

# 次数制約付き最小生成木を構築する分散協調探索手法の検討

## A Study of A Distributed Cooperative Search Method for Degree-Constrained Minimum Spanning Tree

伊藤 翼  
Tsubasa Ito

名古屋工業大学, 〒 466-8555 名古屋市昭和区御器所町  
Nagoya Institute of Technology, Gokiso-cho Showa-ku Nagoya 466-8555, Japan  
tsubasa@matlab.nitech.ac.jp

松井 俊浩  
Toshihiro Matsui

(同 上)  
matsui.t@matlab.nitech.ac.jp

松尾 啓志  
Hirosi Matsuo

(同 上)  
matsuo@matlab.nitech.ac.jp

**keywords:** distributed cooperative search, multi-agent system, degree-constrained minimum spanning tree, combinatorial optimization

### Summary

A degree-constrained minimum spanning tree (d-MST) of a graph is a well-studied problem that has the importance in the design of communication and the electric power networks. In this study, we propose the formalization and distributed cooperation methods for d-MST problem in multi-agent systems. The proposed exact/approximate methods resemble the approaches that apply Distributed Constraint Optimization Problem. In the exact method, each agents propagate messages that represent set of sub-trees in a bottom-up manner. To reduce the number of sub-trees constraints of the d-MST are considered. Also, each agent partially drops the sub-trees based on several heuristics in approximate method. We experimentally compare the proposed techniques from the viewpoints of the quality of the solutions and the cost for tree construction.

### 1. はじめに

通信網を介する分散協調システムでは、通信や競合解決のために木構造を利用することがしばしば必要となる。一般に、木構造の多くは最小生成木 (MST: Minimum Spanning Tree) として研究されている [Prim 57, Kruskal 56, Gallager 83, Khan 09]。

その一方で、実際的なネットワークにおいては、多くの通信リンクが、ある特定のノードの物理ネットワークインターフェースを利用する場合、そのノードは次数制約を課せられることが通例である。例えば、交換機やスイッチの多くは限られた数の通信リンクしか物理的には接続できない。そこで、次数制約付き最小生成木 (d-MST: Degree-Constrained Minimum Spanning Tree) がネットワーク資源の消費を削減する重要な概念として研究されている [Narula 80, Boldon 96, Zhou 97, Knowles 00]。一般的に d-MST は MST とは違い、多項式時間アルゴリズムを用いて解くことができないとされる。そのため、多くの近似解法が提案されているが、それらのほとんどは集中的なアルゴリズムであるため、大域的な情報の全てを单一ノードに集約するコストの増加や、そのノードが單一故

障点となる問題が考えられる。その一方で、関連研究では、電力網におけるフィーダ木を構築する問題を分散制約最適化の枠組みで解く手法が提案されている [Kumar 09]。この手法は、特定の領域に特化しているが、これに類似する手法を d-MST のような、より一般的な制約付きの生成木構築問題に適用する余地があると考えられる。そこで、本稿ではネットワーク中の各ノードをエージェントとみなし、マルチエージェントの分散協調問題の枠組みで d-MST 問題を形式化し、問題を解く。

分散型 d-MST については、従来研究における木構築問題のための分散制約最適化手法に類似する厳密解法、および近似解法を提案する。いずれの手法においても各エージェントが最大でも 1 本の接続辺を決定することで大域的な d-MST の解を成す。これらの 2 種類の手法を解の質、精度、木の構築コスト、探索空間の規模といった観点から比較評価する。

2 章では次数制約付き最小生成木問題について、3 章では、関連研究について述べる。4 章では問題の形式化について、5 章では提案手法について述べる。6 章では分散協調手法の比較・評価について述べ、7 章でまとめる。

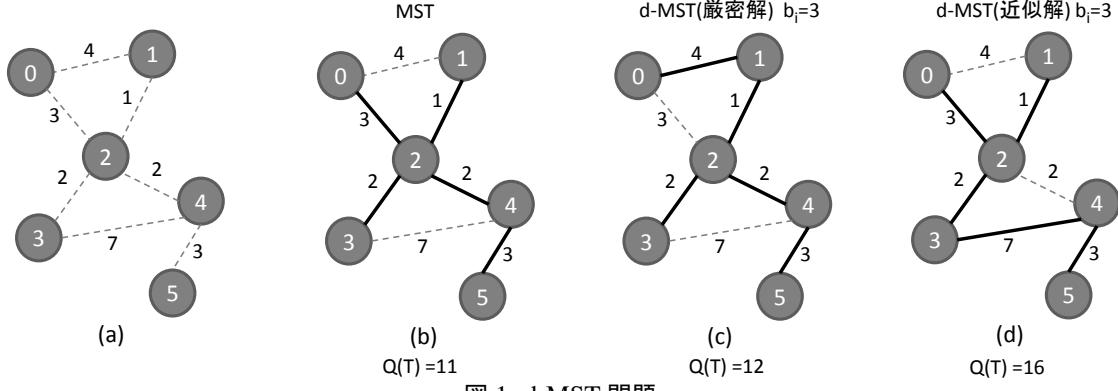


図 1: d-MST 問題

## 2. 次数制約付き最小生成木問題

本章では、本研究で対象としているネットワーク、および、次数制約付き最小生成木問題の詳細について述べる。

本研究において、ネットワークとは、マルチエージェントがメッセージ交換により相互作用する通信ネットワークを想定する。通信ネットワークにおける通信リンクは全二重かつ対称的であり、エージェント間で自由にメッセージの送受信が可能である。

MST 問題は、組み合わせ最適化問題の 1 つである。MST とは、入力として重み付き無向グラフ  $G = (V, E, c)$  が与えられ、そのグラフを構成する  $n$  個の頂点(エージェント)の集合  $V = \{i | i = 0, 1, \dots, n - 1\}$ 、および各頂点  $i, j$  を結ぶ辺(通信リンク)の集合  $E = \{(i, j) | i, j \in V \wedge i \neq j\}$ 、辺のコスト関数  $c: R$  が与えられるときに、閉路を含まず、かつ、 $\sum_{(i,j) \in E} c(i, j)$  を最小化する連結無向成分  $T = (V_T, E_T)$ 、( $E_T \subset E, V = V_T$ ) である。それに加えて、d-MST 問題は、各頂点の次数  $d(i)$  に対し、次数の制限数  $b_i$  が与えられ、構築する次数制約付き生成木  $T = (V_T, E_T)$  に含まれる各エージェント  $i$  の次数を  $d_T(i)$  とするとき、 $d_T(i) \leq b_i$  を満たし、かつ、 $\sum_{(i,j) \in E} c(i, j)$  を最小化する  $T$  を求める問題である。解の質を  $Q(T) = \sum_{(i,j) \in E} c(i, j)$  と定義する。

