

Kautz ダイグラフ上の cycle-rooted tree を用いた マルチソースブロードキャストイング

津野 崇寛* 柴田 幸夫*

概要

計算機相互結合網上での情報散布における問題の一つとして、幾つかの計算機が持っている別々の情報を全ての計算機に伝えるマルチソースブロードキャストイングの研究が行なわれている。本稿では相互結合網のモデルの一つである Kautz ダイグラフに対し、その内部に埋め込まれている全域 cycle-rooted tree を構成するための方法を示す。また、構成された全域 cycle-rooted tree を用いて効率良くマルチソースブロードキャストを行なう手法について考察を行なう。

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, Kautz ダイグラフ

Multi-source broadcasting on the Kautz digraph using cycle-rooted tree

Takahiro Tsuno* Yukio Shibata*

Abstract

In the information dissemination problems on interconnection networks, the problem of multi-source broadcasting, which deals with the dissemination of informations distributed on some (but not all) units to every other units, has been studied. In this paper, we show a method to construct a spanning cycle-rooted tree in the Kautz digraph which is one of the models of interconnection networks, and we investigate the method for multi-source broadcasting using cycle-rooted trees on the Kautz digraph.

Keywords information dissemination, multi-source broadcast, Kautz digraph

1 はじめに

並列計算機を設計する上でその性能を決定する要因の一つとして相互結合網の選択が挙げられ、その性能を評価するために相互結合網をグラフでモデル化して考えることがなされている。評価の要因としては直径や次数、一様性等が挙げられるが、Kautz ダイグラフは相互結合網のモデルとしてこれら多くの要因に対して良い性質を持っている。

一方、複数の計算機を用いて並列分散処理を行なう際に、情報の送受信を効率良く行なうことが全体の処理効率の向上に繋がる。そのため、情報散布に対する様々な問題が研究されている [3]。情報散布問題の一つとして、全てではない幾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イングと呼ばれ、[8] で導入されている。

本研究では、マルチソースブロードキャストイングに有用である cycle-rooted tree を Kautz ダイグラフの上の部分ダイグラフとして構成するための方法を示す。また、その上でのマルチソースブロードキャストイングのアルゴリズムを与え性能の評価を行なう。

2 諸定義

この節では本稿で扱う用語の定義を行なう。なお、ここに書かれていない用語の定義については [2] に従う。

2.1 Kautz ダイグラフ

d 進 n 桁 Kautz ダイグラフ $K(d, n)$ は頂点集合としてどの隣り合う桁の数字も異なる長さ n の $d+1$ 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が $0 \leq i \leq n-2$ なる i に対し $v_i = u_{i+1}$ となるダイグラフである。 $K(d, n)$ は d 正則であり、そのオーダは $d^n + d^{n-1}$ である。図 1 に Kautz ダイグラフ $K(2, 3)$ を示す。

2.2 有向木

ダイグラフ G が有向木であるとは、 G が弱連結で根と呼ばれる入次数 0 の頂点を 1 つ持ち、他の頂点の入次数が全て 1 であることをいう。また、有向木 G の中で出次数が 0 の頂点は葉と呼ばれる。定義より有向木は根から他の任意の頂点までの有向パスが一意に定まる。根から任意の頂点 v までの有向パスの

* 群馬大学大学院工学研究科 情報工学専攻

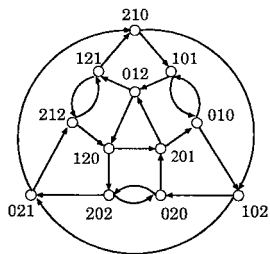


図 1: Kautz ダイグラフ $K(2,3)$

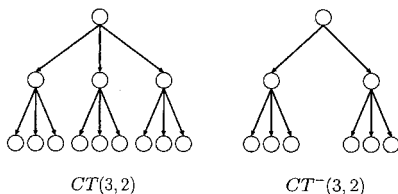


図 2: $CT(3,2)$ と $CT^-(3,2)$

長さを v の深さと呼び、深さの最大値を G の高さと呼ぶ。有向木の 2 頂点 u, v に対し、有向辺 (u, v) が存在するとき u は v の親、 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の出次数が d である有向木を完全有向 d 分木と呼び、高さが h の完全有向 d 分木を $CT(d, h)$ で表す。また、 $h \geq 1$ のとき $CT(d, h)$ から $CT(d, h-1)$ の部分木を 1 つ削除することで得られる有向木を $CT^-(d, h)$ と表す。図 2 に $CT(3,2)$ と $CT^-(3,2)$ を示す。

2.3 cycle-rooted tree

定義 1 [4] ダイグラフ G が cycle-rooted tree であるとは、 G が root-cycle と呼ばれるただ 1 つの有向サイクルを持ち、各頂点の入次数が全て 1 であるときをいう。

root-cycle 上の頂点を cycle-vertex と呼び、更に root-cycle を構成する有向辺を削除することで得られる cycle-vertex を根とする有向木を collateral tree と呼ぶ。

2.4 情報散布問題の設定

本研究では相互結合網全体に大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され、一つの情報を送受信するための単位時間をラウンドとして表す。

情報の送受信モデルとしては [1] で扱われている simultaneous send/receive model を採用する。このモデルにおける制約条件は以下に表される。

- 各頂点は 1 ラウンドに高々一つの情報を受信できる。
- 各頂点は 1 ラウンドに高々一つの情報を送信できる。

- 各頂点は 1 ラウンド毎に送信と受信の両方を同時に行なうことができる。
- 各頂点は 1 ラウンド毎に隣接する頂点に情報を送信でき、隣接される頂点から情報を受信できる。ただし、情報の送信先の頂点からの受信はできない。

