

分散制約充足問題の観測資源割当てへの適用

松井俊浩 松尾啓志 岩田彰

名古屋工業大学 電気情報工学科
〒 466-8555 名古屋市昭和区御器所町
tmatsui@mars.elcom.nitech.ac.jp

あらまし 複数カメラを用いた観測システムにおけるカメラの割当ての, 分散制約充足問題としてのモデル化を提案する. 提案手法では, 観測システムの目的を, 対象の位置の推定, およびその上に立脚する特定の処理として捉える. 各目的に応じた, 対象へのカメラの割り当て方を制約充足問題の枠組みで記述する. このような記述により, 様々な目的が一般化される. さらに, 変数が各カメラノードに分散した分散制約充足問題としてモデル化し, 分散制約充足問題のための協調的なアルゴリズムによって解く. シミュレーションによる実験を行い, 提案手法の応用の可能性を示す.

キーワード カメラ割当て 分散制約充足問題 協調処理

The modeling of the camera allocation as distributed CSP

Toshihiro Matsui, Hiroshi Matsuo and Akira Iwata

Nagoya Institute of Technology
Gokiso-cho, Showa-ku, NAGOYA 466-8555, JAPAN
tmatsui@mars.elcom.nitech.ac.jp

Abstract Camera allocation for the observation system using the multiple cameras is modeled as a Distributed Constraint Satisfaction Problem (distributed CSP). The purposes of the observation system are considered as the estimation processing of the object position and the other specialized processing. The allocation of the cameras for objects in proportion to the purposes is described as constraints and variables. By the description, many purposes for the allocation are generalized. Then, the variable is modeled as Distributed CSP in which it is distributed for each camera node. The problem is solved by the cooperative algorithm for Distributed CSP. The result of the simulation shows the feasibility of the proposed modeling.

Keywords Camera allocation, Distributed CSP, Cooperative processing

1 はじめに

最近、複数のカメラを接続した計算機網を用いた観測システムが研究されている [1] [2] [3] [4] [5] [6] . このようなシステムは遠隔監視, 遠隔講義, 自動撮影などへの応用が期待される. 特に, パン, チルト等による視野の制御可能なカメラを用いたシステムは, 柔軟な撮影が可能である反面, その視野の制御の方法が課題となる.

視野の制御において考慮すべき目的として,

- 対象の位置の把握: カメラの視野を決定するためには, 対象の位置情報を得ることが必要である. 例えば, カメラが奥行き情報を得られない場合, 複数のカメラで対象を観測し, 対象の位置を推定しなければならない.
- 特定の目的のための観測: 特定の目的のためにカメラ資源を用いる. 例えば, 特定の対象を注視しなければならない.

があり, これらの目的を満足するようなカメラの割当て問題を解くことが必要となる.

幾つかのシステムでは各カメラを制御するための複数の計算機を用いることから, 協調的処理の枠組みによってカメラの制御を行う手法が提案されている. 例えば, 各カメラに明示的に大域的な観測や注視の役割を与える [4], 観測対象や特定のタスクごとに複数のカメラを組織化する [1] [3], などがある.

これらの手法は, 特定の目的のための状態遷移モデルと通信プロトコルに基づく. このような設計は, 処理速度の点で効率的である. しかし, 様々な目的に応じた状態遷移モデルとプロトコルの設計は煩雑である. とりわけ, 分散処理の設計は難しくなる傾向がある.

分散協調問題解決の枠組みとして, 分散制約充足問題 (分散 CSP) [7] が研究されている. 分散 CSP の目的は制約充足問題を分散処理により解決することである. 一方で, マルチエージェントシステムにおける理論的な基礎としても重要である.

本稿では, カメラ割当てを重み付き分散 CSP としてモデル化する. 問題を CSP としてモデル化することにより様々な目的を一般化できる. カメラ割当て問題を形式化することは, 協調処理手法を特化する上での基礎的検討としても有用であると考え.

