

## cycle-rooted tree を用いた複数情報の散布について

入野 裕章\* 田中 勇樹† 河合 博之‡ 大澤 新吾§ 柴田 幸夫§

### 概要

計算機相互結合網上での情報散布の問題として、一つの計算機の持つ情報を相互結合網上の全ての計算機に伝えるブロードキャストと、全ての計算機が個別に保持する情報を他の全ての計算機と共有するゴシップが広く研究されている。本稿では、いくつかの計算機が持つ別々の情報を、効率良く相互結合網上の全ての計算機に伝えるマルチソースブロードキャスト問題に対し、cycle-rooted tree を用いた情報散布手法について考察を行う。また、de Bruijn ダイグラフ上でのマルチソースブロードキャストについても考察し、このグラフ族がマルチソースブロードキャスト問題に対して良い性質を持つことを示す。

キーワード 情報散布問題, マルチソースブロードキャスト, de Bruijn ダイグラフ

## Broadcasting multiple messages by cycle-rooted tree

Hiroaki Irino\* Yuuki Tanaka† Hiroyuki Kawai‡ Shingo Osawa§ Yukio Shibata§

### Abstract

In information dissemination problem on interconnection networks, the problems of broadcasting and gossiping have been widely studied. In this paper, we study the other problem, called multi-source broadcasting, defined as follows: there are some (but not all) units each of which has a unique item of information and needs to disseminate to every other unit. We apply the multi-source broadcasting to the cycle-rooted tree and the de Bruijn digraph as a communication network.

**Keywords** information dissemination, multi-source broadcast, de Bruijn digraph

## 1 まえがき

並列分散処理などの分野において、情報の送受信を効率良く行うことはシステム全体の処理効率向上に繋がる。そのため、様々な相互結合網に対して多くの研究がされている。代表的なものに、一つの計算機が持つ情報を相互結合網上の全ての計算機へ散布するブロードキャストと、各々の計算機が持つ情報を全ての計算機が共有するように情報の交換、散布を行うゴシップが挙げられる [4]。

全てではないいくつかの計算機が個々に持つ情報を相互結合網上の全ての計算機へ送信する問題はマルチソースブロードキャストと呼ばれ、[7] で言及されている。[7] ではいくつ

かの分散ディスク環境の上で問題を考えており、あるディスクは任意の他のディスクに情報を直接送信できる、つまり、相互結合網として完全対称ダイグラフを用いている。本研究では、相互結合網として木を拡張した cycle-rooted tree を用い、その上でのマルチソースブロードキャストのアルゴリズムを与える。また、cycle-rooted tree と密接な関係にある de Bruijn ダイグラフについても考察を行う。

## 2 諸定義

この節では本稿で扱う用語の定義を行う。なお、ここに書かれていない用語の定義については [8] に従う。

\* (株) 日立カーエンジニアリング

† 群馬大学アドバンス・テクノロジー高度研究センター  
Email: ytanaka@atec.gunma-u.ac.jp

‡ 函館工業高等専門学校 情報工学科

§ 群馬大学大学院工学研究科 情報工学専攻

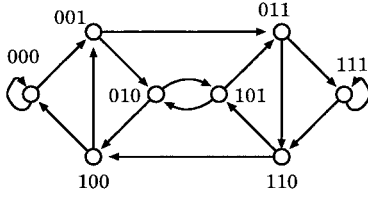


図 1: de Bruijn ダイグラフ  $DB(2,3)$

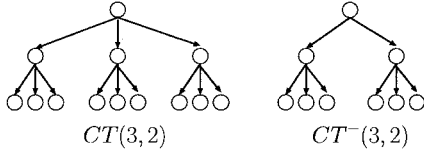


図 2:  $CT(3,2)$  と  $CT^-(3,2)$

## 2.1 de Bruijn ダイグラフ

$d$  進  $n$  桁 de Bruijn ダイグラフ  $DB(d,n)$  は、頂点集合として長さ  $n$  の全ての  $d$  進列を持ち、頂点  $u = u_0u_1 \cdots u_{n-1}$  が  $v = v_0v_1 \cdots v_{n-1}$  へ隣接するための必要十分条件は  $0 \leq i \leq n-2$  なる  $i$  に対し、 $v_i = u_{i+1}$  となることである。図 1 に 2 進 3 桁 de Bruijn ダイグラフ  $DB(2,3)$  を示す。