本稿で扱うマルチソースブロードキャストの定義は以下のように与えられる。

定義 2 G を頂点数 n のダイグラフとし、 $S = \{s_1, s_2, \dots, s_k\} \subset V(G)$, $1 < k < n$ とする。 S の各頂点をソース頂点、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なう。

ある頂点 v が情報 m_i を受信したラウンドを t_1 とし、 v が m_i の送信を開始したラウンドを t_2 とする。このとき $t_2 - (t_1 + 1)$ で表されるラウンドの差を v における遅延と呼ぶ。

3 Kautz ダイグラフにおける全域 cycle-rooted tree の構成

この章では Kautz ダイグラフ $K(d, n)$ を cycle-rooted tree により同型因子分解し、それらを繋ぎ合わせることで様々な root-cycle を持つ全域 cycle-rooted tree を構成するための方法を示す。

3.1 Kautz ダイグラフの同型因子分解

あるダイグラフ G に対し、その有向辺集合を A_1, A_2, \dots, A_k に分割したとき、それぞれの分割集合から誘導される部分ダイグラフが G の全域部分ダイグラフであり、かつそれらが全て同型である場合 G は同型因子分解されるという。Kautz ダイグラフ $K(d, n)$ に対して以下の定理が示されている。

定理 1 [7] Kautz ダイグラフ $K(d, n)$ は高さ n の cycle-rooted tree により同型因子分解される。

定理 1 を基とする $K(d, n)$ の同型因子分解は、実際には cycle-vertex が 0 と i ($1 \leq i \leq d$) の交互列である 2 つの頂点のみからなるような d 個の cycle-rooted tree が因子として得られる。それぞれの因子を上述の i の値に応じて F_i とし、各 F_i が持つ 2 つの cycle-vertex のうち、末尾が i である頂点を dv_i 、末尾が 0 である頂点を sv_i と定める。更に、 dv_i を根とする collateral tree を $Co-dT_i$ 、 sv_i を根とする collateral tree を $Co-sT_i$ と呼ぶことにする。各 F_i に対し、 $Co-dT_i$ は $CT^-(d, n)$ と同型であり、 $Co-sT_i$ は $CT^-(d, n-1)$ と同型である。従って、この方法で得られる cycle-rooted tree の葉以外の出次数は全て d である。図 3 に $K(d, n)$ を同型因子分解した場合の各因子の構造を示す。

F_i の全ての葉を削除したダイグラフをそれぞれ S_i ($1 \leq i \leq d$) と表し、 S_i の集合を S と表す。ま

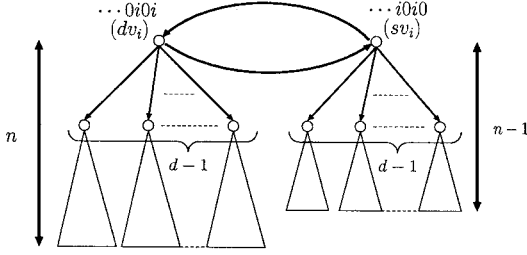


図 3: $K(d, n)$ の F_i の構造

た, 各 S_i に対し, dv_i を根とする collateral tree を $Co-dT_i^-$ で表し, sv_i を根とする collateral tree を $Co-sT_i^-$ と表す. このとき, 次のことが成り立つ.

補題 1 S の任意の 2 つの異なる要素 S_i, S_j ($1 \leq i, j \leq d$) に対し, $V(S_i) \cap V(S_j) = \emptyset$ であり, かつ $\bigcup_{1 \leq i \leq d} V(S_i) = V(K(d, n))$ である.

証明: S_i の頂点集合には末尾が i の, 0 と i の交互列である頂点から $K(d, n)$ 内の長さが $n-1$ の任意の有向パスに含まれている頂点のみが含まれている. 従って S_i に含まれる頂点は全て $i^* \dots$ または $0i^* \dots$ で表される. よって $i \neq j$ のとき $V(S_i) \cap V(S_j) = \emptyset$ が成り立つ.

次に $\bigcup_{1 \leq i \leq d} V(S_i) = V(K(d, n))$ を示す. $K(d, n)$ の頂点の中に $i^* \dots$ で表される頂点は d^{n-1} 個存在し, $0i^* \dots$ で表される頂点は d^{n-2} 個存在する. 各 S_i の頂点数は $Co-dT_i^-$ 及び $Co-sT_i^-$ の形から $d^{n-1} + d^{n-2}$ であるので,

$$\begin{aligned} \sum_i |S_i| &= d(d^{n-1} + d^{n-2}) \\ &= d^n + d^{n-1} \\ &= |V(K(d, n))| \end{aligned}$$

が成り立つ. 従って, $K(d, n)$ 内で $i^* \dots$ または $0i^* \dots$ で表されるような全ての頂点が S_i に含まれている. よって $\bigcup_{1 \leq i \leq d} V(S_i) = V(K(d, n))$ が成り立つ. \square

補題 1 から, ダイグラフ $\bigcup_{1 \leq i \leq d} S_i$ は Kautz ダイグラフの全域な非連結部分ダイグラフであり, 成分として S_1, \dots, S_d を持つ. その各成分は cycle-rooted tree であるから, 任意の cycle-vertex に対してそこから同じ成分内の頂点への有向パスが一意的に存在する. これはすなわち S の各要素の木としての形を崩すことなく, その cycle-vertex を少なくとも 1 つずつ含むような有向サイクルを S の異なる要素間に有向辺を加えることで構成可能であれば, それを root-cycle とする全域 cycle-rooted tree が構成できることを意味している.

