

分散型制約充足アルゴリズムの一評価

3P-2

福本 晃 北村泰彦 辰巳昭治 奥本隆昭
大阪市立大学 工学部

1 はじめに

分散協調問題解決では、複数のエージェントが協調的に動作することによって問題解決が行なわれる。これらのエージェントの組織化や問題解決アルゴリズムに関して、これまでにいくつかの方式が提案されている。しかし、これらの方式を統一的な基準によりその効率を評価した例は少ない。本稿では、一般的な分散型問題の一つである分散制約充足問題 [1] を取り挙げ、これに対する解法として分散集中型と分散協調型のアルゴリズムを示す。また、階層構造を持つエージェントの組織化(階層型)の効果についても述べる。さらに、制約充足問題として知られる 8-queens 問題を用いてそれぞれの手法の解決時間、稼働率、通信量を測定し、その評価を行う。

2 分散制約充足問題の定式化

まず、制約充足問題は一般に次の形で定義される [1].

n 個の領域 D_1, \dots, D_n があり、各 $D_i (i=1, \dots, n)$ は、変数 x_i のとりうる値から成り立っている。また、変数間の制約 $C(x_{k_1}, \dots, x_{k_j})$ が存在し、関係する各変数の領域の直積空間に対して真偽が定義されている。この時、 $C(x_1, \dots, x_n)$ が真となる変数の値の組を見つけることが制約充足問題の解を求めることである。本稿では制約 $C(x_{k_1}, \dots, x_{k_j})$ が偽の場合に、 $m > j$ なる全ての整数 m に対して必ず $C(x_{k_1}, \dots, x_{k_j}, \dots, x_{k_m})$ が偽となる性質を持つものと仮定している。分散制約充足問題とは、これらの変数領域がエージェントに分散して存在しており、解を求めるためにはエージェント間での協調の必要な問題である。

3 分散制約充足問題の解法

本稿では、制約充足問題の解法の一つである、バックトラッキング法を取りあげ、これを分散制約充足問題に拡張する。ここで、エージェントモデルとして次のような仮定をおく。(1) 集中型を除き、複数のエージェントが存在し、各エージェントに1つの変数の割り当てのみが可能である。(便宜上、変数の添字をエージェント番号とする。)(2) 任意の2つのエージェント間で、メッセージ通信により通信が可能である。

3.1 集中型 (C)

1台のエージェントで問題を解く従来の方式である。 $d_1 \in D_1, d_2 \in D_2, \dots$ の順に値を決定して、 $C(d_1, d_2, \dots)$ が偽になると最近の値の決定を取り消して別の値を割り当てる。割り当てべき値がなくなった場合には一つ前の変数に遡り、値の割り当てをやり直す。このようにして、 $C(d_1, \dots, d_n)$ が真となるものが見つければ終了する。 D_1 から新たに割り当てべき値がない場合は、解は存在しないとして終了する。

3.2 分散集中型 (DC)

スーパーバイザとなるエージェントを1つ決める。このエージェントは集中型に準ずる動作を行なう。他のエージェントに割り当てられている変数の値を用いる時には、そのエージェントに値を送信するように依頼する。値の送信依頼を受けたエージェントは、未送信の値をスーパーバイザに送信する。送信されてきた値は、スーパーバイザのローカルメモリに記憶しておく。未送信の値がない場合にはそのことをスーパーバイザに知らせる。スーパーバイザは、あるエージェントから変数のすべての値を得た後には、エージェント間通信を行なわずにメモリから直接、値を得る。

3.3 分散協調型 (DCo)

それぞれのエージェントが、部分的な制約を真にするように割り当てられた変数の値を決めて、制約関係のあるエージェントすべてにその値を送信する。ただし、エージェント番号の小さいものから大きなものにだけ送信するものとする。また $1, \dots, (i-1)$ 番目までの変数の値が確定しているとき、エージェント i はカレントエージェントであるという。この情報も“change-current”メッセージにより知らせる。

他のエージェントから値を受け取ったエージェントは、制約が真のままであるかどうか調べる。制約が偽になる場合、変数の値の決定をやり直し、新しい値を再び関係のあるエージェントすべてに送信する。新しい値がない場合は“backtrack”メッセージを送る。ただし、“backtrack”メッセージはカレントエージェントだけが発信できるので、カレントエージェントでない場合は、カレントエージェントになるまで待つ。これによって、アルゴリズムの完全性が保証される。

3.4 階層型 (H)

エージェントをいくつかのエージェント群に分割して問題解決を行なう。1つのエージェント群は分散集中型あるいは分散協調型の方式によって部分解を構成する。

An Evaluation of Distributed Constrained Satisfaction Algorithms

Akira Fukumoto, Yasuhiko Kitamura, Shoji Tatsumi and Takaaki Okumoto

Osaka City University

3-3-138, Sugimoto, Sumiyoshi-ku, Osaka, 558, Japan

また、エージェント群の間のメッセージの送信、受信がそれぞれ特定のエージェント間で行なわれる。これによって、1つのエージェント群を1つのエージェントと見なしてさらに階層化を進めることができる。

