RF-003

分散制約最適化問題の異なる非厳密解法の統合手法
Integrating Different Incomplete DCOP Algorithms

谷藤拓麻 † **松井俊浩** † **松尾啓志** †
Takuma Yato Toshihiro Matsui Hiroshi Matsuo

1. はじめに

複数の自律的に動作する要素である「エージェント」が協調して問題を解決するマルチエージェントシステムは，分散協調型のシステムのための主要な研究分野である．マルチエージェントシステムにおける協調問題解決の基本的な枠組みとして，分散制約最適化問題 (Distributed Constraint Optimization，DCOP)[1, 2, 3, 4, 5] が研究されている．分散制約最適化問題は，変数，制約および評価関数が複数のエージェントに分散して配置された最適化問題である．この問題では，エージェント間のメッセージ通信を伴う分散協調アルゴリズムにより最適解を得る．多くのマルチエージェントシステムの問題，特にセンサ網 [6] 等の分散資源割り当て問題が，分散制約最適化問題として定式化されている．

分散制約最適化問題の解法は，厳密解法と非厳密解法に分類される．厳密解法は，必ず最適解を求めることができる．しかし一般に，変数，制約および評価関数の数が多くなり，問題の規模や複雑さが大きくなるにつれて計算量，メッセージサイズが指数関数的に増加するという問題点がある．これに対し，いくつかの非厳密解法は，最適解を求めることができるとは限らないが，問題の規模に対する計算量，メッセージサイズの増加が線形に抑制される．そのため，大規模かつ複雑な問題に対しては非厳密解法が適用される．

非厳密解法で得られる解品質を議論するために，t-optimality[1]，p-optimality[2] という解の最適性の指標が提案された．t-optimal な解とは，制約網において，各エージェントを中心に t ホップ内にあるエージェントが持つ変数値を変更しても利得の総和を向上できない解のことである．任意のパラメータ t に対して，t-optimal な解を求めることができる非厳密解法として，局所最適解を繰り返し求める DALO-t[1] が提案された．p-optimal な解とは，元の問題から木幅を p に制限した問題に対する最適解のことである．p-optimality により木幅を制限した問題の最適解を元の問題の近似解として用いる手法が提案された [2]．t-optimality に基づく解法によって得られる解品質は，局所情報のみに依存するため解品質の向上が制限される．p-optimality に基づく解法によって得られる解品質は，元の問題にサイクルが多く含まれる場合大きく悪化するという問題点がある．

そこで，本研究では，解品質の向上を図るために，これらの異なる解の最適性の指標に基づく非厳密解法を統合する手法を提案する．すなわち，p-optimality に基づく解法を実行することで得られた情報のもとで，t-optimality に基づく解法を実行することにより，両者を相補的に用いることを目指す．実験により，提案手法を既存の手法と比較しその有効性を評価する．

以下に本論文の構成を示す．2 章では分散制約最適化問題について述べる．3 章，4 章では非厳密解法の解の最適性の指標について述べる．5 章では提案手法について述べる．6 章では実験による評価を示し，7 章ではまとめと今後の課題について述べる．

2. 分散制約最適化問題とその解法

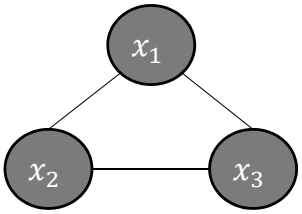
2.1. 分散制約最適化問題

分散制約最適化問題は，変数の集合 X，制約の集合 C，評価関数の集合 F，およびエージェントの集合から構成される．各変数 $x_i \in X$ は，離散有限集合 D_i に含まれる変数値をとる．変数は複数のエージェントに分散して配置される．本論文では，一つのエージェントに対して一つの変数が配置されるものとする．i 番目の変数 x_i がとる変数値を d_i と表し，制約 $c_{i,j}$ は変数 x_i, x_j 間に制約があることを表す．また，$c_{i,j}$ に対応する評価関数を $f_{i,j}(d_i, d_j)$ と表す．ただし，$c_{i,j} \in C$，$f_{i,j} \in F$ である．それぞれの変数への割り当て $A = [d_1, d_2, d_3, ..., d_n]$ に関する評価関数の利得の総和は以下のように表される．

$$R(A) = \sum_{f \in F} f_{i,j} \tag{1}$$

分散制約最適化問題の目的は，$R(A)$ を最大化するような割り当て A^* を求めることである．

変数をノード，変数間にある制約を辺として表現したグラフ (ネットワーク)$G =< X, E >$ を制約網という．この制約網において，各ノードは自身に関連する制約を知っている．また，隣接しているノードとのみ通信することができる．図 1 に 3 つのノード，3 つの制約辺がある制約網の例を示す．この制約網において，評価関数の利得の総和の最大値は $R(A) = 6$ であり，$A = [1, 1, 1]$ が最適解 A^* である．



x_i	x_j	f
0	0	1
0	1	0
1	0	0
1	1	2

図 1: 分散制約最適化問題の例

†名古屋工業大学, Nagoya Institute of Technology

2.2. 分散制約最適化問題の解法

分散制約最適化問題の解法は，厳密解法と非厳密解法の二つに分類することができる．

2.2.1. 厳密解法

厳密解法は，必ず最適解を求めることができる．厳密解法として，制約網に対応する擬似木を利用したメッセージ交換により最適解を求める DPOP[3] 等が提案されている．これらの解法は，変数，制約および評価関数の数が多くなり問題の規模や複雑さが大きくなるにつれて，計算量，メッセージ数およびメッセージサイズのいずれかが指数関数的に増加するという問題点がある．従って，大規模な問題に対しては，厳密解法を適用することは困難である．