3.2 節では S の異なる 2 つの要素間の有向辺の性質について述べ, 3.3 節で実際に S の各要素の形を崩さない全域部分ダイグラフの構成方法を示しそれが全域 cycle-rooted tree であることを示す.

3.2 S の要素間の隣接関係

この節では S の 2 つの異なる要素間の隣接関係に関するいくつかの性質を述べる.

S の各要素は葉以外の全ての頂点の出次数が d である. 従って, 2 つの要素間には一方の葉から出ている有向辺しか存在しない. S の各要素の cycle-vertex を含む有向サイクルを構成するにはその cycle-vertex へ接続するような有向辺が重要な役割を持つ.

S の各要素の cycle-vertex へ接続する有向辺に関して, n が奇数であるとき次のような性質が得られる.

補題 2 n が奇数のとき, S の異なる 2 つの要素 S_i, S_j ($1 \leq i, j \leq d$) 間で以下が成り立つ.

- (i) $Co-sT_i^-$ の葉から sv_j への有向辺がただ 1 つ存在する.
- (ii) $Co-dT_i^-$ の葉から dv_j への有向辺がただ 1 つ存在する.

証明: n が奇であるから, sv_i は $0i \dots i0$ であり, dv_i は $i0 \dots 0i$ である. sv_i の子は $i0 \dots i0k$ ($k \neq 0, i$) で表される. $Co-sT_i^-$ は $CT^-(d, n-2)$ と同型であるので, その各葉はそれぞれ $i0k^* \dots$ で表され, この中に頂点 $i0j0 \dots 0j$ が存在する. この頂点は $0j0 \dots 0j0$, すなわち, sv_j への有向辺を持つ. よって, $1 \leq i \leq d, i \neq j$ である全ての i に対し $Co-sT_i^-$ の葉から sv_j への有向辺が存在する. 補題 1 よりこれら $d-1$ 本の有向辺は全て異なるものであり, $K(d, n)$ は d 正則であるから, これら $d-1$ 本の有向辺と dv_j から sv_j への有向辺 (dv_j, sv_j) を除くと他に sv_j への有向辺は存在しない. 従って (i) が成り立つ.

同様に, dv_i の子は $0i \dots 0ik$ ($k \neq 0, i$) で表される. $Co-dT_i^-$ は $CT^-(d, n-1)$ であるから, その各葉は $ik^* \dots$ で表され, この中に頂点 $ij0j \dots j0$ が存在する. この頂点は $j0 \dots j0j$, すなわち, dv_j への有向辺を持っている. (i) と同様に dv_j への有向辺は (sv_j, dv_j) を除くと $Co-dT_i^-$ の葉からしか接続されない. 従って (ii) が成り立つ. \square

n が偶数の場合についても, 類似した性質が同様に示せる.

補題 3 n が偶数のとき, S の異なる 2 つの要素 S_i, S_j ($1 \leq i, j \leq d$) 間で以下が成り立つ

- (i) $Co-sT_i^-$ の葉から dv_j への有向辺がただ 1 つ存在する.
- (ii) $Co-dT_i^-$ の葉から sv_j への有向辺がただ 1 つ存在する.

補題 2 と 3 から, S の 2 つの異なる要素間で一方の要素の cycle-vertex から もう一方の要素の cycle-vertex への有向パスが必ず存在することが分かる.

3.3 S を用いた全域 cycle-rooted tree の構成

この節では実際に $K(d, n)$ の全域 cycle-rooted tree を S を基にして構成するための方法を示す。

補題 2 と 3 から得られる S の 2 つの異なる要素間での cycle-vertex から cycle-vertex への有向パスを繋ぎ合わせることで 1 つの有向サイクルを構成することを考える。もし上記の条件を満たす有向サイクルが構成できればその有向サイクルを root-cycle とし、その各 cycle-vertex に対し、それが属する S の要素の中で root-cycle に含まれていない子以降の部分各 cycle-vertex の collateral tree とすることで全域 cycle-rooted tree を構成することができる。以下にその構成方法を示す。

ある cycle-vertex から別の cycle-rooted tree の cycle-vertex への有向パスのみで構成される部分ダイグラフを H とおく。この H が単一の有向サイクルとなり得る条件を与えるために、まず S の要素 S_i 上の root-cycle を構成する要素が H 内にどのように含まれるかによって S を以下の 5 つの集合に分割する。

- R_d : dv_i は H に含まれているが、 sv_i は含まれない。
- R_s : sv_i は H に含まれているが、 dv_i は含まれない。
- R_{ds} : 有向辺 (dv_i, sv_i) が H に含まれている。
- R_{sd} : 有向辺 (sv_i, dv_i) が H に含まれている。
- R_b : dv_i と sv_i が H に含まれるが、その間の有向辺は含まれない。

R_d, R_s に属する S の要素に対しては 2 つの cycle-vertex の片方が H に含まれ、もう片方が collateral tree の一部になっている。これに対して、残りの分割集合に属する要素は両方とも H に属しており、その違いは間の有向辺の含まれ方である。特に、 R_b に属する要素は構成される部分ダイグラフ内で 2 つの cycle-vertex が接続されていないことに注意する。

上記のように S を 5 つの集合に分割したとき、 H が有向サイクルとなる条件を示す。ここで、有向サイクルを構成する際に H を順序列 s_1, s_2, \dots, s_x を用いて表現する場合がある。各 s_i ($1 \leq i \leq x$) は S の要素であり、 s_i から s_{i+1} への有向パスはお互いが属している分割集合の条件を満たすものとする。このとき、 R_b に属する要素は構成される部分ダイグラフ内で 2 つの cycle-vertex が接続されていないため、順序列内に 2 回出てくることに注意する必要がある。

H が有向サイクルとなり得る条件を n の偶奇により 2 つの定理として示す。

定理 2 n を奇数とする。 S を $R_d, R_s, R_{ds}, R_{sd}, R_b$ の 5 つの集合に分割したとき、 H が有向サイクルとなり得るための必要十分条件は次の条件のいずれかを満たすことである。

- (i) $|R_{ds}| = |R_{sd}| \neq 0$.
- (ii) $|R_d| = d$.
- (iii) $|R_s| = d$.