図 1 の (a) のようなグラフが与えられるとき、(b) のような MST、(c)、(d) のような d-MST を得る。(b) の MST は、次数制約がないため、最小の値  $Q(T) = 11$  をとる。(c) は d-MST の厳密解、(d) は近似解であり、それぞれ  $Q(T) = 12, Q(T) = 16$  となる。すなわち、同じ次数制約を満たす木でも  $Q(T)$  に差がある。例えば、逐次的に辺を接続するような手法において、(d) のような解となる。仮に (a) において始点頂点を 0 として、そこから順々に辺を接続していくとする。まず、最初に頂点 0 は、隣接辺の中で最小のコスト値 3 を持つ辺 (0, 2) を選択する。次に、頂点 0, 2 の隣接辺の中で、まだ選択していない辺と頂点に対して、最小のコスト値 1 を持つ辺 (2, 1) を接続する。同様に頂点 0, 2, 1 の隣接辺を選択するが、辺 (2, 3) と边 (2, 4) が同等のコスト値 2 を持つため、どちらか 1 つを選択しなければならない。そこで、例えば、頂点番号による優先順序をつけた場合、辺

(2, 3) が選択され、結果として、(d) とは違う木となる。MST 問題には、多くの決定的アルゴリズムが存在する [Prim 57, Kruskal 56]。Prim や Kruskal のアルゴリズムは貪欲法を用いることで、多項式時間で MST 問題を解決可能である。それに対して、d-MST 問題は、一般的に  $2 \leq b_i \leq n - 1$  のときに NP 困難な問題とされている。そのため、多くの近似的な解法が提案されている [Narula 80, Boldon 96, Zhou 97, Knowles 00]。この問題は、特に  $b_i = 2$  のときに、最小ハミルトン路問題と同等な問題として扱われる [Garey 79]。また、d-MST 問題は、ユークリッド距離で辺のコストを表す問題と、ランダムな値で辺のコストを表す 2 種類に分かれる [Knowles 00]。ユークリッド平面における d-MST は  $b_i = 3$  の場合と  $b_i = 4$  の場合、共に NP 困難であるとされるが、 $b_i = 4$  の場合の方が比較的簡単に解けるとされている [Papadimitriou 84]。本研究ではランダムな値で辺のコストを表す問題を扱い、6 章で、 $\forall i \in V$  に対して  $b_i = 3$  として実験を行う。

## 3. 関連研究

d-MST 問題および MST 問題に関する関連研究について 3・1 節、3・2 節で述べる。これらを基礎とする手法を 6 章の評価における比較手法として用いる。また、制約付き生成木構築問題に類似する分散制約最適化問題の研究として、配電網におけるフィーダツリー構築問題に関する研究について 3・3 節で述べる。提案手法は、この問題の解法と類似すると考えられる。

### 3・1 d-MST の解法

[Narula 80]において、primal method, dual method, branch and procedure の 3 つの手法が提案されている。

primal method は、Prim のアルゴリズム [Prim 57] を d-MST の解法として活用した手法 (d-prim) である。Prim のアルゴリズムのように、 $G = (V, E, c)$  が与えられるとき、ある一つの始点ノードのみからなる部分木  $T' = (V_{T'}, E_{T'})$  から処理を開始する。その部分木に属する頂点  $i \in V_{T'}$  と、部分木に属さない頂点  $j \in V - V_{T'}$  を繋ぐ辺のコスト  $c(i, j)$  が最小となる頂点  $j$  と辺  $(i, j)$  を部分木  $T'$  に加えていく。ただし、その過程で  $d(j) < b_j$  を満たす頂点  $j$  のみを選択し、部分木に加える。最終的

に  $V = V_{T'}$  となるとき、低コストな次数制約付き生成木 (DCST: Degree Constrained Spanning Tree)  $T$  を得る。その後、DCST における次数の制限を超えないように  $T$  の辺の組み換えを行い、 $Q(T)$  を最適解に近付ける。2 章でも述べたように、この手法のように逐次的に辺を選択し、バックトラックを行わない近似アルゴリズムの場合、 $Q(T)$  は処理を開始する頂点や、辺の選択順序に依存する。従って、 $Q(T)$  は最適解である保証はない。

dual method は、primal method と同様に Prim のアルゴリズムを d-MST の解法として活用した手法である。まず最初に、prim のアルゴリズムを用いて最小生成木  $T = (V, E)$  を作成する。その後、 $T$  に属する頂点  $i \in V$  の中で、 $d_T(i) > b_i$  である頂点が成す辺  $(i, j)$  を削除し、 $d_T(r) + 1 < b_r$ かつ  $d_T(s) + 1 < b_s$ かつ  $c(i, j) - c(r, s)$  が最小となるような辺  $(r, s)$  を新たに  $T$  に加えることで  $Q(T)$  を最適解に近付ける。この手法も primal method と同様に最適解を得る保証はない。

branch and bound procedure は、分枝限定法を用いることで、部分解(部分木)をバックトラックしながら探索し、DCMST の最適解を得る厳密解法である。解の探索の際に、下界値として MST のコストの総和値、上界値として DCST のコストの総和値を用いることで枝刈りを行う。本研究では、この手法の様に分枝限定法を用いた手法を実装し、得た解を基に 6 章において、各手法の解の最適解からの誤差を評価する。