## 2.2 有向木, 完全有向 $d$ 分木

ダイグラフ  $G$  が有向木であるとは、 $G$  が連結で root と呼ばれる入次数が 0 の頂点の一つを持ち、他の頂点の入次数が全て 1 であるときをいう。出次数が 0 の頂点は葉と呼ばれる。有向木はその定義から、root から他の任意の頂点までのパスが唯一に定まる。root から頂点  $v$  までの距離を  $v$  の深さといい、深さの最大の値を有向木の高さという。有向木の 2 頂点  $u, v$  に対して、有向辺  $(u, v)$  が存在するとき、 $u$  は  $v$  の親であるといい、 $v$  は  $u$  の子であるという。

葉の深さが全て等しく、葉以外の全ての頂点が  $d$  個の子を持つ有向木を完全有向  $d$  分木という。高さ  $h$  の完全有向  $d$  分木を  $CT(d, h)$  と表す。 $h = 0$  では root 自身が葉となり、これは一つの頂点のみからなる自明なダイグラフとなる。 $h \geq 1$  の  $CT(d, h)$  から  $CT(d, h-1)$  の部分木を一つ削除して得られる有向木を  $CT^-(d, h)$  と表す。図 2 に  $CT(3,2)$  と  $CT^-(3,2)$  を示す。

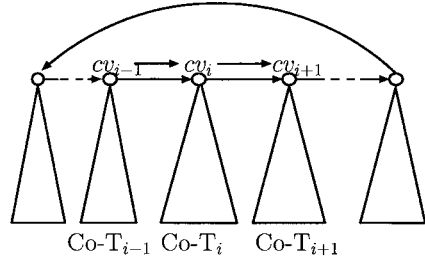


図 4: cycle-rooted tree の構造

## 2.3 cycle-rooted tree

**定義 1** [3] ダイグラフ  $G$  が cycle-rooted tree であるとは、 $G$  が root-cycle と呼ばれるサイクルをちょうど一つ持ち、全ての頂点の入次数が 1 であるときをいう。

root-cycle 上の頂点を cycle-vertex と呼び、 $cv$  と表す。root-cycle の頂点数が  $\alpha$  のとき、各  $cv$  を  $cv_i, i = 1, 2, \dots, \alpha$  と表す（以降、root-cycle の頂点数を  $\alpha$  で表すものとする）。root-cycle を構成する有向辺を消去して得られる有向木を collateral tree と呼び、Co-T と表す。root が  $cv_i$  である Co-T を、 $Co-T_i$  と表す。各  $Co-T_i$  の葉を、単に cycle-rooted tree の葉と呼ぶ。各  $Co-T_i$  の高さを  $h_i$  で表し、その最大値  $\max_{i=1,2,\dots,\alpha} h_i$  を  $h^+$  と表す。 $h^+$  を単に cycle-rooted tree の高さと呼ぶ。図 3,4 に例を示す。

## 2.4 相互結合網のモデルと問題の設定

本研究では、相互結合網全体の全域時計が設定されている同期システムを考える。また、情報の大きさは固定であり、一つの情報の送受信にかかる時間をラウンドという単位で表す。本研究では情報散布が完了するまでのラウンド数が小さいほど効率的な情報散布であるとする。

情報の送受信モデルとして [1] で扱われている simultaneous send/receive model を採用する。このモデルでの制約条件は次のようなものである。

- 各頂点は、1 ラウンドで高々一つの情報を受信できる。
- 各頂点は、1 ラウンドで高々一つの情報を送信できる。
- 各頂点は、1 ラウンドで送信と受信を同時に行うことができる。