証明: H が有向サイクルであるときに (i), (ii), (iii) の条件のいずれも満たしていないと仮定する。このとき、考えられるのは以下の 2 通りである。

- $|R_{ds}| > 0, |R_{sd}| > 0$ かつ $|R_{ds}| \neq |R_{sd}|$
- $|R_{ds}| = |R_{sd}| = 0$ かつ、 $|R_d| > 0, |R_s| > 0$
- $|R_{ds}| = |R_{sd}| = 0$ かつ、 $|R_b| > 0$

どちらの場合にも矛盾が生じることを示す。

1 つ目の場合、 $|R_{ds}| > |R_{sd}|$ と仮定する。 H を順序列で表すと、 n は奇数であるから補題 2 より H 内で 2 つの R_{ds} の要素の間には必ず R_{sd} の要素が含まれていなければならない。しかしながら、 $|R_{ds}| > |R_{sd}|$ であるから、ある 2 つの R_{ds} の要素間には R_{sd} が含まれておらず、矛盾する。 $|R_{sd}| > |R_{ds}|$ の場合も、順序列における 2 つの R_{sd} の要素間には R_{ds} が含まれていなければならない。同様に矛盾が起こる。

2 つ目及び 3 つ目の場合、 $S_i \in R_s$ または $S_i \in R_b$ であるような S_i に対し、 sv_i が H に含まれるはずである。一方、 $S_j \in R_d$ または $S_j \in R_b$ であるような S_j は、 dv_j が H に含まれるはずである。しかしながら、 $|R_{ds}| = |R_{sd}| = 0$ であるので補題 2 よりこの 2 つの頂点を同時に含む有向サイクル H は存在しない。

次に、 S を 5 つの集合に分割したとき、いずれかの条件を満たしていれば有向サイクルとなるように H を構成できることを示す。それぞれの場合について考える。

(i) を満たす場合、 $|R_{ds}| = |R_{sd}|$ である。ここで、 $R'_d = R_d \cup R_b$, $R'_s = R_s \cup R_b$ として定義する。 $R'_d, R'_s, R_{ds}, R_{sd}$ の要素をそれぞれ $r'_d, r'_s, r_{ds}, r_{sd}$ とおき、次のような順序列を考える。

$$r_{ds} (*r'_s) r_{sd} (*r'_d) r_{ds} \cdots r_{ds} (*r'_s) r_{sd} (*r'_d)$$

ただし、 $(*r'_s), (*r'_d)$ とはそれぞれ R'_s, R'_d の要素を任意の数だけ並べたものとする。このとき、隣り合う 2 つの要素は、補題 2 より左の要素の葉から右の要素の根への有向辺が存在する。さらに、右端の R'_d あるいは R_{ds} の要素の葉から左端の R_{ds} の根への有向辺も存在するので、この列から有向サイクルとなるような H が構成可能である。

(ii) を満たす場合、 S の全て要素は R_d に属している。補題 2 より、任意の異なる 2 つの R_d の要素 S_m, S_n に対し、 dv_m から dv_n への有向パスを持つので、全ての $S_i \in S$ に対し dv_i を含むような有向サイクル H が構成できる。

(iii) を満たす場合、 S の全て要素は R_s に属している。補題 2 より任意の異なる 2 つの R_s の要素 S_p, S_q に対し sv_p から sv_q への有向パスを持つので、全ての $S_j \in S$ に対し sv_j を含むような有向サイクル H が構成できる。 \square

定理 3 n を偶数とする。 S を $R_d, R_s, R_{ds}, R_{sd}, R_b$ の 5 つの集合に分割したとき、 H が有向サイクルとなり得るための必要十分条件は次の条件のいずれかを満たすことである。

- (i) $|R_d| = |R_s| \neq 0$.

- (ii) $|R_{ds}| = d$.
- (iii) $|R_{sd}| = d$.

$$|R_d| = |R_s| = 0 \text{ かつ,}$$

- (iv) $|R_b| > 2$.
- (v) $|R_b| = 2$ かつ, $|R_{ds}| \geq 2$ または $|R_{sd}| \geq 2$.
- (vi) $|R_b| = 1$ かつ, $|R_{ds}| \geq 1$, $|R_{sd}| \geq 1$.

証明: H が有向サイクルであるときに上記の条件を全て満たしていないと仮定する。このとき、次の4つに場合分けすることができる。

- $|R_d| \neq |R_s|$, $|R_d| \neq d$, $|R_s| \neq d$.

$$|R_d| = |R_s| = 0 \text{ かつ,}$$

- $|R_b| = 0$ かつ, $|R_{ds}| > 0$, $|R_{sd}| > 0$.
- $|R_b| = 2$ かつ, $|R_{ds}| < 2$, $|R_{sd}| < 2$.
- $|R_b| = 1$ かつ, $|R_{ds}| = 0$ または $|R_{sd}| = 0$.

