

## アクティブ情報統合のための動的分散制約充足プロトコル

平山 勝敏  
神戸商船大学  
hirayama@ti.kshosen.ac.jp

北村 泰彦  
大阪市立大学大学院工学研究科  
kitamura@info.eng.osaka-cu.ac.jp

### 概要

本論文では、Web上の情報を収集統合する Intelligent Ticker システムの概要を述べ、このシステムの構成要素である情報統合エージェント間で用いられる動的分散制約充足プロトコルについて述べる。本システムは、それぞれ複数の情報統合エージェントと情報収集エージェントからなり、情報統合エージェントは情報収集エージェントが収集する情報を統合する。その際、利用者から与えられるタスクを達成するためにどの情報収集エージェントにどの Web 情報源を監視させるべきかに関して情報統合エージェント間で合意を形成する必要がある。ここでは、この問題を動的分散制約充足問題として定式化し、これを解くためのプロトコルを提案する。

### Dynamic Distributed Constraint Satisfaction Protocol for Active Information Integration

Katsutoshi Hirayama  
Kobe University of Mercantile Marine

Yasuhiko Kitamura  
Graduate School of Engineering, Osaka City  
University

### Abstract

We present a dynamic distributed constraint satisfaction protocol that can be used to resolve conflicts among agents of the Intelligent Ticker System, which can gather and integrate various information on the Web. This paper briefly describes the overall system, identifies a conflict resolution problem in the system, shows how to map the problem into a dynamic distributed constraint satisfaction problem, and presents a general protocol for the dynamic distributed constraint satisfaction problem.

#### 1. アクティブ情報収集・統合システム: Intelligent Ticker システム

##### 1.1 要求される機能

アクティブ情報収集とは、インターネットのような動的で大規模な情報源から

利用者に対して必要な情報を監視・提供することをいう。これを実現するアクティブ情報収集システムには次のような機能が求められる。

**情報監視機能:** 頻繁に更新される情報源

に対してその変化を監視し，変化に応じた何らかの動作を行う機能．ある Web ページを監視し，その変更点を表示する AIDE システム[3]，データウェアハウスにおける情報源の変化を検知し，そのメンテナンスを行う DataFoundry システム [1,2]には情報監視機能を実現するモジュールが組み込まれている．

**差分表示機能：**情報源が変化したときに，どのように変化したか利用者に対してわかりやすく表示する機能．上述の AIDE システムには HtmlDiff という 2 つの Web ページの違いを際立たせて表示するモジュールが組み込まれている．

**評価機能：**同様の情報を扱う複数の情報源の中から，さまざまな要因を考慮しながら適切な情報を選択する機能．

**統合機能：**複数の情報源から得られる情報を統合する機能．

## 1.2 Intelligent Ticker システム

頻繁に更新される情報源からの情報収集を考えた場合，一般に更新されるそれぞれの情報の量はそれほど大きくはないと考えられる．我々はこのように少量で速報性のある情報オブジェクトを Ticker とよび，それらを Web 情報源から収集して統合することで利用者の意思決定や問題解決を支援する Intelligent Ticker システムを提案する．本システムは情報収集エージェントと情報統合エージェントという 2 種類のエージェント群からなるマルチエージェントシステムとして構成される．

### (1) 情報収集エージェント

情報収集エージェントは定期的にある特定の Web ページにアクセスして Web 情報源を監視し，その変化に対して

Ticker を生成する．Ticker はオブジェクト，タイムスタンプ等からなり，オブジェクトは更新された情報の断片そのもの，タイムスタンプはそれが更新された時刻を表す．

### (2) 情報統合エージェント

情報収集エージェントは 1 つの Web 情報源を監視し変化が生じたときに Ticker を生成する．一方，情報統合エージェントは，利用者から与えられるタスクに応じて複数の情報収集エージェントからの Ticker を統合する．

## 1.3 情報収集エージェント数の最適決定

情報収集エージェントは 1 つの Web 情報源を監視し続けるため，通信資源を消費する．よって情報収集エージェントを多数用意して多くの Web 情報源を監視させた場合，システムの通信コストは大きくなる．

一方，情報統合エージェントは利用者から与えられるタスクに応じて Ticker を集めて統合する必要があるため，十分な数の情報収集エージェントがないとタスクそのものを実行することができない．

以上より，利用者から与えられたタスクを実行するのに最低限必要な情報収集エージェントの数はいくらかという最小化問題を解くことは非常に有益だと考えられる．本論文では，この問題への最初の取り組みとして，利用者から与えられたタスクをある一定の数の情報収集エージェントで実行可能かどうか判定する決定問題を解くことを目指す．情報収集エージェントの数を変えながらこのような決定問題を繰り返し解くことにより，先の最小化問題の解を得ることができる．

