

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

Sensor network resource allocation using distributed stochastic optimization method

貝嶋 浩次* 松井 俊浩* 松尾 啓志*

Kouji Kaishima Toshihiro Matsui Hiroshi Matsuo

1 はじめに

複数のセンサを用いて対象物(ターゲット)を多方向から観測する,分散協調処理に基づく観測システムに関する研究が行われている.このようなシステムは自動監視,遠隔講義などの様々な用途に応用可能であり,複数の対象物を観測するための基礎となる技術である.分散協調処理に基づく観測システムを実装するためには,複数のセンサを観測対象に割り当てる資源割り当て問題を解く必要がある.

一方で,分散協調システムにおける問題解決の基礎的な研究として,分散制約最適化問題(Distributed Constraint Optimization Problems;DCOP)[1]の研究が行われている.この問題を解くための手法には完全性が保証されている手法と確率的な手法があり,確率的な手法であるDSA(Distributed Stochastic Algorithm)[2],DSTS(Distributed Stochastic Tabu Search)[3]は大規模な問題に対して有効であることが示されている.

分散制約最適化手法を分散協調処理に基づく観測システムに適用した研究はこれまでに多く行われている[4, 5, 6].しかし,これらの研究では問題の規模が大きくなった時に解の探索時間が大幅に増加するため,実際の規模のシステムへの適用には課題がある.そこで,本稿では監視・自動撮影などを分散センサ網で行うことを目的とし,観測資源割り当て問題に対して解の探索時間の改善を行うため,確率的な分散制約最適化手法を適用する手法を提案する.

2 分散制約最適化問題(DCOP)

分散制約最適化問題(DCOP)は n 個の変数 x_1, x_2, \dots, x_n , 各変数が持つ有限で離散的な値の領域 D_1, D_2, \dots, D_n , 変数 x_i, x_j 間の制約 $c_{i,j} \in C_s$ および制約に対応した評価関数 $f_{i,j}$ からなる離散最適化問題である[1].変数と制約は複数のノードに分散して配置される.各ノードは他のノードとメッセージ通信を行い,評価関数の値を大域的に最適化する変数の値の組み合わせを求める.この時,制約 $c_{i,j}$ に対応した評価関数 $f_{i,j}$ から割り当てられた値に対するコストが各ノードで求められる.全ての制約に対するコストの総和を大域コストとし,大域コストを最小にする変数の値の集合が最適解となる.

制約網で表現される変数と制約は,変数をノード,制約をノード間のリンクとみなすことができる.各ノードに対してリンクで接続されたノードの集合をそのノードの近傍という.

3 分散協調処理センサ網

3.1 分散協調処理による複数対象物の観測

分散協調処理による複数対象物の観測ではターゲットを複数のセンサを用いて多方向から観測する[7].各センサには有限の視野があり,視野内にあるターゲットに対して追跡するセンサの集合を割り当てる.各センサはメッセージ通信を行い割り当てを決定する.この時,センサは視野,観測可能なターゲット数など有限な資源の中でセンサ間の割り当ての整合性を保つ必要がある.

3.2 DCOP への定式化

センサをグリッド状に配置した問題は,センサ網の基礎的な問題として有用であり,関連研究でも多く用いられている[5].図1にグリッド状のセンサ網の例を示す.この例ではセンサ $s_1 \sim s_6$ が各頂点に配置された環境にターゲット t_1, t_2 が存在する.各センサの視野は各センサに隣接する四角形の領域である.図1のセンサ s_1, s_2 の部分を定式化すると図2の様に表現できる.

図2の変数 T_j^i はセンサ i が知るターゲット j に関するセンサの割り当てを表す.例えば $T_1^1 = s_1s_2s_4$ の時,センサ s_1 は s_1, s_2, s_4 の3つのセンサが t_1 を追跡するように割り当てられていることを知る¹.