本研究で用いる分枝限定法のアルゴリズムでは、d-MST のある部分問題の部分木を  $T' = (V_{T'}, E_{T'})$  とし、 $(i, j) \in E - E_{T'}$  を  $T'$  に追加することで生成される最小生成木を  $T''$  とするとき、下界値を  $Q(T') + Q(T'')$  とする。また、それまでに探索して得た完全解のコストの最良値を上界値として用いる。処理の概要は以下の通りである。

- (1) 部分木  $T' = (V_{T'}, E_{T'}) = (\{\}, \{\})$  を作成する。
- (2)  $d(i) < b_i$ かつ  $d(j) < b_j$  を満たし、コストが最小な頂点  $i, j \in V - V_{T'}$  と辺  $(i, j) \in E - E_{T'}$  を 1 組選択する。
- (3) もし、 $T'$  が閉路を含まず、かつ連結であれば、部分木  $T'$  に追加する。
- (4) もし、 $V_{T'} \neq V$  ならば(2)からの処理を繰り返す。  
 $V_{T'} = V$  ならば次数制約付き生成木  $T$  を得る。 $Q(T)$  が上界値より小さければ解と上界値を更新し、別の部分木が生成できる所までバックトラックし、(2)からの処理を繰り返す。ただし、その時点の下界値が現在の上界値以上ならば枝刈りする。

### 3.2 MST の近似解法

[Khan 09] では、分散配置されたセンサーが、少ないメッセージ交換で、MST を構築する近似解法(nnt アルゴリズム)を提案している。nnt アルゴリズムでは、各エージェント  $i$  は、 $rank(i)$  というランダムな値を持つ。リーダーとなるエージェント  $s$  から近傍のエージェント間で

メッセージを交換することで、初期木が構築される。この初期木に基づいて、全てのエージェントは自身の rank 値を決定し、その rank 値を基に順序付けられる。リーダー以外の全てのエージェント  $i$  は、 $rank(i) < rank(j)$  となる近傍エージェント  $j$  の中で、 $c(i, j)$  が最小となる辺  $(i, j)$  を 1 つだけ接続することで、コストを最小化できる可能性の高い経路を選択する。これにより、閉路検出の処理を完全に省いて生成木を構築することができる。

5 章において、nnt アルゴリズムをベースとする d-MST の近似解法について述べる。

### 3.3 電力網の経路割り当て問題のフィーダ木の構築

[Kumar 09]において、分散制約充足/最適化問題 [Modi 05, Petcu 05, Kumar 09] を電力網の経路割り当て問題に適用する手法が提案されている。電力網は、電力を供給するソース、消費するシンク、送信する配電線からなる。全てのシンクに対して電力を供給するために、配電経路はソースを根とする生成木とすること、配電線に流れる電力は配電線の最大容量を違反しないことが、制約条件となる<sup>\*1</sup>。この木制約を満たすために、各エージェント  $i$  は、供給元変数  $d_i$  を持つ。このような変数は生成木の有効辺の情報を持つので、本研究ではディレクション変数と呼ぶこととする。 $d_i = j$  のとき、エージェント  $i$  はエージェント  $j$  から電力を供給されることを表す。ディレクション変数に対して、評価関数  $f_i(d_i, d_j)$  が定義されている。 $d_i$  と  $f_i$  による評価は以下のように行う。

- $d_i = j \wedge d_j = i$  のとき、 $f_i(d_i, d_j) = \infty$  となり、制約違反を表す。
- $d_j \neq i$  のとき、 $f_i(d_i, d_j) = 0$  となり、 $j$  は  $i$  から電力を供給されないことを表す。
- $d_j = i$  のとき、 $f_i(d_i, d_j) = c(i, j)$  となり、 $j$  は  $i$  から  $c(i, j)$  分の電力を供給されることを表す。

また、[Kumar 09]では、分散制約最適化問題に対する最適解を求める厳密解法である DPOP アルゴリズム [Petcu 05] のような解法を応用し、フィーダ木の生成問題のために特化させることで、電力網の経路割り当て問題を解いている。

本研究では、4 章において、変数  $d_i$  に類似したディレクション変数を用いた形式化を、5 章において、その形式化に基づいた [Kumar 09] の解法に類似する分散型の厳密解法を提案する。

### 4. 分散 d-MST 問題の形式化

多くの d-MST の既存手法は集中型の解法であるが、本研究では d-MST のような組み合わせ最適化問題は、例えば通信ネットワークの設計に適用する場合、分散された資源(通信リンク)を基に問題を解くため、マルチエージェントシステムの協調問題解決の枠組みにおいて解くことは有用であると考える。そこで、各エージェントに

<sup>\*1</sup> なお、文献 [Kumar 09] の問題では、複数のソースがある場合には、複数の生成木にネットワークが分割される。

変数と評価関数が分散して配置された分散 d-MST 問題として d-MST 問題の形式化を行う。なお、本研究の形式化はエージェントに分散して配置された変数と制約・評価関数の情報を基に分散アルゴリズムにより解を得る分散制約最適化問題への適用に向けた基礎的な位置付けとする。

3 章で述べたように、分散制約最適化問題を扱う [Kumar 09] に類似した形式化を行う。[Kumar 09] が電力網の経路割り当て問題におけるフィーダ木の構築に特化した形式化を行うのに対し、本研究では、より一般的な木の構築問題である d-MST 問題に類似の形式化を適用する。分散 d-MST 問題の構成要素を以下のように定義する。

- マルチエージェントシステムに含まれる  $n$  個のエージェントの集合を  $A = \{i | i = 0, 1, \dots, n-1\}$  で表す ( $V$  と同等)。
  - 変数の集合を  $X = \{x_i | i = 0, 1, \dots, n-1\}$  で表す。各エージェント  $i \in A$  は、自身のディレクション変数  $x_i \in X$  を持つ。
  - エージェント  $i$  の隣接エージェント集合を  $Nbr_i$  で表す。
  - 変数  $x_i$  の値域を  $D_i$  で表す。 $D_i = Nbr_i \cup \{\phi\}$  とし、 $x_i$  の変数値  $d_i \in D_i$  は、エージェント  $i$  のみが決定可能であるとする。
  - 各エージェント  $i$  は評価関数  $f_i$  を持つ。 $f_i$  は自身と関連する変数値から評価値への写像を表す。
- 上記の構成要素のもとで、以下の制約を満たす。
- 木制約:  $T = (A_T, E_T)$  は生成木を成す ( $T$  は、連結かつ閉路がなく、 $A = A_T$ )。
  - 次数制約:  $\forall i \in A$  に対して、 $d(i) \leq b_i$  を満たす。
- 各エージェント  $i$  は、以下の評価関数を持つ。

$$f_{cost_i}(i) = \begin{cases} c(i, x_i) & i \neq x_i \\ 0 & x_i = \phi \\ \infty & \text{otherwise} \end{cases} \quad (1)$$

$$f_{deg1_i}(i) = \begin{cases} 1 & f_{cost_i}(i) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$f_{deg2_i}(i, j) = \begin{cases} 1 & x_j = i \wedge x_i \neq j \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