2.2.2. 非厳密解法

非厳密解法は，必ず最適解を求めることができるとは限らないが，比較的少ない計算量，メッセージ数およびメッセージサイズで近似解を求めることができる．そのため，大規模かつ複雑な問題には，非厳密解法を適用することが一般的である．非厳密解法として，局所的な山登り法に基づく DSA[4] 等が提案されている．また，非厳密解法によって得られる解のために，解の最適性の指標が提案されている [1, 2]．本研究では，t-optimality および p-optimality と呼ばれる指標に基づく解法に注目する．

3.t-optimality

t-optimality は，分散制約最適化問題を解くことで得られる解の最適性の指標である．t-optimal な解とは，各ノードを中心として，t ホップ内にあるどのノードの変数値を変更しても利得の総和 $R(A)$ を向上できない解のことである．$T(x_i, x_j)$ を x_i，x_j 間の距離，$\Omega_t(x_i) = \{u|T(u, x_i) = t\}$ を t ホップ内にあるノードの集合 (グループ)，$D(A, A')$ を異なる変数値を取るノードの集合とすると，t-optimal な解 A は，以下のように定義される．

$$R(A) - R(A') \geq 0,$$

$$\forall A' \text{ where } D(A, A') \subseteq \Omega_t(x_i) \text{ for some } x_i \in X \quad (2)$$

制約網には，最大でノード数 n 個のグループが存在する．もし，$\exists x_i, x_j \in X$, $\Omega_t(x_i) \subseteq \Omega_t(x_j)$ となる場合，存在するグループの数は，n 個より少なくなる．図 1 を例に t-optimal な解について説明する．$A = \{x_1, x_2, x_3\} = [0, 0, 0]$ は，自身の変数値のみを変更，つまり自身のノードのみからなるグループの変数値を変更しても利得の総和を向上できないため，A は 0-optimal な解である．しかし，1-optimal な解ではない．この時，各ノードが作るグループは，$\Omega_1(x_1) = \Omega_1(x_2) = \Omega_1(x_3) = \{x_1, x_2, x_3\}$ となり，3 つのノードの変数値を変更したとき，利得の総和が 3 から 6 に向上するからである．ここで，各ノードが 1 をとる解 $A = \{x_1, x_2, x_3\} = [1, 1, 1]$ は，1-optimal な解であり，この制約網の最適解となる．

ノード間の制約を二項制約とし，$f \geq 0$ とすると，t-optimality では，$R(A)$ を t-optimal な解の利得の総和，$R(A^*)$ を最適解の利得の総和とすると，以下のように誤差の上界が計算される [1].

$$R(A) \geq \frac{2 + t - 1}{|X|} R(A^*) \quad (3)$$

3.1.DALO-t

DALO-t は，分散制約最適化問題の t-optimality に基づく非厳密解法である．すなわち，任意のパラメータ t に対して t-optimal な解を得る．この解法は，大域的な同期を必要としない非同期アルゴリズムであり，以下の三つのフェーズからなる．

1. グループの生成
2. グループ内の最適な割り当ての計算
3. 割り当ての実行

この解法はフェーズ 2，フェーズ 3 を繰り返し実行することで解を求める anytime 解法であり，利得の総和は単調に増加する．次の節からは，各フェーズの具体的な処理について説明する．

3.1.1. グループの生成

各ノードは，自身に関連する制約の情報を t ホップ先のノード，自身の現在の変数値を $t+1$ ホップ先のノードまで同報送信する．メッセージには，自身のノード id を付与し一意に識別できるようにする．このメッセージ交換によって，各ノードは，t ホップ内にあるノードの制約の情報，$t+1$ ホップ内にあるノードの変数値を知ることができる．ここで，各ノードは自身をリーダとし自身を中心に，t ホップ内にあるノードをグループのメンバ，$t+1$ ホップ先にあるノードをフリンジとよぶ．図 2 に例を示す．$t = 1$ の場合，x_2 が作るグループのメンバ $\Omega_1(x_2) = \{x_1, x_2, x_3, x_5\}$，フリンジ $R(x_2) = \{x_4, x_6, x_8\}$ となる．各ノードは，一つのグループを生

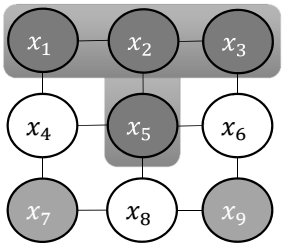


図 2: x_2が作るグループ ($t = 1$ のとき)

成する．ここで，$\exists x_i, x_j \in X$, $\Omega_t(x_i) \subseteq \Omega_t(x_j)$ となる場合があるため，各ノードは自身のグループのメンバ id をメッセージとして隣接するノードに送信し，冗長なグループを削除する．

3.1.2. グループ内の最適な割り当ての計算

各グループのリーダは，フェーズ 1 で得られた制約の情報と変数値の情報をもとに，フリンジの変数値は変更されないと仮定して，グループ内の利得の総和が最

大となるようにメンバへの最適な変数値の割り当てを計算する．最適な割り当てを計算する際には，DCOP の厳密解法を用いる [3]．ここで，各ノードは非同期に動作するため，オーバラップしているグループが計算した割り当てを実行し，フリンジの変数値が変更される可能性がある．このような変更を他のノードに知らせるために，各ノードは変数値を変更した時，$t+1$ ホップ先のノードまで現在の変数値を同報送信する．フリンジからこのメッセージを受け取ったリーダはその計算を中止し，フリンジから受け取った変数値の情報をもとに，再度最適な割り当てを計算する．このフェーズでは，計算のみにとどめ，実際に変数値の変更は行わない．