いずれの場合に対しても矛盾が生じることを示す。始めに, $R'_d = R_d \cup R_b$, $R'_s = R_s \cup R_b$ として定義する。

1つ目の場合, $|R_d| > |R_s|$ と仮定すると $|R'_d| > |R'_s|$ である。 n は偶数であるから, 補題3より H 内で2つの R'_d の要素の間には必ず R'_s の要素が含まれていなければならない。しかしながら, $|R'_d| > |R'_s|$ であるから, H の順序列を考えると, ある2つの R'_d の要素間には R'_s が含まれておらず, 矛盾する。 $|R_s| > |R_d|$ の場合も, $|R'_s| > |R'_d|$ であるから, 2つの R'_s の要素間には R'_d が含まれていなければならず, 同様に矛盾が起こる。

$|R_d| = |R_s| = |R_b| = 0$ かつ $|R_{ds}| > 0$, $|R_{sd}| > 0$ の場合, 補題3より, H 内のある R_{ds} の要素とある R_{sd} の要素の間には必ず R'_d の要素が少なくとも1つ含まれていなければならず, $|R_d| = |R_b| = 0$ であることに矛盾する。同様に, ある R_{sd} の要素とある R_{ds} の要素の間には必ず R'_s の要素が少なくとも1つ含まれていなければならず, $|R_s| = 0$ に矛盾する。

$|R_d| = |R_s| = 0$, $|R_b| = 2$ かつ, $|R_{ds}| < 2$, $|R_{sd}| < 2$ の場合, 明らかに $2 \leq d \leq 4$ である。 R_b に属する2つの S の要素を S_i, S_j とおく。このとき, $|R_{ds}| < 2$ かつ $|R_{sd}| < 2$ であるから, H が有向サイクルとなるには必ず S_i の一方の collateral tree の葉から S_i のもう一方の collateral tree の根への有向辺, あるいは S_j の一方の collateral tree の葉から S_j のもう一方の collateral tree の根への有向辺が存在しなければならず, H の条件に矛盾する。

$|R_d| = |R_s| = 0$, $|R_b| = 1$ かつ, $|R_{ds}| = 0$ または $|R_{sd}| = 0$ の場合, まず $|R_{ds}| = 0$ のとき, R_b に属する S の要素を S_i とする。このとき H を有向サイクルにするには, H 内に $Co-sT_i^-$ の葉から dv_i への有向辺が存在しなければならず H の条件に矛盾する。同様に $|R_{sd}| = 0$ のとき, H を有向サイクルにするには H 内に $Co-sT_i^-$ の葉から dv_i への有向辺が存在しなければならず, H の条件に矛盾する。従って, どちらの場合も矛盾が生じる。

次に, S を5つの集合に分割したときいずれかの条件を満たしていれば有向サイクルとなる H を構成できることを示す。それぞれの場合に分けて考える。

(i) を満たす場合, $R'_d, R'_s, R_{ds}, R_{sd}$ の要素をそれぞれ $r'_d, r'_s, r_{ds}, r_{sd}$ とおき, 次のような順序列を考える。

$$r'_d (*r_{sd}) r'_s (*r_{ds}) r'_d \cdots r'_d (*r_{sd}) r'_s (*r_{ds})$$

ただし, $(*r'_s)$, $(*r'_d)$ とはそれぞれ R'_s , R'_d の要素を任意の数だけ並べたものとする。このとき, 隣り合う2つの要素は, 隣り合う r'_d と r'_s が同じ要素である場合を除いて, 補題3より左の要素の葉から右の要素の根への有向辺が存在し, 右端の R'_s あるいは R_{ds} の要素の葉から左端の R_d の根への有向辺も存在する。ここで, $|R_d| = |R_s| \neq 0$ であるからどの隣り合う r'_d と r'_s も異なる要素であるような並び方が存在するので, この列から有向サイクルとなるような H が構成可能である。

(ii) を満たす場合, S の全て要素は R_{ds} に属している。補題3より, 任意の異なる2つの R_{ds} の要素 S_m, S_n に対し, dv_m から dv_n への有向パスを持つので, 全ての $S_i \in S$ に対し, dv_i を含むような有向サイクル H が構成できる。

(iii) を満たす場合, S の全て要素は R_{sd} に属している。補題3より任意の異なる2つの R_{sd} の要素 S_p, S_q に対し, sv_p から sv_q への有向パスを持つので, 全ての $S_j \in S$ に対し sv_j を含むような有向サイクル H が構成できる。

(iv) を満たす場合, (i) の場合と同様に順序列を考える。 $|R_b| > 2$ であるから, どの隣り合う r'_d と r'_s も異なる要素であるような並び方が存在するので, この列から有向サイクルとなるような H が構成可能である。

(v) を満たす場合も同様に R_b の要素を S_i, S_j とおくと, $|R_{ds}| \geq 2$ または $|R_{sd}| \geq 2$ であるから順序列は

$$S_i (*r_{sd}) S_j r_{ds} (*r_{ds}) S_j (*r_{sd}) S_i r_{ds} (*r_{ds}), \\ S_i r_{sd} (*r_{sd}) S_i (*r_{ds}) S_j r_{sd} (*r_{sd}) S_j (*r_{ds})$$

のどちらかで表すことができる。従って, 補題3よりこの列から有向サイクルとなるような H が構成可能である。

(vi) を満たす場合, R_b の要素を S_i とおくと, $|R_{ds}| \geq 1$ かつ $|R_{sd}| \geq 1$ であるから順序列は

$$S_i r_{sd} (*r_{sd}) S_i r_{ds} (*r_{ds})$$

と表すことができる。従って, 補題3よりこの列から有向サイクルとなるような H が構成可能である。□

定理2, 3より, Kautz ダイグラフ $K(d, n)$ における規則性を持った形の全域 cycle-rooted tree が構成できることが示された。

4 Kautz ダイグラフ上でのマルチソースブロードキャスト

この章では Kautz ダイグラフにおける 全域木及び 全域 cycle-rooted tree を用いたマルチソースブロー

ドキャストイングについて述べる。Kautz ダイグラフ $K(d, n)$ 上でのマルチソースブロードキャストイングを行なう方法として、全域木 F_i と、前章で構成した cycle-rooted tree を用いて考え、両者及び既存の方法の性能について比較する。また、今回 $K(d, n)$ 上に配置されるソースの数は d 個までとする。

