

## 分散制約充足問題：特定の制約網に特化した 変数順序付けヒューリスティックの提案

沖 本 天 太<sup>†1</sup> 岩 崎 敦<sup>†1</sup> 横 尾 真<sup>†1</sup>

分散制約充足問題とは、制約充足問題における変数および制約が複数のエージェントに分散された問題である。既存の分散制約充足アルゴリズムのほとんどは、任意の制約網で動作することを保証している。しかし、特定の制約網、たとえば、ハブを含むようなスケールフリー的な制約網を対象とする場合、対象とする制約網に特化したアルゴリズム/ヒューリスティックが有効となることが予想される。我々の研究は、特定の制約網に特化したアルゴリズムの開発を目的とする。本論文では、その第1歩として、スケールフリー的な制約網に特化した静的な変数順序付けヒューリスティックを提案し、その有効性を示す。実験では、非同期バックトラッキングを用い、エージェントの優先順位の決定法が、スケールフリー的な制約網ではランダム構造の制約網より、アルゴリズムの性能に大きな影響を与えることを示す。さらに、エージェントの優先順位を提案手法と次数順に基づくヒューリスティックによって決定した、非同期バックトラッキングの性能を比較し、提案手法では最大29%の性能向上が得られることを示した。

### Effect of DisCSP Variable-ordering Heuristic for Particular Network Structure

TENDA OKIMOTO,<sup>†1</sup> ATSUSHI IWASAKI<sup>†1</sup>  
and MAKOTO YOKOO<sup>†1</sup>

A Distributed Constraint Satisfaction Problem (DisCSP) is a constraint satisfaction problem in which variables and constraints are distributed among multiple agents. Various algorithms for solving DisCSPs have been developed, which are intended for general purposes, i.e., they can be applied to any network structure. However, if a network has some particular structure, e.g., the network structure is scale-free, we can expect that some specialized algorithms or heuristics, which are tuned for the network structure, can outperform general purpose algorithms/heuristics. In this paper, as an initial step toward developing specialized algorithms for particular network structures, we

examine variable-ordering heuristics in scale-free networks. We use the classic asynchronous backtracking algorithm as a baseline algorithm and examine the effect of variable-ordering heuristics. First, we show that the choice of variable-ordering heuristics is more influential in scale-free networks than in random networks. Furthermore, we develop a novel variable-ordering heuristic that is specialized to scale-free networks. Experimental results illustrate that our new variable-ordering heuristic is more effective than a standard degree-based variable-ordering heuristic. Our proposed heuristic reduces the required cycles by 29% at the critical point.

#### 1. はじめに

人工知能で扱われる問題の多くは、探索という要素を含んでおり、探索/制約充足問題として定式化できる。制約充足問題とは、有限で離散的な領域から値をとる複数の変数に、制約を満たす値を割り当てる問題である。分散制約充足問題とは、制約充足問題における変数および制約が複数のエージェントに分散された問題である<sup>1),2)</sup>。各エージェントは自分の持つ変数の値を決定しようとするが、異なるエージェントの持つ変数間に制約があり、すべてのエージェントの変数への値の割当てが、すべての制約を満たす必要がある。分散制約充足問題は変数をノードに、制約をノード間のエッジに対応させることにより、制約網(制約ネットワーク)を用いて表すことができる。

分散制約充足問題の代表的なアルゴリズムに非同期バックトラッキング(Asynchronous Back Tracking, ABT)がある。このアルゴリズムは分散制約充足問題を解く、最初の非同期かつ完全なアルゴリズムとして提案された<sup>1),2)</sup>。各エージェントはグローバルな制御なしに、並行、非同期に動作することができる。各エージェントは擬似木内でメッセージ通信を行いながら解を求める。擬似木とは、制約網と同一のエッジとノードによって生成される擬似的な木である。

近年スケールフリーと呼ばれるネットワークの性質が Barabási と Albert によって発見され、インターネット、タンパク質相互作用、SNS、論文引用ネットワーク等、現実世界の様々なネットワークがスケールフリー性を持つことが分かってきた<sup>3)-5)</sup>。従来これらのネットワークはランダムグラフとしてモデル化されていた。しかし、従来のモデルでは現実世

<sup>†1</sup> 九州大学大学院システム情報科学府情報学専攻横尾研究室  
Graduate School of ISEE, Kyushu University

界のネットワークの形態や特徴的な性質を生成することができない。たとえば、スケールフリー性を持つ現実世界のネットワークでは次数分布がベキ乗則に従う。これに対し、ランダムグラフでは次数分布がポアソン分布に従う。

制約充足問題の研究では、制約網の構造に着目した研究がいくつか存在する<sup>6)–8)</sup>。たとえば、Walsh はレジスタ割当て問題、時間割問題、準群問題等の応用事例が、制約網の構造としてスモールワールド性を持つことを示した<sup>6),7)</sup>。また、Devlin らは周波数帯域割当て問題や大規模な論理回路の検証で、制約網の構造としてスケールフリー性を持つ問題があることを示した<sup>8)</sup>。さらに、彼らはスケールフリー性を持つ問題では、ランダム構造を持つ問題より、次数順に基づく探索法の効果が大きいことを示した。

分散制約充足問題の研究では、制約網の構造に着目した研究は少なく、特にスケールフリー的な制約網を対象とした研究はほとんどない。文献 6)–8) で示されているように、様々な制約充足問題の応用事例が、実際に制約網の構造としてスモールワールド性やスケールフリー性を持つことが分かった。もし、これらの問題に関する知識が複数のエージェントに分散されている問題を考えると、分散制約充足アルゴリズムを用いて問題を解く必要がある。

さらに、ソーシャルネットワークで、各個人の代理人としてエージェントが存在する状況を考える<sup>9)</sup>。ソーシャルネットワークはスケールフリー性を持つ傾向があるため、このような制約網でミーティングスケジューリング問題<sup>10)</sup>を解決するとき、分散制約充足アルゴリズムを用いて問題を解くことができる。

我々の研究は特定の制約網に特化したアルゴリズム/ヒューリスティックの開発を目的とする。本論文は、その第 1 歩として、スケールフリー的な制約網に特化した変数順序付けヒューリスティックを提案し、その有効性を示す。非同期バクトラッキングをはじめ、既存の分散制約充足アルゴリズムのほとんどは、任意の制約網で動作することを保証している。しかし、特定の制約網、たとえばスケールフリー的な制約網を対象とする場合、対象とする制約網に特化したアルゴリズム/ヒューリスティックが有効となることが予想される。

本論文では、分散制約充足問題の解法として古典的な非同期バクトラッキングを用いる。分散制約充足アルゴリズムには様々な効率的な洗練されたアルゴリズムがある<sup>11)–14)</sup>。しかし、本論文はスケールフリー的な制約網に特化した変数順序付けヒューリスティックの開発を目的とするため、最も簡単かつ明確な非同期バクトラッキングを用いることとする。本論文で提案する変数順序付けヒューリスティックは擬似木の構造を考慮しているため、他の擬似木に基づく分散制約充足アルゴリズムに対しても、有効となることが予想される。

はじめに、エージェントの優先順位の決定法が、スケールフリー的な制約網ではランダム

構造の制約網より、アルゴリズムの評価基準であるサイクル数に大きな影響を与えることを示す。スケールフリー的な制約網では次数分布がベキ乗則に従う。これに対し、ランダム構造の制約網では次数分布がポアソン分布に従う。これらの特徴的な違いから、エージェントの優先順位の決定法がサイクル数に与える影響は異なると予想される。実験では、分散制約充足問題においても、エージェントの優先順位がスケールフリー的な制約網では、サイクル数に大きな影響を与えることが分かった。この結果は文献 8) で報告されている制約充足問題における実験結果と一致している。