この決定問題を解くにあたり次のことを仮定する。

- 情報収集エージェントは、有限個の Web 情報源の中から 1 つだけ選択し、それに関する Ticker を生成する。
- 情報収集エージェントはある一定の数だけ存在する。
- 情報統合エージェントは、利用者から与えられるタスクを実行するために、どういう状況でどの情報収集エージェントにどの Web 情報源を監視してもらいたいかを知っている。
- 一般に情報統合エージェントは複数存在し、各情報統合エージェントは異なるタスクを担当する。それらのタスクは複数の利用者由来し、かつ、利用者が公開したくない情報を含む。

以上のように仮定した上で、次節では、この決定問題を動的分散制約充足問題 [7] として定式化する。

## 2. 動的分散制約充足問題としての定式化

### 2.1 動的分散制約充足問題

制約充足問題とは、有限かつ離散的な値域をもつ変数の集合と変数間に定義された制約の集合からなり、すべての制約を満たす変数への値の割り当てを求める問題である。

一方、分散制約充足問題 [10] とは、制約充足問題における変数と制約が複数のエージェントに分散された問題である。マルチエージェントシステムにおける多くの問題は分散制約充足問題として定式

化可能である。そのため、近年、分散制約充足問題を解く分散アルゴリズムがいくつか提案されている [8]。

動的分散制約充足問題 [7] とは、外部環境の状態を表す変数（環境変数）の値に応じて制約が追加あるいは削除されるという設定を分散制約充足問題に導入したものである。 [7] では分散制約充足問題における制約を  $(p, c)$  の 2 つ組で表し、環境変数  $p$  が真のとき制約  $c$  を追加し、 $p$  が偽のとき制約  $c$  を削除するよう分散制約充足問題の枠組みが拡張されている。

### 2.2 定式化

1.3 節で説明した決定問題は次のようにして動的分散制約充足問題として定式化することができる。

- 1 つの情報収集エージェントに対し 1 つの変数  $x$  を対応させる。変数  $x$  の値域  $D$  は、対応する情報収集エージェントが選択可能な Web 情報源の集合であり、変数の値  $d$  は、対応する情報収集エージェントが選択する 1 つの Web 情報源である。
- 情報統合エージェントが利用者から与えられたタスクを実行するために情報収集エージェントに監視してもらいたい Web 情報源の組み合わせを制約  $c$  とする。なお、どの Web 情報源を監視すべきかは状況に依存するため、状況を表現する環境変数  $p$  を導入し、制約を  $(p, c)$  の 2 つ組で表現する。恒真の論理定数 true に対し、 $(\text{true}, c)$  は  $c$  が任意の状況で満足されなければならないことを意味する。
- 情報統合エージェントを動的分散制約充足問題を構成するエージェントとする。情報統合エージェントは他

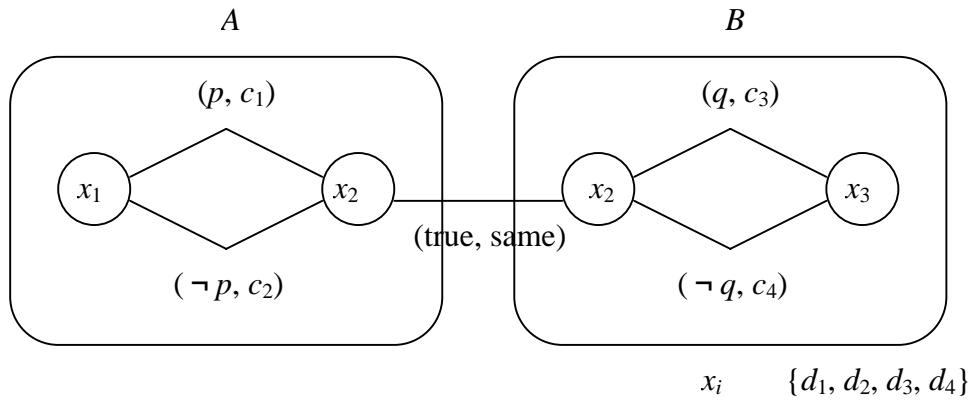


図 1: 動的分散制約充足問題の例

の情報統合エージェントと後述のプロトコルを用いて通信を行い, どの情報収集エージェントにどの Web 情報源を割り当てるかに関して合意を形成する.