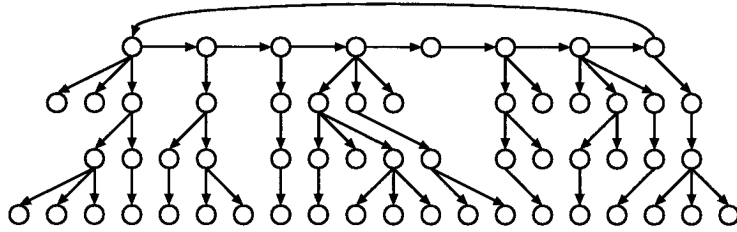


図 3: cycle-rooted tree

- 各頂点は、1 ラウンドで隣接する頂点へ情報を送信でき、同時に隣接される頂点から情報を受信できる。ただし、情報の送信先となる頂点からの受信はできない。

また、各頂点は自身の持つ情報のインデックスを識別することができ、自身が持つ情報が複数ある場合、どの情報を送信するかを選択することができるものとする。

本稿では情報散布問題の一つとして、マルチソースブロードキャストイングを扱う。定義は次のように与えられる。

**定義 2 (マルチソースブロードキャストイング)**  
 $G$  を頂点数  $n$  のダイグラフとし、 $S = \{u_1, u_2, \dots, u_k\} \subset V(G)$ ,  $1 < k < n$  とする。 $S$  の各頂点  $u_i$  はソース頂点、あるいは単にソースと呼び、それぞれ情報  $m_i$  を持つ。ここで  $i \neq j$  に対し  $m_i \neq m_j$  である。 $V(G)$  の全ての頂点が情報  $m_1, m_2, \dots, m_k$  を持つように情報散布を行う。

### 3 既存の結果

完全対称ダイグラフに対する効率的なアルゴリズムは [7] で示されている。これは理論的なラウンド数の下界と高々3しか変わらないことに加え、既存の手法を組み合わせたものであり、簡潔に記述できる。以下にアルゴリズムの概略を示す。ここでは、散布すべきデータの個数を  $i$  で与えており、初期状態では一つの頂点に二つ以上のデータが含まれないものとする。

1. 頂点集合を  $i$  個の、ほぼ同じ大きさに分割する。このとき、各分割集合にはデータを持つ頂点がちょうど一つ含まれるようにする。
2. 各分割集合内でそれぞれブロードキャストイングを行う。これにより各頂点はそれぞれ一つずつデータを持つことになる。

3. 大きさが  $i$  あるいは  $i+1$  になるように頂点集合を分割する。このとき、各データに対して、そのデータを持つ頂点が全ての分割集合に少なくとも一つずつ含まれるようにする。

4. 各分割集合内でゴシップングを行う。これにより、全ての頂点が全てのデータを取得できる。

[7] で示されているアルゴリズムは完全対称ダイグラフの上で与えられており、1. や 3. で行われた分割の後も各分割集合は完全対称ダイグラフとなっているため、ラウンド数の上界が容易に計算できる。しかし、一般のグラフに対しては 2. のブロードキャストイングおよび 4. のゴシップングに必要なラウンド数を見積ることが難しく、また、それぞれの情報散布を効率良く行うためにグラフをどのように分割するかという計算にも非常に大きな手間がかかる。

また、与えられた相互結合網と完全対称ダイグラフとの対応関係を先に定義し、そこからラウンド数の見積りを出す方法にも問題点がある。相互結合網への対応を優先して考えると、あるラウンドで、情報の送受信に用いる完全対称ダイグラフ上の複数の弧が、与えられた相互結合網上の一つの弧に対応してしまう可能性があり、このような場合ラウンド数を増やさなくてはならない。このような状態を回避するための対応づけを構成するには、完全対称ダイグラフ上での送受信を考慮する必要があり、最適化するのは非常に困難であると考えられる。次節では、cycle-rooted tree の上でのマルチソースブロードキャストイングアルゴリズムを提案する。

## 4 cycle-rooted tree を用いたマルチソースブロードキャストリング

本研究でマルチソースブロードキャストリングに用いる cycle-rooted tree は頂点数と弧数が等しく、完全対称ダイグラフと比較して弧の数が極端に少ない。このため、相互結合網が与えられたとき、その上でどのような cycle-rooted tree を構築するか、即ちどの通信リンクを用いて情報散布を行うかを決定することは、複数の弧を一つの通信リンクに対応づける必要のある完全対称ダイグラフと比べて容易であり、また cycle-rooted tree 上の複数の弧を一つの通信リンクに対応づける場合でも、同時に情報を送受信するかどうかの解析が容易であると考えられる。

4.1 節では、cycle-rooted tree 上でマルチソースブロードキャストを行う単純なアルゴリズムを提案し、ラウンド数の見積りを行う。4.2 節では 4.1 節のアルゴリズムを改良したものを示す。なお、ソース頂点は全て root-cycle 上に存在するとする。

### 4.1 サイクル上の送受信を優先したアルゴリズム

ここでは、cycle-rooted tree を基本とするマルチソースブロードキャストリングアルゴリズムを提案する。このアルゴリズムの基本的な流れは [7] と逆である。まず root-cycle の上のみでマルチソースブロードキャストリングを行い、全ての cycle-vertex に全ての情報を持たせる。次に、cycle-vertex それぞれの Co-T に対して複数の情報を順次ブロードキャストする。simultaneous send/receive model を基本とした、大きさ  $\alpha$  の cycle 上のマルチソースブロードキャストリングは  $\alpha-1$  ラウンドで終了することはすぐに分かる。