4 解法の評価

4.1 評価基準

前節で紹介した各種の解法を適用して、制約充足問題の一つである 8-queens 問題を解き、それらの効率を評価する。通信路は 1 本の待ち行列を持つサーバによりモデル化され、1つのメッセージを処理する時間(通信コスト)は一定であると仮定している。

評価の尺度として、(1)問題を解くのに必要な解決時間 (2)エージェントの稼働率(3)全メッセージ数を用いた。ただし、解決時間は制約チェックにかかる時間を 1 単位時間として計算し、計算時間と通信時間を含んでいる。稼働率はエージェント毎に、稼働時間を解決時間で割ることにより求められる。

4.2 シミュレーション結果と考察

シミュレーション結果を図 1, 2 および表 1 に示す。階層型は分散集中型で構成した 4 エージェントの群 2 組による組織化を採用した。エージェント 1~4 の群で部分解を構成した後、エージェント 5~8 の群に送信する。エージェント 4 とエージェント 8 がそれぞれのグループのスーパーバイザとなっており、さらにそのエージェント間で通信が行われる。

この実験結果から明かなことは分散協調型の通信オーバーヘッドは極めて大きいという点である。これは三つの手法の中で最もメッセージ数は大きくなっており、通信コストが大きくなるにつれ急激に性能が低下する。他の手法に比べて各エージェントの稼働率は高いが、制約情報の部分性により、部分解が他のエージェントからの新たな変数割り当てによって制約を満たさなくなり、新たな割り当てを行わなくてはならない場合が多い。

次に分散集中型ではほとんどスーパーバイザしか動作しておらず、並列性は低い。ただし、通信コストに関係なくメッセージ数が一定なので、通信オーバーヘッドによる性能の悪化は穏やかである。

階層型はちょうど上記の二つの手法の中間的な性質を示す。すなわち、エージェント 1~4 の群が部分解を送信した後も、新たな部分解を見つけるために解決を続けるので、並列性が高くなっている。通信量の増加は分散集中型に比べて 40%程度であり、通信コストが低い間は解決時間は三つの手法の中で最もよい。ただし、通信コストが大きくなるにつれ通信オーバーヘッドが大きくなり、分散集中型と逆転する。

以上の結果を総合すると、並列性を上げ、通信オーバーヘッドを小さくする上で組織化の効果が大きいことが明

らかになった。

5 まとめ

分散制約充足問題について、集中型、分散集中型、分散協調型、階層型の各手法を示し、8-queens 問題を用いたシミュレーション実験により各手法の比較を行った。その結果、適切な組織化を行うことにより全体性能が向上することが明らかになった。今後の課題としては、異なる性質を持つ問題や様々な組織化に関する評価を行うことが挙げられる。

参考文献

- [1] 横尾他: “分散制約充足による分散協調問題解決の定式化とその解法”, 信学論, Vol. J75-D-I, No.8, pp.704-713.(1992)

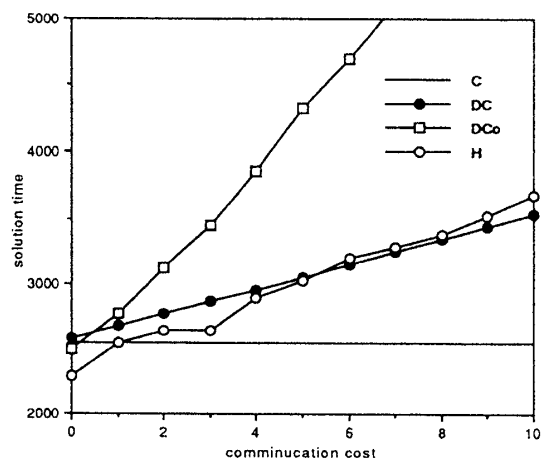


図 1: 解決時間による各手法の比較

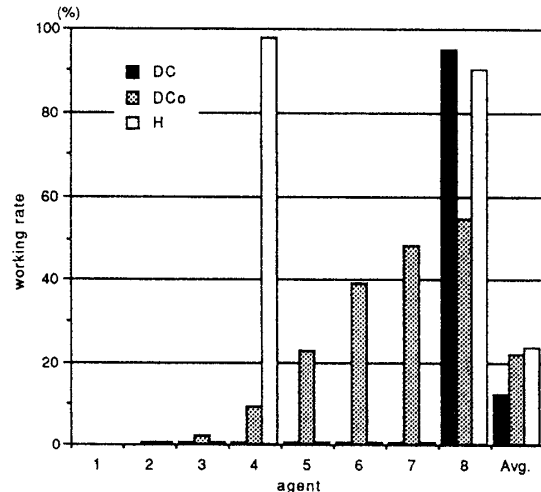


図 2: 稼働率による各手法の比較

表 1: 通信量による各手法の比較

手法 \ 通信コスト	0	1	2	...	10
分散集中型	95	95	95	...	95
分散協調型	377	428	441	...	467
階層型	133	135	135	...	137