さらに、スケールフリー的な制約網で、エージェントの優先順位によって決定される、擬似木の深さやバックエッジ数とサイクル数との関連性を調べる。非同期バクトラッキングでは、エージェントの優先順位を基に擬似木が作成される<sup>1),2)</sup>。深さやバックエッジ数は擬似木の構造に大きな影響を与えるため、これらの要因はサイクル数に影響を与えると予想される。しかしながら、実験では、異なる深さやバックエッジ数とサイクル数との間に特徴的な関連性はみられなかった。

最後に、スケールフリー的な制約網に特化した静的な変数順序付けヒューリスティック (ALH) を提案し、その有効性を示す。具体的には、パラメータの異なるスケールフリー的な制約網で、ALH が次数順に基づくヒューリスティックより、サイクル数を減少させることを示す。エージェント/変数の優先順位には、静的な変数順序付け (static variable ordering) と動的な変数順序付け (dynamic variable ordering) の 2 つがある。静的な変数順序付けは、探索が始まる前にエージェント/変数の順序を決定する手法であり、通常のバクトラッキング法で用いられることが多い。動的な変数順序付けは、探索時に次のエージェント/変数の順序を現在の探索状況に基づき動的に決定する手法である。分散制約充足問題における変数順序付けヒューリスティックを扱う研究は多い<sup>15)–17)</sup>。しかし、本論文のように、スケールフリー的な制約網を想定した変数順序付けヒューリスティックに着目した研究は、著者らの知る限りほとんどない。

本論文の構成を以下に示す。2 章では、分散制約充足問題の定義および非同期バクトラッキングについて概説する。3 章では、スケールフリーの性質について述べる。4 章では、優先順位の決定法が与える影響を、スケールフリー的な制約網とランダム構造の制約網で調べる。5 章では、スケールフリー的な制約網に特化した変数順序付けヒューリスティックを提案し、その有効性を示す。6 章では、本論文のまとめと今後の課題について述べる。

## 2. 分散制約充足問題の定式化

制約充足問題 (Constraint Satisfaction Problem, CSP) とは, 有限で離散的な領域から値をとる複数の変数に, 制約を満たす値を割り当てる問題である. この問題は一般に, 次のように定義される.  $m$  個の変数  $x_1, x_2, \dots, x_m$  と, それぞれの変数が値をとる有限で離散的な領域  $D_1, D_2, \dots, D_m$  および制約の集合が存在する. 制約は述語論理によって内包的に定義されるとする. すなわち, 制約  $p_k(x_{k1}, \dots, x_{kj})$  は, 直積  $D_{k1} \times \dots \times D_{kj}$  に対して定義され, これらの変数の値が互いに整合のとれている場合に真となる. 制約充足問題の解を求めることは, すべての制約を満足する変数の値の組を求めることである.

分散制約充足問題 (Distributed Constraint Satisfaction Problem, DisCSP)<sup>1),2)</sup> とは, 制約充足問題における変数および制約が複数のエージェントに分散された問題である. 各エージェントは自分の持つ変数の値を決定しようとするが, 異なるエージェントの持つ変数間に制約があり, すべてのエージェントの変数への値の割当てが, すべての制約を満足する必要がある. 形式的には, エージェントの集合  $\{1, 2, \dots, m\}$  が存在し, 各変数  $x_j$  に対して, その属するエージェント  $i$  が定義される ( $\text{belongs}(x_j, i)$  と書く). 制約に関する情報も同様にエージェント間に分散される. エージェント  $l$  が制約  $p_k$  を知っていることを  $\text{know}(p_k, l)$  と書く.

次の場合に, 分散制約充足問題が解けたという.

- すべてのエージェント  $i$  において,  $\forall x_j \text{ belongs}(x_j, i)$  について,  $x_j$  の値が  $d_j$  に決定される. そして, すべてのエージェント  $l$  について,  $\forall p_k \text{ know}(p_k, l)$  なる制約が,  $x_1 = d_1, x_2 = d_2, \dots, x_n = d_n$  のもとで真となる.

制約充足問題の代表的な例に  $n$  クイーン問題がある. この問題で, 各クイーンに対応する独立なエージェントが存在し, 各エージェントがそれぞれ自分のクイーン的位置を決定しようとしているとする. このとき, この問題は分散制約充足問題として定式化される.

非同期バックトラッキング (Asynchronous Back Tracking, ABT) は, 分散制約充足問題を解く, 最初の非同期かつ完全なアルゴリズムとして提案された<sup>1),2)</sup>. 問題を簡単化するため, 以下の議論では次の仮定をおく. これらの仮定の一般化は容易であり, アルゴリズムの一般性は失われていない.

- 各エージェントの持つ変数は唯一である.
- すべての制約は binary (2 変数間) である.
- 各エージェントは自分に属する変数が関係する制約をすべて知っている.

オリジナルの非同期バックトラッキングでは, 擬似木は前処理で構成されるのではなく, エージェント間の優先順位を用いて, アルゴリズムの実行中に動的に構成される. 本論文では, 簡単のため, 前処理によって, 制約網の構造を考慮して適切な優先順位を決定すると同時に, 擬似木も構成することを仮定している. 擬似木をあらかじめ構成することで, オリジナルの非同期バックトラッキングが必要であった, 動的なエージェント間の関係の追加が不要になり, アルゴリズムがシンプルになる. 同様な前提は, 分散制約最適化問題を解く他のアルゴリズムでも用いられている<sup>18),19)</sup>. 擬似木とは, 以下の性質をすべて満たすような擬似的な木である.

- 擬似木は元の制約網と同一のエッジとノードからなる.
- 擬似木は元の制約網の生成木のいずれか 1 つに対応する. 元の制約網のすべてのエッジは, 生成木のエッジ (ツリーエッジ) かそれ以外のエッジ (バックエッジ) に分類される.
- 生成木の根ノードからいずれか 1 つの葉ノードへのパスに含まれる 2 ノード間のみ, バックエッジがある. 異なるパスに含まれる 2 ノード間には, バックエッジはない.

本論文では, 擬似木を以下のように構成する. エージェントの優先順位を  $o = x_1, x_2, \dots, x_n$  として, (1)  $x_n$  の親ノードを,  $x_n$  とエッジで関連するエージェント中で, 最も優先順位の低いエージェントとする. (2) 以下,  $x_{n-1}, x_{n-2}$  の順に, ノード  $x_i$  の親ノードを以下のように決定する.  $x_i$ , および  $x_i$  の子孫ノードが関連する,  $x_i$  より優先順位の高いエージェント中で, 最も優先順位の低いエージェントを,  $x_i$  の親ノードとする.

非同期バックトラッキングでは, 擬似木内の各エージェントがメッセージ通信を行いながら解を求める. エージェント間では, 次の 2 種類のメッセージが用いられる.

- ( $\text{ok?}, (x_j, \text{value})$ ): 変数  $x_j$  の値の割当てが  $\text{value}$  であることを伝えるメッセージ.
- ( $\text{nogood}, x_j, \text{nogood}$ ): 新たに発生した  $\text{nogood}$  を伝えるバックトラックメッセージ.  $\text{nogood}$  は制約違反を起こす値の組合せである. たとえば,  $\text{nogood}\{(x_i, d_i), (x_j, d_j)\}$  は,  $(x_i, d_i)$  と  $(x_j, d_j)$  が制約違反を起こすことを表す.