3.1.3. 割り当ての実行

各グループのリーダは，フェーズ 2 で行った最適な割り当てを計算した結果，現在のグループ内の利得の総和よりも大きい場合，グループのメンバの変数値の変更を試みる．ここで，オーバラップしているグループが同時に変数値を変更し，利得の総和が減少することを避けるために，lock/commit 非同期プロトコルを用いる [1]．各ノードは，自身の変数値を変更した時，$t+1$ ホップ先のノードまで現在の変数値を送信する．

このフェーズ 2, 3 を繰り返してゆくことで，t-optimal な解を求めることができる．

3.1.4.DALO-t の問題点

この解法は，各ノードがパラメータ t によって決められた範囲の局所情報を収集し，局所的な最適解を繰り返し求める解法である．したがって，解品質は局所情報のみに依存し，解品質の向上が制限されるという問題点がある．また，解品質を向上させるためにパラメータ t を大きくすると，各ノードが解く問題の規模が大きくなり，計算コストが指数関数的に増加する可能性がある．

4.p-optimality

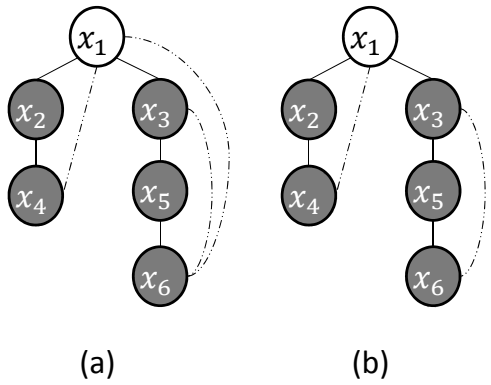


図 3: 木幅が 3 の擬似木と木幅を 2 に制限した擬似木

p-optimality は，t-optimality とは異なる視点に基づく解の最適性の指標である．p-optimality では，制約網に対応する擬似木を用いる．図 3 の (a) に擬似木の例を示す．擬似木は，制約網に含まれるノード間に順序関係を与えるグラフ構造であり，制約網における生成木の一つに対応する．生成木に含まれる辺を木辺，それ以外の辺を後退辺と呼ぶ．擬似木において，各ノードを根とする部分木に含まれるノードのいずれかと，制約辺で関連している上位ノード数を誘導幅 $w(x_i)$ と呼ぶ．また，全ノードについて最大の誘導幅を木幅 $W(G)$ と呼ぶ．ここで，与えられたパラメータ p と木幅が等しくなるまで後退辺を除去する．このようにして得られた擬似木 $G^{'} =< X, E^{'} >$ の最適解を p-optimal な解という．すなわち，p-optimality に基づく解法では，木幅をパラメータ p によって制限した問題の最適解を元の問題の近似解として用いる．

f_{max} を評価関数の最大値とすると，p-optimal な解 A の最適解 A^* に対する誤差の上界は以下のように計算される [2].

$$R(A^*) - R(A) \leq f_{max} \times \sum_{k=1}^{W(G)-p} (|X| - (k+1)) \quad (4)$$

4.1.p-optimal アルゴリズム

p-optimality に基づく解法である，p-optimal アルゴリズム [2] は以下の二つのフェーズからなる．

1. 擬似木の木幅を p に削減
2. 木幅 p の擬似木に厳密解法を適用

4.1.1. 擬似木の木幅を p に削減

フェーズ 1 では，元の擬似木 $G =< X, E >$ から，与えられたパラメータ p と木幅が等しくなるまで後退辺を除去する．この手順を Algorithm 1 に示す．

Algorithm 1 フェーズ 1

```
1: set E' to E
2: repeat the following procedure W(G) − p times
3: each x_i ∈ X, p + 1 ≤ w(x_i) ≤ W(G)
4: remove the first backedge in G' from E'
   if there is one.
5: returnG' = (X, E')
```

ここで $first\ backedge$ とは，各ノードにおいて，一番上位のノードと関連している後退辺のことをいう．図 3 の (b) は，(a) の擬似木から後退辺を除去し，木幅を 2 に制限した擬似木である．

4.1.2. 木幅 p の擬似木に厳密解法を適用

フェーズ 2 では，生成した擬似木に対して，厳密解法を適用して最適解を求める．この解が p-optimal な解となり，p-optimal な解を元の問題の近似解として用いる．p-optimal な解を求める時，擬似木が得られるので DPOP[3] 等の擬似木に基づく解法を用いる．本論文では，DPOP について説明する．

DPOP は動的計画法に基づく解法であり，擬似木の葉ノードから順にボトムアップに利得を集計し，根ノー

ドからトップダウンに変数値を決定する．この解法は以下の 2 つのフェーズからなる．

1. UTIL メッセージの生成，伝搬
2. VALUE メッセージの生成，伝搬

ここで，擬似木のノード x_i に関連するノードの表記を以下に示す．

- $P(x_i)$：木辺で x_i と関連する親ノード
- $PP(x_i)$:後退辺で x_i と関連する祖先ノードの集合
- $C(x_i)$：木辺で x_i と関連する子ノードの集合
- $PC(x_i)$：後退辺で関連する子孫ノードの集合
- $\overline{PP}(x_i)$：x_i を根とする部分木に含まれるいずれかのノードと関連する x_i の祖先ノード，および $P(x_i)$ からなる集合