この形式化においては、各エージェント  $i$  の変数  $x_i$  の値域  $D_i$  は接続すべき近傍エージェント集合と空集合の和  $Nbr_i \cup \{\phi\}$  となる。例えば、図 1 のグラフ (a) の場合、エージェント 2 に関して  $x_2 = \{0, 1, 3, 4, \phi\}$  であり、 $x_2 = 0$  と変数値を決定するとき、自身の接続辺を  $(0, 2)$  に決定することを意味する。 $x_2 = \phi$  とするとき、エージェント 2 は、辺を接続しないことを意味する。ある 1 つのエージェント  $s$  が自身の持つディレクション変数値を  $x_s = \phi$  と決定し、それ以外の全てのエージェント  $i$  が  $x_i$  を  $d_i \in Nbr_i$  に決定しなければならない。これは

$|E_T| = |A_T| - 1$  個の辺が接続されなければ、生成木の条件を満たさないためである。全てのエージェント  $\forall i$  の集合  $A_T$  と、辺  $(i, d_i)$  の集合  $E_T$  からなる木  $T = (A_T, E_T)$  は、生成木を表す。例えば、(a) において、各エージェントが、それぞれ  $x_0 = 1, x_1 = 2, x_2 = 4, x_3 = 2, x_4 = 5, x_5 = \phi$  のような変数値の決定を行った場合、 $T$  は (c) の d-MST と同等である。

d-MST の、ある部分問題における各エージェント  $i$  の集合  $A_{T'}$  と変数値の組み合わせが表す部分木  $T' = (A_{T'}, E_{T'})$  において、制約による組み合わせの削除が可能である。例えば、図 1 の (a) のグラフにおいて、部分木  $T' = (A_{T'} = \{0, 1, 2\}, E_{T'} = \{(0, 1), (1, 2), (2, 0)\})$  のような部分木が構築されるとする。この部分木においては、 $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$  のような閉路が存在し、その時点では木制約を満たさないため、その変数値の組み合わせを削除する。また、 $E_{T'} = \{(0, 1), (1, 2), (2, 1)\}$  のように複数のエージェントが同じ辺を選択する場合は、最終的に全てのエージェントが連結されなくなるため、これも削除する。この制約は式 1 を用いて、式 4 のように表される。

$$\forall i \in A_{T'}, f_{cost_i}(i) \neq \infty \quad (4)$$

また、ある時点で複数のエージェントが  $x_i = \phi$  と決定した場合、 $|E_{T'}| > |A_{T'}| - 1$  となり、全てのエージェントが連結されなくなるため削除する。この制約は式 2 を用いて、式 5 のように表される。

$$\sum_{i \in A_{T'}} f_{deg1_i}(i) \geq |A_{T'}| - 1 \quad (5)$$

また、ある時点で、次数の制限数を超えるような変数の組み合わせについても削除する。この制約は式 2 と式 3 を用いて、式 5 のように表される。

$$\forall i \in A_{T'}, f_{deg1_i}(i) + \sum_{j \in Nbr_i \cup A_{T'}} f_{deg2_i}(i, j) \leq b_i \quad (6)$$

上記の制約を満たした上で、式 7 に示す式を最適化する木  $T$  を目的とする。

$$Q(T) = \min \sum_{i \in A} f_{cost_i}(i) \quad (7)$$

## 5. 提案手法-分散協調探索手法-

本章では、4 章の形式化に基づく分散型 d-MST の厳密解法 (dd-mst) を提案する。また、dd-mst の探索空間を削減する手法と nnt のアルゴリズムをベースに d-MST を構築する 2 つの近似解法についても示す。

### 5.1 厳密解法

dd-mst は各エージェントがメッセージ交換を伴う協調探索により d-MST の厳密解を求める手法である。[Kumar 09] の手法に類似する手法を d-MST の構築に適用させたものといえる。各エージェントは、隣接エージェントから受信する部分解 (部分木集合) と自身の変数値との組み

```

1 Phase1: ordered tree generation
2 generate ordered tree;
3 set  $prt_i$ ,  $child_i$ ;
4
5 Phase2: partial solution message propagation
6 if  $|child_i| == 0$  then
7   foreach  $d_i \in D_i$  do
8      $T'_i(k_i) \leftarrow (V_{T'_i(k_i)} \leftarrow \{i\}, E_{T'_i(k_i)} \leftarrow \{(i, d_i)\})$ 
9     if  $T'_i(k_i)$  satisfies all constraints then
10       $S_{T'}(i) \leftarrow S_{T'}(i) \cup T'_i(k_i)$ ;
11    else
12      execute Upon receipt of set  $S_{T'}(j)$ ;
13
14 Phase3: optimal answer message propagation
15 execute Upon receipt of set  $Opt$ ;
16
17 Upon receipt of set  $S_{T'}(j)$  from agent  $j$ :
18    $S_{T'}(i) \leftarrow Calculate\_Subtrees(S_{T'}(j))$ ;
19   if  $prt_i != NULL$  then
20     send message  $\langle S_{T'}(i) \rangle$  to  $prt_i$ ;
21   else
22      $Opt \leftarrow Make\_Optimal\_Set(S_{T'}(i))$ ;
23      $x_i \leftarrow Choose\_Optimal(Opt)$ ;
24
25 Upon receipt of set  $Opt$  from agent  $j$ :
26    $x_i \leftarrow Choose\_Optimal(Opt)$ ;
27   if  $|child_i| != 0$  then
28     send message  $\langle Opt \rangle$  to  $child_i$ ;
29   halt;
30
31 procedure Calculate_Subtrees( $S_{T'}(j)$ ):
32   foreach  $T'_j(k_j) \in S_{T'}(j)$  ( $k_j = 0, 1, \dots, |S_{T'}(j)| - 1$ ) do
33     foreach  $d_i \in D_i$  do
34        $T'_i(k_i) \leftarrow (V_{T'_i(k_i)} \leftarrow V_{T'_j(k_j)} \cup \{i\},$ 
35          $E_{T'_i(k_i)} \leftarrow E_{T'_j(k_j)} \cup \{(i, d_i)\})$ ;
36       if  $T'_i(k_i)$  satisfies all constraints then
37          $S_{T'}(i) \leftarrow S_{T'}(i) \cup T'_i(k_i)$ ;
38     return  $(T'_i(k_i))$ ;
39
40 procedure Make_Optimal_Set( $S_{T'}(i)$ ):
41   foreach  $T'_i(k_i) \in S_{T'}(i)$  do
42     if  $T'_i(k_i)$  has  $\min\{Q(T'_i(k_i))\}$  then
43       foreach  $a$  such  $(a, d_b) \in E_{T'_i(k_i)}$  do
44          $value_a \leftarrow d_b$ 
45          $Opt \leftarrow Opt \cup value_a$ 
46     return  $(Opt)$ ;
47
48 procedure Choose_Optimal( $Opt$ ):
49   foreach  $value_k \in Opt$  do
50     if  $(i == k)$  then
51        $x_i \leftarrow value_k$ ;
52   return  $(d_i)$ ;

