

分散制約最適化手法を適用した
協調カメラ網の実装

加藤伸哉

平成 22 年 2 月 3 日

目次

第 1 章	はじめに	1
第 2 章	関連研究	3
2.1	協調センサ網	3
2.2	AVA	4
2.3	分散制約最適化問題	5
2.3.1	解の探索手法	6
2.3.2	問題の形式化	7
第 3 章	提案手法	12
3.1	観測資源の要求と制限に関する制約	12
3.1.1	SAV (Sensor As Variable)	12
3.1.2	探索空間の比較	13
3.2	安定的な資源割り当てを保持するための制約	15
3.2.1	保持制約	15
3.2.2	制約間の優先関係	16
第 4 章	提案手法を適用した観測システムの実装	19
4.1	全体の処理	19
4.2	対象物検出	20
4.3	座標推定	22
4.4	対象物同定	24
4.5	制約網の生成	25
4.6	解の探索	26

第 5 章 評価	29
5.1 システム構成	29
5.2 解の探索のパラメータ	29
5.3 シミュレーションによる評価	31
5.3.1 収束サイクルと探索空間の関係	32
5.3.2 DSA による解の探索	32
5.4 実システムによる評価	32
5.4.1 実験環境	33
5.4.2 実行時間	33
5.4.3 単独のスナップショットに対するシステムの挙動	34
5.4.4 連続したスナップショットに対するシステムの挙動	34
第 6 章 まとめ	47

第1章

はじめに

近年，マルチエージェントシステムの応用分野として，分散センサ網が注目されている [9]．分散センサ網は，自律的なセンサ群が相互に接続されたシステムであり，各センサが協調して情報収集やシステム制御をおこなうことを求められる．分散センサ網の応用例として，ロボットサッカー [4] や協調的な監視カメラ網 [11][6] などが提案されている．

その一方で，マルチエージェントシステムの協調問題解決における基本的な枠組みとして，分散制約最適化問題が研究されている [1][2][5][10]．分散制約最適化問題では，エージェントの状態が変数として表現され，エージェント間の協調が変数間の評価関数として表現される．各エージェントは互いに情報を交換し，評価関数を大域的に最大化する変数値を決定する．マルチエージェントシステムの目的を分散制約最適化問題として形式化することにより，そのシステムで解決すべき問題を明示的に記述することができる．

特に分散センサ網については，観測資源を観測対象物にどのように割り当てるかを分散制約最適化問題へ形式化する手法の研究が行われている [6][11]．

実際の観測システムでは，センサの注視方向の変化や観測対象物の移動など，環境は時刻とともに変化する．そのため，環境の変化にシステム全体が追従するために，各センサに対する観測資源の割当ても環境の変化にともなって変更する必要がある．

時刻とともに変化する環境における資源割当て問題を分散制約最適化問題へ形式化する場合，ある時刻での大域的なスナップショットをつなげた時系列的な割当て問題として考えることができる．各時刻におけるスナップショットに対して割ける時間は

限られているので、短時間で解を発見することができる確率的解法を用いることが有効である。しかし、各エージェントの協調と資源割当てをおこなう問題は複雑であり、解の精度の改善と探索時間の削減の両立には課題がある。

本研究は、協調カメラ網に含まれる比較的小規模な部分問題への適用を目的とする。短時間で解を得るために、従来手法より簡単な分散制約最適化問題への形式化を用いる。

また、実際的なシステムにおいては、時系列的な割当てについて、あるエージェントの動作が頻繁に切り替わることは好ましくない。一度決定した割当てを次の割当て問題の解として可能な限り保持するため、直前の問題の解と、現在の問題の解を同一にたもつことを目的とする「弱い」制約を導入した。

最後に、提案手法を実際の分散協調カメラ網へ適用し、解の精度、実行時間などの評価を行った。

以降、2章では協調センサ網の関連研究について触れ、3章で提案手法について述べる。4章で実装について説明し、5章で評価を述べる。

第2章

関連研究

本章では、対象とする協調センサ網の特徴と、解決すべき問題および関連研究について述べる。

2.1 協調センサ網

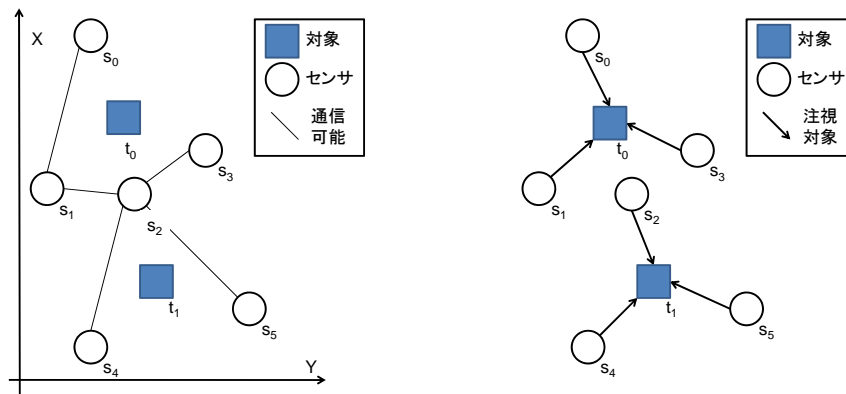
協調センサ網とは、自律的な複数のセンサが協調して観測対象物の情報収集をおこなうシステムである。

複数のセンサと複数の観測対象物が存在する協調センサ網では、各観測対象物を注視するセンサを決定する資源割り当て問題を解決する必要がある。図 2.1 に協調センサ網における割り当て問題の例を示す。図 2.1 (a) のようなセンサと観測対象物の配置がある場合、例えば全ての観測資源を平等に観測対象物へ割り当てることを目的とすれば、図 2.1 (b) のように、各観測対象物を 3 センサが注視するような割り当てが考えられる。

本研究では、協調センサ網のうち、次の特徴を持つものを観測対象物とする。

- 観測対象物は計算資源を持たない
- 各センサが観測できる範囲に制限がある
- 各センサが同時に観測できる観測対象物の数に制限がある

これらの特徴を持つ協調センサ網には監視カメラ網や協調的な天文観測などがあげられる。その一方で、ロボットサッカーや協調カーナビゲーションシステムなど、観



(a) センサと観測対象物の二次元平面上における配置

(b) 割り当て例

図 2.1: 協調センサ網の例

測対象物が計算・通信資源を持ち，協調的な処理の構成要素に含まれるものは，本研究の対象としない．

以降では，協調センサ網の資源割り当て問題解決について，エージェントによる協調モデルと，分散制約最適化問題への形式化による協調モデルを述べる．

2.2 AVA

本研究で対象とする協調センサ網について，エージェントの集合であるエージェントを用いた協調モデルが提案されている [7][8]．エージェントによる協調モデルでは，エージェント内のマネージャと呼ばれるエージェントがメンバとして参加するエージェントを制御する．

エージェントによる協調モデルを使用する手法では，システムは自律的に動作する能動カメラエージェント (Active Vision Agent, AVA) の集合として構成される．各 AVA は，観測対象物を検出すると観測対象物ごとにエージェントを生成する．エージェントは観測対象物ごとに統合され，各エージェントにはただひとつのエージェントマネージャが選出される．各エージェントマネージャは通信により観測資源の割り当てを決定し，他の AVA はエージェントマネージャの決定に従う．すなわち，このモデルで

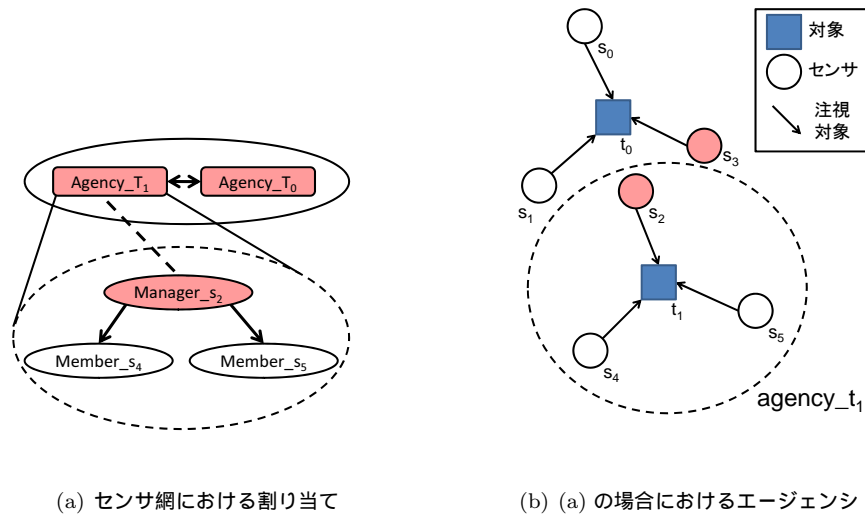


図 2.2: エージェントによる協調モデル

は、協調センサ網における資源割り当て問題を、各観測対象物の観測センサを代表するエージェントマネージャが協調して解決する。

図 2.2 にエージェントによる協調モデルの例を示す。センサ網の割り当てが図 2.2(a) の場合、例えばセンサ s_2 がマネージャとなり、観測対象物 t_1 を注視するセンサ集合であるエージェント t_1 の代表として他エージェントと協調する。センサ s_4 , s_5 は代表であるセンサ s_2 の決定に従う。この関係を図 2.2(b) に示す。

エージェントによる協調手法を用いたシステムは実機を用いた小規模な環境で実証されているが、その協調プロトコルを最適化問題として考えた場合の解の精度については明確ではない。

2.3 分散制約最適化問題

分散制約最適化手法とは、マルチエージェントシステムを各エージェントに変数、評価関数が分散して配置される分散制約最適化問題へ形式化する手法である [1][2][5][10]。分散制約最適化問題は、マルチエージェントシステムにおける協調問題解決のための基礎的な枠組みとして研究されている。分散制約最適化問題は以下のように形式化される最適化問題である。

- マルチエージェントシステムに含まれるエージェントの集合を A とする
- 変数の集合を X とし, それぞれの変数 $x_i \in X$ はいずれかのエージェント $a_j \in A$ によって管理される
- 値域の集合を D とし, 各変数 $x_i \in X$ の値域を $d_i \in D$ とする. 各変数の値域は離散的かつ有限である
- 制約の集合を C とし, 制約 c_k はひとつもしくは複数の変数に対する制約を表す
- 制約 c_k には評価関数 f_k が対応する. 評価関数は, 制約に関連する変数から評価値を計算する
- 評価関数全体の集合を F とすると, 分散制約最適化問題は評価関数の総和 $\sum_{f_i \in F} f_i$ を最小化する変数値 $x_0, x_1, \dots, x_{|X|-1}$ を計算する問題である

各エージェントは自身が管理する変数と制約で関係する変数を管理する他のエージェントと通信し, メッセージを交換することで解の探索をおこなう. メッセージ交換をともなう分散処理により解を求めるためのアルゴリズムが提案されている.

2.3.1 解の探索手法

分散制約最適化問題の解の探索手法は, 最適解を求めることが保証される厳密解法と, 最適解を求める保証がない非厳密解法に分類される.

分散制約最適化問題の厳密解法として, ADOPT(Asynchronous Distributed Constraint Optimization)[1], DPOP[2] が提案されている.

どちらの手法も, あらかじめ深さ優先探索などを用いて, 解くべき制約網に対する生成木を構築し, その生成木に対応する疑似木を解の探索に用いる. どちらの手法も確実に最適解を求めることができるが, 解の探索に用いる疑似木の induced-width [3] の増加に従って必要なサイクル数/メッセージサイズが指数的に増大するという問題がある. このため, 短時間かつ省メモリで解を探索する必要があるシステムへ応用する場合には課題があるといえる.