各ノード x_i は，$\overline{PP}(x_i)$ に含まれるノードの全ての変数値の組について，自身を根とする部分木の利得値の表 $UTIL_{x_i,P(x_i)}$ を計算する．この表は，UTIL メッセージとして，親ノード $P(x_i)$ に送信される．このような計算がボトムアップに行われる．

UTIL メッセージが根ノードまで伝搬された後，各ノード x_i は最適な変数値 d_i^* を決定する．これは，根から葉に向けてトップダウンに計算される．Algorithm 2 に擬似コードを示す．

Algorithm 2 UTIL メッセージの生成と VALUE メッセージの生成

```
[Compute_UTIL]
X̄ ← P̄P(x_i)
for each d̄ ∈ ∏_{x_j∈X̄} D_{x_j} do
    UTIL_{x_i,P(x_i)}(d̄)
    ← max_{d_i∈D_i}{f_{i,P(x_i)}(d_i, d_{P(x_i)})+
      Σ_{x_j∈PP(x_i)} f_{i,j}(d_i, d_j)+
      Σ_{x_k∈C(x_i)} UTIL_{x_k,x_i}((d̄, d_i))}
end for
[Choose_Optimal]
d_i^* ← argmax_{d_i∈D_i}{f_{i,P(x_i)}(d_i, d_P(x_i)^*)+
  Σ_{x_j∈PP(x_i)} f_{i,j}(d_i, d_{x_j}^*)+
  Σ_{x_k∈C(x_i)} UTIL_{x_k,x_i}((d̄, d_i))}
```

4.1.3.p-optimal アルゴリズムの問題点

DPOP は擬似木の木幅について計算量，メッセージサイズが指数関数的に増加するという問題点がある．そのため，p-optimality のように，木幅を制限した問題の最適解を元の問題の近似解として用いる手法が研究されてきた．しかし p-optimality における解品質は除去される制約辺に依存するため，制約網にサイクルが多く含まれるような問題では解品質が大きく悪化する可能性がある．

5. 提案手法

本章では，解品質の向上を図るために，異なる解の最適性の指標 p-optimality，t-optimality に基づく非厳密解法の統合を提案する．

5.1. 提案手法の概要

p-optimality に基づく解法は，サイクルが多く含まれる問題の場合，解品質が大きく悪化する．t-optimality に基づく解法は，局所情報のみに依存するため，解品質の向上が制限される．そこで，解品質の向上を図るために，p-optimality に基づく解法と t-optimality に基づく解法を統合する．まず，制約網に対応する擬似木を生成し，その木幅をパラメータ p に制限する．そして，DPOP の最初の処理である UTIL メッセージの伝搬を行う．これにより，擬似木の根ノードは大域的な利得の集計結果を得る．しかし，他のノードはこの恩恵を受けることがない．そこで，同一の擬似木について，各ノードを根として，それぞれ UTIL メッセージの伝搬を行う．このようにして得られた利得を DALO-t の計算で用いることにより，局所的かつ詳細な情報と広範囲の情報を活用し，解品質の向上を目指す．

5.2. 提案手法のアルゴリズム

提案手法のアルゴリズムは大きく以下の二つのフェーズからなる．

1. p-optimal 解法の実行
2. t-optimal 解法の実行

5.2.1.p-optimal 解法の実行

p-optimal 解法は，以下の三つフェーズからなる．

1. 制約網から擬似木を生成
2. 擬似木の木幅をパラメータ p に制限
3. 各ノードを根として，それぞれ UTIL メッセージの伝搬

まず，制約網から深さ優先探索によって擬似木を生成し，p-optimal アルゴリズムのフェーズ 1 と同様の手続きによって，木幅をパラメータ p に制限する．次に，同一の擬似木に対して，それぞれのノードを根として DPOP の UTIL メッセージを伝搬する．しかし，本来は根ではないノードを強引に根とすると矛盾が生じる．このような例を図 4 に示す．

図 4(b) のように，x_2 を根ノードとした場合，部分木間に制約辺が存在する．擬似木に基づく解法は，部分木間に制約辺がないことに基づく分割統治法であるため，このままでは正しく計算ができない．そこで，このような擬似木に対処する手法 [5] と類似する方法を用いる．

図 5 に例を示す．x_1, x_4 間に制約辺があるため，x_4 の複製 $x_4^{'}$ を生成する．そして，その複製を部分木に分かれる部分のノード x_2 を管理しているエージェントに配置する．そして，x_1, x_4 間にある制約 $c_{1,4}$ および評価関数 $f_{1,4}$ を削除し，$x_1, x_4^{'}$ 間に同一の制約 $c_{1,4'}$，お

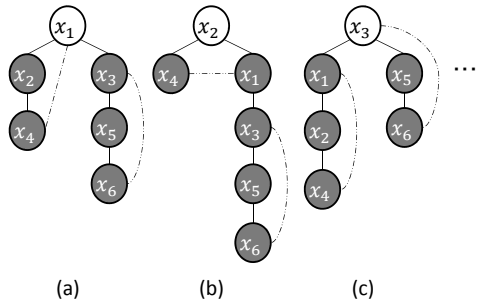


図 4: 各ノード根として，UTIL メッセージを伝搬

よび評価関数 $f_{1,4'}$ を追加する．また，$x_4, x_4^{'}$ 間に制約 $c_{4,4'}$，等式評価関数 $f_{4,4'}$ を追加する．
この手続きを Algorithm 3 に示す．ノード x_i に関連するノードについて以下に表記する．