```

図 2: dd-mst

合わせを新たに部分解として計算し、次エージェントに引き継ぐ。3章で述べた [Petcu 05, Kumar 09] の手法のように、各エージェントの部分解は、あらかじめ順序付けられた構造に従ってボトムアップに集計される。最終的には、最上位のエージェントが大域的な解(d-MST)を得る。そして、最上位のエージェントが最適解を下位のエージェントに伝播させることで、全てのエージェントがるべき変数値を決定する。この手法は(1)順序木の生成(2)部分解の伝播(3)最適解の伝播の3つのフェーズからなる。図2にdd-mstの処理を表す擬似コードを示す。エージェント*i*を主体とした処理とする。

### §1 順序木の生成

フェーズ(1)では、フェーズ(2)、フェーズ(3)のための前処理を行う。先に述べたように、本手法では、部分解の伝播を行うために何らかの順序を必要とする。従つ

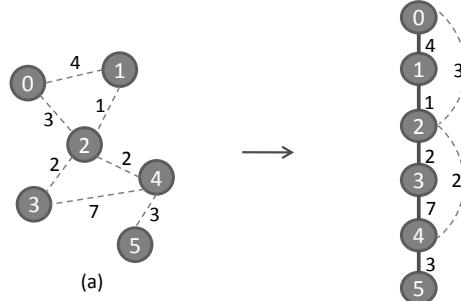


図 3: 順序木の生成

て、あらかじめ生成木のような初期構造を構築することで、問題を解くための順序をエージェントに与える。本手法では、これを順序木と呼ぶ。例えば、2章で示した図1のグラフ例題(a)から、図3で示されるような各エージェントが1列に繋がれた生成木を作成する。なお、4章の形式化に基づく場合、エージェントの順位付けとして、任意の生成木を用いることができると言われるが、本研究では、簡単のために最も単純な線形なグラフを用いる。順序木の親、子、葉、根にあたるエージェントをそれぞれ、親エージェント、子エージェント、葉エージェント、根エージェントと呼ぶ。また、エージェント*i*の親エージェントを $prt_i$ 、子エージェントを $child_i$ で表す。全てのエージェント*i*は、dd-mstの開始時に図2の1-3行目に示すordered tree generationを実行し、自身の $prt_i$ 、 $child_i$ を決定することで、順序性を得る。

### §2 部分解の伝播

フェーズ(2)では、各エージェントが自身の値域を基に計算した部分解(部分木集合)の伝播を行う。フェーズ(2)は、葉エージェント*l*から開始される。葉エージェント*l*は、図2の5-10行目に示すように、自身の変数 $x_l$ のとりうる値の組み合わせから $|D_l|$ 通りの部分木 $T'_l(k_l)$ 、( $k_l = 0, 1, \dots, |D_l| - 1$ )を計算し、新たな部分木集合を生成する。なお、9, 36行目のall constraintsとは、4章で述べた部分木を表す各ディレクション変数の整合性や辺の数、次数の制約である。もし、部分木 $T'_l(k_l)$ が、全ての制約を満たせば、その部分木を、部分木の集合 $S_{T'}(l)$ に追加し、 $S_{T'}(l)$ を親エージェント $prt_l$ へ送信する。また、葉エージェント以外のエージェント*i*は、31-38行目に示すprocedure Calculate\_Subtreesにおける処理に従って、子エージェント $child_i$ から受信した部分木集合 $S_{T'}(child_i)$ と、自身の変数 $x_i$ のとりうる値との組み合わせから $(|S_{T'}(child_i)| \times |D_i|) - 1$ 通りの部分木 $T'_i(k_i)$ 、( $k_i = 0, 1, \dots, |S_{T'}(child_i)| \times |D_i| - 1$ )を計算する。もし、部分木 $T'_i(k_i)$ が、全ての制約を満たせば、その部分木を部分木の集合 $S_{T'}(i)$ に追加し、 $S_{T'}(i)$ を親エージェント $prt_i$ へ送信する。

### §3 最適解の伝播

フェーズ(3)では、フェーズ(2)の部分解の伝播により導き出される最適解の伝播を行う。根エージェント*r*は、図2の21-23行目に示すように、次数制約を満たし、かつ $Q(T'_r(k_r))$ を最小化する生成木を最適解(d-MST)とし

