

## 分散制約最適化問題の非厳密解法の改良に関する研究

学籍番号 23413574 氏名 柳田 大輝

指導教員名 松尾 啓志

## 1 はじめに

近年、複数の自律的なエージェントが協調して動作するマルチエージェントシステムが注目されている。マルチエージェントシステムにおける協調問題は、分散制約最適化問題として定式化できる。

分散制約最適化問題を解くアルゴリズムは、厳密解法と非厳密解法に分類される。本研究では、最適解を必ず発見できるとは限らないが計算量やメッセージサイズなどを比較的小さく抑えることができる非厳密解法に注目する。非厳密解法の中でも、KOPT アルゴリズム[1]とDUCTアルゴリズム[2]という2つのアルゴリズムに注目する。

KOPTアルゴリズムは、 $k$ -Optimalと呼ばれる最適性の指標にそった解を得ることができる。このアルゴリズムには、得られる解の品質と探索時間にトレードオフが存在する。これに対し、複数のパラメータを設定した探索を同時に実行することで、解品質と探索時間の抑制を両立する手法を提案する。

DUCTアルゴリズムは、サンプリングを行うことにより、精度のよい解が得られることが期待される。しかし、問題グラフにより、エージェントが使用するメモリ量が大きくなる問題がある。この問題を解決する方法として2つの手法を提案する。

## 2 分散制約最適化問題とその解法

分散制約最適化問題は、制約最適化問題の変数をエージェントに分散して配置したものである。 $n$  個の変数  $x_1, \dots, x_n$  と制約の集合から構成される。各制約には評価関数  $f_{i,j}$  が存在し、変数  $x_i, x_j$  の割当てに応じて評価値を返す。目的は、すべての制約についての評価値の和が最大または最小となるような、変数値の割当て  $a$  を求めることである。

## 3 従来手法：KOPT アルゴリズム

KOPT アルゴリズムは、任意の  $k$  についての  $k$ -Optimal な解を得られるアルゴリズムである。 $k$ -Optimal な解とは、 $k$  個以下のエージェントが持つ変数値を変更しても評価値が向上しない解である。

動作は、周囲のエージェントと情報の交換、エージェントがグループを構成し最適な割当てを計算、グループ内のエージェントで同意を形成し変数値を更新という3つのフェーズを繰り返す。

KOPT アルゴリズムでは、 $k$  をエージェント数  $n$  に近づけるほど、最適解に近い解が得られるが、 $k$  を大きくすると、メッセージ通信ステップ数と計算量が増える。また、合意を形成することが難しくなる。そのため、評価値の向上が遅くなると考えられる。

## 4 KOPT アルゴリズムを改良する

## 提案手法：探索の多重化

既存の KOPT アルゴリズムは、パラメータ  $k$  を1つだけ選択して実行する。これに対して、本研究で提案する多重化を行う手法では、複数のパラメータを選択し、並列に実行する。そして、一定の通信ステップ数毎に、並列実行したそれぞれの探索結果のうち、最も高い評価値を獲得した変数の割当てを選択する割当ての同期を行う。同期処理に必要な通信ステップ数は、グラフの直径に比例する。この同期に必要な通信を探索に必要な通信とオーバーラップさせることで、同期に必要な通信ステップ数を隠ぺいする。ただし、パラメータ  $k$  のうち最大の  $k_{max}$  とグラフの直径  $d$  の関係は、次の式を満たす必要がある。

$$d \leq 2 \times \left\lfloor \frac{k_{max}}{2} \right\rfloor + k_{max} + 1$$

## 5 評価：KOPT 探索の多重化の効果

提案する探索の多重化による効果を不規則な構造を持つグラフを対象に、既存の KOPT アルゴリズムとの比較を行った。(図1)

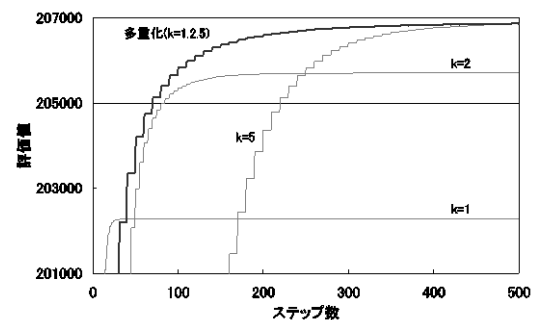


図1: ステップ数に対する評価値の推移

既存の KOPT アルゴリズムに対して、評価値が早く向上し、最終的に得られる解は  $k_{max}$  を選択した既存手法と同等であった。また、図2に示す通りオーバーラップにより、評価値の向上が早くなることを確認した。