以下ではまず, 観測システムのモデルを示す. そしてカメラ割り当てを重み付き分散 CSP としてモ

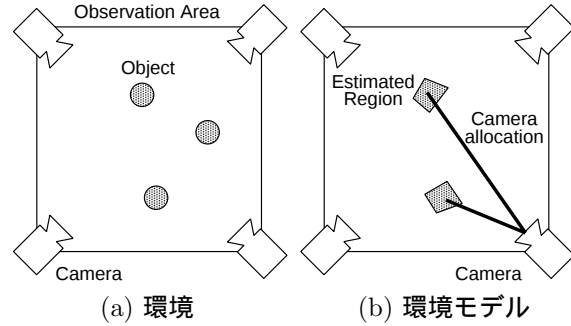


図 1: 観測システムのモデル

デル化する. 提案手法について, シミュレーションによる評価を示す.

2 観測システムの形式化

観測システムをモデル化する. 室内を移動する複数の人物の位置を, 複数のカメラにより観測するシステムを想定する. 本稿では初期の検討として簡単なモデルを用いる.

この観測システムは次の処理を反復する.

1. 各カメラノードは観測データを得る. 観測データは他のノードにも送信される.
2. 各ノードは観測データにより環境モデルを更新する.
3. 各ノードは協調的にカメラの割当てを決定する.
4. カメラの割当てに従って各カメラの観測方位を更新する

2.1 目標とカメラ

対象とカメラの配置を図 1(a) に示す. 環境は 2 次元平面としてモデル化される. また, 障害物は存在しないものとする.

対象は観測領域の内部を移動する. 対象の数は一定であるが, その数は未知である. 各対象の半径は定数である. 対象の移動する方向は予測できないが, 速度に上限 v を仮定する.

カメラは観測領域内の対象を観測する. カメラの数を定数 N^{cam} で表す. 各カメラの視点は固定される. カメラの視野は自由に変更できるものとする. しかし, 本稿では具体的なカメラの視野の制御では無く, 対象に対するカメラの割り当てのみを考

慮する.

カメラは, 各方向において対象が存在するか否か, を観測する. カメラ i の観測するデータは, 観測方位 θ に対して, $f_i(\theta) = \{0, 1\}$ により表される. 観測における誤差は考慮しない.

カメラは対象の距離を観測できない. また, カメラはそれぞれの対象を区別できない. 従って, 対象の存在する領域を推定するために, 複数のカメラからの観測データを統合しなければならない.

2.2 環境モデル

各カメラノードは環境モデルを持つ. 環境モデルは, 対象領域と, カメラ割当てから成る (図 1(b)). 対象領域は, 対象が存在すると推定された領域である. 観測システムは, 観測されたデータが得られる時に, 対象領域を推定する. (2.3 節および 2.4 節) そして, 対象領域に対するカメラ割当てを決定する.

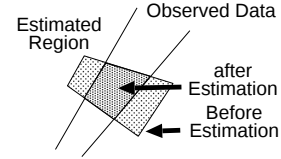
各対象領域 j に対するカメラ i の割り当ては, 変数 $x_{i,j}$ ($i = 0, \dots, N^{Cam} - 1, j = 0, \dots, N^{Rgn} - 1$) として表される. N^{Rgn} は対象領域の数である. 領域が選択されるならば, $x_{i,j}$ は 1 である. もしそうでなければ, $x_{i,j}$ は 0 である.

推定の結果, 対象領域は消去, 分割, 統合される場合がある. そのような場合, その領域に対する割り当ても, 同様に変更する. すなわち, 推定前の対象領域に対する割り当ては, 推定後の対象領域にも反映される.

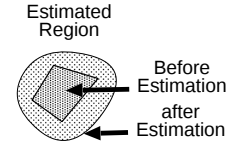
2.3 観測データによる推定

カメラにより観測されたデータが得られたときに, 対象領域を削減する. 観測されたデータは各方位における対象の有無を表す. そこで, 観測されたデータを 2 次元平面上に投影し, 対象領域との集合積を求める (図 2(a)).

各カメラの観測データについて, そのカメラが割当てられていない領域は削減されないものとする. 従って, 選択されない領域の方向は観測しなくてよい. このことにより, カメラの視野を限定することは可能である. しかし, 本稿ではカメラの視野の其他的な制御は考慮しない.



(a) 削減



(b) 増加

図 2: 対象領域 (Estimated region) に対する演算

2.4 対象の移動の推定

対象の速度は v 以内であると仮定した. そこで, 時間経過に応じて領域を増加することで, 対象の移動する範囲を推定する. (図 2(b)).