て選び，その d-MST を構成する子孫のノードのとるべき変数値の集合  $Opt = \{value_v | v = 0, 1, \dots, n - 1\}$  を生成する．そして，自身の変数  $x_r$  を  $value_r \in Opt$  に決定し，子エージェント  $child_r$  へ送信する． $prt_i$  から  $Opt$  を受信したエージェント  $i$  は，自身の変数  $x_i$  を  $value_i \in Opt$  に決定し， $Opt$  を  $child_i$  へ送信する．最終的に，葉エージェント  $l$  が  $x_l$  を決定したとき，処理を終了する．

## 5.2 近似解法

5.2.1 節では，dd-mst における貪欲的手法の適用のための初期の検討として，探索空間削減手法の簡易な方法を示す．5.2.2 節では，それらの手法に対して，探索空間の規模と解の質とのトレードオフの議論を行うために，3 章で述べた nnt アルゴリズムに 4 章で示した形式化を適用し，次数制約を追加した手法を示す．

### §1 dd-mst における探索空間削減手法

dd-mst は，仮に制約を一切考慮しない場合，探索すべき最大の組み合わせは， $\prod_{i \in A} |D_i|$  となり，指数関数的に増加する．前述のように，各エージェントにおいて制約違反する部分木を削除することにより，その組み合わせは削減される．しかし，削減が不十分であれば，組み合わせは爆発的に増大する．従って，実際には各エージェントの送受するメッセージ容量や，データの記憶領域の容量を考慮し，保持する部分木の数を制限することが望ましい．そこで，本研究では，dd-mst における探索空間を削減する初期の検討として以下の 2 手法について示す．

- (1) 各エージェント  $i$  が生成する部分木の組み合わせを，最初の  $k$  個のみに制限し， $k$  個を超える組み合わせは切り捨てる手法．
- (2) 各エージェント  $i$  が生成する部分木  $T'_i(k_i)$  に関して  $Q(T'_i(k_i))$  の小さい順に優先順位を付け，その上位  $k$  個のみを選択する手法．

(1)，(2) の両手法は，分散探索手法における探索空間の大きさと，解の質，精度とのトレードオフの関係に対する基本的なヒューリスティクスとして用いる．ただし， $k$  による制限を加える場合，最終的に解がないという可能性が考えられる．その場合はエージェント順序を定める生成木が次数制約を満たすものであれば，ベースラインとして，その木を流用できるため， $x_i = prt_i$  することで解を確保する．

### §2 nnt をベースとする分散構築手法

3 章において述べた nnt アルゴリズムは MST を構築する手法であるが，これに簡単なヒューリスティクスを適用することで次数制約を満たす分散型 d-MST の近似解法とする (d-nnt)．nnt アルゴリズムでは，リーダー以外の全てのエージェントが rank を用いて，辺の接続先を 1 つだけ選ぶ．そのため，4 章で述べたディレクション変数を用いる形式化を適用可能である．d-nnt も dd-mst と同様，順序木に基づく順序関係をランク値の代わりとして用いる．nnt とは違い，辺の接続を決定する過程で  $d(i) = b_i$  となったエージェント  $i$  は，それ以降，他エージェント

からの接続を拒否し，拒否されたエージェントは別の辺に接続を行うことで，次数制約を満たす．ただし，最後まで辺を決定できないエージェントが存在する可能性があるため，その場合は 5.2.1 節で述べたように，あらかじめ生成した順序木の関係に基づき， $x_i = prt_i$  として，解を確保する．各エージェントが受信した部分木集合と自身の値域との組み合わせを計算する dd-mst に対し，d-nnt では各エージェントが自身の値域に関してのみ計算を行うことで，探索空間の増大を線形に保つ．

## 6. 実験・評価

6.1 節で提案手法における探索空間の削減に関する予備評価を行う．6.2 節では，提案手法の探索空間に関する評価を，6.3 節では，厳密解法と近似解法に関する比較評価を行う．

本実験では，分散型の厳密解法が問題を解ける範囲で実験を行うために，辺数  $en$  を小さい値に設定し，極めて規模の小さい(辺密度の小さい)問題を作成した．また，グラフの辺に割り当てるコスト値の種類  $ctype$  は，一様分布に従う 10 から 100 の範囲の整数値 ( $L$ ) と一様分布に従う 10 から 500 の範囲の整数値 ( $H$ ) の 2 種類とした．各エージェント数  $n$  ( $5 \leq n \leq 30$ ) に対して，10 個の問題インスタンスを用意し，10 個の問題を解いた解に関する各評価指標の平均値を評価する．いずれの問題グラフも 2 章で述べたような近似解法の解を近似解に誘導可能なグラフである．また，全てのエージェント  $i$  に対して， $b_i = 3$  の次数制約を与える．評価の対象とする手法は，以下の通りである．

- dd-prim: d-prim を分散処理化した近似解法
- d-nnt: nnt に次数制約を追加した近似解法 (d-nnt)
- dd-mst: 分散型 d-MST の厳密解法 (dd-mst)
- dd-mst-cl:  $k$  個の組み合わせのみを保持する dd-mst
- dd-mst-tp:  $Q(T)$  の小さい上位  $k$  個の組み合わせのみを保持する dd-mst

分散協調手法である dd-mst の比較対象として，逐次的な解法である d-prim のアルゴリズムを分散処理化した手法を dd-prim として用意する．dd-prim では，リーダーとなるエージェント  $s$  を中心とし，メッセージ交換を伴う辺の探索，辺情報の集約を反復的に繰り返すことで d-MST を構築する．

### 6.1 探索空間の削減手法に関する予備実験

dd-mst-cl と dd-mst-tp の 2 手法に関して，解の探索空間とパラメータ  $k$  の関係について予備実験を行った．エージェント数  $n$  は 30 である．dd-mst-cl は  $k = 100, 1,000, 10,000, 50,000, 100,000$  の場合について，dd-mst-tp は  $k = 10, 100, 1,000, 10,000, 30,000$  の場合について，解の質  $Q(T)$  と実組み合わせ総数とエラー回数の平均値を評価した．実組み合わせ総数は，各エージェントが最終的に保持する組み合わせの総数であり，エラー回数は  $k$  による組み合わせの制限によって，解が出なかった回数