始めに、いくつかの有向木に対するブロードキャストイングについて示す。

定理 4 ダイグラフ D の任意の頂点 v に対し、ある情報 m_i を受信したラウンドと次の情報 m_{i+1} を受信したラウンドの差を c とし、 $\deg^+ v = d$ とする。このとき、 v で遅延が発生しないための必要条件は $c \geq d$ を満たすことである。

定理 5 [6] $CT^-(d, h)$ のブロードキャストイング時間は $dh - 1$ ラウンドである。

系 1 $K(d, n)$ の各 S_i に対して、 dv_i をソース頂点としたときのブロードキャストイング時間は $d(n - 1) - 1$ である。

証明: dv_i から情報を送る順番として、まず $Co-dT_i^-$ の全ての子に情報を送信してから sv_i に情報を送るものとする。このとき、 $Co-dT_i^-$ の全ての頂点に送信するのに必要なブロードキャストイング時間は $Co-dT_i^-$ が $CT^-(d, n - 1)$ であることと定理 1 から $d(n - 1) - 1$ である。一方、 $Co-sT_i^-$ の全ての頂点に送信するのに必要なブロードキャストイング時間は sv_i に情報が送信されるのが d ラウンド目であること、 $Co-sT_i^-$ が $CT^-(d, n - 2)$ であることから $d + d(n - 2) - 1 = d(n - 1) - 1$ である。従って、 S_i のブロードキャストイング時間は $d(n - 1) - 1$ である。□

マルチソースブロードキャストイングを行なうための手順として、最初にそれぞれの情報を特定の頂点に送信してから残りの頂点へ全ての情報を送信することを考える。

1つ目の方法として、全域木 F_i を用いた $K(d, n)$ 上のマルチソースブロードキャストイングを考える。 $K(d, n)$ の直径は n であるから、任意の2頂点間の距離は高々 n である。また、各頂点は1ラウンドに1つの情報を受信し、別の1つの情報を送信することのみが行なえるため、各情報が目的の頂点に送信されるまでに発生する遅延は高々 $d - 1$ ラウンドの遅延である。従って、 d 個の情報を全て単一の頂点に送信するために必要なラウンド数は高々 $d + n - 1$ ラウンドであることがわかる。

これを基に、 dv_i から他の全ての頂点に全ての情報を送信するために必要なラウンド数を考える。

定理 6 Kautz ダイグラフ $K(d, n)$ に埋め込まれている全域木 F_i に対して、 dv_i から d 個の情報を全ての頂点に送信するために必要なラウンド数は高々 $d^2 + dn - d - 1$ である。

証明: dv_i が情報を送信する順番として、まず1つの情報 m_i を全ての子に送信してから次の情報を送信し始めるものとする。同様に、 dv_i 以外の頂点に対し

てもある情報を送信中に他の情報を受信しても、今送信している情報を全ての子に送信してから次の情報の送信を行なうものとする。また、 dv_i の情報の送信順序としてまず sv_i を除く全ての子に送信してから、最後に sv_i に情報を送信するものとし、各頂点がそれぞれの情報を子に送る順番は全て同じであるとする。

まず、 sv_i を除いた dv_i の子を根とする部分木に全ての情報を送信するのに必要なラウンド数を考える。それぞれの部分木に対して、根に1つ目の情報が送信されるのは i ($1 \leq i \leq d - 1$) ラウンド目であり、次からの情報は d ラウンド毎に送られる。また、各部分木は葉以外の全ての頂点の出次数が d であることから完全 d 分木であり、各情報を送信する際に遅延は発生しない。従って、各部分木が全ての情報を送信するのに必要なラウンド数は、最初の情報が送られるまでにかかるラウンド数 i と最後の情報が根に送られるまでにかかるラウンド数 $(d - 1)d$ 及び最後の情報を部分木の全てに送信するために必要なラウンド数 $d(n - 1)$ の和 $d^2 + dn - 2d + i$ である。よって、最もラウンド数が遅れる部分木でも $d^2 + dn - d - 1$ ラウンドで全ての情報を送信することができる。

次に $Co-sT_i^-$ に全ての情報を送信するのに必要なラウンド数を考える。 sv_i に最初の情報が送信されるのは d ラウンド目であり、次からの情報は d ラウンド毎に送られる。根と葉以外の全ての頂点の出次数は d である。更に根の出次数は $d - 1$ であるから木の上ではどの頂点においても遅延が発生しない。また、 $Co-sT_i^-$ は $CT^-(d, n - 1)$ と同型であるから、全ての情報を全ての頂点に送信するために必要なラウンド数は $d + d(d - 1) + d(n - 1) - 1 = d^2 + dn - d - 1$ となる。

以上より、 dv_i から d 個の情報を全ての頂点に送信するために必要なラウンド数は高々 $d^2 + dn - d - 1$ である。□

定理 6 より次も明らかとなる。

系 2 $d \geq 2, n \geq 2$ に対し Kautz ダイグラフ $K(d, n)$ のソースが高々 d 個のマルチソースブロードキャストイングに必要なラウンド数は高々 $d^2 + dn + n - 2$ ラウンドである。