$\text{ok?}$ メッセージは値を送信する (優先度の高い) エージェントから, 制約を評価する (優先度の低い) エージェントに送られるものであり, 選択した値が許容できるかを尋ねるものである.  $\text{nogood}$  メッセージは制約を評価するエージェントから値を送信するエージェントに送られるもので, 制約評価エージェントが制約条件違反を発見したことを示す. 非同期バックトラッキングでは,  $\text{ok?}$ メッセージを受信することにより, エージェント  $x_i$  は自分より優先順位の高いエージェントとの間にある制約を満足する値を見つける. 制約を満足

する値がなければ，エージェント  $x_i$  は，自分より優先順位の高いエージェントのうち，最も優先順位の低いエージェントへ *nogood* メッセージを送信する．非同期バクトラッキングでは，アルゴリズムの完全性を保証しつつも，全体の制御なしで各エージェントが非同期，並行的に行動できる．

### 3. スケールフリー

近年スケールフリーと呼ばれるネットワークの性質が Barabási と Albert によって発見され，インターネット，タンパク質相互作用，SNS，論文引用ネットワーク等，現実世界の様々なネットワークがスケールフリー性を持つことが分かってきた<sup>3)-5)</sup>．従来これらのネットワークはランダムグラフとしてモデル化されていた．しかし，従来のモデルでは現実世界のネットワークの形態や特徴的な性質を生成することができない．たとえば，スケールフリー性を持つ現実世界のネットワークでは次数分布がベキ乗則に従う．これに対し，ランダムグラフでは次数分布がポアソン分布に従う．

ランダムグラフとは，Erdős らの定義によれば，任意のノードの組が一定の確率で隣接するグラフを指す<sup>20)</sup>．このグラフでは，ノードが持つエッジ数に典型的な値が存在し，多くはその値から離れない．ノード数を  $N$ ，次数を  $k$ ，ノードの出現頻度を  $p(k)$  とすると，ランダムグラフでは次数分布が以下のポアソン分布に従う<sup>7)</sup>．

$$p(k) = e^{-\lambda} \lambda^k / k!, \text{ where } \lambda = N \binom{N-1}{k} p^k (1-p)^{N-1-k}.$$

現実世界の様々なネットワークはスケールフリー性を持つ．スケールフリーネットワークでは，一部のノードが膨大なエッジを持ち，大半のノードはごくわずかなノードとしか隣接しない．たとえば，スケールフリーネットワークでは，エッジを 2, 3 本程度しか持たないノードが大半を占め，エッジを 1,000 本程度持っているノードがいくつか存在する．他のノードと比べ，極端に多くのエッジを持つノードをハブと呼ぶ．スケールフリーの性質を以下にまとめる<sup>5)</sup>．

- ノードの次数分布がベキ乗則に従う．次数を  $k$ ，ノードの出現頻度を  $p(k)$  とすると，ベキ乗則は以下の式で表される． $\gamma$  はベキ指数とする．

$$p(k) \propto k^{-\gamma}$$

- 他のノードと比べ，極端に多くのエッジを持つハブが存在する．
- ノードが持つエッジ数には特徴的なサイズが存在しない．
- ネットワーク障害に対する頑強性は高いが，ハブを狙った攻撃に対しては脆弱である．

### 4. スケールフリー的な制約網における変数順序付けヒューリスティックの影響

本章では，エージェントの優先順位の決定法が，スケールフリー的な制約網ではランダム構造の制約網より，サイクル数に大きな影響を与えることを示す．さらに，スケールフリー的な制約網で，異なるエージェントの優先順位によって決定される擬似木の深さやバックエッジ数とサイクル数との関連性を調べる．

ここでは，離散イベントのシミュレータを用いて，エージェントの並行動作のシミュレートを行う．すなわち，各エージェントは自分自身のクロックを管理し，送信されたいくつかのメッセージを受信し，局所的な計算を行う．そして他のエージェントに対して，メッセージを送信するごとにクロックの値を 1 つ増加させる（これを 1 サイクルと呼ぶ）．送信されたメッセージは次のサイクルで，他のエージェントで受信されるとする．このシミュレータを用いて，解を得るのに必要とされるサイクル数を異なる難易度で測定する．ここでの難易度とは，各制約に關係する変数への割当ての組の総数を  $n$ ，制約違反となる割当ての組の総数を  $m$  とし， $m/n$  で与えられる．また，制約違反となる割当ての組はランダムに選んだ．以降では解法が要するサイクル数が最も大きくなるような難易度をピークと呼ぶ．

本論文では，スケールフリー的な制約網は形成ツール<sup>21)</sup>を用いて生成した．このプログラムは，与えられたパラメータ（ノード数，最低次数，ベキ指数）に対して，次数分布がベキ乗則に従うよう，各ノードに次数を割り当てた，ベキ乗則リストを生成する．次に，エッジをランダムに張り，各ノードが最低次数のエッジを持つ制約網を生成する．ここではノード数 100 程度のスケールフリー的な制約網を生成した．このモデルで生成されるスケールフリー的な制約網のクラスタ係数は小さく，BA モデルで生成されるスケールフリーグラフに類似している．本研究は，スケールフリー的な制約網を対象とした場合の，分散制約充足アルゴリズムに関する研究である．このため，複雑ネットワークに関する従来研究と比較して，ネットワーク（制約網）の規模が小さいことは本質的に重要な問題ではない．

エージェントの優先順位を変えながら，非同期バクトラッキングにおけるサイクル数をランダム構造の制約網とスケールフリー的な制約網で評価する．実験では，10 個のランダム構造を持つ制約網 ( $r_1, \dots, r_{10}$ ) およびスケールフリー的な制約網 ( $s_{f1}, \dots, s_{f10}$ ) をそれぞれ生成した．各制約網で分散制約充足問題の難易度を 0.1 から 0.9 とし，各難易度で 100 個のランダムな分散制約充足問題を解かせ，そのときのサイクル数の平均値を求めた．また，ドメイン数は 3 とし，各制約網でエージェントの優先順位をランダムに 10 通り決定した．スケールフリー的な制約網はノード数 100，最低次数 2，ベキ指数 1.8 のパラメータ

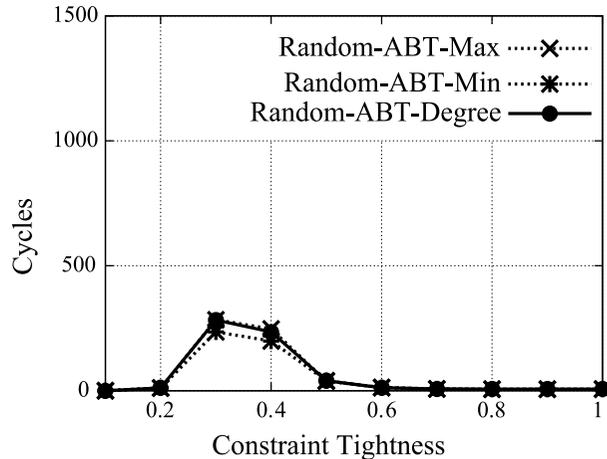


図1 ランダム構造の制約網 ( $r_1$ ) における非同期バクトラッキングの評価  
Fig.1 Performance of ABT in random network  $r_1$ .