である。なお、解が出ない場合は、あらかじめ生成した順序木を解として、 $Q(T)$  を算出する。結果を表 1 に示す。

`dd-mst-cl` と `dd-mst-tp` のどちらもパラメータ  $k$  の値を減少させると、解の質が低下し、エラーの発生頻度が増え、 $k$  の値を増加させると、実組み合わせ数が増加する。記憶領域の容量やメッセージサイズを小さく抑えたい場合、 $k$  の値は小さい方が望ましいが、解の質は低下する。反対に解に高い質や精度を求める場合、 $k$  の値は大きい方が望ましいというトレードオフの関係がある。

6・2 節と 6・3 節の実験では、エラーがなく、かつ本研究の実験環境における実験時間とメモリ制限を考慮したパラメータ  $k$  の値を設定する。`dd-mst-cl` に関しては 100,000 に、`dd-mst-tp` に関しては 30,000 に設定する。

## 6・2 探索空間の比較

`dd-mst`, `dd-mst-cl` と `dd-mst-tp` の 3 手法に関して、エージェント数  $n = 10, 15$  のときの探索空間の規模の比較実験を行った。なお、6・1 節で述べたように `dd-mst-cl` は  $k = 100,000$ , `dd-mst-tp` は  $k = 30,000$  に設定し、実組み合わせ総数、組み合わせ総数、最大組み合わせ総数、最大削除数を評価した。実組み合わせ総数は 6・1 節で述べたものと同等であり、組み合わせ総数とは制約を何も考慮しない場合の各エージェントの組み合わせ数の総数である。最大組み合わせ数とは各エージェントが保持する組み合わせの最大値であり、最大削除数とは各エージェントが制約や  $k$  によって組み合わせを削減した最大の数である。結果を表 2 に示す。

$n = 10, 15$  のいずれの場合も `dd-mst`, `dd-mst-cl`, `dd-mst-tp` の順に組み合わせ数が多い。純粹に d-MST に関する制約だけを考慮する `dd-mst` は、最大組み合わせ数が最も大きいため、最も大きな記憶領域が必要となるが、最大でも 30,000 の組み合わせのみを保持する `dd-mst-tp` は最も少ない記憶領域で済む。従って、送受するメッセージサイズも小さく抑えることが可能であるが、6・1 節で述べたようなトレードオフの関係によって、解の質、精度は低下する可能性がある。なお、`d-nnt` に関しては、各エージェントの探索空間の規模の増大は線形で、記憶領域は一定であるため、今回は評価を省いた。

## 6・3 厳密解法と近似解法の比較

`dd-prim`, `d-nnt`, `dd-mst`, `dd-mst-cl` と `dd-mst-tp` の 5 手法に関して、 $n = 5$  から  $n = 30$  まで 5 ノードずつについて、 $ctype=H, L$  の 2 種類の問題インスタンスを解く比較実験を行った。 $gap$ (最適解からの  $Q(T)$  の誤差) と  $cycle$ (サイクル数) の平均値を評価した。サイクルとはエージェントがメッセージを受信し、そのメッセージに基づく局所的な処理を実行し、その処理に基づくメッセージを他エージェントに送信する一連の動作の単位である。このサイクル数を分散協調探索のメッセージ送受による構築コストを見積もるための評価指標として用いる。なお、`ddmst-cl` と `ddmst-tp` は上限数  $k$  により解が

表 1: 探索空間の削減手法に関する予備実験

問題(インスタンス数: 10)	解法		
	<code>dd-mst-cl</code>		
$n: 30$ en: 42 ctype: L	$k$	$Q(T)$	実組み合わせ総数
	100,000	1,324	1,615,427
	50,000	1,325	844,136
	10,000	1,331	191,657
	1,000	1,362	22,388
	100	1,401	2,512
<code>dd-mst-tp</code>			
$n: 30$ en: 42 ctype: L	$k$	$Q(T)$	実組み合わせ総数
	30,000	1,311	365,751
	10,000	1,312	191,669
	1,000	1,327	21,845
	100	1,465	2,418
	10	1,493	250

表 2: 探索空間の比較

問題(インスタンス数: 10)	解法		
	<code>dd-mst</code>	<code>dd-mst-cl</code>	<code>dd-mst-tp</code>
$n: 10$ en: 18 ctype: L	実組み合わせ総数/ 組み合わせ総数	90,320/ 7,683,061	71,163/ 7,683,061
	最大組み合わせ数	44,556	33,606
	最大削除数	214,550	156,826
$n: 15$ en: 20 ctype: L	実組み合わせ総数/ 組み合わせ総数	136,032/ 490,991,877	130,522/ 490,991,877
	最大組み合わせ数	41,429	41,073
	最大削除数	144,288	133,706

出ない場合があるが、本実験では必ず解が出る問題を選ぶ。結果を表 3 に示す。なお、表 3 の「-」は現実的な計算時間による解が算出不可能なことを表す。

$n$  が小さいとき、`dd-mst`, `dd-mst-cl`, `dd-mst-tp` の  $gap$  値は他手法に比べて小さく、質の高い d-MST を構築できた。厳密解法である `dd-mst` が最も質の高い d-MST を構築可能であるが、探索空間の複雑度は各エージェントの繋がる順序木の深さと隣接辺数に対して指數関数的に増加するため、このような小規模な問題インスタンスを用いても現実的な時間で解が得られない場合が存在した。`dd-mst-tp` は平均的に小さい  $gap$  値を示したため、ある程度の解の質を保ちつつ、探索空間の規模を小さく抑えるような手法の適用の有効性を示すことができた。その一方で、エージェント数が増えてくると、例えば  $n = 20$ ,  $ctype = L$  のときや  $n = 25$ ,  $ctype = H$  のときのように、`dd-prim` の方が小さい  $gap$  値をとる。すなわち、考慮すべき組み合わせの上限数  $k$  を設定するというような基本的なヒューリスティクスを用いる場合、問題の規模の増大によって解の質が劣化する可能性が高まる。また、どのエージェント数  $n$  のときも、`d-nnt` の  $gap$  は、1 番大きくなつたため、極端な探索空間の削減はあまり有効ではないと考える。ただし、今回は小規模な問題インスタンスのみを解いたため、規模の大きい問題における議論が必要である。 $cycle$  に関しては、`dd-prim` が最も多いため、メッセージ回数が多く通信遅延が大きくなる可能性