領域の増加と, 観測データによる領域の削減と均衡させることにより, 環境中の対象の位置の変化が対象領域に反映される.

3 観測資源割当ての制約充足問題としての形式化

カメラの割当ては, 対象の存在位置の推定と, それに立脚する特定の処理のための観測のトレードオフなど, 様々な条件によって決定される. これらの条件を正確に定式化し, さらに, 複雑な目的のための分散アルゴリズムの設計することは簡単ではない.

そこで, 問題を重み付き制約充足問題としてモデル化し, さらに問題を分散 CSP へ展開する.

3.1 重み付き分散 CSP としての定式化

まず, カメラ割当てを重み付き制約充足問題 (X, D, C, S, φ) [9] として表現する. X は変数の集合である. 対象領域 j に対するカメラ i の割り当てを変数 $x_{i,j} \in X$ で表す. D は変数に関連付けられた値域の集合である. $x_{i,j}$ は値域 $d_{i,j} \in D$ 中の値を取る. $d_{i,j} = \{0, 1\}$ である. C は制約の集合である. $c \in C$ は各変数のどの値が互いに整合するかを特定する.

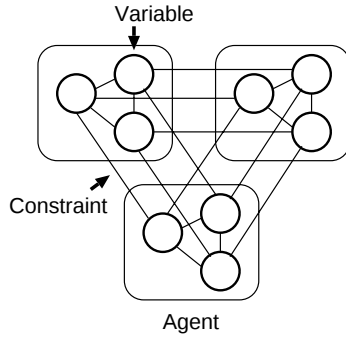


図 3: 分散 CSP の制約ネットワーク

S, φ は過制約な問題を許すために加えられる. $S = (E, \otimes, \succ)$ は評価構造と呼ばれる. E は評価値の集合である. $\otimes$ は評価値の結合演算を定義する. $\succ$ は評価値の順序を定義する. $\varphi(c) \in E$ は制約 $c \in C$ の評価である. 重み付き CSPP と変数の割当て A に対する評価は次のようである.

$$V_P(A) = \bigotimes_{c \in C, A \text{ violates } c} \varphi(c)$$

重み付き CSP の解は $V_P(A)$ が最小となる割当て A である.

分散 CSP は, CSP の分散処理への拡張である. 分散 CSP のシステムは複数のノード (エージェント) と各ノード間の通信チャネルから構成される. 本稿のモデルではカメラノード i が分散 CSP のノードに対応する. 各ノード i は自分の変数 $x_{i,0}, \dots, x_{i, Nrgn-1}$ を持つ. また, 各ノードは自分の変数に関連する制約を持つ. 分散 CSP を制約ネットワークとして表した例を図 3 に示す. 制約ネットワークは変数をノード, 2 変数制約を辺として表現したグラフである. 本稿で用いる制約は多変数に関連するので, 実際には hyper-graph となる.

各ノードは自分の変数の値のみ変更できる. そのため, 各ノードは通信しながら大域的な解を求めなければならない.

3.2 分散制約充足アルゴリズムに基づく解法

問題を解くために, 分散 CSP における代表的な解法である, 非同期弱コミットメントアルゴリズムに基づく解法を用いる. 非同期弱コミットメントアルゴリズムの概要は次のようである.

- 各ノードは, 自分の変数および関連する制約に加えて, view と呼ばれる他ノードの変数の情報および, nogood と呼ばれる新たに導出された違反制約の集合を持つ. 違反制約は, 制約に違反するような部分解を表す.
- 各ノード間で送受される基本的なメッセージは, 自分の変数値変更の通知メッセージ *ok?* および, 違反制約の通知メッセージ *nogood* である.
- 各ノード間の通信におけるメッセージの順序は保存される.
- 各ノードは他ノードから *ok?* 受信したとき, view を更新する. また, *nogood* メッセージを受信したとき, 違反制約の集合を更新する.
- 各ノードは他ノードからメッセージを受信した後, 必要に応じて view との整合が取れるように自分の変数の値を変更し, 他ノードへ *ok?* メッセージを送信する. 変数の値の整合性の評価には, もとの制約の集合および, 新たに導出された制約の集合を用いる.
- 各ノードは自分の変数の値を変更しても view と整合が取れないとき, 違反制約を導出し, 他ノードへ *nogood* メッセージを送信する.