また、複数の情報を root に持つ有向木でのブロードキャストリングについては次の定理が与えられている。

**定理 1** [9]  $T$  を  $CT(d, h)$  とする。このとき、root から  $m$  個の情報をブロードキャストするのに必要なラウンド数は  $(m + h - 1)d$  である。

**系 1**  $T$  を  $CT^-(d, h)$  とする。このとき、root から  $m$  個の情報をブロードキャストするのに必要なラウンド数は  $(m + h - 1)d - 1$  である。

**証明:**  $CT(d, h)$  の上で  $m$  個の情報をブロードキャストするとき、全ての頂点が情報を送信す

る必要があるとき、かつ情報を送信できる状態のときは常に情報を送信しているとすると、最後の情報を  $(m + h - 1)d$  ラウンド目に受信する頂点はただ一つの葉である。その葉を含む、root の子を root とする部分木を削除することで、全ての頂点が最後の情報を高々  $(m + h - 1)d - 1$  ラウンド目までに受信することが出来る。逆に、root の子のうち、最初の情報を一番最後に受信する頂点を root とする部分木は、最初の  $d - 1$  ラウンドでは何の送受信も行われぬ。よって、この部分木は全ての頂点が全ての情報を受信するには少なくとも  $(m + (h - 1) - 1)d + (d - 1)$  ラウンド必要である。□

これらの結果から、cycle-rooted tree 上のマルチソースブロードキャストリングに関して、ラウンド数の上界を与えることができる。

**定理 2** root-cycle の大きさを  $\alpha$ 、情報の個数を  $k$  としたとき、 $(\alpha - 1) + (k + h - 1)d$  ラウンドで終了する cycle-rooted tree 上のマルチソースブロードキャストリングアルゴリズムが存在する。

### 4.2 サイクル上の送受信と Co-T 上の送信を組み合わせたアルゴリズム

定理 2 の方法では、root-cycle 上でのマルチソースブロードキャストリングを最優先に行うが、いくつかの root-cycle 上の頂点は、他の頂点が全ての情報を持っていないときでも全ての情報を持つ場合がある。また、サイクル上でマルチソースブロードキャストリングを行っている際、送信状態あるいは受信状態になっていない頂点が存在することがある。そのような頂点は、自身が持ついくつかの情報を root-cycle 上にない頂点へ、即ち  $Co-T_i$  へ送信することで、サイクル上での送受信の終了を待つことなく情報散布でき、処理効率の向上が期待できる。

図 5 に、root-cycle 上の各頂点の情報散布の様子を示す。root-cycle 上の頂点  $cv_i$  はあるラウンドで情報を受け取ると、次のラウンドで  $cv_{i+1}$  へ情報を送信し、その次のラウンドから  $Co-T_i$  に含まれる頂点への送信を順次行う。

図 6 では、 $cv_i$  が  $Co-T_i$  へある情報を送信している最中に別の情報が root-cycle によって送信されてきたときの様子を示している。ラウンド 0 で  $cv_i$  が情報  $m_x$  を受け取り、ラウンド 1 からラウンド 4 で  $Co-T_i$  への送信を行っているが、ラウンド 4 で次の情報  $m_{x+1}$  を受け取ると、ラウンド 5 では  $Co-T_i$  への送信を行わず、 $cv_{i+1}$  へ

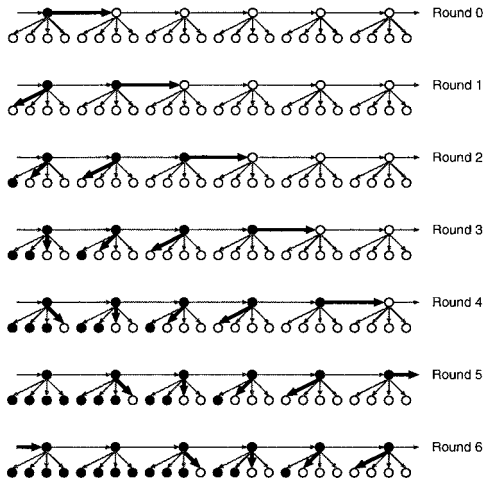


図 5: root-cycle とその直下の頂点に対する送受信の流れ

$m_{x+1}$  を送信する。ラウンド 6 では  $m_x$  を Co- $T_i$  上の、まだ送信していない頂点へ送信し、全ての隣接頂点へ  $m_x$  を送信したら、次のラウンド (図ではラウンド 7) で  $m_{x+1}$  を送信する。