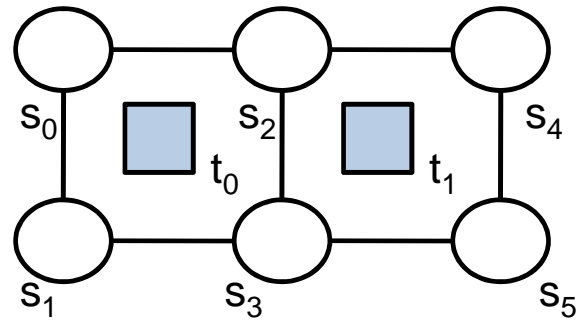


図 2.3: グリッドモデルによる表現

最適解を求める保証がないものの、比較的高速に一定の精度の解を発見できる方法として、確率的な解法である DSA(Distributed Stochastic Search Algorithm)[5] および DSTS(Distributed Stochastic Tabu Search)[10] が提案されている。

DSA は各エージェントが自身の管理する変数に対して、その変数値の近傍を探索し、確率的な値の更新をおこなう手法で、確率的手法を導入した山登り法を分散処理として実行する。DSTS は DSA に対して、タブーサーチを追加したものであり、複雑な問題における解の精度を高めることができる。

本研究で用いるシステムでは、各時刻におけるスナップショットに対して割ける時間は限られる。本研究では比較的短い時間で近似解を得る性質を重視し解の探索アルゴリズムに確率的解法である DSA/DSTS を用いることとする。

2.3.2 問題の形式化

マルチエージェントシステムとしての協調センサ網を分散制約最適化問題へ形式化する手法が複数提案されている。

それらでは、協調センサ網における資源割り当て問題の基礎的な表現として、グリッドモデルを用いている。

グリッドモデルは、センサの可視範囲や観測対象物の同定などを簡略化し、センサの資源割り当て問題のみを考える際に用いられるモデルである。グリッドモデルによ

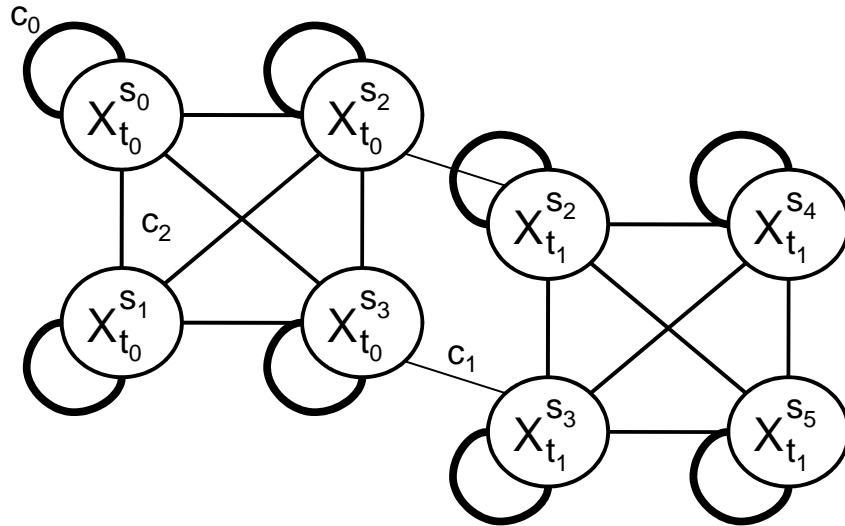


図 2.4: 資源割り当て問題の STAV による形式化

るセンサ網の表現を図 2.3 に示す．グリッド内に観測対象物は高々 N 個存在し，センサの可視範囲はセンサが隣接するブロックまでである．関連研究では， N を 1 としている．例えば，図 2.3 で，センサ s_0 は観測対象物 t_0 のみを注視可能であり，センサ s_2 は観測対象物 t_0, t_1 を注視できる．

実際に分散制約最適化問題に形式化する方法として，STAV と TAV を示す．

STAV

STAV[6] はセンサ s_i と観測対象物 t_j の組を変数 $x_{t_j}^{s_i}$ として形式化する手法であり，その値域は t_j を注視できるセンサのベキ集合である．例として，図 2.3 の協調センサ網を STAV を用いて分散制約最適化問題へ形式化すると，図 2.4 のような制約網となる．

図 2.4 中に示される STAV の制約は次のように表される．

- c_0 : 観測対象物への資源要求

観測対象物は N センサ以上で追跡されることを表す単項制約である． N として 3 を採用し，割り当てが 3 センサに満たない場合に違反となる．この制約 c_0 に対する評価関数 f_{c_0} は以下のように表される．ただし， $w_i^{c_0} (i = 0, \dots, N - 1)$ は違反度を表す定数である．

$$f_{c_0}(x_{s_i, t_j}) = \begin{cases} w_0^{c_0} & |x_{s_i, t_j}| = 0 \\ w_1^{c_0} & |x_{s_i, t_j}| = 1 \\ w_2^{c_0} & |x_{s_i, t_j}| = 2 \\ 0 & otherwise \end{cases} \quad (2.1)$$

- c_1 : 観測資源の制限

各センサの観測資源が M 観測対象物を同時に注視できないという二項制限である． M として 1 を採用し，センサが 2 つ以上の観測対象物に割り当てられるとき違反となる．この制約 c_1 に対する評価関数 f_{c_1} は以下のように表される．ただし， w^{c_1} は違反度を表す定数である．

$$f_{c_1}(x_{s_i, t_j}, x_{s_i, t_k}) = \begin{cases} w^{c_1} & x_{s_i, t_j} \cap x_{s_i, t_k} \neq \emptyset \\ 0 & otherwise \end{cases} \quad (2.2)$$

- c_2 : 意志決定の統一

同一観測対象物に対する各センサの決定がシステム全体で同一であるという二項制約である．センサ間で決定が異なる場合に違反となる．この制約 c_1 に対する評価関数 f_{c_2} は以下のように表される．ただし， w^{c_2} は違反度を表す定数である．

$$f_{c_2}(x_{s_i, t_j}, x_{s_k, t_j}) = \begin{cases} w^{c_2} & x_{s_i, t_j} \neq x_{s_k, t_j} \\ 0 & otherwise \end{cases} \quad (2.3)$$

TAV

TAV は観測対象物 t_j を変数 x_{t_j} として形式化する方法である．割り当て探索問題における変数 x_{t_j} の値域は観測対象物 t_0 を注視できるセンサの集合を $\{s_0, \dots, s_n\}$ とすると，その可能な組み合わせ全てであり，すなわち注視できるセンサの集合のベキ集合である．

割り当て探索問題における制約は，次に示す 2 つが存在する．

- c_0 : 観測対象物への資源要求

観測対象物は N センサ以上で追跡されることを表す制約である． N として 3 を

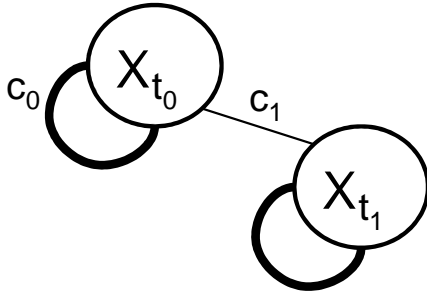


図 2.5: TAV による資源割り当て問題の形式化

採用し、割り当てが 3 センサに満たない場合に違反となる。この制約 c_0 に対する評価関数 f_{c_0} は以下のように表される。ただし、 $w_i^{c_0} (i = 0, \dots, N - 1)$ は違反度を表す定数である。

$$f_{c_0}(x_{s_i, t_j}) = \begin{cases} w_0^{c_0} & |x_{t_j}| = 0 \\ w_1^{c_0} & |x_{t_j}| = 1 \\ w_2^{c_0} & |x_{t_j}| = 2 \\ 0 & otherwise \end{cases} \quad (2.4)$$

- c_1 : 観測資源の制限

各センサの観測資源が M 観測対象物を同時に観測できないという制限である。 M として 1 を採用し、センサが 2 つ以上の観測対象物に割り当てられるとき違反となる。この制約 c_1 に対する評価関数 f_{c_1} は以下のように表される。ただし、 w^{c_1} は違反度を表す定数である。

$$f_{c_1}(x_{s_i, t_j}, x_{s_i, t_k}) = \begin{cases} w^{c_1} & x_{t_j} \cap x_{t_k} \neq \emptyset \\ 0 & otherwise \end{cases} \quad (2.5)$$

TAV の割り当て探索における制約網を図 2.5 に示す。

これら既存の手法には、解の精度が低下するほか、時間的に連続した問題間の解についての制約など、実システムにおける課題を省略しているといった問題が存在する。

前者については、STAV ではエージェント間の合意を明示的に形式化するために制約網が複雑になり、解の精度や探索時間に課題が残る。TAV は STAV と比較して、ある観測対象物 t_j に対応する変数は x_{t_j} ひとつであるため、制約網が単純になっている。

しかし、対象とするシステムでは観測対象物 t_j そのものは計算資源を持たないので、TAV を直接適用することはできない。観測対象物の計算資源をセンサが肩代りし、肩代りするセンサを選出するリーダー選出問題と、センサ割り当て問題を階層的に結合する手法 [11] も提案されている。

後者については、既存の形式化における制約には、注視観測対象物を変更することによるオーバーヘッドなど、実際のシステムへ応用する場合の実際的な制約が存在していない問題がある。また、これらの研究ではグリッドモデルが用いられ、各センサの視野や配置は単純化されているために、各センサの配置や観測対象物の数が固定的であって、実際的なシステムのモデルとしては不適合であると考えられる。

第3章

提案手法

本研究では，STAV では制約として含まれていた，各エージェント間の観測対象物に対する割り当ての合意を決定する問題を考慮しない．協調センサ網における割り当て問題を，比較的小規模な問題へ適用することを前提として，より単純な分散制約最適化問題へ形式化する．

また，時系列的に変化する環境において，各センサが観測対象物への観測資源の割り当てを変更する回数を削減するために，観測対象物への観測資源の割り当てを変更することに対して，実際的な制約である保持制約を追加する手法を提案する．

3.1 観測資源の要求と制限に関する制約

協調センサ網の各センサ割り当てを変数値として分散制約最適化問題へ形式化する手法について述べる．各センサは他のセンサの割り当てについて直接は関与しないために，STAV や TAV に対してより単純な形式化となる．

3.1.1 SAV (Sensor As Variable)

SAV では，センサ s_i に対して，センサが注視する観測対象物の集合を変数 x_{s_i} として形式化する．すなわち，センサ s_i はその変数の要素 $t_k \in x_{s_i}$ を注視する．

各センサが一度に注視できる観測対象物数には制限があるため，本研究では，各センサが一度に注視できる観測対象物の最大数をひとつとする．各変数 x_{s_i} の値域 d_{s_i} はセンサ s_i が注視できる観測対象物の集合を $\{t_0, \dots, t_m\}$ とすると $\{\emptyset, \{t_0\}, \dots, \{t_m\}\}$ であり，値域の総数は $m + 1$ となる．