このアルゴリズムは解が無ければ, いずれかのノードが空の違反制約を検出し, 解があれば全ノードがメッセージ待ちの安定状態になる.

重み付き制約充足問題は, 過制約な問題であるため, 解が充足できない場合に, 幾つかの制約を緩和しなければならない. 過制約な分散制約充足問題の為に, 制約の強化または緩和を行うスケジュールにより, 非同期弱コミットメントアルゴリズムを反復的に実行する解法が提案されている [8].

ここでは段階的な制約の緩和を行う手法を適用することとした¹. また, アルゴリズムが安定状態に到達したことを検出するために, 大域停止検出手法 [12] を仮定した.

3.3 制約の設計

本稿では, 問題を簡単にするために, 割当ての目的として

¹横尾らの手法 [8] ではある閾値よりも優先度が低い制約は全て緩和されるが, 本稿では最適解を求めるように緩和を制御することとした

1. 対象の存在する領域を, 十分な精度で推定する.
すなわち, 対象領域を十分削減する.
2. 観測するカメラ台数を削減する. すなわち, 部分的な観測 (注視) のために使用できる未使用のカメラを用意する.

のみを考慮する. 従って, 最低限のカメラ割当てによって対象領域を十分に削減することが目的となる. これらについて以下の制約を設計した.

[対象領域の削減]

観測データは方位情報しか持たないため, 各対象領域を削減するためには, その対象領域にについて2つ以上のカメラからの観測データが必要である. また, カメラの視線は十分な交差角を持つ必要がある (図 4(a)). ここで視線は, カメラの座標から, 領域の重心座標へのベクトルとして定義する.

$c_{0,j}$: 各対象領域 j に対して, カメラは2つ以上割当てられなければならない. いずれか2つのカメラの視線の内積は閾値より小さくなければならない.

[オクルージョンの回避]

本稿では対象領域間のオクルージョンを考慮する (図 4(b)). オクルージョンは, 対象領域の推定において観測データとの対応付け誤りの原因となる. 対象領域に対するオクルージョンの無いカメラからの観測データのみを用いれば対応付け誤りは起こらない. しかし, オクルージョンを回避できない場合がある. このような場合, 次善の努力として, その領域をできるだけ削減するために, 全てのカメラを割当てる.

$c_{1,j}$: 領域 j にオクルージョンの無いカメラのみ割当てる. または, 領域 j に全てのカメラを割当てる.

$c_{2,j}$: 領域 j にオクルージョンの無いカメラのみ割当てる.

[観測するカメラの削減]

観測するカメラの数を削減するためには, カメラは不要な領域の選択をしなければよい.

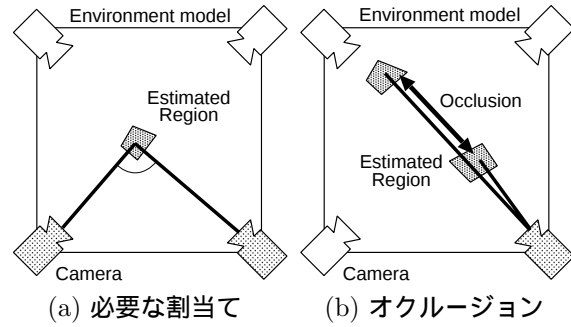


図 4: 制約の設計

$c_{3,i}$: カメラ i はいずれの領域も選択してはならない.

上記の制約について, 対象領域の削減, オクルージョンの回避, カメラ数の削減の順に優先されるものとする. 各制約の評価に次の順序を与える.

$$\begin{aligned} \varphi(c_{0,j}) &> \varphi(c_{1,j}) \\ \varphi(c_{1,j}) &> \varphi(c_{2,j}) \\ \varphi(c_{2,j}) &> \bigotimes_i \varphi(c_{3,i}) \end{aligned}$$

4 評価

シミュレーションにより評価を行った. 観測領域の大きさは $6m \times 6m$ とした. カメラは観測領域の4隅に配置した. 対象は観測領域内を速度 $1.2m/s$ で周回するものとした. 制約 $c_{0,j}$ の閾値は内積 $\cos(\pi/12)$ とした. 領域の演算は増加演算における等方性を考慮し6近傍のビットマップを用いて近似した. セルの数は観測領域全体に対して 100×100 である. 1時間ステップを $50ms$ である.