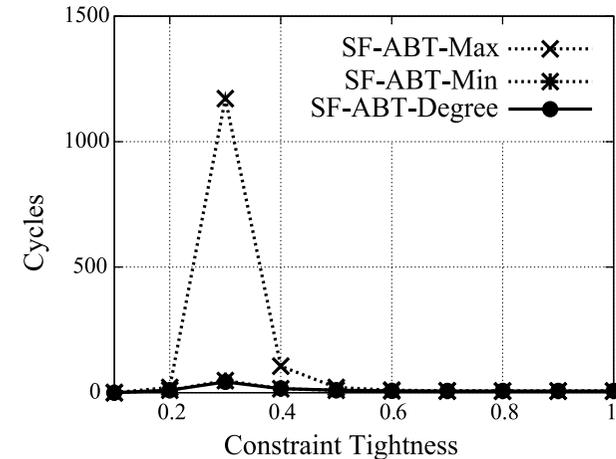


図2 スケールフリー的な制約網 ( $sf_1$ ) における非同期バクトラッキングの評価  
Fig.2 Performance of ABT in scale-free network  $sf_1$ .

で生成した\*1。ランダム構造の制約網は、スケールフリー的な制約網と同じノード数およびエッジ数になるように生成した。図1と図2にランダム構造の制約網 ( $r_1$ ) とスケールフリー的な制約網 ( $sf_1$ ) における実験結果を示す。ここでは、3つの特徴的な結果を示すエージェントの優先順位を与えた場合を選んでいく。x軸は問題の難易度を表し、y軸は各難易度でのサイクル数の平均値を表す。Random-ABT-Max (-Min) とSF-ABT-Max (-Min) は、それぞれの制約網における、ピークでのサイクル数が最大 (最小) となる非同期バクトラッキングのサイクル数の平均値を表す。Random-ABT-Degree とSF-ABT-Degree はエージェントの優先順位を次数順 (降順) に基づくヒューリスティックによって決定した非同期バクトラッキングのサイクル数の平均値を表す。

ランダム構造の制約網では、エージェントの優先順位が変わっても、ピークでのサイクル数はほとんど変化しなかった。これに対し、スケールフリー的な制約網では、ピークでのサイクル数が大きく変化した。図1と図2では制約網の構造にかかわらず、難易度が上がるにつれサイクル数はいったん増加し、難易度0.3付近でピークをむかえ減少していることが

分かる。このことは難易度0.3までの問題には解が存在し、難易度0.3の問題が解の存在する最も難しい問題であることを表している。また難易度0.4以上では解が存在しないことを表している。図1では、どのような優先順位を与えても、ピークでのサイクル数は235から283までの値となり、特徴的な変化はみられなかった。これに対し、図2では、ピークでのサイクル数が47から1,171まで大きく変化した。この結果から、スケールフリー的な制約網ではランダム構造の制約網より、エージェントの優先順位がサイクル数に大きな影響を与えることが分かった。これらの結果は他の制約網 ( $r_2, \dots, r_{10}, sf_2, \dots, sf_{10}$ ) でも変わらなかった。またスケールフリー的な制約網でピークでのサイクル数が最も小さかったとき、エージェントの優先順位は次数順 (降順) に従っていた。

さらに、スケールフリー的な制約網で、エージェントの優先順位をランダムに決定し、擬似木の深さが異なる非同期バクトラッキングを評価した。ここではドメイン数を10とし、スケールフリー的な制約網はノード数100、最低次数2、ベキ指数1.8のパラメータで生成した。表1に擬似木の深さとバックエッジ数、そしてピークでのサイクル数を示す。ここでは、深さが異なる6つの特徴的な結果を選んでいく。Pseudo-tree 4の深さ22はPseudo-tree 1の深さ17より大きい。しかし、Pseudo-tree 4のサイクル数1,673はPseudo-tree 1のサイクル数10,253より小さかった。この結果から、深さが小さくなるようなエージェントの

\*1  $\gamma = 2.2, 2.6, 3.0$  の場合でも、本質的な結果 (エージェントの優先順位の決定法がサイクル数に大きな影響を与えること) は変わらなかった。

表 1 特徴的な結果を示すエージェントの優先順位を与えた場合の、擬似木の深さとバックエッジ数、そしてピークでのサイクル数

Table 1 Depth and number of backedges of pseudo-trees and required cycles at critical point.

Pseudo-tree	深さ	バックエッジ数	サイクル数
Pseudo-tree 1	17	207	10,253
Pseudo-tree 2	14	176	7,815
Pseudo-tree 3	15	220	2,279
Pseudo-tree 4	22	327	1,673
Pseudo-tree 5	13	173	777
Pseudo-tree 6	12	175	380

優先順位を与えても、サイクル数が減少するとは限らないことが分かった。

同様に、バックエッジ数が少なくなるようなエージェントの優先順位を与えても、サイクル数が減少するとは限らないことが分かった。表 1 より、Pseudo-tree 1 のバックエッジ数 207 は Pseudo-tree 4 のバックエッジ数 327 より少ない。しかし、Pseudo-tree 1 のサイクル数 10,253 は Pseudo-tree 4 のサイクル数 1,673 より大きかった。

以上の結果から、スケールフリー的な制約網では、エージェントの優先順位がサイクル数に大きな影響を与えることが分かった。ランダム構造の制約網では、制約網に特徴的な構造がないため、エージェントの優先順位をどのように選んでも、サイクル数に大きな差はなかったと考えられる。一方、スケールフリー的な制約網では、特徴的なエージェント（ハブ）が存在し、これらの特徴的なエージェントの優先順位を、どう決定するかでサイクル数に大きな差が出ていると考えられる。また、異なる深さやバックエッジ数を与えるエージェントの優先順位とサイクル数との間に特徴的な関連性はみられなかった。このことは、深さやバックエッジの数は小さい方が良いという予想に反する結果となっている。この理由として、他の要因（ハブの優先順位等）の方が支配的となっていると考えられる。

## 5. スケールフリー的な制約網に特化した変数順序付けヒューリスティックの提案

本章では、スケールフリー的な制約網に特化した静的な変数順序付けヒューリスティック (variable ordering with Average Length between Hubs, ALH) を提案し、その有効性を示す。具体的には、パラメータの異なるスケールフリー的な制約網で、ALH が次数順に基づくヒューリスティックより、サイクル数を減少させることを示す。4 章の実験で、ピークでのサイクル数が最も小さかったとき、エージェントの優先順位は次数順（降順）に従っていた。ALH は、ハブが擬似木内で異なるブランチに位置するように、各エージェントの優

先順位を決定する。このとき、ハブをルートとする部分木どうしは独立している（制約を持たない）ため、これらの部分木を独立した部分問題と見なすことができる。各部分木ごとの独立した部分問題を別々に解くことは効率的であると考えられる。ALH では、各エージェントの優先順位を、与えられたハブまでの最短距離の平均値を基に決定した。その際、ハブ間の最短距離のパス上に位置するエージェントを要素とする、境界集合を定義し、その優先度を高くした。このように、ALH では各エージェントの優先度を、次数順ではなく、与えられたハブから自分がどれくらい近いかによって決定する。

### 5.1 ALH の概要

$G = (N, E)$  をスケールフリー的な制約網とする。 $N = \{n_i | i \in \mathbb{N}\}$  をノード/エージェントの有限集合とし、 $E = \{e(n_i, n_j) | n_i, n_j \in N, n_i \neq n_j\}$  をエッジ/制約の有限集合とする。非同期バクトラッキングでは、解を得るために前処理として、エージェントの優先順位を基に擬似木が作成される。ALH によってエージェントの優先順位を決定した場合、境界集合の要素は擬似木の上位に位置し、ハブは擬似木内で異なるブランチに位置する。