簡単な例題を示す. 2 つの情報統合エージェント  $A, B$ , 3 つの情報収集エージェント  $x_1, x_2, x_3$  がいて,

- 情報統合エージェント  $A$  は, Web 情報源  $d_1$  がある状態(環境変数  $p$  が真)のとき, 情報収集エージェント  $x_1$  には Web 情報源  $d_1$  を, 情報収集エージェント  $x_2$  には Web 情報源  $d_2$  を監視させたい. なお, この制約を  $c_1$  と表し,  $x_1=d_1$  かつ  $x_2=d_2$  のとき, かつ, そのときに限り  $c_1$  は真になるものとする(以下同様). 一方, Web 情報源  $d_1$  がある状態にない(環境変数  $p$  が偽)のとき, 情報収集エージェント  $x_1$  には Web 情報源  $d_3$  を, 情報収集エージェント  $x_2$  には Web 情報源  $d_4$  を監視させたい(制約  $c_2$ ).
- 情報統合エージェント  $B$  は, Web 情報源  $d_3$  がある状態(環境変数  $q$  が真)のとき, 情報収集エージェント  $x_2$  に

は Web 情報源  $d_2$  を, 情報収集エージェント  $x_3$  には Web 情報源  $d_3$  を監視させたい(制約  $c_3$ ) が, Web 情報源  $d_3$  がある状態にない(環境変数  $q$  が偽)のとき, 情報収集エージェント  $x_2$  には Web 情報源  $d_2$  を, 情報収集エージェント  $x_3$  には Web 情報源  $d_4$  を監視させたい(制約  $c_4$ ).

とする. この例題は図 1 に示す動的分散制約充足問題として定式化することができる. 図 1 でグラフの節点は変数, 枝は制約を表し, 制約(true, same)は常に値が同じであることを意味する. この例題には,  $p$  が真のときには  $q$  の値に関わらず解が存在するが,  $p$  が偽のときには解が存在せず, 3 つの情報収集エージェントだけではタスクを実行することができない.

### 2.3 プロトコル

動的分散制約充足問題を解くためのプロトコル(分散アルゴリズム)について考察する. まず, [7]で提案されているアルゴリズムでは, 通常分散制約充足問題を解くアルゴリズムの 1 つである非同期弱コミットメント探索アルゴリズム

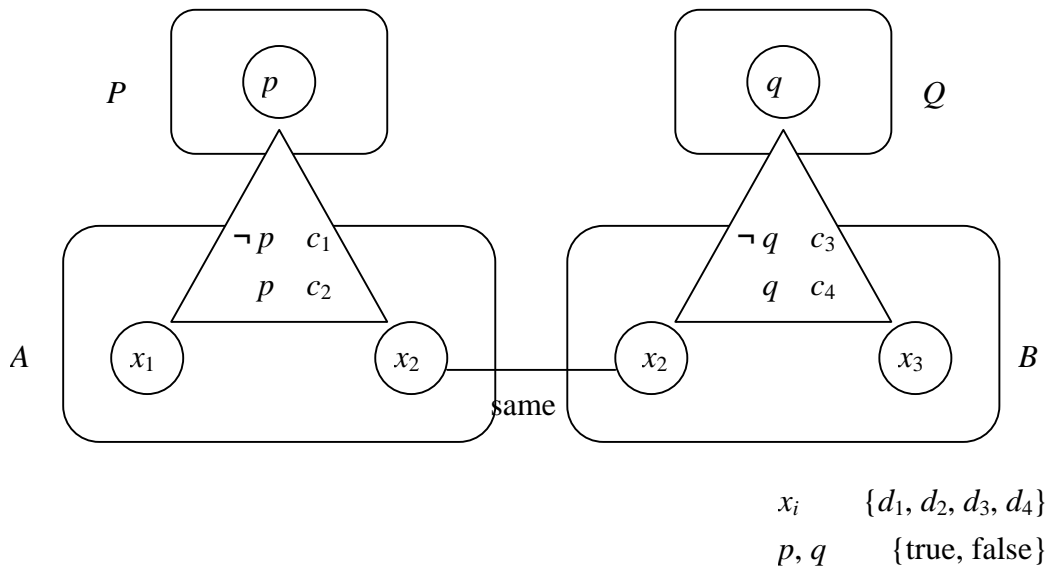


図 2: 分散制約充足問題への変換

[9]を動的分散制約充足問題用に拡張している。しかしながら，nogood（処理の過程で生成される新しい制約）の扱いに関する手続きに誤りがあるため，そのままの形で利用することはできない。本論文では，別のアプローチとして，動的分散制約充足問題を通常の分散制約充足問題に変換し，既存の分散制約充足アルゴリズムを適用することを提案する。