次に、cycle-rooted tree を用いてマルチソースブロードキャストイングを行なう方法を提案し、その性能を評価する。

今回 $K(d, n)$ 上のマルチソースブロードキャストイングを行なう全域 cycle-rooted tree のモデルとして、前節で構成したものをを用いる。 n が奇数ならば全ての $S_i \in S$ が R_d に属している全域 cycle-rooted tree を用い、 n が偶数ならば全ての $S_i \in S$ が R_d に属している全域 cycle-rooted tree を用いることにする。

それぞれの場合における各 S_i の cycle-vertex と collateral tree を図 4 に示す。

全域 cycle-rooted tree を用いてマルチソースブロードキャストイングを効率よく行なうために、前処理として各ソースの情報を root-cycle 上に送信し、情報をバランスよく再配置する必要がある。この前処理に関して次のことがいえる。

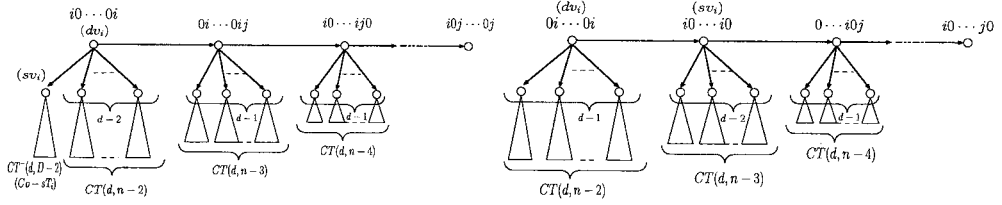


図 4: (左) n が奇数の場合における $S_i \in R_d$ の構造 (右) n が偶数の場合における $S_i \in R_{d_s}$ の構造 (各一部)

補題 4 d 個の情報を各 dv_i に 1 つずつ送信するために必要なラウンド数は高々 $n+1$ ラウンドである。

証明: d 個の情報をそれぞれ m_1, m_2, \dots, m_d とおき, それぞれの情報を持っているソース頂点を s_1, s_2, \dots, s_d とおく. また s_i ($1 \leq i \leq d$) から dv_i への次のような有向パスを \vec{P}_i とおく.

$s_i = x_1 x_2 \dots x_n$ に対し, 以下のような 1, 2 の手順から作られる有向パスに従って各情報を送信することを考える.

1. $x_n = i$ ならば s_i から dv_i への最短パス.
2. $x_n \neq i$ ならば 有向辺 (s_i, s'_i) ($s'_i = x_2 \dots x_n, i$) を通った後, s'_i から dv_i への最短パス.

s_i を除く \vec{P}_i の頂点は全て $cdotsi(*0i)$ または $\dots i0(*i0)$ で表せる. 従って, 任意の 2 つの有向パス \vec{P}_i, \vec{P}_j における頂点の共通部分は s_i あるいは s_j のみが考えられる. しかしながら s_i が m_i を送信するのが 1 ラウンド目であるのに対して, s_i が m_j を受信するのは 2 ラウンド目以降である. 従って各情報を送信する際に遅延は起こらず, \vec{P}_i の長さは n が奇数の場合は高々 n , n が偶数の場合は高々 $n+1$ であることから命題が成り立つ. \square

補題 4 を用いて前処理を行なった後, 実際にマルチソースブロードキャストを行うために必要なラウンド数に対する定理として, 次が与えられる.

定理 7 各 dv_i が高々 1 つずつ情報を持っているとする. このときマルチソースブロードキャストに必要なラウンド数は $d < n$ の場合高々 $2dn - d - n - 1$ であり, $d \geq n$ の場合高々 $d^2 + dn - 2d$ である.

証明: 葉以外の全ての頂点が情報を送信する順序の規則として, 情報 m_i, m_j をこの順番で受信したとき全ての子に m_i を送信してから m_j を送信し始めること, 自身が cycle-vertex であるならばまず隣接する cycle-vertex に情報を送信すると定める. また, dv_i が子に情報を送信する順序として sv_i が cycle-vertex でない限り必ず最後にするようにする. この送信規則において, 各情報を dv_i が送信してから次の dv_j が受信するまでにかかるラウンド数は n の偶奇によらず n となる.

ここで, S の各要素は全て同じ部分集合に属しているため, 全域 cycle-rooted tree 内で同じ構造を持っている. すなわち, ある S の全ての頂点に全ての情報を送信するために必要なラウンド数が得られれば,

それが $K(d, n)$ のマルチソースブロードキャストに要するラウンド数となる. これを, $d < n$ と $d \geq n$ の 2 つの場合に分けて求める.

dv_i が全ての子に 1 つの情報を送信するために d ラウンドを要する. $d < n$ の場合, dv_i がある情報を受信するまでにその前に受信した情報を全ての子に送信し終えていることは明らかである. よって, dv_i における遅延は発生せず, 残りの頂点も遅延が発生しないことから S_i 内での遅延は発生しない.

従って S_i の全ての頂点に全ての情報を送信するのに必要なラウンド数は d 個目の情報を dv_i が受信するまでにかかるラウンド数 $(d-1)n$ と d 個目の情報を S_i の全ての頂点に送信するのにかかるラウンド数 $d(n-1) - 1$ の和である $2dn - d - n - 1$ ラウンドとなる.

$d \geq n$ の場合, dv_i がある情報を全ての子に送信し終える前に次の情報を受信してしまう. これは言い換えれば dv_i がある情報を全ての子に送信し終えたときに必ず次の情報を既に受信しているということの意味する.