ある定数  $c \in \mathbb{N}$  以上のエッジを持つノードをハブと呼ぶ。 $H$  をハブの集合とし以下のように定義する。

$$H = \{n_i | n_i \in N, \deg(n_i) \geq c\}.$$

$\deg(n_i)$  はノード  $n_i$  の次数を表す。ALH では、各エージェントが自分は  $H$  に属するかを知っているとす。

$G$  上の直接つながっていない、任意の 2 つのノード  $n_i$  と  $n_j$  をつなぐエッジの列をパスと呼ぶ。 $n_i$  と  $n_j$  間のパス中の、エッジの個数が最小となる値を最短距離と呼び、 $dis(n_i, n_j)$  と記述する。 $H$  内のハブまでの最短距離の平均値を  $n_i^{av}$  とし、以下のように定義する。

$$\forall n_i \in N, n_i^{av} = \sum_{n_j \in H} dis(n_i, n_j) / |H|.$$

またハブ間の最短距離の平均値を  $h^{av}$  とし、以下のように定義する。

$$\forall n_i \in H, h^{av} = \sum_{n_i \in H} n_i^{av} / |H|.$$

条件式  $n_i^{av} \leq h^{av}$  を満たすノードの集合を境界集合  $BS$  と呼び、以下のように定義する。

$$BS = \{n_i | n_i^{av} \leq h^{av}\}.$$

Algorithm 1 に提案手法 ALH を示す。(i) では、定数  $c \in \mathbb{N}$  を与え、ハブの集合  $H$  を決定する。(ii) では、すべてのノード  $n_i \in N$  に対し、ハブまでの最短距離の平均値  $n_i^{av}$  を求める。(iii) では、境界集合  $BS$  を決定する。(iv) では、エージェントの優先順位を、境界集合を基に以下のように決定する。各エージェントの優先度は境界集合を基に決定され、境界集合内のハブ以外のエージェントから高い優先度を付けていく。その際、最短距離の平

**Algorithm 1** ALH

---

```

set  $H = \{n_i | n_i \in N, deg(n_i) \geq c\}$ ; —(i)
FOREACH node  $n_i \in N$ 
 $n_i^{av} \leftarrow \sum_{n_j \in H} dis(n_i, n_j) / |H|$ ; —(ii)
FOREACH  $n_i \in N$ 
if  $n_i^{av} \leq h^{av}$  then
   $BS \leftarrow BS \cup \{n_i\}$ ; —(iii)
end if
determine priority of  $n_i \in BS$  according to  $n_i^{av}$ ; —(iv)
determine priority of  $n_i \in N \setminus BS$  according to  $\sum_{n_j \in BS} dis(n_i, n_j)$ ;

```

---

平均が小さいエージェントほど高い優先度を付ける．もし平均値が同じ場合は，両者の次数を比較し，次数の大きい方に高い優先度を付ける．次にハブに優先度を付ける．その他のエージェントに関しては，境界集合の要素までの最短距離の和を基に優先度を付けていく．すなわち，ノード  $n_i$  に対して，最短距離の和  $\sum_{n_j \in BS} dis(n_i, n_j)$  を求め，値の小さいエージェントから大きな優先度を付ける．ここでも同じ値を持つエージェントが存在する場合は，次数を比較し，次数の大きい方に高い優先度を付ける．提案手法では，各エージェントはハブを知っているため，非同期，並行的に行動することができる．各エージェント（ハブを含む）は，ハブまでの最短距離の平均値  $n_i^{av}$  を求める．ハブは全エージェントに対して，ハブ間の最短距離の平均値  $h^{av}$  を送信する．各エージェントは， $h^{av}$  を基に，自分が境界集合の要素かを知る．すべてのエージェントは，ブロードキャストしながら，自分の優先度を決定する．

ハブどうしの間にエッジがないスケールフリー的な制約網について考える．簡単化のため，ハブの個数が2の場合で議論を進める．この制約網では，ハブどうしの間にエッジがないため，ハブどうしはパスでつながっている．ハブ間の最短距離のパス上には，境界集合の要素となるノードが少なくとも1つ存在する．ALHでは，境界集合内のハブ以外のノードから高い優先度を付けていく．次にハブに優先度を付ける．この優先順位を基に擬似木を作成すると，境界集合内のハブ以外のノードは，ハブの祖先となり，擬似木の上位に位置し，ハブはやや上位に位置する．ハブどうしの間には制約がないため，擬似木内でハブは異なるブランチに位置する．たとえば，図3(c)はALHによって決定したエージェントの優先

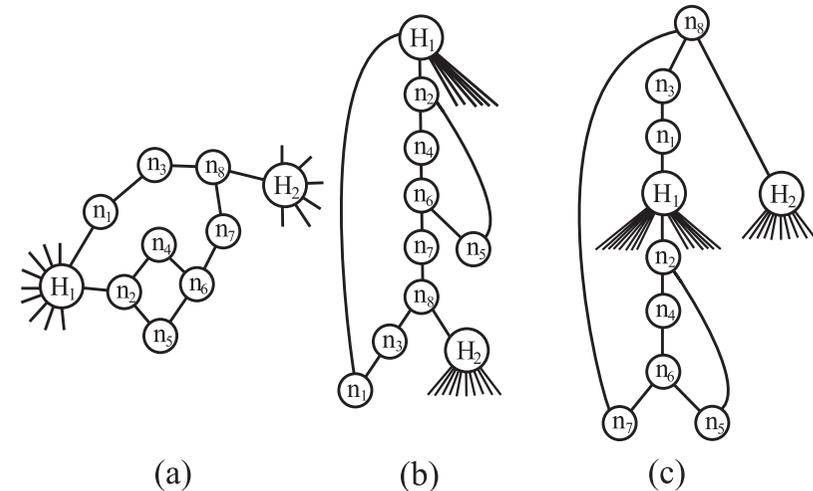


図3 (a) スケールフリー的な制約網． $H_1, H_2$  をハブとする．(b) エージェントの優先順位が度数順（降順）に基づくヒューリスティックによって決定された擬似木．(c) エージェントの優先順位がALHによって決定された擬似木

Fig. 3 (a) Constraint network representing a DisCSP where  $H_1$  and  $H_2$  are hubs, (b) Pseudo-tree determined by degree-based heuristic, and (c) Pseudo-tree determined by ALH.

順位を基に生成した擬似木を表す．図中の  $H_1$  と  $H_2$  はハブを表し，擬似木内で異なるブランチに位置している．ハブをルートとする部分木どうしは独立している（制約を持たない）ため，これらの部分木を独立した部分問題と見なすことができる．各部分木ごとの独立した部分問題を別々に解くことは効率的であると考えられる．ALHは，このようなハブが異なるブランチに位置する擬似木を想定し，各エージェントの優先順位を決定する，変数順序付けヒューリスティックである．これに対し，ハブどうしの間にエッジがあるスケールフリー的な制約網では，境界集合の要素がハブのみとなる．境界集合には，ハブ以外のノードは存在しないため，ハブから高い優先度を付けていく．ハブ以外のノードの優先順位は，ハブまでの最短距離の和によって決定される．この優先順位を基に擬似木を作成すると，優先順位が度数順（降順）に従った擬似木と類似する．すなわち，次数の最も大きいハブが擬似木のルートとなり，その他のハブはルートとなるハブの子孫ノードとなる．また，ハブは擬似木の上位に位置する．提案手法では，各エージェントは，ハブを知っているものとしているため，ハブまでの距離を幅優先探索で求める．このとき，各エージェントは非同期，並行に行