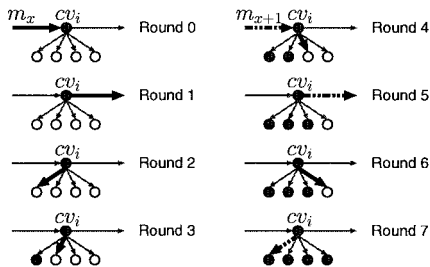


図 6:  $cv_i$  に次の情報が送信されてきた場合の処理の流れ

本節で示した手法は、定理 2 で与えられるものよりもラウンド数が大きくなることはない。

## 5 de Bruijn ダイグラフ上のマルチソースブロードキャスト

de Bruijn ダイグラフは cycle-rooted tree と非常に深い関係があり、また相互結合網に適したグラフの族として広く研究がされている。この節では、de Bruijn ダイグラフに含まれるある特定の全域 cycle-rooted tree を扱い、それをを用い

たマルチソースブロードキャストについて考察する。はじめに、de Bruijn ダイグラフに含まれる木についての補題を与える。

**補題 1** de Bruijn ダイグラフ  $DB(d, n)$  の弧のうち、接続している両端点のラベルの最も左の桁が異なる 2 頂点に接続しているものを全て削除して得られるダイグラフは、 $d$  個の有向木であり、各有向木の root は自己ループに接続している頂点である。

**証明:** 最も左の桁が異なる 2 頂点に接続している全ての弧を削除することから、残ったダイグラフには少なくとも  $d$  個の連結成分を含むことは明らかである。ある  $x \in \mathbb{Z}_d$  に対し、頂点  $xx \cdots x$  から任意の  $t_i \in \mathbb{Z}_d, 1 \leq i \leq n-1$  に対し、 $xt_1 t_2 \cdots t_{n-1}$  への有向パスが  $xx \cdots x, x \cdots xt_1, x \cdots xt_1 t_2, \dots, xt_1 t_2 \cdots t_{n-1}$  の形で存在することから、連結成分の数は  $x$  の選択枝だけ存在し、高々  $d$  個である。よって連結成分はちょうど  $d$  個存在する。

次に各成分が自己ループを除きサイクルを持たないことを示す。題意より、削除される弧はラベルの最も左の桁と、その次の桁が異なる頂点から接続しているものである。そのような頂点は  $DB(d, n)$  中に  $d \times (d-1) \times d^{n-2}$  個存在し、出次数は  $d$  なので削除される弧の本数は  $(d-1) \times d^n$  である。 $DB(d, n)$  の弧数は  $d^{n+1}$  本であり、自己ループは  $d$  本存在するので、削除された辺の本数と自己ループの本数を全体から引くと  $d^n - d$  本となる。今、頂点は削除をしていないことから、連結成分の個数が  $d$  であることから、このダイグラフは自己ループを除きサイクルが存在しないことが言える。

de Bruijn ダイグラフ上の自己ループを含まないある頂点  $x \cdots xt_1 t_2 \cdots t_k$  に対し、その頂点へ隣接している頂点は、命題で示されている弧を削除すると  $xx \cdots xt_1 t_2 \cdots t_{k-1}$  のみである。よって自己ループを含まない頂点は全て入次数が 1 である。また、自己ループに接続している頂点は、自己ループを考慮しないと入次数が 0 である。これは有向木の定義に合致することから、題意を満たすことが言える。  $\square$

**補題 2** de Bruijn ダイグラフ  $DB(d, n)$  には、頂点  $00 \cdots 0, 11 \cdots 1, 22 \cdots 2, \dots, d-1 d-1 \cdots d-1$  を含む長さ  $dn$  のサイクルが存在する。

**証明:** de Bruijn ダイグラフの定義より、ある  $x \in \mathbb{Z}_d$  に対し、 $xx \cdots x$  から  $x \oplus 1 x \oplus 1 \cdots x \oplus 1$  への長さ  $n$  のパスを次のように構成できる：