観測資源の要求と制限に関する制約を、「追跡制約」として次のように形式化する。

追跡制約 $c_0^{t_i}$

観測対象物 t_i に割り当てられたセンサ数が 2 に満たない場合に違反となる。この制約は多項制約であり、制約 $c_0^{t_i}$ に対する評価関数 $f_{c_0^{t_i}}$ は以下の式として与えられる。 n_{t_i} は t_i に割り当てられたセンサ数を表す。また、 X_{t_i} を観測対象物 t_i を注視できるセンサ集合とする。 $w_{IGN_0}^0, w_{IGN_1}^0$ は制約の違反度を示す定数値である。

$$f_{c_0^{t_i}}(X_{t_i}) = \begin{cases} w_{IGN_0}^0 & n_{t_i} = 0 \\ w_{IGN_1}^0 & n_{t_i} = 1 \\ 0 & otherwise \end{cases} \quad (3.1)$$

また、制約 $c_0^{t_i}$ は 1 以上のセンサ数で追跡するという制約と、2 以上のセンサ数で追跡するという制約に分解することができる。以降では、特に必要がある場合には 1 以上のセンサ数で追跡するという制約を $c_{00}^{t_i}$ 、2 以上のセンサ数で追跡するという制約を $c_{01}^{t_i}$ と記述する

図 3.1 (a) のような、4 センサがそれぞれ 2 つの観測対象物を注視できる配置をセンサ割り当てを変数とする本提案手法で形式化すると、図 3.1 (b) となる。図 3.1 (b) 中の追跡制約 c_0 には、観測対象物 t_0, t_1 の両方についての追跡制約が含まれる。すなわち、例えば変数 x_{s_0} と変数 x_{s_1} は 2 つの制約でつながっている。また、この形式化では視野を共有するセンサを表す変数間全てに制約辺が存在する。

また、図 3.2(a) のように、注視できる観測対象物が異なるようなセンサが存在する場合の制約網を図 3.2(b) に示す。この場合、センサ s_0 とセンサ s_4 との間には制約辺が存在しない。

SAV を用いて形式化された制約網は、観測対象物 t_j への割り当てを決定するための完全グラフが連結された構造となる。各完全グラフの次数は、その観測対象物を注視できるセンサ数である。

3.1.2 探索空間の比較

SAV, STAV, TAV について、システム全体のセンサ数を N 、各対象を注視できる最大のセンサ数を n 、注視すべき観測対象物の数を M 、各センサが注視できる最大の対象数を m とした時の変数と値域、および探索空間の規模を表 3.1 に示す。

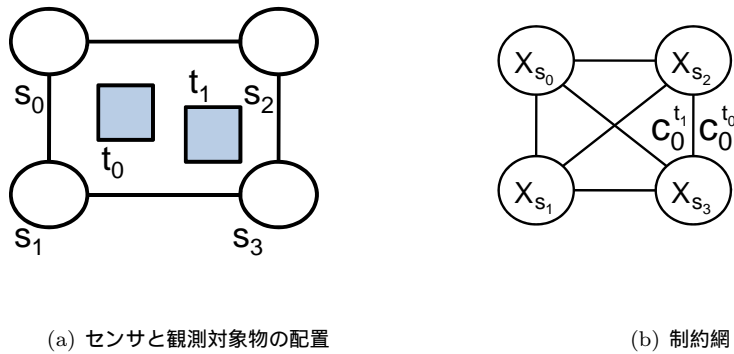


図 3.1: SAV による形式化

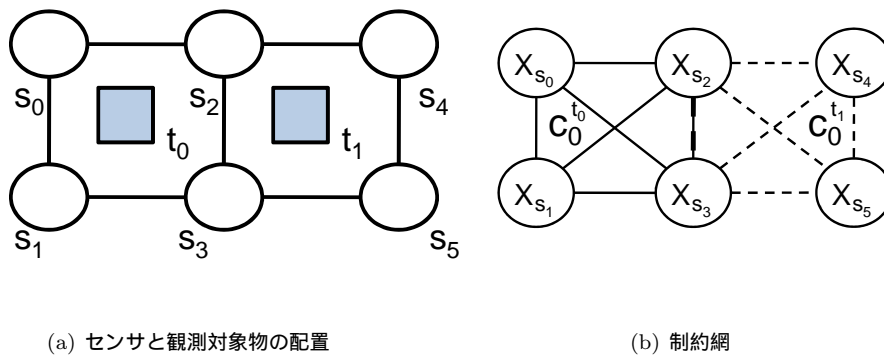


図 3.2: 視野を共有しないセンサが存在する場合の SAV による形式化

提案である SAV では、変数の数はセンサ数 N に対して $O(N)$ であり、一般に $N > M$ であるから、TAV よりも変数の数は多くなるが、 $(\log_2 m)/M < n/N$ であるならば、探索空間の規模は小さくなる。この関係は、システム全体のセンサ数が少なく、対象数が多い場合に満たされる。

また、STAV は変数の数が $O(NM)$ であることから、SAV の方が明らかに探索空間の規模は小さくなる。

表 3.1: 形式化の探索空間

形式化	変数	値域	探索空間
SAV	$O(N)$	$O(m)$	$O(m^N)$
TAV	$O(M)$	$O(2^n)$	$O(2^{nM})$
STAV	$O(NM)$	$O(2^n)$	$O(2^{nNM})$

3.2 安定的な資源割り当てを保持するための制約

提案システムでは観測対象物に対する割り当てはスナップショットごとに決定される。各スナップショット間で決定する割り当ての間に明確な関係はないが、実際的には、各スナップショットにおける割り当てが頻繁に変更されることは望ましくない。例として、センサの割り当てが変更される場合、これまでに注視していたセンサが収集した情報を共有するための通信が発生する、また注視のために視線方向を制御する場合、視線方向の制御中には観測が不可能になるなどという問題が存在する。

本研究では、連続したスナップショット間で割り当ての変更が少ない解を得るために、前スナップショットでの割り当てと、現在のスナップショットでの割り当てが同一であるための制約を導入する。

3.2.1 保持制約

直前のスナップショットにおけるカメラ s_i の割り当てを定数 a_{s_i} とする。あるセンサ s_i が直前のスナップショットにおける割り当てと、現在のスナップショットにおける割り当てが同一であることが望ましい場合というのは、以下の条件を両方とも満たす場合である。

- 直前のスナップショットでカメラ s_i へ割り当てられた観測対象物は現在のスナップショットにおいてもカメラ s_i が注視できる観測対象物である
- 直前のスナップショットでカメラ s_i に観測対象物が割り当てられていた

よって、安定的な資源割り当てを保持するための制約を、「保持制約」として次のように形式化する。保持制約 c_1

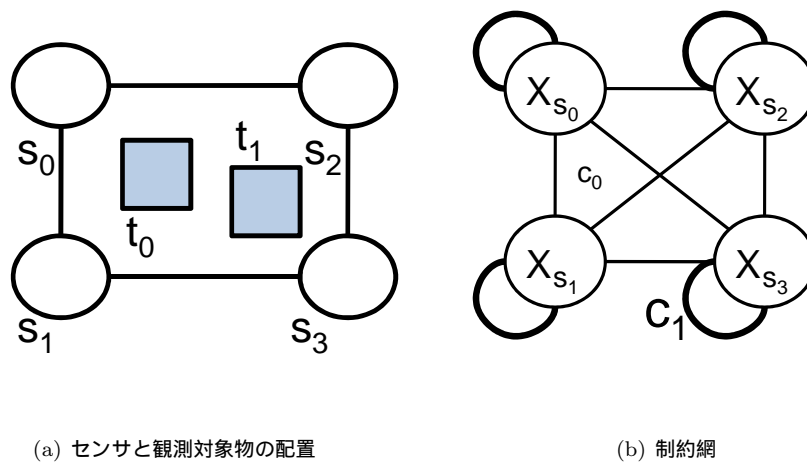


図 3.3: 保持制約を追加した場合の形式化

一度、あるスナップショットで決定した割り当てである a_{s_i} を次以降のスナップショットでも保持するための制約である。センサ s_i に割り当てられた観測対象物 t_i を変更する場合に違反となる。この制約は単項制約であり、制約 c_1 に対する評価関数 f_{c_1} は以下の式として与えられる。 w^{c_1} は制約の違反度を示す定数値である。

$$f_{c_1}(x_{s_i}) = \begin{cases} w^{c_1} & (x_{s_i} \neq a_{s_i}) \wedge (\{a_{s_i}\} \in (d_{s_i} \setminus \{\emptyset\})) \\ 0 & otherwise \end{cases} \quad (3.2)$$

観測対象物の数が 2、センサ数が 4 であり、全てのセンサがいずれかの観測対象物を注視するような状態である図 3.3(a) を保持制約を追加した提案手法を用いて形式化すると、図 3.3(b) のようになる。

3.2.2 制約間の優先関係

分散制約最適化問題には、その最適解において評価関数値の総和がゼロである保証はない。例えば、センサ数 2、観測対象物数 2 ではセンサ数が不足しており、明らかに追跡制約を満足することはできない。また、センサ数が 4、観測対象物の数が 2 の場合も、例えば図 3.4 のような制約網が作成される場合については、保持制約を満足することができない。新たにセンサ s_0 が観測対象物 t_1 を注視する割り当て以外は保持制約に違反してしまうが、センサ s_0 は観測対象物 t_1 を注視できないためである。セ

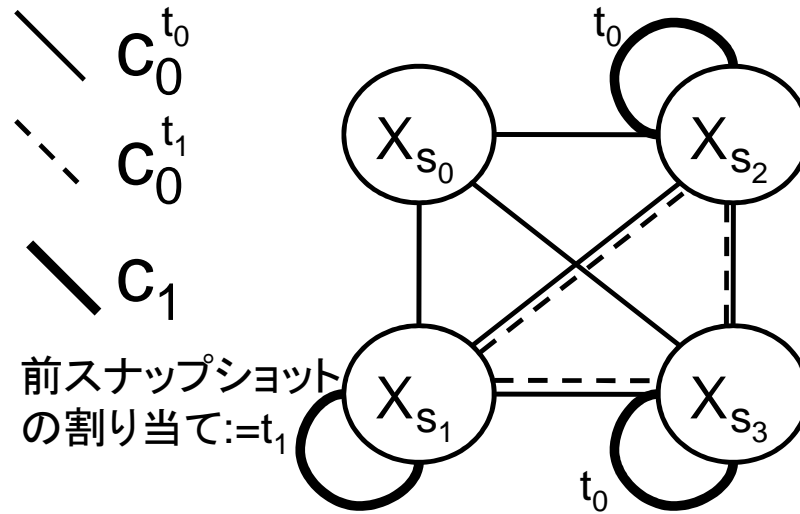


図 3.4: 制約違反が残る変数の組が最適解であるような制約網の例

ンサ s_0 が観測対象物 t_0 へ注視する観測対象物を変更し、かつセンサ s_2 もしくは s_4 のどちらかのみが観測対象物 t_0 から観測対象物 t_1 へ注視する観測対象物を変更することが合理的であるが、その場合は変数 x_{s_2} もしくは x_{s_3} のどちらかが保持制約に違反する。

また、本研究では解の探索手法として確率的解法を用いるために、そもそも最適解に到達する保証はない。

追跡制約と保持制約について、満足できない場合の重大さは、追跡制約の方が保持制約よりも大きい。その理由は、追跡制約を満足できなければ、観測対象物に関する情報は偶然による以外に一切得られないが、保持制約を満足できなくても他のセンサが観測対象物の注視をおこなうために、観測対象物に関する情報の一部は得ることができるためである。

以上の理由から， $A \succ B$ を B よりも A の制約を優先する記号として，これらの制約の優先度は $c_{00}^{t_i} \succ c_{01}^{t_i} \succ c_1$ とする．すなわち，保持制約よりも追跡制約を優先し，また，追跡制約のうち，観測対象物に 1 カメラ以上を割り当てる制約を優先する．