対象領域の削減を評価するために, 初期状態では全観測領域を対象領域とし, 観測データにより対象領域を削減するものとした.

図 5 は, ステップ 0 から 3 までの対象領域とカメラ割当てを示す. はじめは, 領域の断片によってオクルージョンが発生する. しかし, 領域が十分削減されると, カメラの割り当てが削減される.

図 6 は, 20, 35 ステップにおける, カメラ割り当てを示す. 対象の移動によって発生するオクルージョンを回避するように, カメラ割当てが変更される. 35 ステップでは, 領域 1 に対して, 他のカメラが割り当てられる.

図 7 は、割り当てられるカメラの数を示す。割り当てられるカメラの数はオクルージョンに追従している。しかし、ステップ 2 以後は対象領域数は 3 に維持された。

シミュレーションの結果により、対象の存在範囲を絞り込みつつ、オクルージョンの回避、カメラの使用台数を削減を行う処理を、重み付き分散 CSP の枠組みで表現できることが示された。

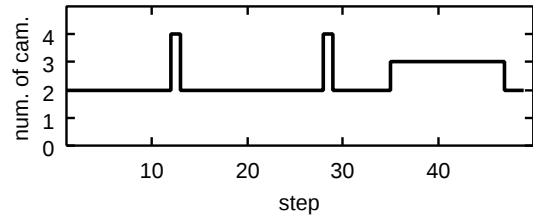


図 7: Number of allocated cameras

5 考察

本稿では例題として簡単な観測モデルを用いた。また、カメラ割当てにおける目的も簡単なものに限定了。このモデルは、複数の方位データからの位置推定、対象の移動の推定、観測データと推定データの対応付けなどの基本的な要点を含むモデルとして採用した。実際のシステムにおいては、カメラの、観測誤差、画角の限界、障害物の影響などを考慮する必要がある。

シミュレーションにおいては小規模な問題を仮定し、アルゴリズムの処理時間を無視した。カメラ数や対象領域数が増加した場合など、大規模な問題において大域的な最適解を求める場合、実時間処理は困難になる。そのような場合は、期限に応じて重要な目的のみを充足する手法 [11] などの適用が必要である。また、環境の変化が小さい場合には、全解の探索における情報の再利用により、探索を効率化する手法 [10] を適用する余地があると考えられる。

各ノードは同一の環境モデルを持つ。さらに、割当て問題の更新処理、および解探索の開始時点と終了時点は同期している。比較的大規模な分散処理への適用のためには、これらを緩和する必要がある。従来の分散 CSP のためのアルゴリズムに対し、ノード間に問題の不整合がある場合を許すような拡張が必要となる。

各カメラノードを CSP におけるノードとした。一方で、より重要な制約は、対象領域の観測精度に関するものであり、それらは各カメラに分散した各変数に関連する。メッセージ通信回数の削減の点では、各対象領域単位に CSP のノードを設定する方が効率的であると考えられる。しかし対象領域ごとのグループ化は計算機網の実体と合致しない。このようなモジュールの特化は、観測対象や特定のタスクごとに複数のカメラを組織化するエージェンシのような概念 [1] [3] と関連すると考えられる。