定式化した制約網には以下の3種類の制約が存在する.
intra-agent 制約 センサ i が知る変数 T_j^i が持つ制約で,

同時に複数のセンサを追跡できないことを表す.変数 T_j^i が持つ intra-agent 制約と評価関数 $c_{intra(i,j)} \in C_{intra}$, $f_{intra(i,j)}$ は次の通りである.ただし, $\phi(i)$ はセンサ i が持つ変数の中でセンサ i が追跡する割り当てを取っている変数の数, $weight_{intra(i,j)}$ は変数 T_j^i が持つ intra-agent 制約の重みである.

$$c_{intra(i,j)} := \{\phi(i) \leq 1\}$$
$$f_{intra(i,j)} = \begin{cases} weight_{intra(i,j)} & (\overline{c_{intra(i,j)}}) \\ 0 & (c_{intra(i,j)}) \end{cases}$$

¹各センサの視野内にあるターゲットに関する情報は,近傍のセンサと事前に交換されるものとする.このような前提は関連研究[4, 5, 6]と同様である.

*名古屋工業大学 Nagoya Institute of Technology

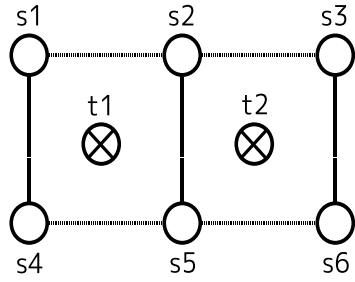


図 1: 配置例

inter-agent 制約 同じターゲット j に関する変数 T_j^i , T_j^k 間の制約で, 同じ割り当てを取らなければならないことを表す. 変数 T_j^i が持つ inter-agent 制約と評価関数 $C_{inter(i,j),(k,j)} \in C^{inter}$, $f_{inter(i,j),(k,j)}$ は次の通りである. ただし, $weight_{inter(i,j),(k,j)}$ は変数 T_j^i が持つ inter-agent 制約の重みである.

$$C_{inter(i,j),(k,j)} := \{T_j^i = T_j^k\}$$

$$f_{inter(i,j),(k,j)} = \begin{cases} weight_{inter(i,j),(k,j)} & (\overline{C_{inter(i,j),(k,j)}}) \\ 0 & (C_{inter(i,j),(k,j)}) \end{cases}$$

ignore 制約 ターゲットを追跡しているセンサの個数に関する制約で, n 個以上のセンサで追跡する値を取らなければならないことを表す. 変数 T_j^i が持つ ignore 制約と評価関数 $C_{ignore(i,j)} \in C^{ignore}$, $f_{ignore(i,j)}$ は次の通りである. ただし, $\varphi(T_j^i)$ は変数 T_j^i に割り当てられた値が示すターゲット j を追跡するセンサの数, $weight_{ignore(i,j)}$ は変数 T_j^i が持つ ignore 制約の重みである.

$$C_{ignore(i,j)} := \{\varphi(T_j^i) \geq n\}$$

$$f_{ignore(i,j)} = \begin{cases} weight_{ignore(i,j)} & (\overline{C_{ignore(i,j)}}) \\ 0 & (C_{ignore(i,j)}) \end{cases}$$

C^{intra} はセンサの資源, C^{inter} はセンサ間の割り当ての整合性を表す制約である. C^{intra} , C^{inter} が充足されない観測資源の割り当ては実行不可能であるため C^{intra} , C^{inter} が充足している割り当てを求めなければならない.

4 確率的な分散制約最適化手法

分散制約最適化手法には, 完全性を持つ手法である Adopt[6], DPOP[8], 確率的な手法である DSA[2], DSTS[3] などが提案されている. 完全性を持つ手法は与えられた問題の最適解を求めることができる. しかし, 変数や制約が多い問題では探索時間, またはメッセージのサイズが指数関数的に増加するため適用可能な問題に制限がある. Adopt は分枝限定法に基づく解探索手法であるため, 最大次数のノードを優先して生成された「深さ優先探索木」の深さが大きい場合, 解の探索時間が大きくなる可能性が高い. 一方, 確率的な手法は求められ

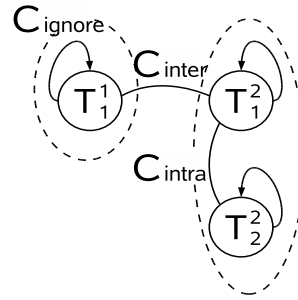


図 2: 制約網の表現

た解が最適解である保証はないが, 変数や制約の多い問題における探索時間を完全性を持つ手法に比べ改善できることから大規模な問題への適用が期待される.