制約間の優先度を満足させるためには，優先度の高い制約にひとつ違反した場合の違反度よりも優先度の低い制約全てに違反した場合の違反度が下回る必要がある．

システム中のカメラ数を N ，システムが注視できる最大の観測対象物の数を M とし，違反度の値は以下の関係を満たすように決定した．

- $w_0^{c_0} \geq w_1^{c_0} * (M + 1)$

c_{01} の制約は，常に観測対象物数だけ存在する． M 個の観測対象物について，1 センサだけ割り当てることで c_{01} に違反する場合よりも 1 個の観測対象物について 0 センサを割り当てることで c_{00} に違反する方が違反度が大きい必要がある

- $w_0^{c_0} \geq w^{c_1} * (N + 1)$

c_1 の制約は，最大でセンサ数だけ存在する． N 個のセンサが全て観測対象物を変更し c_1 に違反するよりも，1 個の観測対象物について 1 センサだけ割り当てることで c_{01} に違反する方が違反度が大きい必要がある

第4章

提案手法を適用した観測システムの 実装

本章では、提案手法を実システムに適用するために必要な機能の詳細と実装方法について述べる。

4.1 全体の処理

本研究で実装する観測システムは、それぞれが独立したカメラエージェントによって構成される。ひとつのカメラエージェントはひとつのカメラを制御し、図 4.1 に示すように、対象物検出、位置推定、対象物同定、割り当て決定、カメラ制御を反復する。以下に各動作の概要を示す。

- 対象物検出

カメラから取得した画像の色形式を HSV 形式へ変換し、色の色相を主に用いることで明るさの変化に強い検出をおこなう。

- 座標推定

カメラ制御のため、注視する観測対象物の位置を画像中の位置とパン・チルト角から観測対象物の床位置を求める

- 対象物同定

色情報を用いて、他のエージェントが検出した観測対象物と自エージェントが検出した観測対象物との同定をおこなう。また、画像中の位置とパン・チルト角から観測対象物の床位置を求め、その平均位置を観測対象物の位置とする

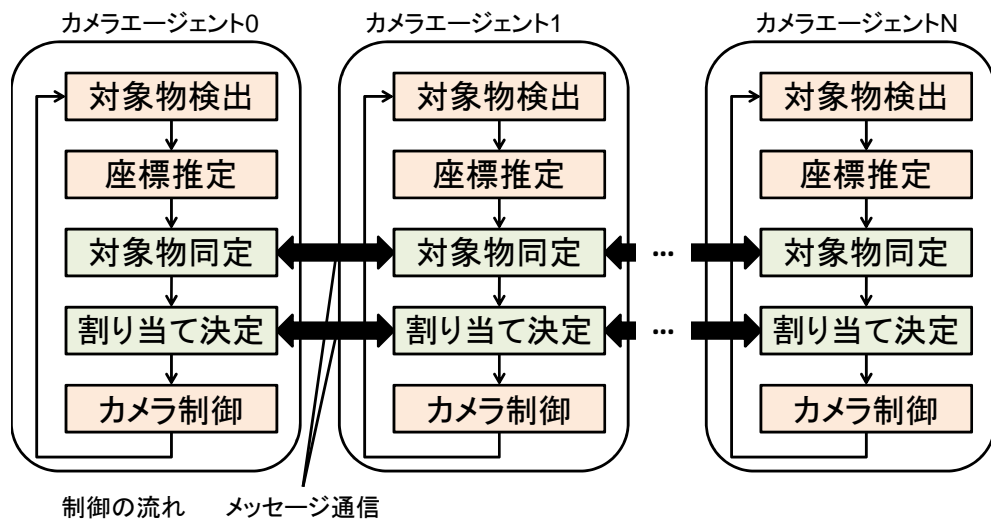


図 4.1: カメラエージェントの動作

- 割り当て

提案手法を用いてセンサ割り当てを決定する。

- 制約網の生成

各カメラエージェントは、変数、値域、評価関数の計算に必要な情報を収集する

- 解の探索

各カメラエージェントは実際に評価関数を計算し、解の探索をおこなう

- カメラ制御

観測対象物の床位置からパン・チルト角を求め、その角へカメラを制御する。この時、パン・チルト角の推定は対象物同定に用いる床位置の推定の逆変換である。

また、各カメラエージェント同士の情報交換はメッセージパッシングとして実装した。

4.2 対象物検出

本研究では、色の色相を用いて明るさの変化に強い対象物検出を行った。

また、色情報は観測対象物の同定の際に使用する。

表 4.1: 観測対象物とする色の範囲

色名	色相 (H)	彩度	明度
黄	30 ~ 60	0.3 ~ 1.0	0.3 ~ 1.0
青	190 ~ 230	0.2 ~ 1.0	0.2 ~ 0.6

前処理として, RGB から HSV へ変換する. HSV は色を

- H, 色相 (色の種類, 赤/黄/緑/水色/青/紫...)
- S, 彩度 (鮮やかさ)
- V, 明度 (明るさ)

で表す形式である.

RGB から HSV への変換式を以下に示す.

r, g, b はそれぞれ 1 画素の rgb 値で範囲は $[0..255]$, h, s, v は $0 \leq h < 360, 0 \leq s, v \leq 1$ の範囲である.

$$max = \max(r, b, g) \quad (4.1)$$

$$min = \min(r, g, b) \quad (4.2)$$

として,

$$h = \begin{cases} 0 & max = min \\ 60 * (g - b) / (max - min) & max = r \\ 60 * (b - r) / (max - min) + 120 & max = g \\ 60 * (r - g) / (max - min) + 240 & max = b \end{cases} \quad (4.3)$$

$$s = \begin{cases} 0 & max = 0 \\ (max - min) / max & otherwise \end{cases} \quad (4.4)$$

$$v = max \quad (4.5)$$

表 4.1 の条件を全て満たす最大の連結領域を観測対象物として検出する. また, 色が異なる領域は別観測対象物として認識する.

4.3 座標推定

カメラ座標系とワールド座標系の相互変換については座標変換の理論が存在するが、本研究ではより単純な線形補間をおこなう。

床面の点とその点を画像中心に捉えるパン・チルト角について、いくつかの対応点を取り、それらの線形補間を用いて、対応点にない値についてもパン・チルト角から床面座標、もしくは逆の床面座標からパン・チルト角への変換をおこなう。

この手法を用いる場合には、誤差を十分抑えるために、以下の条件を満たす必要がある

- 各対応点同士の間隔が十分に短いこと
- 各対応点の外側に対しておおきく外れた点を補間することがなく、対応点は床座標、パン・チルト各がとりうる値の範囲全体をある程度網羅すること

パン・チルト角から床面座標、もしくは逆の床面座標からパン・チルト角への変換のどちらも同じ変換式を使うので、以降ではパン・チルト角から床面座標へ変換する場合を例として用いる。また、この手法では X, Y 座標を独立に線形補間して求めるので、以降では X 座標を求める場合について記述する。すなわち、測定した対応点の組 $(p_i, t_i, x_i), i = 0, \dots, N$ のリストと求めるべきパン・チルト角 p_{in}, t_{in} から床面 X 座標 x_{out} を求める。

床座標、パン・チルト角を格子状に区切り、その交点で対応するパン・チルト各、床座標をとれば、求めたい点を囲む 4 点を用いて補間することができる。しかし、パン・チルト角から床座標を得たい場合は、測定誤差が大きくなるおそれがある。

よって、本システムでは求めたい値に近い 3 点によって決まる平面を用いて補間をおこなう。

近い 3 点は、 p, t に垂直に写像した場合の平面ユークリッド距離を用いる。入力を $P_{in} = (p_{in}, t_{in})$ 、サンプル点を $P_j = (p_j, t_j)$ とすると、その距離 $|P_{in} - P_j|$ が小さい順にサンプル点を 3 つ選ぶ。ただし、

- サンプル点同士が近い場合

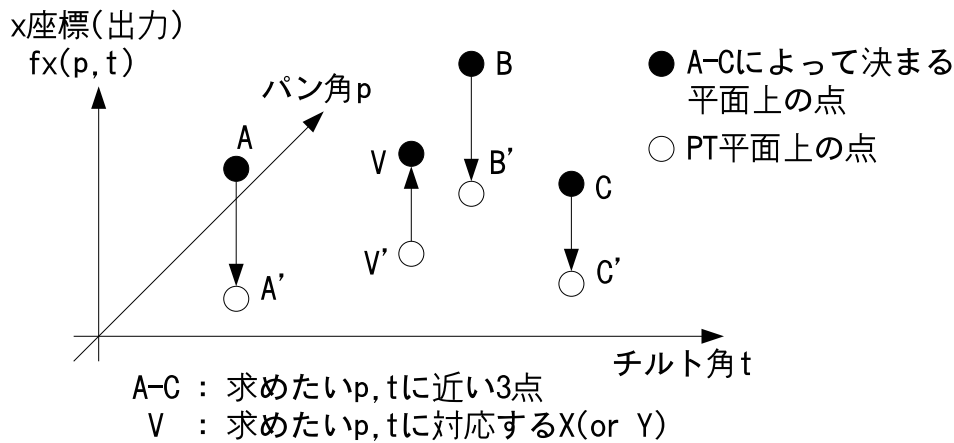


図 4.2: 近傍の測定点と求める値

- 3つのサンプル点が直線に並ぶ場合

については、3点による平面を決定できないので、次に近い点をとる。それらのサンプル点を以降では $A = (p_a, t_a, x_a)$, $B = (p_b, t_b, x_b)$, $C = (p_c, t_c, x_c)$ と記述する。 x_{out} の値は、 x_a, x_b, x_c の値に重みをつけて結合した値となる。実際の重みと式は以下のおりである。

パン・チルト角 p, t が決定すると、その時画像中心と床面との交点はただひとつに定まり、その際の X 座標を $fx(p, t)$ とする。図 4.2 のように、3点によって決まる平面上に、求めたい $V = (p_{in}, t_{in}, x_{out})$ があると仮定する。すなわち、次の式が成り立つ

$$\vec{AV} = \alpha \vec{AB} + \beta \vec{AC} \quad (4.6)$$

これらの式にあるベクトルを全て PT 平面へ垂直に降ろせば、次の式が成り立つ

$$\vec{A'V'} = \alpha \vec{A'B'} + \beta \vec{A'C'} \quad (4.7)$$

よって、 $\vec{A'B'}$, $\vec{A'C'}$ と内積をとれば、それぞれ次の式が成り立つ

$$\vec{A'V'} \cdot \vec{A'B'} = \alpha \vec{A'B'} \cdot \vec{A'B'} + \beta \vec{A'C'} \cdot \vec{A'B'} \quad (4.8)$$

$$\vec{A'V'} \cdot \vec{A'C'} = \alpha \vec{A'B'} \cdot \vec{A'C'} + \beta \vec{A'C'} \cdot \vec{A'C'} \quad (4.9)$$

以上の α, β に関する連立方程式は、容易に解くことができる。求めた α, β を 4.6 式へ代入することで、 \vec{AV} の値を求め、 V を求める。

4.4 対象物同定

システム全体で整合のとれた動作をするためには、カメラごとに検出した観測対象物について同定をおこなう必要がある。本研究では、検出の際に色情報を保持し、色情報を用いて観測対象物の同定をおこなう。