### (1) 分散制約充足問題への変換

2.2 節で説明した動的分散制約充足問題を次のようにして通常の分散制約充足問題に変換する。

- 各環境変数を 1 つの仮想的なエージェントに割り当てる。
- 制約 $(p, c)$ を $\neg p \quad c$ に置き換える。なお，これにより制約 $(\text{true}, c)$ は $c$ に置き換わる。

以上の操作により，例えば図 1 に示した動的分散制約充足問題は図 2 のような分散制約充足問題に変換される。図 2 では，環境変数  $p, q$  はそれぞれエージェント  $P,$

$Q$  に割り当てられ，エージェント  $A$  は  $P$  と制約 $(\neg p \quad c_1) (p \quad c_2)$ を共有し，エージェント  $B$  は  $Q$  と制約 $(\neg q \quad c_3) (q \quad c_4)$ を共有する。さらにエージェント  $A$  と  $B$  は互いの変数  $x_2$  の値が等しいという制約を共有する。

### (2) 分散制約充足アルゴリズムの適用

変換後の分散制約充足問題を解く際には基本的に既存の分散制約充足アルゴリズムを適用することができる。なお，一般に，ある情報統合エージェントは複数の情報収集エージェントに対応する複数の変数をもつため，1 エージェントが複数の変数をもつ場合に対応した分散制約充足アルゴリズム[5, 6, 9]を用いる必要がある。

現在，これらのアルゴリズムを前述の問題に適用することを検討中である。適用にあたっては，主にアルゴリズムの効率を上げるために手続きの細かい部分を若干修正する必要があると考えられる。例えば，図 2 の分散制約充足問題には，

環境変数のみをもつエージェント  $P, Q$  のように変数の値を自分では変更できないエージェントが存在する。詳細は省略するが、そのような場合、例えば[6, 9]のアルゴリズムではそのようなエージェントを越えて nogood を送信した方が明らかに効率がよい。

### 3. まとめと今後の課題

本論文では、Web 上の情報を収集統合する Intelligent Ticker システムの概要を述べ、情報統合エージェント間で用いられる動的分散制約充足プロトコルについて述べた。今後の課題としては、プロトコルの実現とその評価、および、このプロトコルを用いて最小化問題を解くこと等が挙げられる。後者については、[4]で提案されている手法を適用できると考えられる。

### 参考文献

[1] Adam, N., Adiwijaya, I., Critchlow, T., Musick, R.: Detecting Data and Schema Changes in Scientific Documents, *In IEEE Advances in Digital Library* (2000)

[2] Critchlow, T., Fidelis, K., Ganesh, M., Musick, R., Slezak, T.: DataFoundry: Information Management for Scientific Data, *IEEE Trans InfTechnol Biomed*, 4(1): 52-57 (2000)

[3] Douglis, F., Ball, T., Chen, Y.-F., Koutsofios, E.: The AT&T Internet Difference Engine: Tracking and Viewing Changes on the Web, *World Wide Web*, 1: 27-44 (1998)

[4] Hirayama, K. and Yokoo, M.: An Approach to Over-constrained Distributed Constraint Satisfaction Problems: Distributed Hierarchical Constraint Satisfaction, *Proc. of ICMAS-2000*, 135-142 (2000).

[5] Hirayama, K., Yokoo, M.: Local Search for Distributed SAT with Complex Local Problems, *Proc. of AAMAS-2002* (to appear) (2002)

[6] Hirayama, K., Yokoo, M., Sycara, K.: The Phase Transition in Distributed Constraint Satisfaction Problems: First Results, *Proc. of the International Workshop on Distributed Constraint Satisfaction* (2000)

[7] Modi, P. J., Jung, H., Tambe, M., Shen, W., Kulkarni, S.: A Dynamic Distributed Constraint Satisfaction Approach to Resource Allocation, *Proc. of CP-2001*, 685-700 (2001)

[8] Yokoo, M., Hirayama, K.: Algorithms for Distributed Constraint Satisfaction: A Review, *Autonomous Agents and Multi-Agent Systems*, 3(2): 189-212 (2000)

[9] 横尾真, 平山勝敏: 複雑な局所問題に対応する分散制約充足アルゴリズム, *人工知能学会誌*, 15(2): 348-354 (2000)

[10] Yokoo, M., Durfee, E. H., Ishida, T., Kuwabara, K.: The Distributed Constraint Satisfaction Problem: Formalization and Algorithms, *IEEE Trans. on Knowledge and Data Engineering*, 10(5): 673-685 (1998)