- $ANC(x_i)$: x_i の祖先ノードの集合
- $DES(x_i)$: x_i の子孫ノードの集合
- $N(x_i)$: c_i の近傍ノードの集合
- $rep(x_i)$: x_i の複製ノード

Algorithm 3 部分木間に後退辺がまたがる問題の対処

1: 葉ノードからボトムアップ $ANC(x_i)$ を集計
2: 根ノードからトップダウンに $DES(x_i)$ を集計
3: **if** $x_j \in N(x_i)$ が $x_j \notin ANC(x_i) \cap x_j \notin DES(x_i)$ **then**
4: $rep(x_j)$ を x_k $where$ $x_i \in DES(x_k) \cap x_j \in DES(x_k)$ に設置
5: 制約 $c_{x_j.rep(x_j)}$，等式関数 $f_{j,rep(x_j)}$を追加
6: 制約 $c_{i,j}$および $f_{i,j}$と同一の $c_{i,rep(x_j)} f_{i,rep(x_j)}$を追加
7: $c_{i,j}$, $f_{i,j}$を削除
8: **end if**

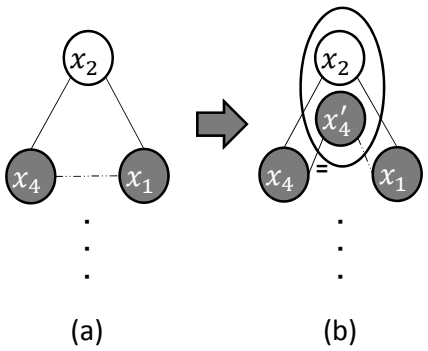


図 5: 部分木間に制約辺が存在する問題の対処

ここで，複製を置かれたエージェントは，自身が元々管理していたノードと複製のノードとの組み合わせを評価する必要がある．エージェントが持つ複製ノードの集合を $x_r \in cop(x_i)$ とすると，Algorithm 4 のように UTIL メッセージを生成する．

Algorithm 4 複製を置かれたエージェントにおける UTIL メッセージの生成

1: $[Compute\_Util]$
2: $UTIL_{x_r.x_i}$
$\leftarrow max_{d_r \in D_r}\{\sum_{x_k \in C(x_i)} UTIL_{x_k,x_i}(d_i, d_r)\}$
3: $\overline{X} \leftarrow \overline{PP}(x_i)$
4: **for each** $\overline{d} \in \prod_{x_j \in \overline{X}} D_{x_j}$ **do**
5: $UTIL_{x_i,P(x_i)}(\overline{d})$
$\leftarrow max_{d_i \in D_i}\{f_{i,P(x_i)}(d_i, d_{p(x_i)}) +$
$\sum_{x_j \in PP(x_i)} f_{i,j}(d_i, d_j) +$
$\sum_{x_r \in cop(x_i)} UTIL_{x_r,x_i}((\overline{d}, d_i))\}$
6: **end for**

ここで一つのノードのみを根とした場合，変数値を決定をするために，VALUE メッセージを生成し，伝搬する必要がある．提案手法では，それぞれのノードを根として UTIL メッセージを生成するため VALUE メッセージを伝搬することなく，各ノードは変数値を決定する．しかし，ノードが取りうるそれぞれの変数値に対して利得が同じになる対称解ができてしまう可能性がある．この場合，各ノードは最適な変数値の決定ができない．そこで，各ノードに，微小な乱数の値を持つ単項関数を与えておく [7]．これにより，各ノードは高確率で無矛盾に最適な変数値を決定できる．

5.2.2. t-optimal 解法の実行

t-optimal 解法は以下の三つフェーズからなる．

1. グループの生成
2. グループ内の最適な割り当ての計算
3. 割り当ての実行

各ノードは，DALO-t と同様に，制約の情報を t ホップ先，自身がとる変数値を $t+1$ ホップ先まで同報送信する．それに加えて，フェーズ 2 で生成した UTIL メッセージも $t+1$ ホップ先まで同報送信する．よって各ノードは，グループのフリンジが生成した UTIL メッセージの情報を知ることができる．

次に，各ノードはグループのメンバへの最適な変数値の割り当てを DPOP を用いて求める．この時，フリンジを葉ノードとする木を生成しフリンジが生成した UTIL メッセージを利用する．$t = 1$ にした時の例を図 6 に示す．ノード x_2 が作るグループのメンバは，$\Omega_1(x_2) = \{x_1, x_2, x_3, x_5\}$，フリンジは，$R(x_2) = \{x_4, x_6\}$ である．この時，x_2 を根ノードとした場合にフリンジ x_4, x_6 が生成した UTIL メッセージも利用して，グループ内のメンバへの最適な割り当てを計算する．

最後に計算した割り当てを実行するグループを決定する．ここでは，DALO-t と同様の手法を用いて，割り当てを実行するグループを決定する. 変数値を変更したノードは現在の変数値を $t+1$ ホップ先のノードまで送信する．