本研究で用いるカメラは、パン・チルト可能なカメラと固定カメラの2種類が存在する。固定カメラは、その視線方向を制御できないので、その画像中に映る観測対象物のみを注視可能とする。一方でパン・チルト制御が可能なカメラについては、現在の取得画像に写っていない観測対象物についても、カメラを制御することで注視が可能になる可能性がある。パン・チルトを考慮した注視可能な範囲を用いて注視可能性を判断するために、観測対象物の位置について合意をとった後、その位置を注視できるか各カメラエージェントが判断し、可視メッセージを送信する。

観測対象物の同定により、自エージェントがパン・チルト制御も含めて注視可能な観測対象物とその床位置について他エージェントと合意をとり、注視に必要なパン・チルト角を推定することが可能になる。

観測対象物の床位置は、観測した全てのエージェントが推定した床位置の算術平均を用いる。

観測対象物の床位置が判明すれば、前章により床位置とパン・チルト角の相互変換が可能であり、そのパン・チルト角が制御可能な範囲内である観測対象物を注視可能とする。

各カメラエージェントは、以下の手順で観測対象物の情報をまとめ、位置を推定する。観測対象物の情報は、(色, カメラ ID, 床位置) の組で表され、この観測対象物の情報を互いに交換することで注視可能な観測対象物についての合意を形成する。

- 観測対象物情報のリスト化をおこなう
 - － 自身が観測した色の観測対象物の情報を注視可能リストに追加する
 - － 自身が検出した観測対象物のリストを、メッセージ化する
- 他カメラの観測情報から、注視すべき観測対象物の位置推定および注視が可能か判断する

- 他カメラから受け取ったメッセージ中の観測対象物を注視可能リストに追加する
 - 他カメラが検出した観測対象物を色で分別し，それぞれ平均の床位置を求める．平均の床位置は別領域に保持する．
 - それぞれの色をもつ観測対象物について，以下の条件を満たす場合にカメラ制御により注視可能とする
 - * その色の観測対象物は自身で観測していない
 - * 自身のカメラはパン・チルト制御できる
 - * その位置を観測するために必要なパン・チルト角が制御可能な範囲にある
 - カメラ制御により注視可能な色の観測対象物の観測対象物の情報をメッセージ化する．
- 他カメラの注視可能な観測対象物と自分自身が観測可能な観測対象物をリストアップする

以上の手順をまとめた図を，図 4.3 に示す．

4.5 制約網の生成

これまでに得た情報から，制約網を生成する．

各カメラエージェントは，自身の割り当てを示す変数を作成する．

各カメラエージェントは，注視できる観測対象物リストのうち，自身が注視できる観測対象物をリストアップし，それらの観測対象物 ID，およびなにも割り当てないことを示す空集合を示す定数を値域としてリストに登録する．

各カメラエージェントは，自身が注視できる対象 t_i について評価関数 $f_{c_0 t_i}$ を計算できるようにするための情報を収集する．また，同様に評価関数 f_{c_1} を計算できるようにするための情報を収集する．それぞれの評価関数の計算に必要な情報を以下に示す

- c_0 の計算

自センサが注視できる観測対象物に対して，その観測対象物を注視できる他セ

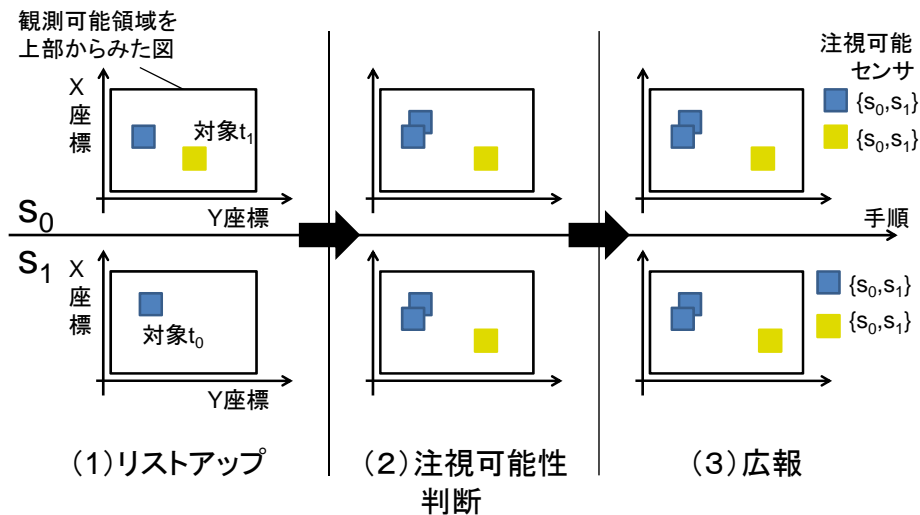


図 4.3: 観測対象物同定の手順

ンサの変数値が必要である．追跡制約の評価関数を各自計算するためには，自センサが注視できる観測対象物 t_j を同様に注視できるセンサについてのみ，メッセージの交換を行えばよい．提案システムでは，各センサの視野については既知であり，視野を共有する，同一の観測対象物を注視できるセンサは各自のセンサで計算できるものとする．

- c_1 の計算

現在のスナップショットにおいて注視できる観測対象物と，前問題で割り当てられた観測対象物との対応が必要である．各カメラエージェントは，現在注視している観測対象物が現スナップショットで注視できるかを，その色情報に基づいて判断する．同一の色情報を持つ観測対象物を，現問題における前問題の観測対象物であるとする．

4.6 解の探索

本提案システムでは，解の探索アルゴリズムとして DSTS[10] を用いた．DSTS は，以前に遷移した値へ遷移しない期間であるタブー期間をゼロとすることで，DSA[5] と

同様の動作をする．

センサ s_i に対する評価関数 F^{s_i} の計算について，疑似コードをアルゴリズム 1 に，実装したアルゴリズムをアルゴリズム 2 に示す．ただし， $P(x)$ を確率 x で *true*，確率 $1 - x$ で *false* となる述語とする．初期値としては，前問題の割り当てを示す値と同一の値を使用する．

また，DSTS は確率的に解を変更していくため，最終的なサイクルでの各変数値の組み合わせが，それまでに探索していた組み合わせのうちの最良のものである保証がない．本来であれば，これまでに探索してきた組み合わせのうち，最良のものを用以てシステムを制御すべきである．

探索済みの各変数値の組から最良の評価関数値を得られる変数の組を得ることが必要な点については，最良解の分散的な検出を行わず，あらかじめ決定しておいたリーダーエージェントが，全探索履歴における最良解を保持し，一定回数の探索後に放送するものとする．

リーダーエージェントによる最良解の計算は，自身の変数値をリーダーエージェントにも送信することで可能となる．解の探索のために，他の視野を共有するエージェントと自身の変数値を交換する際に，同時にリーダーエージェントへも送信することで，オーバーヘッドを隠すことができる．

Algorithm 1 評価関数値 v の計算手順

```

メッセージ ( $cID, tID$ ) を交換
for all ( $cID, tID$ ) do
   $N[tID] \leftarrow N[tID] + 1$ 
end for
for all  $N[i]$  do
  if  $N[i] = 0$  then
     $v \leftarrow v + w_{IGN_0}^{c_0}$ 
  else if  $N[i] = 1$  then
     $v \leftarrow v + w_{IGN_1}^{c_0}$ 
  end if
end for
if  $A_{s_i} \in D_{s_i} \wedge A_{s_i} \neq \emptyset \wedge x_{s_i} \neq A_{s_i}$  then
   $v \leftarrow v + w^{c_1}$ 
end if

```

Algorithm 2 DSTS

```

 $x_{s_i} \leftarrow a_{s_i}$ 
while  $k \leq DSTS\_LIMIT$  do
  メッセージを交換
   $V_{s_i} \leftarrow F^{s_i}(message\_list, x_{s_i})$ 
  if  $s_i = leader$  then
    if  $global\_min > F^{global}(message\_list)$  then
       $global\_min\_set \leftarrow message\_list$ 
    end if
  end if
   $min \leftarrow INT\_MAX$ 
  for all  $j \in A_{s_i} \wedge j \notin TABU\_LIST$  do
    if  $min > F_{all}(message\_list, j)$  then
       $min\_alloc \leftarrow j$ 
       $min \leftarrow F_{all}(message\_list, j)$ 
    end if
  end for
   $\Delta = now\_val - min\_val$ 
  TABU\_LIST を  $x_{s_i}$  で更新
  if  $\Delta \geq 0 \wedge P(p_1)$  then
     $x_{s_i} \leftarrow min\_alloc$ 
  else if  $\Delta < 0 \wedge now\_val > 0 \wedge P(p_2)$  then
     $x_{s_i} \leftarrow min\_alloc$ 
  end if
   $k \leftarrow k + 1$ 
end while

```

第5章

評価

提案手法を実験により評価した。実験は各カメラに対して、提案手法にもとづいて、画像入力、割り当ての決定、パン・チルト制御を行うカメラエージェントを構成した。

まず、解の探索パラメータを決定するための予備実験を行い、最適解までの収束に必要な探索サイクル数を評価した。

そして、実装した観測システムを用いて評価を行った。

5.1 システム構成

システムに使用したカメラ数、観測対象物の数を以下に示す

- カメラ数 N : 6
 - パン・チルト可能なカメラ数 : 4
 - 固定カメラ数 : 2
- 観測対象物の数 M : 2

また、設置したカメラの配置と、床面を注視可能なカメラ数で塗り分けた図を図 5.1 に示す。図 5.1 の 2, 3 番カメラは固定カメラであり、それ以外の 0, 1, 4, 5 番カメラがパン・チルト可能なカメラである。

5.2 解の探索のパラメータ

解の探索に用いるアルゴリズムである DSTS について、そのパラメータを予備実験により決定した。

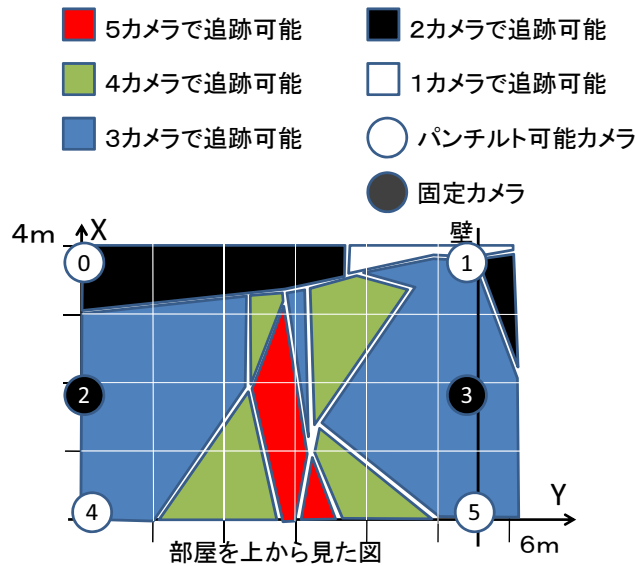


図 5.1: カメラの配置と注視可能なカメラ数

DSTS には、次の 3 パラメータが存在する

- 確率 p_1

自変数のみを考慮して、評価関数を改善する値へ変更する確率である。この確率が小さすぎる場合と大きすぎる場合の問題点を以下に示す。

- － 小さすぎる場合

小さすぎる場合には、値の変更回数に変数全体で少なくなるために、最適解への収束サイクルが増加する

- － 大きすぎる場合

大きすぎる場合には、値の変更が他変数の変更と衝突し、結果として評価関数の値が改善されない場合が増加する

- 確率 p_2

局所解から脱出するため、自変数の値をあえて評価関数値を増加させる値へ変更する確率である。この確率が小さすぎる場合と大きすぎる場合の問題点を以下に示す。

- 小さすぎる場合
小さすぎる場合には，局所解からの脱出に時間がかかる
- 大きすぎる場合
大きすぎる場合には，他のエージェントが評価関数の値を改善するのを待たずに値を変更してしまい，最適解への到達サイクルが増加する