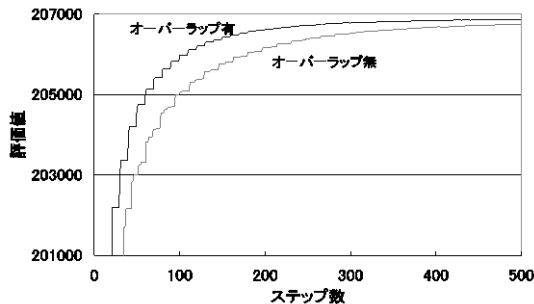


図2: オーバーラップの効果

## 6 従来手法: DUCT アルゴリズム

前処理として、深さ優先探索などを用いて制約網から擬似木を構成する。擬似木上の根に位置するエージェントから順に変数値を選択し子へ伝える。続いて、葉側から順に評価値を計算し根へ向かって伝える。この操作を繰り返す。変数値を選択する際には、コンテキストと呼ばれる過去の変数割当てに対する評価値からバウンド(評価値の境界)を計算し、選択基準とする。この過去の評価値を保存する量は、問題の規模に依存し、また、根に近いエージェントほど多くなる。

## 7 DUCT を改良する提案手法 1:

### コンテキスト中の必要な変数値の保存

保存するコンテキスト量を減らすために、グラフの構造から、冗長な値を判断し、必要な変数値のみを保存する手法を提案する。

## 8 DUCT を改良する提案手法 2:

### 保存するコンテキスト量の制限

保存するコンテキスト量を一定以下に抑える手法を提案する。一定量を超えた場合には、規則に従って保存されているコンテキストをメモリから追出す。追出す選択基準として、コンテキストの受信回数、過去の最大バウンドを提案する。

## 9 評価: DUCT のコンテキスト中の

### 必要な変数値の保存による効果

グラフ構造別のエージェントの深さと保存する変数の数の関係を比較した結果を図3に示す。既存手法では、深さに比例して保存する変数の数が増加する。

これに対して提案手法では、ある深さをピークに、保存する変数の数が減少する。得られる解の精度は同等である。

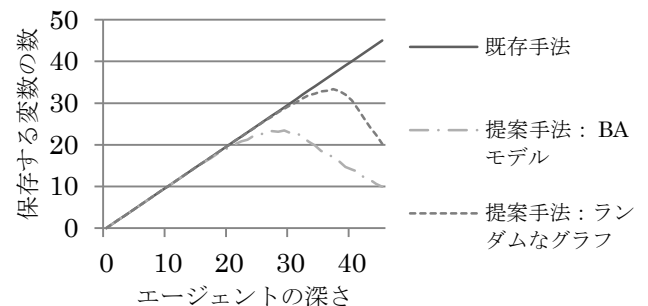


図3: グラフ構造による削減効果の違い

## 10 評価: DUCT の保存するコンテキスト量の制限

保存するコンテキスト数を10から2000まで変化させ、追い出し基準ごとに得られる解コストを比較した。図4は、エージェント数10のBAモデルで生成したグラフに対する結果である。ある程度までコンテキスト数を制限しても、解コストの大きな悪化はみられなかった。

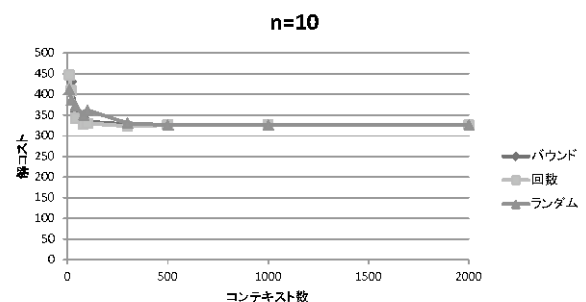


図4: 保存コンテキスト数と解コスト

## 11 おわりに

2種類の非厳密解法に対する改良を提案し、計算機シミュレーションにより評価した。KOPTに対しては、解の精度を落とさず探索速度が向上することを確認した。DUCTに対しては、メモリ量の削減と、既存の指標のみによらない改善の余地があることを示した。

## 参考文献

- [1] Katagishi, H. and P. Pearce, J.: "KOPT: Distributed DCOP Algorithm for Arbitrary k-optima with Monotonically Increasing Utility", DCR07
- [2] Ottens, B., Dimirakakis, C. and Raltings, B.: "DUCT: An Upper Confidence Bound Approach to Distributed Constraint Optimization Problems", AAAI12