DSTS(Distributed Stochastic Tabu Search)[3] は DSA(Distributed Stochastic Algorithm)[2] にタブーサーチを組み込んだ確率的な手法の 1 つである. 各ノードはメッセージ通信により得る近傍ノードが持つ変数の値より, コストの改善量 Δ の最大値とその時の変数の値を求める. Δ の最大値が $\Delta \geq 0$ となる場合は, 確率 $p1$ で変数の値を変更する. 値を変更した場合はタブーリストへ遷移を一定期間 (タブー期間) 禁止する値を書き込み², 変更後の値をメッセージとして近傍へ送信する. Δ の最大値が $\Delta < 0$ となる場合は, 制約違反が存在する場合のみ確率 $p2$ で値を変更する. 値を変更した時の処理は $\Delta \geq 0$ の時と同じである.

5 提案手法

5.1 確率的な分散制約最適化手法の適用

DSTS は評価関数から求められる局所的なコストと改善量 Δ を基に変数の値を変更することで, 大域的な改善を行い全体の割り当てを求め. 3.2 節で示した定式化より, 変数 T_j^i に割り当てられた値に対する評価関数 $f_{(i,j)}$ は以下の式で示される. ただし, $Neighbor(i,j)$ は変数 T_j^i 間と C^{inter} を持つセンサの集合である.

$$f_{(i,j)} = f_{intra(i,j)} + \sum_{k \in Neighbor(i,j)} f_{inter(i,j),(k,j)} + f_{ignore(i,j)}$$

センサの資源割り当て問題は C^{intra} , C^{inter} が充足する解を求めなければならない. しかし, DSTS は制約違反が存在する限り値の変更を行うため, ターゲットの数に対し割り当てに必要なセンサが不足している場合は全ての制約を充足できないため解を求めることができない. そこで, DSTS のパラメータ $p2$ を用いて変数の値を変更するための条件を, $\Delta < 0$ かつ hard constraint (C^{hard}) の違反が存在する場合とする. ただし, C^{hard} は次のよ

²タブーリストに書き込む値, 期間の方式は問題に適応するように選択する [3].

うに定義する .

$$\begin{cases} \bullet C^{hard} = C^{intra} \cup C^{inter} \\ \bullet cost_{intra(i,j)}, cost_{inter(i,j),(k,j)} \gg cost_{ignore(i,j)} \end{cases}$$

ここで, $C^{hard(i,j)} = C^{intra(i,j)}, C^{inter(i,j),(k,j)} \in C^{hard}$ ($k \in Neighbor(i,j)$) を充足した変数 T_j^i に対する DSTS の処理について議論する . $\Delta < 0$ の場合は $C^{hard(i,j)}$ の違反がある場合のみ変更するため変数 T_j^i の値の変更は行われない . また, $C^{hard(i,j)}$ を充足している変数 T_j^i が別の値に変更すると $C^{inter(i,j),(k,j)}$ を違反するため $\Delta \geq 0$ にはなりえない . $C^{hard(i,j)}$ を充足している変数 T_j^i は値の変更を行わないことから, 求めた割り当ては必ず C^{hard} を充足する . よって, この手法を用いて求めた割り当てはシステム上で実行可能である .

5.2 制約の重み付け

変数 T_j^i が持つ各制約の数はセンサの視野と視野内にあるターゲットの数に依存する . あるターゲットが見えているセンサが m 個であるとすると, 変数 T_j^i が持つ制約は C^{intra} が高々1個, C^{inter} が $(m-1)$ 個, C^{ignore} が1個である . C^{intra} と C^{inter} が同等の評価値を持つためには各制約の個数を考慮した重み付けが必要である . また, C^{intra} が充足するためには制約違反が存在する時に $\phi(i)$ が減少する方向に遷移させる必要がある . このことから, 変数 T_j^i が持つ制約 C^{intra} の重み $weight_{intra(i,j)}$ を以下のように設定する . ただし, C^{inter} の重みは全て同じであるとする .

$$\begin{aligned} weight_{intra(i,j)} &= (\phi(i) - 1) \sum_{k \in Neighbor(i,j)} weight_{inter(i,j),(k,j)} \\ &= (m-1)(\phi(i) - 1)weight_{inter(i,j),(k,j)} \end{aligned}$$

図2の T_1^2 について考えると $m = 4$, $\phi(2) \leq 2$ より $weight_{intra(2,1)} = 3weight_{inter(2,1),(k,1)}$ ($k = 1, 4, 5$) となる .