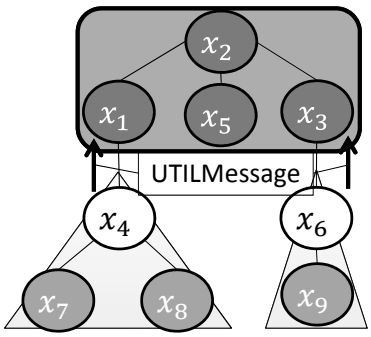


図 6: フリンジが生成した UTIL メッセージも利用して，グループ内のメンバの最適な割り当てを計算

5.3. 提案手法の計算，通信コスト

まず提案手法における，p-optimal 解法の実行に必要な通信，計算コストについて考察する．p-optimal 解法では，擬似木の生成，木幅の制限，UTIL メッセージの伝搬を行う．はじめに，典型的な擬似木である DFS 木の生成と木幅を制限する時の通信コストについて考える．逐次的な深さ優先探索を模倣する分散アルゴリズムは，木辺の数の 2 倍のサイクル数を要する．これをオーバヘッドの上限と考えれば，木を生成する処理に必要なサイクル数は，2 × (木辺の数) である．擬似木の木幅を制限するのに必要なサイクル数は，Algorithm 1 より，$(W(G)-p)\times$(木の最大深さ) となる．p=1 とした時，木幅を制限する処理は必要ない．次に，部分木間にまたがる制約辺の検知，対処にかかる通信コストについて考える．ボトムアップに子孫の集合，トップダウンに祖先の集合を集計するのに必要なサイクル数は，(木の最大深さ) × 2 となる．しかしこれを深さ優先探索，およびそのバックトラックに行えば，サイクル数は隠蔽できる．ノードの複製の設置にかかるサイクル数は，最大で，(木の最大深さ) – (複製を置くノードの深さ) となる．

次に，DPOP の UTIL メッセージの伝搬にかかるサイクル数と計算，通信コストについて考察する．サイクル数は，ボトムアップの計算を並行するため，木の最大深さだけ必要となる．しかし，この計算を深さ優先探索のバックトラックの際に行えば，サイクル数は隠蔽できると考えられる．したがって，サイクル数は増加しない．各ノードが行う，UTIL メッセージの計算量とメッセージサイズは，木幅と値域に対して，指数関数的に増加する．提案手法では，木幅を制限するため．計算量，メッセージサイズを抑制できると考えられる．また，メッセージ数は，それぞれのノードを根とした場合の UTIL メッセージを生成するため，最大 (木辺) × (ノード数) である．そのため，通信のオーバヘッドが大きい環境においては実行時間に影響を与える可能性がある．

次に，提案手法における t-optimal 解法の実行にかかる計算，通信コストについて考察する．UTIL メッセージをフリンジから収集する処理は，変数値を送信するメッセージと統合することができる．したがって，メッセージ数は DALO-t と比べて増加しない．この時，メッセージサイズは増加するが，UTIL メッセージのサイズが小さければ，その増加は重大ではない．また，比較的大きなパケット環境を用いる環境では，パケットに収まるメッセージならばサイズの差が隠蔽されることが期待できる．局所範囲の最適な割り当てを求める時にかかる計算コストは，フリンジの UTIL メッセージも利用するので，DALO-t に比べて計算量は増加する．どの程度増加するかは，フリンジのノード数と UTIL メッセージのサイズに依存する．通信コストが計算コストよりも大きい一般的な環境では，UTIL メッセージのサイズが小さければ，計算コストの増加の影響は，相対的に小さいと考えられる．

6. 実験と評価

本章では，提案手法を実験により評価する．問題の制約網は，一つのノードに対する次数の平均が一定になるまで，ランダムに辺を繋ぐことにより生成した．生成された制約網は単一連結成分からなる．本章では，密度を一つのノードに対する平均の次数と定義する．実験結果は 30 インスタンスの平均値を示す．

6.1. 解品質の比較

提案手法の解品質を既存の手法である p-optimal アルゴリズム，DALO-t と比較した．実験では，p=1，p=2 とした p-optimal アルゴリズム，t=1，t=2 とした DALO-t と (p=1,t=1)，(p=2,t=1)，(p=2,t=2) とした提案手法を比較した．実験における各変数の値域のサイズは 3 とし，各評価関数の利得は 0-1000 の間でランダムに選択した．図 7 は，密度を変えながら，ノード数 40 のグラフにおける実験結果を示す．グラフは，左から p-optimal アルゴリズム，DALO-t，提案手法となっている．また，密度 5，7，9 の t=2 にした DALO-t および提案手法の結果はメモリ不足により実行できなかったため，グラフ上に示していない．実験の結果，提案手法によって得られる解品質は，既存の手法よりも高品質であることが分かった．また，どの手法も密度が上がるにつれて解品質が悪化することが分かった．特に，p-optimal アルゴリズムではその傾向が顕著であり，これは除去される制約辺が多くなるためである．本提案手法は p-optimal 解法と t-opitmal 解法を組み合わているため，密度の上昇に対して，解品質の悪化は DALO-t と同程度に抑えられている．図 8 は，密度 3 のグラフにおける異なるノード数での実験結果を示す．実験の結果，どのノード数においても解品質は同程度であることが分かった．

	p-Max	p-Average	t-Max	t-Average	Total-Max	Total-Average
p=1,t=1	9	6.0	60	32.7	69	38.7
p=2,t=1	27	19.9	60	32.7	87	52.6
p=2,t=2	27	18.3	99	65.4	126	83.7