- タブー期間

同じ解を繰り返し探索することを防ぐために，一度探索した解を記憶する期間の長さである．本提案システムでは，値域が最大 3 通りとなるため，長さは 1 もしくは 0 のみとなる．

また，期間の長さが 0 の場合は DSA と同一のアルゴリズムとなる．

これらのパラメータについて，形式化ごとに最適な値は異なるため，本提案手法を性能を評価する前に最適な値を推定する．

図 5.3 に示す制約網に対して，確率 p_1, p_2 を変化させて最適解への収束サイクルを計測したグラフを図 5.2 に示す．図 5.2 (a) は 確率 p_2 を固定して確率 p_1 を変化させた場合のグラフであり，図 5.2 (b) は 確率 p_1 を固定して確率 p_2 を変化させた場合のグラフである．

また，複数の問題で計測した DSA と DSTS の最適解到達サイクル数を図 5.4 に示す．図 5.2 の結果から，それぞれの手法における確率 p_1 , 確率 p_2 はそれぞれ $p_1=0.6$, $p_2=0.2$ とした。

図 5.4 から，DSTS よりも DSA を用いた方が高速に最適解を発見できている．これは，本提案手法を用いることにより生成される制約網が十分に単純であるということによるものである．

結果より，以降の実験では確率 $p_1=0.6$, 確率 $p_2=0.2$ の DSA を解の探索アルゴリズムとして用いる．

5.3 シミュレーションによる評価

以降では，シミュレーションにより本提案手法における解の探索の性能を測定する．

5.3.1 収束サイクルと探索空間の関係

一般に、分散制約最適化問題の難しさの指標として、一変数あたりの制約数である制約密度や、induced-width [3] が用いられる。

本研究では、これらの指標ではなく、探索空間における最適解の割合を用いて、問題の難しさを概算する。

追跡制約のみを考慮し、保持制約を考慮しない場合、本提案システムにおいて生成される分散制約最適化問題を解いた時の最適解の割合と最適解までの収束サイクルを図 5.5 に示す。

図 5.5 から、探索空間における最適解の割合が少ない問題の方が、求まる最適解の違反度に関係なくより長い収束サイクルがかかることがいえる。また、最適解の割合が 0.1 を越えるあたりから、収束に必要なサイクル数は減少しないているが、これは探索する問題が小規模であり、値の交換と探索に最小限必要なサイクル数に近づいたためと考えられる。

5.3.2 DSA による解の探索

本研究で用いる形式化により求まる、最小の最適解の割合を持つ問題を図 5.6 に示す。また、実システムでは 6 カメラが同時に観測できる領域が存在しないため、実際に作成される制約網のうち、最小の最適解の割合を持つ問題は図 5.7 に示す通りとなる。また、比較のため、より単純な問題である図 5.8 に示す制約網についても評価をおこなった。

これらの各問題について、1000 回の解の探索を行った場合に、最適解に収束したサイクル数の平均値および最悪値を表 5.1 に示す。

この評価から、最も最適解までの収束に時間がかかる問題についても、230 サイクルの探索をおこなうことで確実に最適解に到達することがいえる。

5.4 実システムによる評価

提案手法を実装した観測システムを用いた実験による評価を行った。

表 5.1: 解の探索に必要なサイクル数

問題	最適解の割合	平均探索サイクル	最悪探索サイクル
(a) 理論的に最難問	0.1%	57.771	431
(b) 实际的に最難問	0.3%	34.875	222
(c) 比較的に容易	25%	3.282	13

5.4.1 実験環境

実装したシステムは、図 5.9 に示すように、カメラとカメラサーバ、カメラエージェントを用いて構成した。

実装したシステムにおいては、各カメラは部屋の床中央付近を見下ろす形で設置した。カメラサーバは、各カメラエージェントとカメラの間で、画像の取得やカメラ制御などをより容易にする働きを持つ。

各カメラエージェントは同一の構成をもつ 2 つのマシンに分散して配置した。使用したマシン構成を以下に示す。

- OS: fedora 10 カーネル 2.6.27.
- CPU: Intel Core 2 Duo 2.66GHz
- メモリ: 2GB

また、各カメラエージェント間での通信は、所属するマシンごとにまとめて送受信する。

5.4.2 実行時間

本提案システムを 1000 回のスナップショットに対して実行した場合の各モジュールについての平均実行時間および最悪の実行時間を表 5.2 に示す。画像取得およびカメラ制御のモジュールで長い実行時間がかかっている原因は、各カメラエージェントとカメラ本体が離れて実装されているためである。

割り当ての決定には 1 サイクルあたり 1ms、総合して 200 ms 程度しか必要としないため、十分高速に解を発見できていることがいえる。

表 5.2: 各モジュールの実行時間

モジュール	平均実行時間	最悪実行時間
画像取得	230 ms	53 ms
観測対象物検出	36 ms	17 ms
観測対象物同定	132 ms	50 ms
割り当て決定	238 ms	195 ms
カメラ制御	803 ms	797 ms

表 5.3: 各観測対象物ごとの位置と注視できるカメラ

観測対象物の色	X 座標	Y 座標	注視できるカメラ ID
青	1.0m	4.2m	0,2,4
黄	2.1m	3.0m	0,1,3,4,5

5.4.3 単独のスナップショットに対するシステムの挙動

提案手法を適用した協調センサ網の挙動について、図 5.10 に示す配置を用いて実験をおこなう。

各カメラの入力を図 5.11 に示す。各センサの検出結果を総合し、各観測対象物は表 5.3 に示す位置と推定される。表 5.3 の状態を分散制約最適化問題へ形式化した図を 5.13 に示す。観測対象物の画像中位置や大きさは考慮していないので、カメラ 3 のように端に映るだけでも注視可能とされている。解を用いてカメラを制御した結果が図 5.12 である。パン・チルト可能なカメラはそれぞれ、割り当てられた観測対象物を画像中心へ捉える位置に視線方向を制御しているといえる。

5.4.4 連続したスナップショットに対するシステムの挙動

黄色、青色の対象物を図 5.14 に示すように床に配置し、移動させた場合の実行例を以下に示す。このときの、全体の評価関数の値、および各観測対象物を追跡したカメラ数を図 5.15 に示す。

図 5.15 では、1-230 サイクルまでが 1 番目のスナップショットであり、231 から 460

までが2番目のスナップショットである。全てのスナップショットで最終的な最良解では各観測対象物に2カメラを割り当てており、追跡制約を満足できている。

観測対象物の床位置が変化しても、各カメラが注視できる観測対象物が変化しないような2番目等のスナップショット(サイクル数231-460)では、解の探索は行われるものの、評価関数の値は0となる割り当てのみを探索している。これは保持制約によって、観測対象物を割り当てられているセンサは初期解から値を変更せず、観測対象物を割り当てられていないセンサのみが解の探索をおこなうためにおこる。

また、1, 4, 7番目の各スナップショットでは解の探索が行われている。各スナップショットにおける解の探索の原因と結果を以下に示す。

- 1番目のスナップショット

6カメラ全てが注視できる観測対象物を持つ。全カメラの割り当てが決定していないため、保持制約が全て無効な状態でカメラ割り当てをおこなう解を探索する。この時、問題の最適解の割合は25.0%であり、比較的容易な問題であるといえる。2観測対象物それぞれに2カメラを割り当て、使用しているカメラ数は4である。

- 4番目のスナップショット

5カメラが注視できる観測対象物を持つ。1カメラで、注視していた観測対象物が移動したために注視不可となった。この時、問題の最適解の割合は2.5%であり、比較的困難な問題であるといえる。別の観測対象物を注視していなかったカメラが新たに割り当てられ、使用しているカメラ数は4である。

- 7番目のスナップショット

6カメラ全てが注視できる観測対象物を持つ。1カメラにおいて、注視していた観測対象物が移動したために注視不可となった。この時、問題の最適解の割合は6.5%であり、比較的困難な問題であるといえる。その観測対象物を新たに注視できるカメラは全て他の観測対象物を注視していたが、そのうち1カメラの注視観測対象物を変更した。使用しているカメラ数は4である。

また、観測対象物を部屋中央付近でランダムに移動させ、提案システムを100回の

表 5.4: 同一観測対象物の連続注視スナップショット長

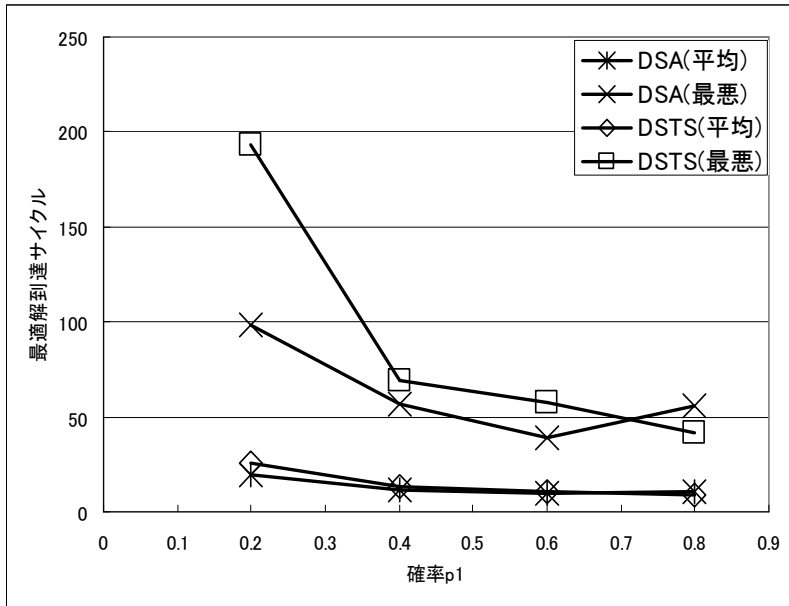
カメラ ID	パン・チルト制御	最長	最短	平均
0	可能	100	100	100
1	可能	97	97	97
2	不可	17	2	9.5
3	不可	13	7	9.7
4	可能	49	23	36
5	可能	96	3	49.5

表 5.5: 解の探索における最適解の割合

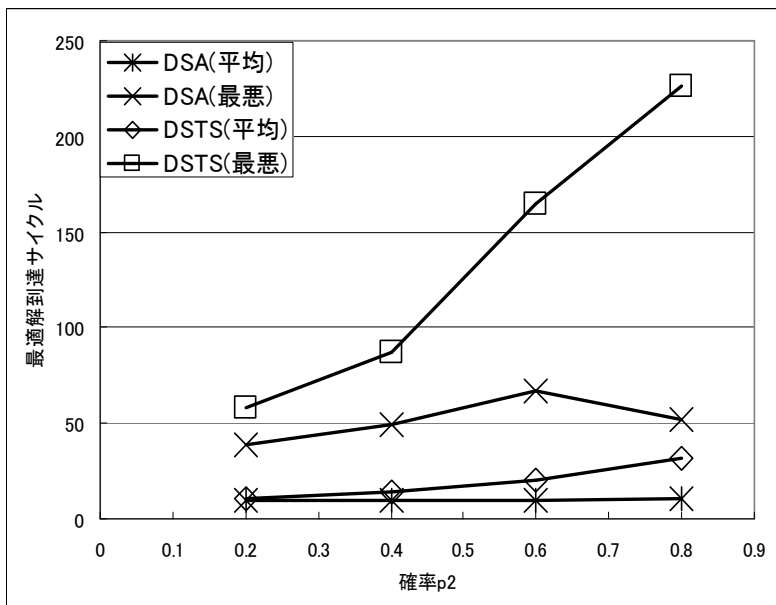
スナップショット番号	最適解の割合
0	27.3 %
3	3.7 %
4	0.4 %
23	2.4 %
30	2.0 %
47	2.4 %
51	0.4 %