6 評価

提案手法の有効性をシミュレーションを用いて評価した . 図3の様にセンサをグリッド状に配置した環境で評価を行った³ . 制約網が一連になるようにターゲットの数を10,15,20,25と変化させ, サイクル数, 解の精度, センサの稼働数を測定した . サイクルとはエージェントが近傍からのメッセージを受信し, メッセージの処理を行い, 必要ならばメッセージを近傍に送信するまでの動作を示した単位である . 以上の動作を全てのエージェントが行うまでを1サイクルとし i サイクル目に近傍エージェントに送信したメッセージは $(i+1)$ サイクル目に受信するものとした . 解の精度は n 個以上 (今回は $n = 3$) のセンサで追跡されているターゲットの数, 稼働数はター

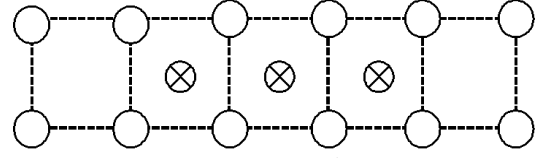


図3: センサとターゲットの配置

表1: ターゲットの数と最適解

ターゲットの数	変数の総数	最適解	
		解の精度	稼働数
10	40	7	22
15	60	10	32
20	80	14	42
25	100	17	52

表2: パラメータの設定値

パラメータ	設定値
a · Num(D)=5	p1=0.50, p2=0.10, タブー期間=1
b · Num(D)=5	p1=0.90, p2=0.20, タブー期間=1
c · Num(D)=16	p1=0.95, p2=0.20, タブー期間=1
d · Num(D)=16	p1=0.90, p2=0.30, タブー期間=1

ゲットを追跡しているセンサの数とした . ターゲットの数と解の精度, センサの稼働数の最適解は表1の通りである . 解を発見するか1000サイクル経過するまでを1回の試行とし, 各パラメータで1000回試行した . 1000サイクル経過しても解を発見できない場合は試行を打ち切り, 1000サイクル, 解の精度0, 稼働数0としてカウントした . また, 値を変更した時にタブーリストへ書き込む値は変更前の値とした .

今回の評価では, 変数の領域による評価を行った . 変数の領域は Num(D)=5, Num(D)=16 の2種類で Num(D) は領域の要素数である . Num(D)=5 は探索領域を制限した問題の形式化であり, 関連研究 [4, 6] で用いられている . Num(D)=5 の変数の領域 D は $D = \{s1s2s3, s1s2s4, s1s3s4, s2s3s4, IGNORE\}$ である . $IGNORE$ はそのターゲットに対する追跡割り当てを行わないことを表す . 一方, Num(D)=16 は探索領域を制限しない問題の形式化である .

実験結果を図4,5,6に示す . 図4は平均サイクル, 図5,6はそれぞれ最適解を1としたときの解の精度, センサの稼働率である . パラメータ a,c は解の精度が最良のもの, パラメータ b,d は平均サイクルが最小である Num(D)=5,16 ごとの結果で, 設定値は表2の通りである . なお, パラメータのタブー期間はサイクル数で示す .

ターゲットの数が25の試行において, 変数の領域を制限しなかったパラメータ c,d の平均サイクル数は, 制限した中で最短であった b · Num(D)=5 の2.1~2.7倍に増加した . これは変数の領域を制限しない探索は探索空

³このセンサとターゲットの位置は文献 [4] に従った . なお, 各ターゲットの重要度は同一とした .