表 3: 厳密解法と近似解法の比較

問題 (インスタンス数: 10)			解法									
			dd-prim		d-nnt		dd-mst		dd-mst-cl		dd-mst-tp	
n	en(avg)	ctype	gap(avg)	cycle(avg)	gap(avg)	cycle(avg)	gap(avg)	cycle(avg)	gap(avg)	cycle(avg)	gap(avg)	cycle(avg)
5	7	L	22	27	35	14	0	16	0	16	0	16
	8	H	80	31	410	20	0	16	0	16	0	16
10	18	L	14	105	189	50	0	36	1	36	0	36
	18	H	47	103	1,003	52	0	36	3	36	0	36
15	20	L	11	199	426	67	0	56	6	56	0	56
	21	H	36	195	847	72	0	56	45	56	0	56
20	25	L	9	325	694	85	0	76	5	76	27	76
	27	H	33	328	1,135	91	0	76	114	76	0	76
25	34	L	17	473	974	111	-	-	33	96	1	96
	33	H	94	467	1,341	111	-	-	131	96	117	96
30	42	L	11	626	1,365	140	-	-	38	126	23	126
	41	H	116	644	1,777	137	-	-	272	126	9	126

がある。これは dd-prim が辺の探索と情報の集約を反復的に繰り返すためである。それに対して、順序木に沿って 1 往復のメッセージ伝播のみを行う dd-mst, dd-mst-cl, dd-mst-tp の cycle は少ないため、メッセージ回数は少なく、通信遅延は少く済む。その一方で、各エージェントの計算すべき部分木集合のサイズが指数関数的に増加し、抑制しても比較的のサイズが大きいため、個々のエージェントの計算時間や通信時間が大きくなることは不利な点であると考える。また、d-nnt に関しては、次数制限によって、接続辺がすぐには定まらない場合、cycle が増加すると考えられる。

## 7. おわりに

本研究では、分散協調問題の枠組みで、次数制約付き最小生成木問題を形式化した。それに対して基礎的な分散協調探索手法を提案し、他の近似的な手法についても検討、評価した。

初期の検討としての比較的小規模な例題を用いた評価実験の結果では、厳密解法において、ある程度の組み合わせを切り捨てる手法で、探索空間を削減しつつ、一定の解の質、精度が得られ、手法の有効性が確認された。厳密解法の探索空間をより効果的に削減し、必要なメモリやメッセージサイズを抑制するために、木の表現や枝刈りの条件の改善の適用が挙げられる。このためには、部分木の集合とコストをボトムアップに集計するのみではなく、トップダウンな集計を伴う分枝限定法を併用する手法の適用が考えられる。また、さらに大規模かつ複雑な問題に対しては、より洗練された近似的な手法の併用が必要であるといえる。

最良優先探索手法による枝刈りの導入や、従来研究の分散制約最適化手法への適用における解の質、精度の向上と探索空間の縮小は、今後の課題である。

## 謝辞

本研究の一部は科学研究費補助金（基盤研究 C（一般）, 課題番号 21500073）の助成による。

## ◇ 参考文献 ◇

- [Boldon 96] Boldon, B., Deo, N., and Kumar, N.: Minimum-Weight Degree-Constrained Spanning Tree Problem: Heuristics and Implementation on an SIMD Parallel Machine, *Parallel Computing*, Vol. 22, No. 3, pp. 369–382 (1996)
- [Gallager 83] Gallager, R. G., Humblet, P. A., and Spira, P. M.: A Distributed Algorithm for Minimum-Weight Spanning Trees, *ACM Transactions on Programming Languages and Systems*, Vol. 5, No. 1, pp. 66–77 (1983)
- [Garey 79] Garey, M. R. and Johnson, D. S.: Computers and Intractability: A Guide To the Theory of NP-Completeness, a (1979)
- [Khan 09] Khan, M., Pandurangan, G., and Kumar, V. A.: Distributed Algorithms for Constructing Approximate Minimum Spanning Trees in Wireless Networks, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 20, No. 1, pp. 124–139 (2009)
- [Knowles 00] Knowles, J. D. and Corne, D.: A new evolutionary approach to the degree-constrained minimum spanning tree problem, *IEEE Transactions on Evolutionary Computation*, Vol. 125-134, No. 2 (2000)
- [Kruskal 56] Kruskal, J. B.: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem, in *Proceedings of The American Mathematical Society*, Vol. 7, pp. 48–48 (1956)
- [Kumar 09] Kumar, A., Faltings, B., and Petcu, A.: Distributed constraint optimization with structured resource constraints, in *Autonomous Agents & Multiagent Systems/Agent Theories, Architectures, and Languages*, pp. 923–930 (2009)
- [Modi 05] Modi, P. J., Shen, W., Tambe, M., and Yokoo, M.: Adopt: asynchronous distributed constraint optimization with quality guarantees, *Artif. Intell.*, Vol. 161, No. 1-2, pp. 149–180 (2005)
- [Narula 80] Narula, S. C. and Ho, C. A.: Degree Constrained Minimum Spanning Tree, *Computers & Operations Research*, Vol. 7, No. 4, pp. 239–249 (1980)
- [Papadimitriou 84] Papadimitriou, C. H. and Vazirani, U. V.: On Two Geometric Problems Related to the Traveling Salesman Problem, *Jurnal of Algorithms*, Vol. 5, No. 2, pp. 231–246 (1984)
- [Petcu 05] Petcu, A. and Faltings, B.: A Scalable Method for Multiagent Constraint Optimization, *9th Int. Joint Conf. on Artificial Intelligence*, pp. 266–271 (2005)
- [Prim 57] Prim, R. C.: Shortest connection networks and some generalizations, *Bell System Technical Journal*, Vol. 36, No. 1, pp. 1389–1401 (1957)
- [Zhou 97] Zhou, G. and Gen, M.: A note on genetic algorithms for degree-constrained spanning tree problems, *Networks*, Vol. 30, No. 2, pp. 91–95 (1997)