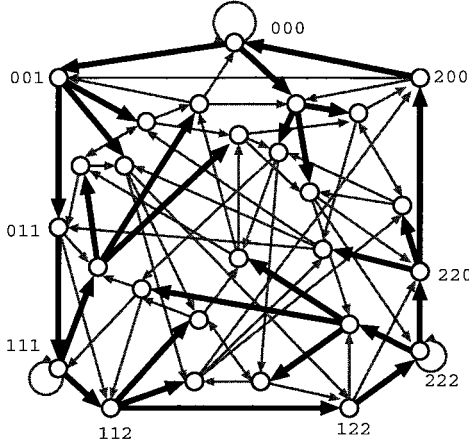


図 7:  $DB(3,3)$  上の SP CRT

$xx \cdots x, x \cdots xx \oplus 1, x \cdots xx \oplus 1 x \oplus 1, \dots, x x \oplus 1 x \oplus 1 \cdots x \oplus 1, x \oplus 1 x \oplus 1 \cdots x \oplus 1$ .

$x \neq y$  なる  $x, y$  に対し,  $xx \cdots x$  から  $x \oplus 1 x \oplus 1 \cdots x \oplus 1$  へのパスと  $yy \cdots y$  から  $y \oplus 1 y \oplus 1 \cdots y \oplus 1$  へのパスは共通する内点 (両端点以外の頂点) を持たないので, このようなパスをつなげることで長さ  $dn$  のサイクルを構成することが可能である.  $\square$

補題 2 で与えられるサイクルを *1-necklace cycle* と呼ぶ.

補題 1 で与えられるいくつかの有向木に,  $x \in \mathbb{Z}_d$  なる全ての  $x$  に対する  $(xx \oplus 1 \cdots x \oplus 1, x \oplus 1 \cdots x \oplus 1)$  という形の弧を追加すると, 補題 2 で与えられる *1-necklace cycle* ができ, これを *root-cycle* とする *cycle-rooted tree* を構成することができる, これを *SP CRT* と呼ぶ.

SP CRT は次の性質を持つ.

系 2 SP CRT 上の頂点

$$\underbrace{xx \cdots x}_{n-k} \underbrace{xx \oplus 1 \cdots x \oplus 1}_k$$

を root とする Co-T に含まれる頂点は, ラベルの左から  $n-k$  以下の桁数だけ  $x$  が続き, その次の桁から  $x \oplus 1$  がちょうど  $k$  桁続くようなラベルを持つ. 更に  $k \neq n$  のとき, Co-T は  $CT^-(d, n - (k + 1))$  となる.

系 3 SP CRT 上の頂点  $xx \cdots x$  を root とする Co-T は,  $CT^-(d, n - 1)$  から  $CT(d, n - 2)$  を更にもう一つ削除した木となる.

図 7 に de Bruijn ダイグラフ  $DB(3,3)$  上の SP CRT を示す. また, 図 8 に, 系 2, 系 3 の内容を図示する.

次に, SP CRT を用いた de Bruijn ダイグラフ上でのマルチソースブロードキャストについての考察する. ここでは散布する情報の個数を高々  $d$  個とし, ソース頂点集合は de Bruijn ダイグラフの任意の頂点部分集合とする. 情報散布を効率良く行うために, 前処理として情報  $m_i$  を  $ii \cdots i$  という頂点へ送信しておく. これにより, SP CRT を用いたマルチソースブロードキャストに必要なラウンド数の見積りが簡潔になるとともに, 均衡のとれたソース配置となる. de Bruijn ダイグラフに関して, 次の補題を与える.

**補題 3**  $x = x_0 x_1 \cdots x_{n-1}, y = y_0 y_1 \cdots y_{n-1}$  を  $x \neq y$  なる  $D(d, n)$  上の頂点とし,  $x$  には情報  $m_i$  が,  $y$  には情報  $m_j$  が保持されているとする. 頂点  $x$  から頂点  $i = ii \cdots i$  への最短パスと頂点  $y$  から頂点  $j = jj \cdots j$  への最短パスには, 共通する頂点は存在しない.

**証明:**  $x$  から  $ii \cdots i$  への最短パスを  $P_i$ ,  $y$  から  $jj \cdots j$  への最短パスを  $P_j$  とする.  $P_i$  に含まれる頂点は,  $x_{n-1} = i$  ならば全ての, そうでないならば  $x$  以外の頂点の最も右の桁  $i$  であり, 同様に  $P_j$  に含まれる頂点も同様に最も右の桁が  $j$  となることから明らかである.  $\square$