スナップショットに対する割り当てを決定するまで動作させた場合の使用カメラ数を図 5.17 に示す。対象は図使用カメラ数が減少しているスナップショットにおいて、解の探索が発生していない理由は、2 カメラを用いて注視する割り当てが追跡制約を満足するためと考えられる。それぞれの解の探索における、最適解の割合を表 5.5 に示す。初期のスナップショットに対して作成される問題が他の問題と比較して簡単であるので、前問題の解によって解の探索の難易度が変化することがいえる。また、実際的に最も難しい問題の最適解の割合が 0.3%であることを考えると、比較的難易度の高い問題が作成されているといえる。

また、同一観測対象物を注視し続けたスナップショット長を表 5.4 に示す。全体的に、固定カメラは短く、パン・チルト可能なカメラは長くなっている。これは視野の広さが関係しているものと思われる。



(a) 確率 p1 による解の探索サイクルの変化



(b) 確率 p2 による解の探索サイクルの変化

図 5.2: パラメータによる解の探索サイクルの変化

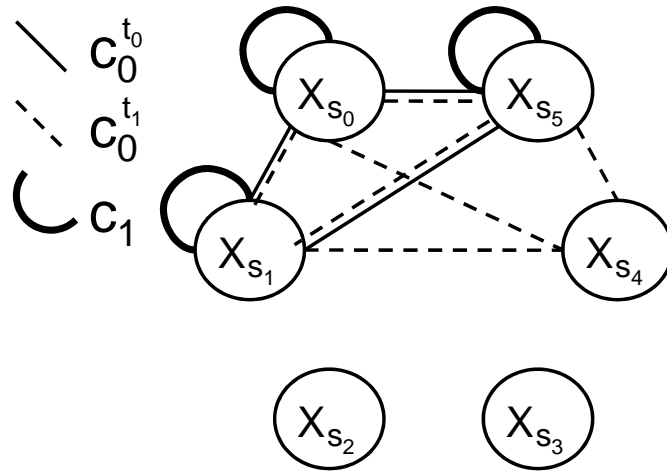


図 5.3: パラメータ決定用の制約網

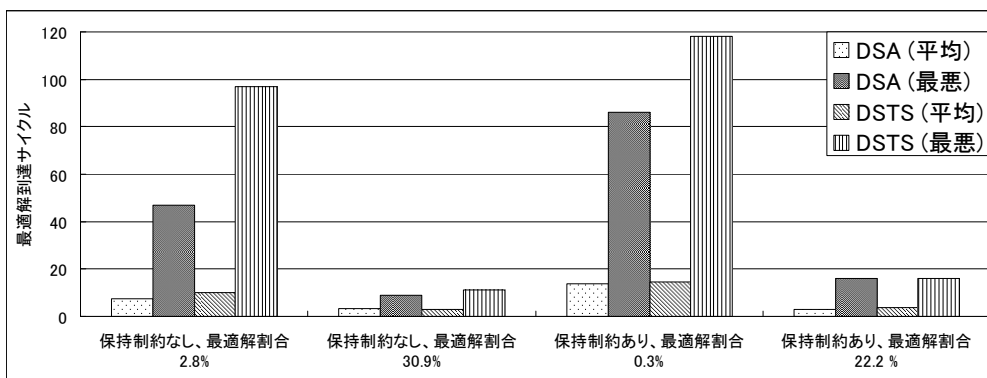


図 5.4: 問題ごとの最適解到達サイクル数

平均サイクル数

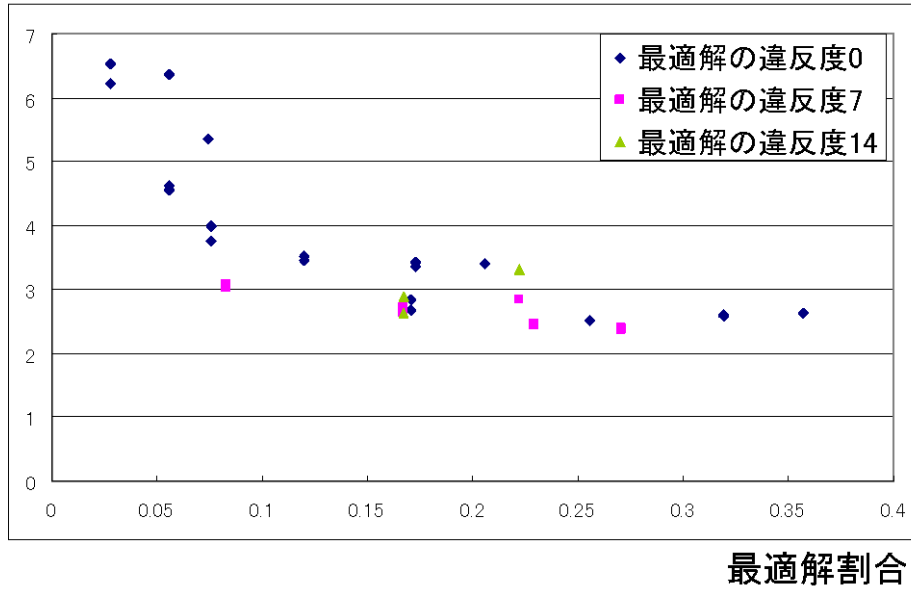


図 5.5: 収束サイクルと最適解の割合の相関

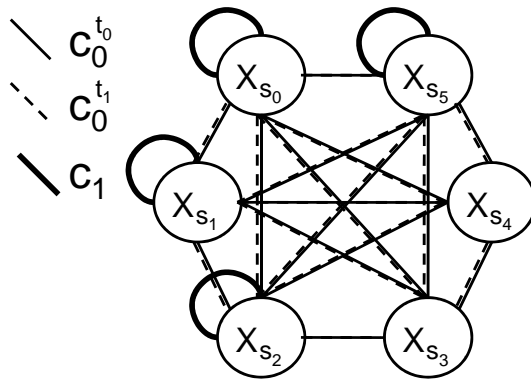


図 5.6: 評価対象とした制約網 (困難, 理論的)

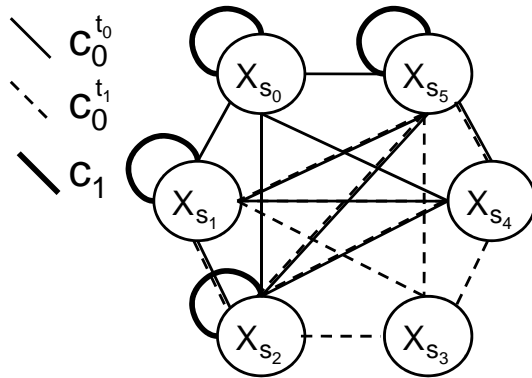


図 5.7: 評価対象とした制約網 (困難, 実際の)

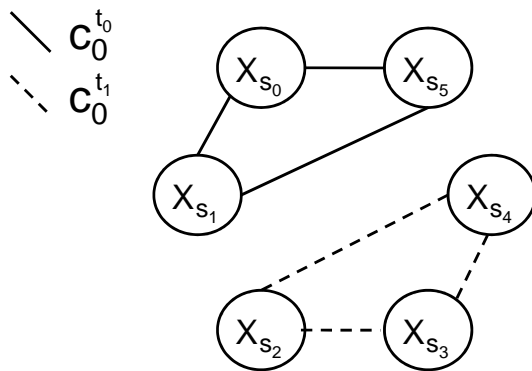


図 5.8: 評価対象とした制約網 (容易)