従って S_i の全ての頂点に全ての情報を送信するのに必要なラウンド数は $d-1$ 個の情報を全ての子に送信するためにかかるラウンド数と, d 個目の情報が S_i の全ての頂点にかかるラウンド数の和となる. ここで, n が奇数ならば dv_i の子の中で sv_i に情報が送信されるのは最後となるため, d 個目の情報を全ての頂点に送信するのに必要なラウンド数は系 1 より $d(n-1) - 1$ ラウンドである. 従って, S_i の全ての頂点に全ての情報を送信するために必要なラウンド数は $d^2 + dn - 2d - 1$ となる.

一方, n が偶数である場合 sv_i が cycle-vertex であることから送信規則により dv_i の子の中で最初に sv_i に情報が送信される. これにより, d 個目の情報を S_i の全ての頂点に送る際に d 番目に $CT(d, n-2)$ の根である dv_i の子に情報を送信することとなり, これに必要なラウンド数は $dn - d$ となる.

以上より, S_i の全ての頂点に全ての情報を送信するために必要なラウンド数は高々 $d(d-1) + dn - d = d^2 + dn - 2d$ となる. \square

補題 4, 定理 7 より次が従う.

系 3 $d \geq 2, n \geq 2$ に対し Kautz グラフ $K(d, n)$ のソースが高々 d 個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 $d < n$ のとき高々 $2dn - d$ ラウンドであり, $d \geq n$ のとき高々 $d^2 + dn - 2d + n + 1$ ラウンドである.

5 比較結果

図5に系2と系3で得られた全域木と全域 cycle-rooted tree を用いたマルチソースブロードキャストにかかるラウンド数のグラフを示す。ただし、図中の x と y はそれぞれ d と n を表し、SPCRT(x, y) は全域 cycle-rooted tree を用いた結果を、ST(x, y) は全域木を用いた結果を表している。実際には $d \leq n - 2$ の場合は全域木を用いたアルゴリズムの方が効率が良いが、 d の値が n に近づく程ラウンド数の差が縮まり、 $d > n - 2$ の場合においては全域 cycle-rooted tree を用いたアルゴリズムの方が効率が良くなる。

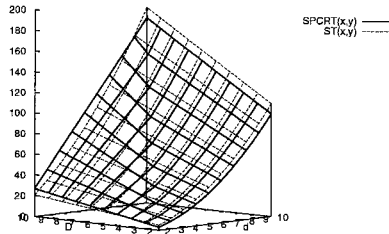


図5: 全域木と全域 cycle-rooted tree のラウンド数の比較

また、Kautz ダイグラフ上でのブロードキャストに関する既存の手法として、次が示されている。

定理 8 [5] 次のラウンド数を満たす Kautz 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。

$$b(K(2, n)) \leq 2n,$$

$$b(K(3, n)) \leq 3n,$$

$$b(K(d, n)) \leq$$

$$\begin{cases} \frac{(d+3)(n+1)}{2}, & \text{if } 4 \leq d \leq 12, d \neq 9 \\ \min\{2n \lceil \log_2 d \rceil, 3n \lceil \log_3 d \rceil\}, & \text{otherwise} \end{cases}$$

ここで、 $b(K(d, n))$ とは $K(d, n)$ の任意の頂点からのブロードキャストにかかるラウンド数の中で最大のものである。

図6に定理8の結果を d 倍したものと、系2及び系3で得られた結果の比を示す。既存の手法に対し、ほぼ0.4以下の比を保っているため、提案した手法の方が効率が良いことが分かる。ただし、 n の値に対し、 d の値が極端に大きくなると、提案した手法の効率が既存の手法よりも下回ってしまうことがある。

6 まとめ

本稿では Kautz ダイグラフ上でのマルチソースブロードキャストの効率の良い実現に適した全域 cycle-rooted tree の構成方法を提案した。また、構成

SPCRT(x, y) ———
ST(x, y) - - - - -
existing(x, y) ·····

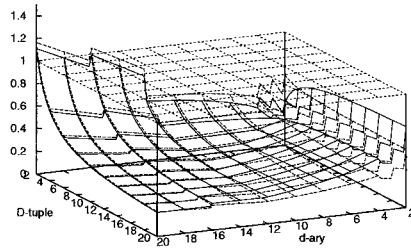


図6: 既存の方法との比較

可能な全域 cycle-rooted treeの中から特定の形を選択し、実際に散布アルゴリズムを与えた。

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] G. Chartrand and L. Lesniak, "Graphs and Digraphs. fourth edition," Chapman & Hall/CRC, 2004.
- [3] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman, "A survey of gossiping and broadcasting in communication networks," *Discrete. Appl. Math.*, 37-38(1992) 297-317.
- [4] T. Hasunuma, "Completely independent spanning trees in the underlying graph of a line digraph," *Discrete Math*, 234(2001) 149-157.
- [5] M. C. Heydemann, J. Opartny, and D. Sotteau, "Broadcasting and spanning trees in de Bruijn and Kautz networks," *Discrete Appl. Math.*, 37-38(1992) 297-317.
- [6] 入野, 田中, 河合, 大澤, 柴田, "cycle-rooted treeを用いた複数情報の散布について," 情報処理学会アルゴリズム研究会研究報告, 2007-AL-112(6), 2007年5月.
- [7] H. Kawai, N. Fujikake and Y. Shibata, "Factorization of de Bruijn digraphs by cycle-rooted trees," *Information Processing Letters* 77 pp.269-275, 2001.
- [8] S. Khuller, Y.-A. Kim, Y.-C. Wan, "On generalized gossiping and broadcasting," *J. Algorithms*, 59(2006) 81-106.