表 1: 各エージェントの問題の規模

また，ノード数 40，密度 3 のグラフに対して，提案

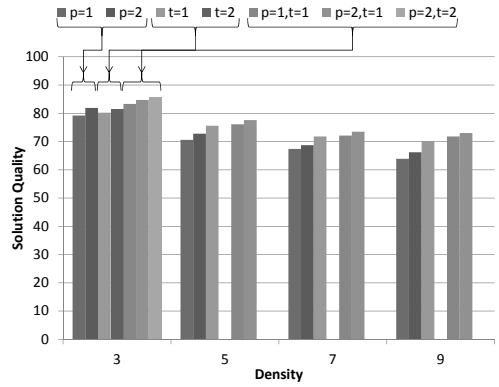


図 7: ノード数 40 のグラフにおける解品質の比較

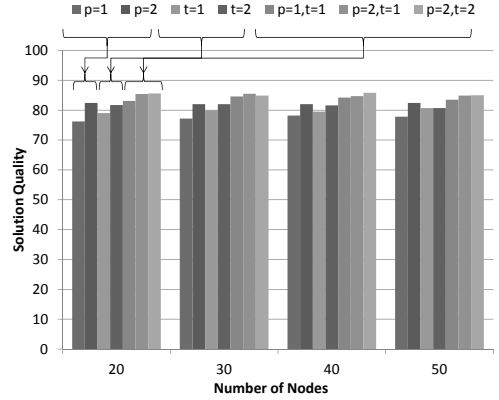


図 8: 密度 3 のグラフにおける解品質の比較

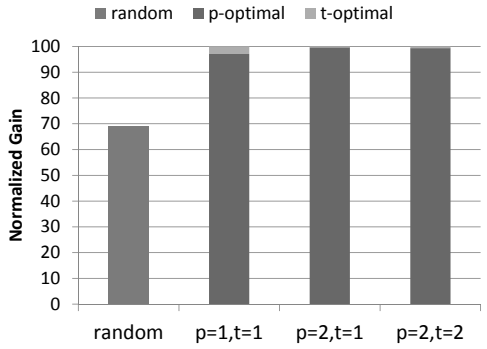


図 9: p-optimal 解法，t-otpimal 解法で得られる利得

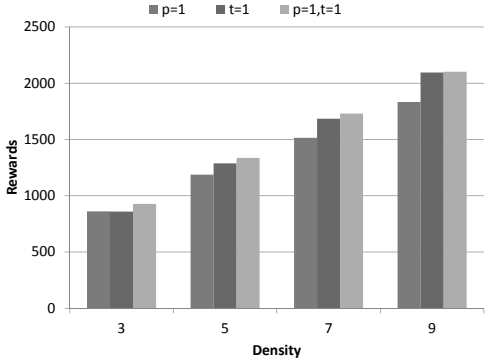


図 10: ノード数 100 のグラフにおける利得の総和の比較

手法における一つのエージェントが解く問題の規模を表 6.1 に示す．ここで問題の規模とは，p-optimal 解法では，UTIL メッセージを生成する時に必要な変数の値域の直積のサイズ，t-optimal 解法では，グループに含まれる変数の値域の直積のサイズを表す．パラメータを上げると，各エージェントの解く問題の規模は増加する．また p-optimal と t-optimal 解法を比較すると，t-opitmal 解法における問題の規模が大きいことが分かった．

6.1.1. t-optimal 解法，p-optimal 解法で得られる利得

提案手法における，p-optimal 解法および t-optimal 解法を実行することで得られる利得について評価した．図 9 は，ノード数 40，密度 3 として実験した結果のグラフであり，グラフの縦軸は，最終的に得られた利得に対して正規化した値を表している．また，比較対象として，ランダムに変数値を割り当てた時の結果を一番左に示してある．実験の結果，p-optimal 解法の実行により得られた利得が全体に占める割合が大きいことが分かった．また，(p=1,t=1) とした提案手法は，t-optimal 解法の実行によって得られる利得が比較的大きいことが分かった．

6.1.2. 頂点彩色問題における比較

続いて頂点彩色問題に対して，(p=1,t=1) とした提案手法と p=1 とした p-optimal アルゴリズム，t=1 とした DALO-t と比較した．各評価関数の利得は，二変数値が同一の場合は利得 0，異なる場合は利得を 1 から 10 の間でランダムに選択した．図 10 に密度を変えながら，ノード数 100 における実験結果を示す．グラフの縦軸は，利得の総和を表している．実験の結果，提案手法によって得られる解品質は既存の手法よりも高品質であることが分かった．密度が上がるにつれて，p-optimal アルゴリズムと DALO-t，および提案手法の解品質の差が大きくなることが分かった．これは，6.1 で述べたように，除去される後退辺の数が多くなるためである．また，同様の理由で，密度が上がるにつれて，提案手法と DALO-t の解品質の差が小さくなったと考えられる．

6.2. 収束速度の比較

(p=1,t=1) とした提案手法と DALO-t の収束速度を比較した．ここで，提案手法における p-optimal 解法の実行サイクル数を (ノード数 $-$ 1) $\times$ 2 と見積り実験を行った．この実験では，部分木間に制約辺が存在しないため，木幅を制限する処理は必要ない．実行サイクル数は 500 で打ちきった．ノード数を 50，密度を 4 とした時の結果を図 11 に示す．実験の結果，提案手法よりも DALO-t の方が早く収束することが分かった．