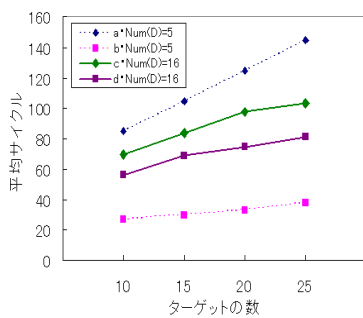


図 4: 平均サイクル

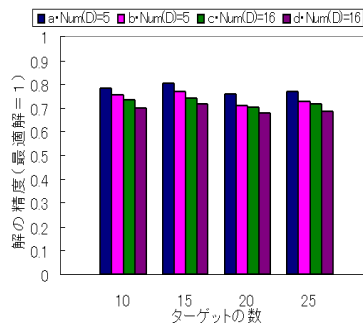


図 5: 解の精度

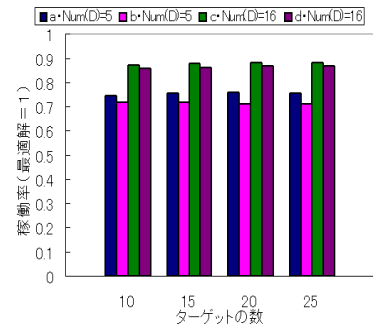


図 6: センサの稼働率

間が大きいためである。一方、ターゲットの数を 10 から 25 に増やした時の探索時間の増加量は 40 ~ 70%程度であった。

変数の領域を制限した探索では解の精度は最適解の約 75%であったがセンサの稼働率は約 70%程度であった。一方、変数の領域を制限しない探索では解の精度は最適解の約 70%であったが、センサの稼働率は約 90%であった。この結果は、変数の領域を制限した探索では *IGNORE* 以外の割り当てがなされたターゲットに関する変数は必ず C_{ignore} を満足するため解の精度が高くなるが、センサの数を限定した割り当てしか行えないためセンサの稼働率が減少したためであると考えられる。変数の領域を制限しない探索では、センサの数に依存しない割り当てが行えるため高い稼働率を得られるが、追跡するターゲットが増加したことにより資源が分散してしまうため C_{ignore} の違反数が増加したと考えられる。ターゲットの数を 10 から 25 に増やした時の解の精度の減少量は 2 ~ 3 ポイント、センサの稼働率の減少量は 1 ポイント未満であった。また、図には示していないがターゲットの数が 25 の試行に対して解の発見率は表 2 に示した全てのパラメータで 100%であったことから、問題の規模を大きくした時の影響は小さい。

これらの評価結果から確率的な手法を適用するための提案手法は有効であると考えられる。

7 まとめ

複数のセンサを用いた分散協調処理に基づく観測システムに確率的な分散制約最適化手法である DSTS を適用し、大規模な問題に対応する手法を提案した。探索時間、解の精度、センサの稼働率の面から提案手法の有効性を確認した。今後の課題として、ターゲットが移動する環境などのさらなる実際的な問題への適用や分散センサ網のための協調プロトコル [9] との比較・検討が挙げられる。

参考文献

[1] A. Petcu. "A Framework for Distributed Constraint Optimization". Proc. *Technical Report (EPFL)*,

No. 2006/001, 2006.

- [2] W. Zhang, G. Wang, and L. Wittenburg. "Distributed Stochastic Search for Constraint Satisfaction and Optimization". In *AAAI-02 Workshop on Probabilistic Approaches in Search*, pp. 53–59, 2002.
- [3] 飯塚泰樹, 鈴木浩之. "Tabu Search を用いたマルチエージェント型分散制約充足手法の提案". 電子情報通信学会技術研究報告, 2006.
- [4] S. Ali, S. Koenig, and M. Tambe. "Preprocessing Techniques for Accelerating the DCOP Algorithm ADOPT". In *AAMAS*, pp. 1041–1048, 2005.
- [5] R. Bejar, C. Domshilak, C. Fernandez, C. Gomes, B. Krishnamachari, B. Selman, and M. Valls. "Sensor networks and distributed CSP". *Artificial Intelligence*, Vol. 161, pp. 117–147, 2005.
- [6] P. J. Modi, W. M. Shen, M. Tambe, and M. Yokoo. "ADOPT: Asynchronous Distributed Constraint Optimization with Quality Guarantees". *Artificial Intelligence*, Vol. 161, No. 1-2, pp. 149–180, 2006.
- [7] R. Bejar, B. Krishnamachari, C. Gomes, and B. Selman. "Distributed Constraint Satisfaction in a Wireless Sensor Tracking System". In *the IJCAI-2001 Workshop on Distributed Constraint Reasoning*, pp. 81–90, 2001.
- [8] A. Petcu and B. Faltings. "DPOP: A Scalable Method for Multiagent Constraint Optimization". In *IJCAI*, pp. 266–271, 2005.
- [9] 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". In 電子情報通信学会論文誌, Vol. J88-D-i, No. 9, pp. 1438–1447, 2005.