動するため、ALH の計算量は  $O(n)$  となる。よって、ALH を用いた非同期バックトラッキングの計算量は従来の計算量とほとんど変わらない。

ここで簡単な例を与える。図 3 (a) にある分散制約充足問題のスケールフリー的な制約網を示す。  $H_1, H_2, n_1, \dots, n_8$  をエージェントとし、  $H_1, H_2$  をハブとする。各エージェントの次数が以下のような次数順で与えられているとする。

$$\begin{aligned} \text{deg}(H_1) &> \text{deg}(H_2) > \text{deg}(n_2) = \text{deg}(n_6) = \text{deg}(n_8) \\ &> \text{deg}(n_1) = \text{deg}(n_3) = \text{deg}(n_4) = \text{deg}(n_5) = \text{deg}(n_7). \end{aligned}$$

図 3 (b) にエージェントの優先順位が次数順（降順）に基づくヒューリスティックによって決定された擬似木を示す。  $H_1$  は最も大きい次数を持つため、この擬似木のルートとなる。これに対し、ALH では以下の境界集合が定義される。

$$BS = \{H_1, H_2, n_1, n_3, n_8\}.$$

また各ノードの最短距離の平均値はそれぞれ  $h^{av} = 2, n_1^{av} = n_3^{av} = n_8^{av} = 2, n_2^{av} = n_4^{av} = n_5^{av} = n_6^{av} = n_7^{av} = 3$  と求まる。この値を基にエージェントの優先順位を決定すると以下ようになる。

$$n_8, n_1, n_3, H_1, H_2, n_2, n_6, n_4, n_5, n_7$$

図 3 (c) にエージェントの優先順位が ALH によって決定された擬似木を示す。この擬似木では、ハブが異なるブランチに位置していることが分かる。次数順（降順）に基づくヒューリスティックによって決定される擬似木では、ハブがルートか擬似木の上位に位置する。

## 5.2 実 験

パラメータの異なる以下の 3 つのスケールフリー的な制約網で、エージェントの優先順位を次数順（降順）に基づくヒューリスティックと ALH によって決定した非同期バックトラッキングをシミュレーションで評価する。

- (SFN 1): ノード数 100, 最低次数 2, ベキ指数 1.8, 平均次数 2.5,
- (SFN 2): ノード数 200, 最低次数 2, ベキ指数 1.8, 平均次数 2.9,
- (SFN 3): ノード数 100, 最低次数 3, ベキ指数 1.8, 平均次数 3.8,

4 章での実験のとき同様、各パラメータに対して、スケールフリー的な制約網をそれぞれ 10 個ずつ生成した。各制約網で分散制約充足問題の難易度を 0.1 から 0.9 とし、各難易度で 100 個のランダムな分散制約充足問題を解かせ、そのときのサイクル数の平均値を求めた。実験結果は 10 個のスケールフリー的な制約網におけるサイクル数の平均値を表す。ここではドメイン数を 10、ハブ数を 2 とした。

SFN 1 における ALH の有効性を示す。図 4 に SFN 1 における実験結果を示す。x 軸は

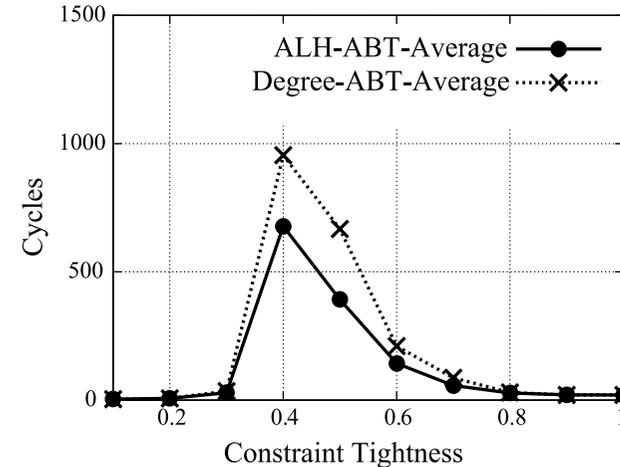


図 4 SFN 1 における、エージェントの優先順位を次数順に基づくヒューリスティックと ALH によって決定した非同期バックトラッキングの評価

Fig. 4 Effect of ALH and standard degree-based heuristics in SFN 1.

分散制約充足問題の難易度を表し、y 軸は各難易度でのサイクル数の平均値を表す。Degree-ABT-Average はエージェントの優先順位を次数順（降順）に基づくヒューリスティックによって決定した非同期バックトラッキングのサイクル数の平均値を表す。ALH-ABT-Average はエージェントの優先順位を ALH によって決定した非同期バックトラッキングのサイクル数の平均値を表す。特徴的なサイクル数の差はピークで現れ、ALH-ABT-Average の有効性が観察された。図 4 では難易度 0.4 付近でピークをむかえ、ALH-ABT-Average のピークでのサイクル数は Degree-ABT-Average の 955 サイクルに対し、約 29%少ない 678 サイクルであった。

SFN 2 における ALH の有効性を示す。この制約網はノード数が 100 から 200 に増えているため、SFN 1 と比べ、より大きな制約網である。図 5 に SFN 2 における実験結果を示す。ノード数を増やしても、ALH の有効性は変わらなかった。ALH-ABT-Average のピークでのサイクル数は、Degree-ABT-Average の 7,486 サイクルに対し、約 19%少ない 6,155 サイクルであった。

SFN 3 における ALH の有効性を示す。この制約網は SFN 1 と比べ、平均次数が 2.5 から 3.8 に増えているため、より複雑な（制約の多い）制約網である。図 6 に SFN 3 における実験

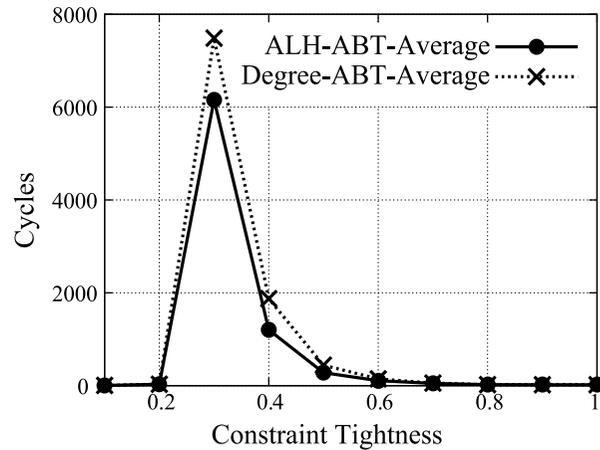


図 5 SFN 2 における，エージェントの優先順位を次数順に基づくヒューリスティックと ALH によって決定した非同期バクトラッキングの評価

Fig. 5 Effect of ALH and standard degree-based heuristics in SFN 2.