6.3. 考察

提案手法について実験結果をもとに考察する．提案手法では，p-optimal 解法の実行後，t-optimal 解法を実行する．従って，p-otpimality，t-optimality を満たす解が得られると期待できる．提案手法の解品質は，p-optimal 解法の実行によって得られた解から，t-optimal 解法を実行するため，高品質な解に収束する可能性が高

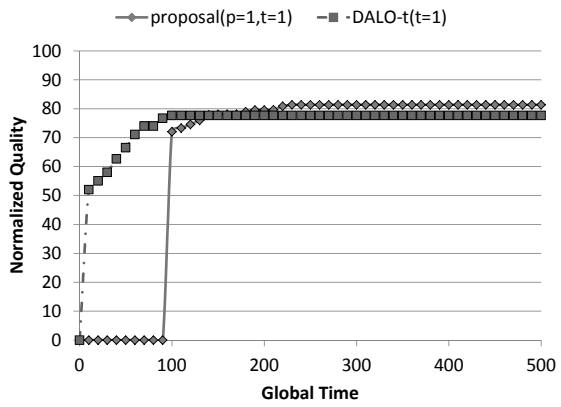


図 11: ノード数 50，密度 4 とした時の DALO-t との実行時間の比較

いと考えられる．また，t-optimal 解法では，局所情報に加えて，p-optimal 解法の実行により生成した UTIL メッセージも利用する．従って，広範囲の情報に基づいて局所範囲の最適な割り当てを求めるため，高品質な解が得られたと考えられる．この影響は，(p=1,t=1) とした提案手法で大きくみられた．

提案手法では，パラメータ p，t を調整することにより，計算および通信コストを犠牲にする代わりに，より高品質な解を得ることができる．ここで, パラメータ p，t の設定について考察する．p-optimal アルゴリズムの解品質は，除去される後退辺に依存する．問題のグラフにサイクルが多く含まれる場合，すなわちグラフ密度が大きい場合，除去される後退辺が多くなるため，解品質が大きく悪化する可能性がある．また，DALO-t の解品質は，グループサイズに依存する．問題のグラフ密度が小さい場合，グループサイズが小さくなるため，解品質が大きく悪化する可能性がある．従って，問題のグラフ密度が大きい場合，パラメータ p を大きく，問題のグラフ密度が小さい場合，パラメータ t を大きく設定することで解品質の悪化を抑制できると考えられる．

提案手法の収束速度は，p-otpimal 解法の実行時間に依存する．本論文では，パラメータ p=1 とし，p-otpimal 解法の実行時間を最悪の場合を想定して (ノード数 -1) $\times 2$ と見積もって実験を行った結果，既存の DALO-t よりも収束速度が悪くなることが分かった．提案手法では，パラメータを上げるにつれてより高品質な解を得ることできるが，収束速度が悪化する．この問題については，提案手法と DALO-t を多重実行する手法を用いる．すなわち，p-optimal 解法を実行している間は，DALO-t を実行することで得られた解を採用することで，この問題を隠蔽できると考えられる．

7. まとめ

本論文では，分散制約最適化問題の非厳密解法によって得られる解品質に着目し，異なる解の最適性の指標 p-optimality[2]，t-optimality[1] に基づく非厳密解法を統合する手法を提案した．すなわち，p-optimality に基づく解法を実行し，得られた情報のもとで，t-optimality に基づく解法を実行することにより両者を相補的に用いる．実験により提案手法の有効性を確認したところ，既存の p-optimal アルゴリズム，DALO-t よりも高品質な解が得られることを示した．

今後の研究課題として，異なる問題に対して提案手法の有効性を調べる．本論文ではランダムなグラフに対してのみ実験を行ったが，その他に実ネットワークに近いスケールフリーグラフに対して実験を行う必要がある．また，センサネットワーク [6] 等の応用問題に対して，解品質，計算，通信コストを調べる必要がある．

謝辞

本研究の一部は，科研費基盤研究 (C)25330257 および平成 23 年度人工知能研究振興財団研究助成による．

参考文献

[1] Kiekintveld, C., Yin, Z., Kumar, A. and Tambe, M.: Asynchronous algorithms for approximate distributed constraint optimization with quality bounds, *AAMAS: volume 1 - Volume 1*, pp. 133–140 (2010).

[2] 沖本天太，ジョヨンジュン，岩崎　敦，横尾　真：擬似木に基づく分散制約最適化問題の精度保証付き非厳密解法の提案，情報処理学会論文誌， Vol. 52, No. 12, pp. 3786–3795 (2011).

[3] Petcu, A. and Faltings, B.: A scalable method for multiagent constraint optimization, *IJCAI*, pp. 266–271 (2005).

[4] Zhang, W., Wang, O. and Wittenburg, L.: Distributed stochastic search for constraint satisfaction and optimization: Parallelism, phase transitions and performance, *in PAS*, pp. 53–59 (2002).

[5] Atlas, J. and Decker, K.: A complete distributed constraint optimization method for non-traditional pseudotree arrangements, *AAMAS*, pp. 111:1–111:8 (2007).

[6] Zhang, W., Wang, G., Xing, Z. and Wittenburg, L.: Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks, *Artif. Intell.*, Vol. 161, No. 1-2, pp. 55–87 (2005).

[7] A.Farinelli, A.Rogers, A.Petcu and N.R.Jennings: Decentralized Coordination of Low-Powered Embedded Devices Using the Max-Sum Algorithm, *AAMAS*, Vol. 2, pp. 639–646 (2008).