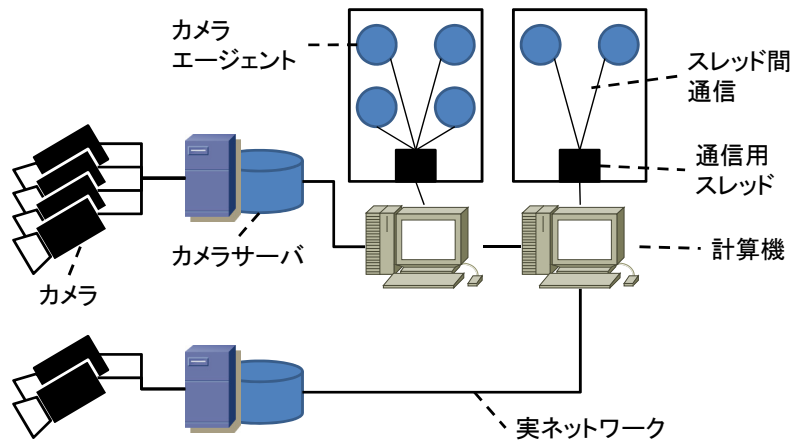


図 5.9: 実装システムの概要

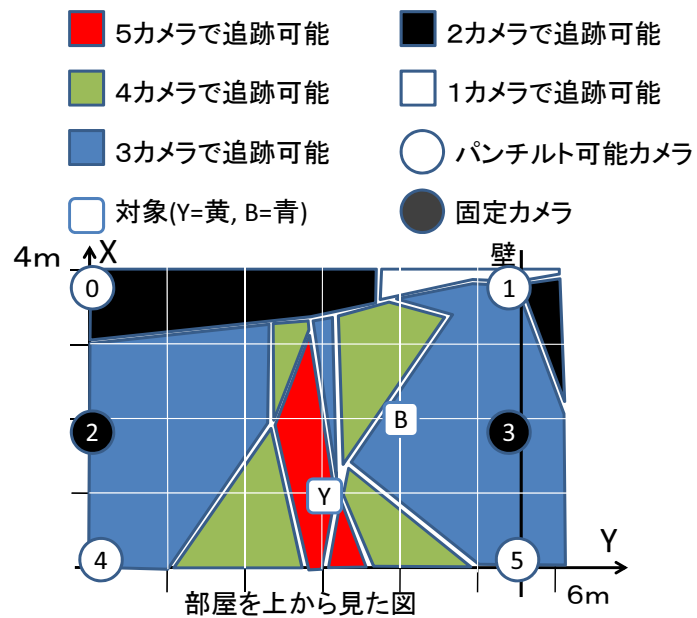


図 5.10: 観測対象物の配置



図 5.11: 入力画像



図 5.12: 制御後画像

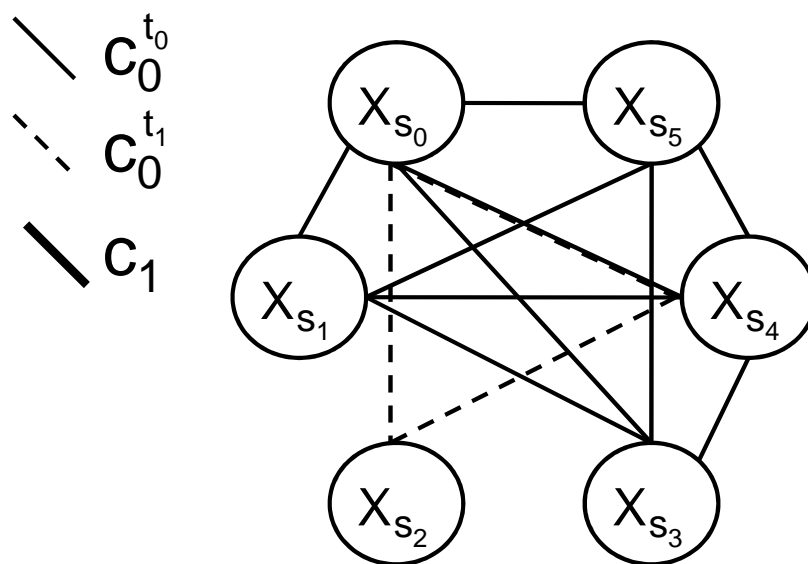


図 5.13: 生成される制約網

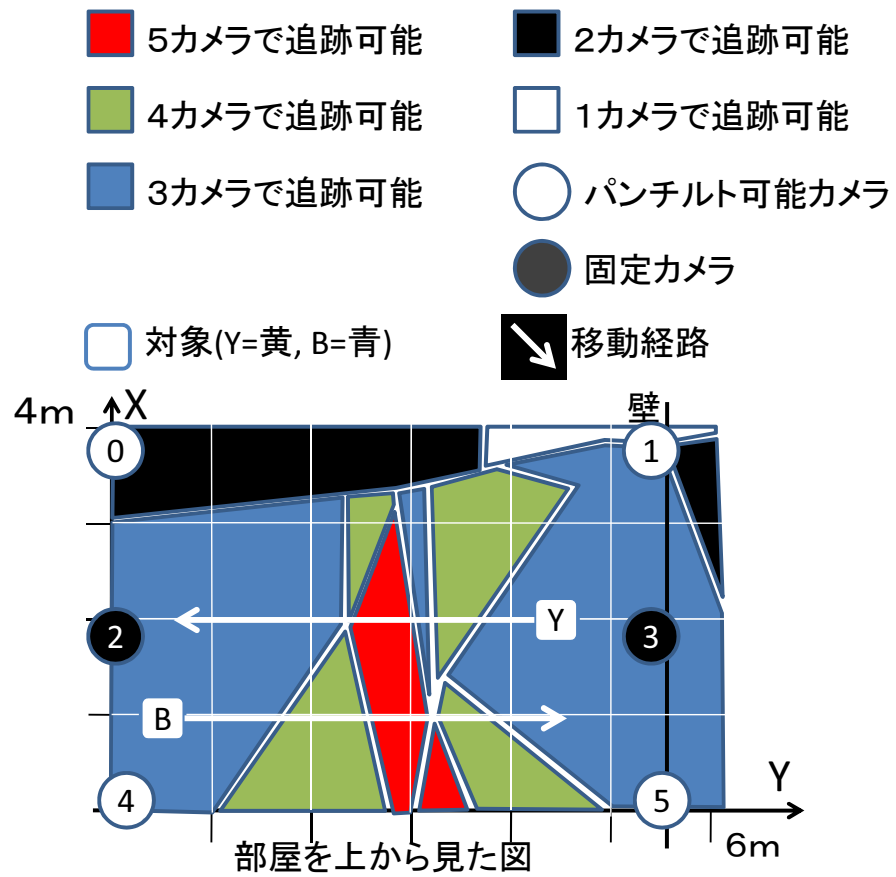
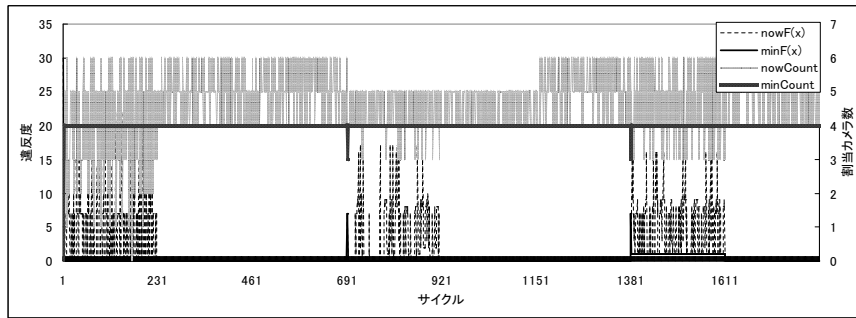


図 5.14: 観測対象物の移動経路



- 図 5.15: 各サイクルにおける違反度と使用カメラ数
- 5カメラで追跡可能
 - 2カメラで追跡可能
 - 4カメラで追跡可能
 - 1カメラで追跡可能
 - 3カメラで追跡可能
 - パンチルト可能カメラ
 - 固定カメラ

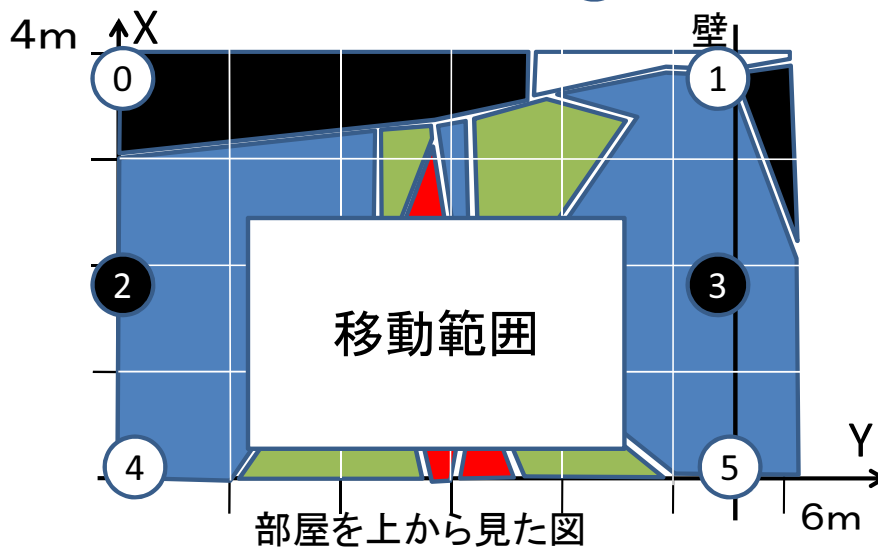


図 5.16: おおまかな対象の移動範囲

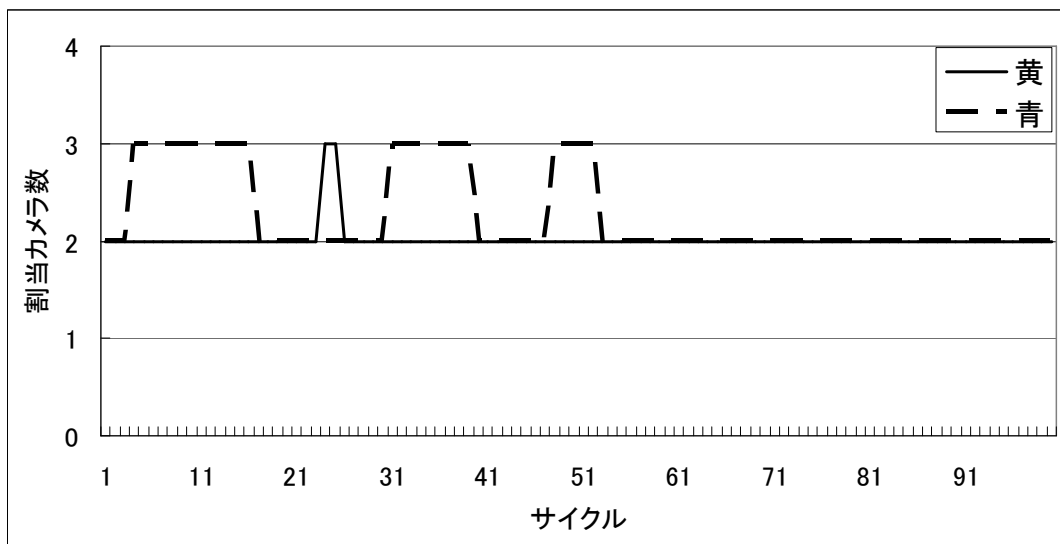


図 5.17: 注視に使用したカメラ数の推移

第6章

まとめ

本研究では，協調センサ網を分散制約最適化問題へ形式化する手法を提案した．

センサ間の合意を得るための形式化や合意を得るための問題を作成することなどを行わず，より単純な形式化をおこなうことで，問題の解空間としての規模を低減した．また，複数のスナップショット間で割り当てを引き継ぐための制約を追加し，初期解として前スナップショットにおける割り当てを用いることで，変化する環境対してもある程度の追従を可能とした．また，提案手法を実システムへ適用し，シミュレーションおよび実システムの両方で実験を行い，その有効性を確認した．

今後の課題としては，システム規模の大規模化に対する検討，解探索手法，形式化の変更による一層の分散協調的な割り当ての決定などがあげられる．

謝辞

本研究のために、多大なご協力をいただき、御指導を賜った名古屋工業大学の松尾啓志教授、津邑公暁准教授、齊藤彰一准教授、松井俊浩助教に深く感謝いたします。

また、本研究の際に多くの助言、協力をしていただいた松尾・津邑研究室の方々に深く感謝いたします。

また、本研究の一部は、科学研究費補助金(基礎研究C, 課題番号 21500073)によります。

参考文献

- [1] Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, Marina Yokoo, and Makoto Yokoo. Adopt: Asynchronous distributed constraint optimization with quality guarantees. *Artifl Intell*, Vol. 161, pp. 149–180, 2005.
- [2] Adrian Petcu and Boi Faltings. Dpop: A scalable method for multiagent constraint optimization. In *IJCAI 05*, pp. 266–271, Edinburgh, Scotland, Aug 2005.
- [3] Adrian Petcu and Boi Faltings. A scalable method for multiagent constraint optimization. *IJCAI*, pp. 266–271, Aug 2005.
- [4] RSJ2009. 外部カメラを用いたヒト型ロボットによるサッカー競技 RoboCup SSL Humanoid の提案と現状, 2009.
- [5] Weixiong Zhang, Ong Wang, , and Lars Wittenburg. Distributed stochastic search for constraint satisfaction and optimization: Parallelism phase transitions and performance. *PAS*, pp. 53–59, 2002.
- [6] 貝嶋浩次, 松井俊浩, 松尾啓志. 確率的な分散制約最適化手法を用いたセンサ網の資源割り当て手法の提案. *情報科学技術レターズ (FIT2007)*, Vol. 43, pp. 173–177, 2007.
- [7] 浮田宗伯. 能動視覚エージェント群の密な情報交換による 多数対象の実時間協調追跡. *信学論*, Vol. J88-D-i, No. 9, pp. 1438–1447, 2005.
- [8] 浮田宗伯, 松山隆司. 能動視覚エージェント群の情報交換による多数対象の実時間協調追跡. *情報処理学会 CVIM 研究会*, Vol. 43, pp. 64–79, 2002.

- [9] 末永俊一郎, 吉岡信和, 本位田真一. 無線センサネットワークにおける複数プログラムの動的配備. 情報処理学会, Vol. 50, No. 1, 2009.
- [10] 飯塚泰樹, 鈴木浩之, 竹内郁. 分散制約充足問題のための multi-agent tabu search 手法の効果. 信学論, Vol. J90-D, No. 9, pp. 2302-2313, 2007.
- [11] 太田和宏, 松井俊浩, 松尾啓志. 階層化された分散制約充足/最適化手法を用いた分散センサ網における観測資源割り当ての検討. 第8回情報科学技術フォーラム講演論文集 (FIT2009), 第2分冊, pp. 67-74, 2009.