**補題 4** 情報  $m_i$  を各々, 頂点  $ii \cdots i$  へ送信する処理には, 高々  $n$  ラウンド必要である.

この節の最後に, de Bruijn ダイグラフ上で SP CRT を用いたマルチソースブロードキャストのラウンド数について上界を見積る. 始めに *1-necklace cycle* 上のマルチソースブロードキャストに必要なラウンド数であるが, 4.1 節で示した一般的なマルチソースブロードキャストとは異なり, ここでは全ての情報が  $xx \cdots x$  の形の *cycle-vertex* に到達した段階までで *cycle* 上の送受信に関するラウンド数とすることができる. 何故ならば, 補題 1 で与えたループ頂点を root とする有向木は  $CT^-(d, n - 1)$  と同型であり, 系 1 より  $d$  個の情報をブロードキャストするのに必要な時間が  $(d + (n - 1) - 1)d - 1$  ラウンドであることが示されていることによる. よって, *1-necklace cycle* 上でのマルチソースブロードキャストに  $d(n - 1)$  ラウンド, ループ頂点を root とする  $CT^-(d, n - 1)$  上への  $d$  個の情報をブロードキャストするのに

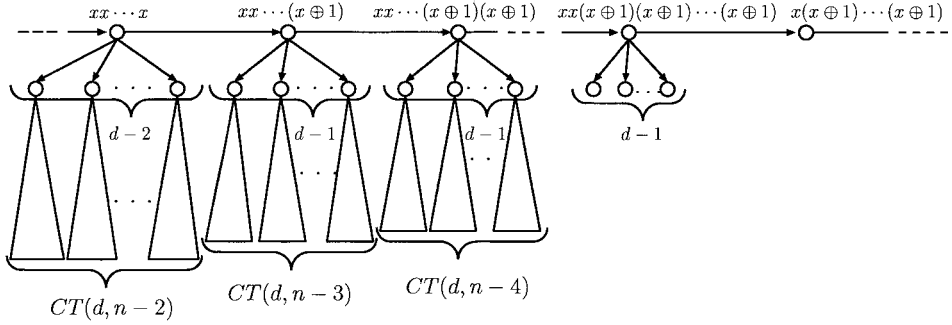


図 8: SPCRT の構造 (一部)

高々  $(d + (n - 1) - 1)d - 1$  ラウンドであることから、補題 4 の前処理を加えた合計の時間が de Bruijn ダイグラフ上での高々  $d$  個の情報に関するマルチソースブロードキャストのラウンド数となる。

**定理 3** de Bruijn ダイグラフ  $DB(d, n)$  上で高々  $d$  個の情報のマルチソースブロードキャストに必要なラウンド数は高々  $2dn + d^2 - 2d - 1$  である。

定理 3 は 4.1 節で与えたように、1-necklace cycle 上のマルチソースブロードキャストにかかるラウンド数と定理 1 による Co-T 上のブロードキャストのラウンド数を独立に計算している。SPCRT では各 Co-T の形が明確であるので、4.2 節では (上界が Co-T の大きさやソース頂点の配置に依存することから) 計算することが困難であったアルゴリズムについても上界の見積りが可能となる。ここでは  $n \geq d$  であるとき、Co-T の頂点への情報の到着間隔が  $d$  以上となることから、4.2 節のように cycle 上での送受信を行っている間に Co-T へ送信するアルゴリズムに必要なラウンド数が定理 3 のラウンド数と大きく異なることを示す。

**定理 4** de Bruijn ダイグラフ  $DB(d, n)$  上で高々  $d$  個の情報のマルチソースブロードキャストに必要なラウンド数は、 $n \geq d$  のとき高々  $2dn - d$  である。

**証明:** root-cycle 上の送受信において、 $n \geq d$  ならば  $cv_i$  での情報の受信間隔は  $d$  以上である。情報  $m_x$  が到達してから  $d$  ラウンドで  $cv_{i+1}$  を含めた全ての隣接頂点への  $m_x$  の送信が完了する。サイクル上に無い頂点も情報を受信する間隔は  $d$  以上であり、図 6 のように隣接頂点への送信中に次の情報を受信することがない。これ

