使い方をされる変数との間で可能な解候補を協調的に発見しなければならない。協調は次節で述べる黑板モデルで行うが、協調によって値が決定された変数の組み合わせは新たな解候補となり、1 つの制約として全エージェントの評価対象となる。全エージェントにおいて変数に値が定まったとき、制約は充足される。また、全解探索は協調した解候補を 1 つずつエージェントに送信することで可能となる。

4. 黑板モデルによる協調

本研究で用いる黑板モデルは、NFS を用いたものとは異なり、データを前述のソケットインタフェースを用いて送受信する。ソケットにより全エージェントに送信することで黑板モデルを実現する（図 1）ことができる。

本研究で提案する方法では、マネージャにより分割された問題をソケットを介して送信し、その問題についての評価を各エージェントにより行なわせる。そして結果をマネージャに送り、それらの結果を協調することで解候補を生成する。こうして生成された新たな解候補を黑板に書いて（ソケットにより全てのエージェントにフィードバックして）解候補の評価を行わせ、最終的な解（真の解）を求めて行く。この際、エージェントは与えられた制約を評価するために必要な条件が揃うまで wait 状態を保つ

ている。そして全エージェントによる評価がすべて制約を満たしている場合、その解候補を真の解とする。

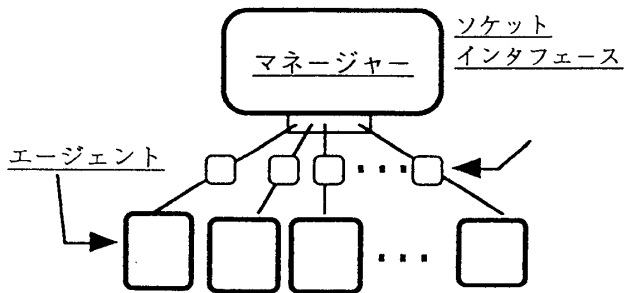


図1 黒板モデル

5. 覆面算による検証

これまで述べてきたモデルを、本研究では覆面算（図2）に対して適用し、有効性の検証を試みる。この検証に用いる覆面算は桁上がりなしという仮定のもとに、各桁の計算を分散プロセスとし、それぞれをマネージャにより1つのエージェントに割り当て（図3）ソケットを介し送信する。

$$\begin{array}{r} 3ABC \\ +) C43A \\ \hline ADDE \end{array}$$

[[3, A, B, C], [C, 4, 3, A], [A, D, D, E]]

図2 覆面算とその入力

- [3, C, A] ... 分散プロセス4 ⇒ エージェント4
- [A, 4, D] ... 分散プロセス3 ⇒ エージェント3
- [B, 3, D] ... 分散プロセス2 ⇒ エージェント2
- [C, A, E] ... 分散プロセス1 ⇒ エージェント1

図3 問題の分割

それぞれのエージェントが自律的に生成した案は、マネージャに送られ協調が図られる（図4）。生成した案は、変数リストを先頭にした、（変数リストとユニフィケーションできる）全ての解候補リストを要素とするリストである。そして協調の結果として生成される新たな解候補を各エージェントにフィードバックする。

エージェントはフィードバックされた解候補が割

り当てられた制約を満たしているかの評価、あるいは新たな案の生成を行ない、その結果を再びマネージャに送り返す。その際、制約を満たしている場合には“[ok]”を返す。これらの処理を繰り返し、全てのエージェントから“[ok]”が返ってきた解候補は真の解とみなされる。

[[C,A], [0,3], [1,4], [2,5], [3,6], [4,7], [5,8], [6,9]]
... エージェント4からの案
[[A,D], [0,4], [1,5], [2,6], [3,7], [4,8], [5,9]]
... エージェント3からの案
[[B,D], [0,3], [1,4], [2,5], [3,6], [4,7], [5,8], [6,9]]
... エージェント2からの案
[wait]
... エージェント1からの案

[[C,A,D,B], [0,3,7,4], [1,4,8,5], [2,5,9,6]]
... 協調の結果（新たな解候補）

図4 マネージャによる協調

6. おわりに

本研究では、複数のエージェント上でCLPを分散実行させ、各エージェントから得られる結果の協調を図って解を求めるモデルを提案し、覆面算へこのモデルの適用を試みた。その結果、本モデルの有効性を確認することができた。

今後はこのモデルを改良し、より現実的な問題に対し適応できるようにして行く予定である。

参考文献

- [1] 田中: “YY-server を中心とする協調分散処理の機構”, マルチエージェントと協調計算 I (中島 編), 近代科学社 (1992), pp.165-177.
- [2] 岸, 山下: “Parallel Prolog による問題解決の一手法”, 1989年度人工知能学会全国大会, pp.303-306.
- [3] 矢吹, 益田: “ソケットインタフェースを利用した Prolog 応用プログラムの並列処理”, INAP'93, pp.5-12.
- [4] M.Henz, G.Smolka, J.Würtz: “Oz - A Programming Language for Multi-Agent Systems”, IJCAI'93, pp.404-409.
- [5] P.R.Cohen, E.A.Feigenbaum: “The Handbook of Artificial Intelligence (Vol.3)” PITMAN BOOKS Limited (1982), pp.11-14.
- [6] 雪田修一 著: UNIX ネットワークベストプログラミング入門, 技術評論社 (1992).