結果を示す．最低次数を増やしても，ALH の有効性は変わらなかった．ALH-ABT-Average のピークでのサイクル数は，Degree-ABT-Average の 2,082 サイクルに対し，約 29%少ない 1,489 サイクルであった．

以上の結果から，スケールフリー的な制約網では，ALH が次数順に基づくヒューリスティックより，サイクル数を減少させることが分かった．スケールフリー的な制約網では，予想どおり，ハブの優先順位をどう決定するかで，サイクル数に差がでていた．また，SFN 3 の制約網での実験では，ALH の有効性は変わらなかった．このため，ALH は，より複雑な（制約の多い）スケールフリー的な制約網においても有効であると予想される．

ここでは，ベキ指数を 1.8 に固定して実験を行ったが，ベキ指数を大きくした場合，1 つのノードにすべてのノードがリンクした制約網が生成されると考えられる．このような制約網では，ALH における境界集合の要素は 1 つのハブのみとする．このとき，エージェントの優先順位は，次数順に従ったものと類似するため，ALH の有効性は次数順に基づくヒューリスティックとほとんど変わらなくなると予想される．

さらに，SFN 3 の制約網で ALH におけるハブの個数を変え，非同期バクトラッキングを評価する．ALH の境界集合はハブまでの最短距離の平均値によって定義された．そこで，ハブの個数とサイクル数との関連性を調べるため，ハブ数を 2 から 5 としたときの ALH に

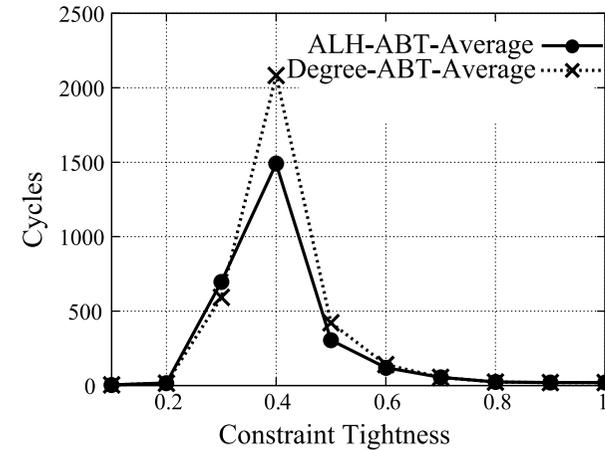


図 6 SFN 3 における，エージェントの優先順位を次数順に基づくヒューリスティックと ALH によって決定した非同期バクトラッキングの評価

Fig. 6 Effect of ALH and standard degree-based heuristics in SFN 3.

よって，エージェントの優先順位を決定した非同期バクトラッキングのサイクル数を評価する．図 7 に SFN 3 における実験結果を示す．ABT-Hub-5 はハブ数を 5 としたときの ALH による非同期バクトラッキングのサイクル数の平均値を表す．ABT-Hub-5 のピークでの 1,048 サイクルは ABT-Hub-2 の 1,004 サイクルとほとんど変わらなかった．この結果から，ALH ではハブの個数を増やしても，サイクル数に影響を及ぼさないことが分かった．このことは，ハブの個数を変えても，擬似木の構造が，それほど変化しなかったためだと考えられる．しかし，この結果は任意のスケールフリー的な制約網に対し，ハブ数をいくつに設定するかを説明しない．任意のスケールフリー的な制約網に対し，ALH でハブ数を表す定数  $c$  の値をどのように決定するかは今後の課題とする．

### 5.3 単純なヒューリスティックとの比較

本節では，ALH における境界集合の要素を，各ハブまでの最短距離の平均値を基に定義する必要があることを示す．ALH では，境界集合の要素は各ハブまでの最短距離の平均値を基に定義した．しかし，より単純な方法を用いて境界集合の要素を決定するヒューリスティックでも，同様の効果が得られるかもしれない．そこで，エージェントの優先順位を境界集合の要素を単純に定義するヒューリスティックと ALH によって決定した非同期バクトラッキングのサイクル数を評価する．

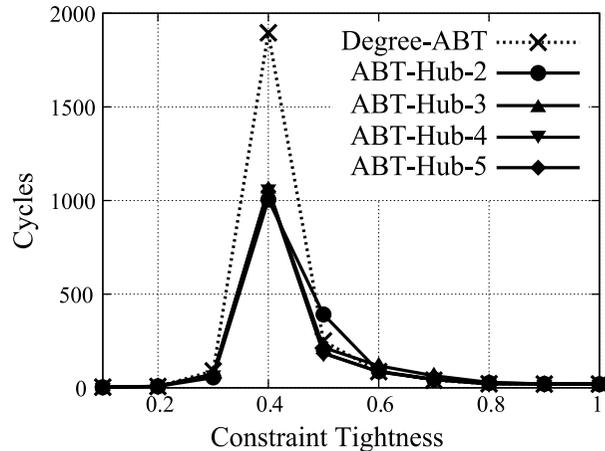


図 7 SFN 3 における，エージェントの優先順位をハブ数が異なる ALH によって決定した非同期バクトラッキングの評価

Fig. 7 Effect of ALH by increasing number of hubs from 2 to 5 in SFN 3.

単純なヒューリスティックでは，境界集合の要素はハブの境界に位置するエージェントを基に定義する．各ノード  $n_i \in N$  から最も近い 2 つのハブ  $h_j, h_k \in H$  までの最短距離を以下のように定義する．

$$h_{i,j}^{min} = \min\{dis(n_i, h_j) | h_j \in H\},$$

$$h_{i,k}^{min} = \min\{dis(n_i, h_k) | h_k \in H \setminus \{h_j\}\}.$$

また以下の条件式を満たすノードの集合を（単純な）境界集合と呼ぶ．

$$\forall n_i \in N, |h_{i,j}^{min} - h_{i,k}^{min}| \leq 1.$$

境界に位置するエージェントはハブまでの最短距離が同じもの ( $h_{i,j}^{min} - h_{i,k}^{min} = 0$ )，もしくは 1 ステップ差のもの ( $|h_{i,j}^{min} - h_{i,k}^{min}| = 1$ ) を選んだ．各エージェントの優先順位は境界集合を基に決定する．ALH 同様，境界集合の要素から高い優先度を付けていく．境界集合に属さないエージェントに関しては，次数の大きい順に優先度を付ける．図 8 に SFN 3 における実験結果を示す．Naive-ABT はエージェントの優先順位を単純なヒューリスティックによって決定した非同期バクトラッキングのサイクル数の平均値を表す．Naive-ABT のピークでの 3,744 サイクルは ALH-ABT の 1,067 サイクルの約 3.7 倍大きかった．この結果から，単純な方法を用いて境界集合を決定するヒューリスティックでは，ALH と同様の効果が得られないことが分かった．したがって，ALH では境界集合の要素を，各ハブま

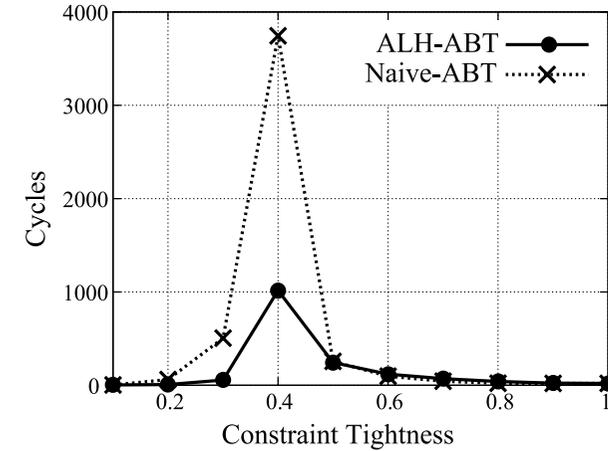


図 8 SFN 3 における，エージェントの優先順位を単純なヒューリスティックと ALH によって決定した非同期バクトラッキングの評価

Fig. 8 Effect of ALH compared to naive heuristic in SFN 3.