は、各情報が他の情報と干渉することなく送受信可能であることを示している。このことから、ループ頂点へのデータの送信に必要なラウンド数とループ頂点  $xx \cdots x$  が情報  $m_{x-1}$  を受信するまでのラウンド数、 $CT^-(d, n-1)$  の一つの情報のブロードキャストに必要なラウンド数を合計したものがマルチソースブロードキャストに必要なラウンド数となる。よって、

$$n + n(d-1) + d(n-1) = 2dn - d$$

ラウンドとなる。  $\square$

## 6 既存の手法との比較

**定理 5** [5] 次のラウンド数を満たす de Bruijn ダイグラフ上のブロードキャストアルゴリズムが存在する。

$$b(D(d, n)) \leq n \left( \frac{5}{4} \lceil \log_2 d \rceil + 3 \right).$$

図 9, 10, 11 は、定理 5 を  $d$  倍 (散布すべき情報の個数) したものに対する、定理 3 で与えたマルチソースブロードキャストのアルゴリズムのラウンド数の比である ( $1 \leq d \leq 10, 1 \leq n \leq 50$ )。値が小さいところほど、提案手法の効率が良いことを示している。概ね 0.3~0.4 を推移しており、提案手法の効率が良いことが分かる。図 9 で  $d \geq 7$  のとき 1 よりも大きな値となっている、すなわち提案手法の方がラウンド数が大きい結果が出ているが、これは  $n = 1$  のときの値であり  $n \geq 2$  に対しては 1 を下回っている。また、図には掲載していないが、 $n$  に対して  $d$  が極端に大きくなると提案手法のラウンド数が既存の結果によるものより大きくなる傾向がある。

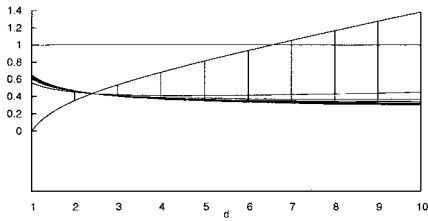


図 9: 次元  $d$  の違いによる速度の比較

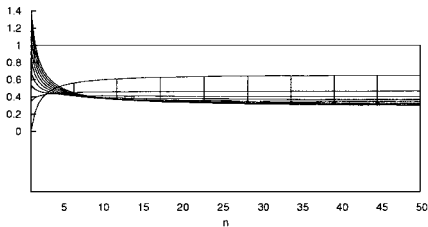


図 10: 桁数  $n$  の違いによる速度の比較

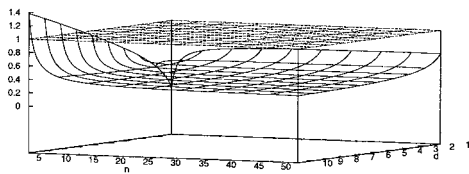


図 11: 図 9,10 の 3 次元図

## 7 まとめ

本稿では相互結合網上に存在する複数の情報を効率良く散布する手法としてマルチソースブロードキャスト問題を考え、cycle-rooted tree による散布アルゴリズムを提案した。また、cycle-rooted tree と関連の深い de Bruijn ダイグラフについても言及し、特定の形の cycle-rooted tree を用いたアルゴリズムを与えた。

## 謝辞

本研究は、日本学術振興会科研費基盤研究 (C) (2) No.16500006 の助成により遂行された。

## 参考文献

- [1] A. Bar-Noy and S. Kipnis, “Broadcasting multiple messages in simultaneous send/receive systems,” *Discrete Appl. Math.* 53, (1994) 95–105.
- [2] K. Diks, A. Lingas, and A. Pelc, “An optimal algorithm for broadcasting multiple messages in trees,” *J. Parallel and Distributed Computing*, 59, 465–474 (1999).
- [3] T. Hasunuma, “Completely independent spamming trees in the underlying graph of a line digraph,” *Discrete Math.*, 234(2001) 149–157.
- [4] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman, “A survey of gossiping and broadcasting in communication networks,” *NETWORKS*, Vol.18, (1988), 319–349.
- [5] M.-C. Heydemann, J. Opatrny, D. Sotteau, “Broadcasting and spanning trees in de Bruijn and Kautz networks,” *Discrete. Appl. Math.*, 37 38(1992),297 317.
- [6] J. Hromkovič, R. Klasing, B. Monien and R. Peine, “Dissemination of information in Interconnection Networks (Broadcasting & Gossiping),” in Ding-Zhu Du and D.Franc Hsu,eds.,Combinatorial Network Theory, pp.125-212,1996.
- [7] S. Khuller, Y.-A. Kim, Y.-C. Wan, “On generalized gossiping and broadcasting,” *J. Algorithms*, 59(2006) 81–106.
- [8] 立花, 奈良, 田澤, グラフ理論への入門, 共立出版, 1991.
- [9] M. Suderman, “Multiple message broadcasting,” Master thesis, Simon Fraser university, 1999. <http://citeseer.ist.psu.edu/468982.html>.