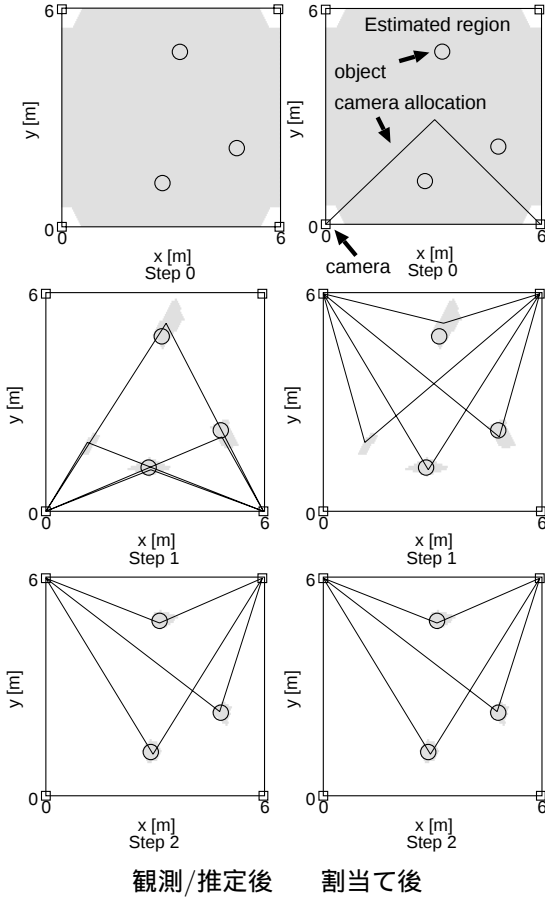


図 5: 環境モデルの変化 (1)

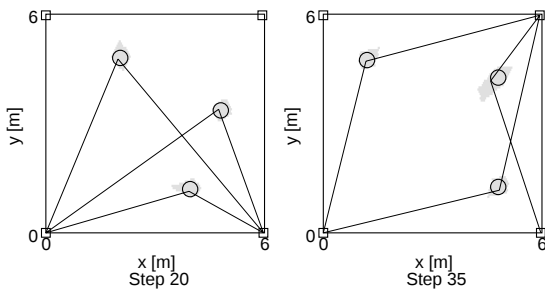


図 6: 環境モデルの変化 (2) (観測/推定後)

6 まとめ

本稿では、カメラ資源の割り当てを重み付き分散 CSP としてモデル化した。シミュレーションにより提案手法の有効性を評価した。

より実地的なモデルの設計および、実機による評価が今後の目標である。

謝辞

本研究の一部は堀情報科学技術振興財団助成による。

参考文献

- [1] 長尾卓, 浮田宗伯, 松山隆司: 能動視覚エージェント群による複数対象の協調的追跡, 情報処理学会研究報告, 2000-CVIM-121-8, pp.57-64 (2000).
- [2] 中澤篤志, 加藤博一, 井口征士: 分散カメラエージェントを用いた広域人物位置検出システム, 情報処理学会論文誌, Vol.41, No.10, pp.2895-2906 (2000).
- [3] 加藤丈和, 向川康博, 尺長健: 安定な顔認識のための分散協調登録, 電子情報通信学会論文誌 Vol.J84-D-II No.3, pp.500-508 (2001).
- [4] 岡嶋正道, 竹内義則, 大西昇: カメラの連携による複数移動物体の追跡, 情報処理学会研究報告, PRMU99-275, pp.105-112 (2000).
- [5] 亀田 能成, 宮崎 英明, 美濃 導彦, 飯塚 重喜, 辻本 雅彦: 講師追跡によるカメラ映像の自動切り替え, 情報処理学会 第 58 回全国大会 Vol.3, 2V-4, pp.219-220 (1999).
- [6] 森大樹, 内海章, 谷内田正彦: 非同期多視点情報を用いた複数人物追跡手法の検討, 情報処理学会研究報告, PRMU98-178, pp.15-22 (1999).
- [7] Katsutoshi Hirayama and Makoto Yokoo, An Approach to Over-constrained Distributed Constraint Satisfaction Problems: Distributed Hierarchical Constraint Satisfaction, 4th International Conference on Multi-agent Systems (ICMAS-2000)", 2000.

- [8] Katsutoshi Hirayama and Makoto Yokoo, An Approach to Over-constrained Distributed Constraint Satisfaction Problems: Distributed Hierarchical Constraint Satisfaction, 4th International Conference on Multi-agent Systems (ICMAS-2000), 2000.
- [9] T. Schiex, H. Fargier and G. Verfaillie, Valued Constraint Satisfaction Problems : Hard and Easy Problems, Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95), pp.631-637, 1995.
- [10] Thomas Schiex and G'erard Verfaillie, No-good Recording for Static and Dynamic Constraint Satisfaction Problems, International Journal of Artificial Intelligence Tools, 3(2) pp.187-207 (1994).
- [11] Garvey A. and Lesser V., A Survey of Research in Deliberative Real-Time Artificial Intelligence, Journal of Real-Time Systems, Vol.6, No.3 (1994).
- [12] Nir Shavit and Nissim Francez, A new approach to the detection of locally indicative stability, Int. Colloq. on Automata, Languages, and Programming, 1986.