での最短距離の平均値を基に定義する必要がある．

## 6. おわりに

本論文では，エージェントの優先順位の決定法が，スケールフリー的な制約網ではランダム構造の制約網より，サイクル数に大きな影響を与えることを示した．実験では，エージェントの優先順位を変えながら，非同期バクトラッキングのサイクル数をランダム構造の制約網とスケールフリー的な制約網で評価した．ランダム構造の制約網では，エージェントの優先順位が変わっても，ピークでのサイクル数はほとんど変化しなかった．これに対し，スケールフリー的な制約網では，ピークでのサイクル数が大きく変化した．

さらに，スケールフリー的な制約網では，エージェントの優先順位によって決定される擬似木の深さやバックエッジ数とサイクル数には，関連性が低いことを示した．実験では異なる深さやバックエッジ数とサイクル数，ピークでのサイクル数を比較したが，特徴的な関連性はみられなかった．

最後に，スケールフリー的な制約網に特化した静的な変数順序付けヒューリスティック (ALH) を提案し，その有効性を示した．実験では，パラメータの異なるスケールフリー的な制約網で，ALH が度数順に基づくヒューリスティックより，サイクル数を減少させるこ

とを示した。また、エージェントの優先順位を ALH によって決定した非同期バクトラッキングでは、回数順に基づくヒューリスティックのときと比べ、ピークでのサイクル数の平均値が最大 29%減少した。

以降では、今後の研究課題について述べる。本論文の実験は具体的な応用問題ではなく、抽象的な、ランダムに生成された問題で行った。問題のパラメータ設定が具体的な応用問題と比較して、現実性がどの程度あるのかに関しては未解明であるため、今後の課題と考えている。また、ハブが受信する *nogood* メッセージ数を調べ、ハブをルートとする部分木で独立した部分問題を、別々に解くことが効率的であることを示す必要がある。さらに、スケールフリー的な制約網に特化した完全性を保証するアルゴリズムおよび動的な変数順序付けヒューリスティックの開発を考えている。

謝辞 本研究の遂行にあたり、本研究は以下の科研費（課題番号 20240015, 23500184, 23650073）の助成を受けました。ここに深く感謝いたします。

### 参 考 文 献

- 1) 横尾 真, エドモンド ダーフィ, 石田 亨, 桑原和宏: 分散制約充足による分散協調問題解決の定式化とその解法, 電子情報通信学会論文誌, Vol.J-75 D-I, No.8, pp.704–713 (1992).
- 2) Yokoo, M. and Hirayama, K.: Algorithms for distributed constraint satisfaction: A review, *Journal of Autonomous Agents and Multi-agent Systems*, Vol.3, No.2, pp.189–211 (2000).
- 3) Barabási, A.-L. and Albert, R.: Emergence of scaling in random networks, *Science*, Vol.286, pp.509–512 (1999).
- 4) Newman, M.: The structure and function of complex networks, *Society for Industrial and Applied Mathematics Review*, Vol.45, pp.167–256 (2003).
- 5) 増田直紀, 今野紀雄: 複雑ネットワークの科学, 産業図書 (2008).
- 6) Walsh, T.: Search in a small world, *Proc. 16th International Joint Conference on Artificial Intelligence*, pp.1172–1177 (1999).
- 7) Walsh, T.: Search on high degree graphs, *Proc. 17th International Joint Conference on Artificial Intelligence*, pp.266–274 (2001).
- 8) Devlin, D. and O’Sullivan, B.: Preferential attachment in constraint networks, *Proc. 21st International Conference on Tools with Artificial Intelligence*, pp.708–715 (2009).
- 9) Chalupsky, H., Gil, Y., Knoblock, C., Lerman, K., Oh, J., Pynadath, D., Russ, T. and Tambe, M.: Electric elves: Agent technology for supporting human organizations, *AI Magazine*, Vol.23, No.2, pp.11–24 (2002).
- 10) Brito, I. and Meseguer, P.: Distributed meeting scheduling, *Artificial Intelligence Research and Development*, Vol.163, pp.38–45 (2007).
- 11) 横尾 真: 弱コミットメント戦略を用いた制約充足問題の解法, 情報処理学会論文誌, Vol.35, No.8, pp.1540–1548 (1994).
- 12) Mailler, R. and Lesser, V.: Asynchronous partial overlay: A new algorithm for solving distributed constraint satisfaction problems, *Journal of Artificial Intelligence Research*, Vol.25, pp.529–576 (2006).
- 13) Zivan, R. and Meisels, A.: Dynamic ordering for asynchronous backtracking on DisCSPs, *Constraints*, Vol.11, No.2-3, pp.179–197 (2006).
- 14) Zivan, R. and Meisels, A.: Message delay and DisCSP search algorithms, *Annals of Mathematics and Artificial Intelligence*, Vol.46, No.4, pp.415–439 (2006).
- 15) Hamadi, Y.: Backtracking in distributed constraint networks, *International Journal on Artificial Intelligence Tools*, pp.219–223 (1998).
- 16) Arbelaez, A. and Hamadi, Y.: Exploiting weak dependencies in tree-based search, *Proc. 24th Annual ACM Symposium on Applied Computing*, pp.1385–1391 (2009).
- 17) Sultanik, E., Lass, R. and Regli, W.: Dynamic configuration of agent organizations, *Proc. 21st International Joint Conference on Artificial Intelligence*, pp.305–311 (2009).
- 18) Petcu, A. and Faltings, B.: A scalable method for multiagent constraint optimization, *Proc. 19th International Joint Conference on Artificial Intelligence*, pp.266–271 (2005).
- 19) Modi, P., Shen, W.-M., Tambe, M. and Yokoo, M.: ADOPT: Asynchronous distributed constraint optimization with quality guarantees, *Artificial Intelligence*, Vol.161, No.1-2, pp.149–180 (2005).
- 20) Erdős, P. and Rényi, A.: On random graphs I, *Publicationes Mathematicae Debrecen*, Vol.6, pp.290–297 (1959).
- 21) Densmore, O.: An exploration of power-law networks (2009), available from (<http://backspaces.net>).

(平成 23 年 3 月 17 日受付)

(平成 23 年 9 月 12 日採録)



沖本 天太 (学生会員)

2008年9月ドイツ・フライブルグ大学情報学部の学部修士一貫教育であるDiplom課程修了。現在、九州大学大学院システム情報科学府博士後期課程在籍中。分散制約充足/最適化問題，ネットワーク科学，群論に関する研究に興味を持つ。人工知能学会学生会員。



岩崎 敦 (正会員)

2002年神戸大学大学院自然科学研究科博士課程修了。同年より2004年までNTTコミュニケーション科学基礎研究所に勤務。2004年より九州大学大学院システム情報科学研究院助教。ゲーム理論，学習，オークション，実験経済学に関する研究に従事。博士(学術)。人工知能学会会員。



横尾 真 (フェロー)

1984年東京大学工学部電子工学科卒業。1986年同大学院修士課程修了。同年NTTに入社。1990～1991年ミシガン大学客員研究員。2004年より九州大学大学院システム情報科学研究院教授。マルチエージェントシステム，制約充足問題に関する研究に従事。エージェントの合意形成メカニズム，制約充足/分散制約充足等に興味を持つ。博士(工学)。1992年，2002年人工知能学会論文賞，1995年情報処理学会坂井記念特別賞，1999年，2005年人工知能学会全国大会優秀論文賞，2004年Association for Computing Machinery (ACM) Special Interest Group on Artificial Intelligence (SIGART) Autonomous Agent Research Award，2005年ソフトウェア学会論文賞，2006年学士院学術奨励賞，2010年人工知能学会業績賞，International Foundation for Autonomous Agents and Multiagent Systems influential paper award受賞。情報処理学会フェロー，人工知能学会，日本ソフトウェア科学会，電子情報通信学会，AAAI各会員。2011年AAAIフェロー。