論　文

広域並列分散システムのブロードキャストスケジューリングについて

田崎　　太[†*]　　田村　　裕[†]　　仙石　正和[††]　　篠田　庄司[†††]

Broadcast Scheduling for Wide Area Parallel Distributed Systems

Futoshi TASAKI[†*], Hiroshi TAMURA[†], Masakazu SENGOKU[††], and Shoji SHINODA[†††]

あらまし　複数のコンピュータをLANやWANなどで結合した並列分散システムが広く利用されるようになってきた．並列分散システムの重要な機能の一つに，システム全体へデータを配信するブロードキャストがある．ブロードキャスト処理は並列分散システムの性能に大きく関与し，この処理を短時間で完了する配信手順が望まれる．そこで，最適な配信手順を見つけ出すことを目的とした最小ブロードキャスト時間問題が数多く検討されてきた．この問題に対する従来の研究においては，複数の並列分散システムを結合してできる広域のシステムを対象とするものは余り多くない．そこで本論文は，トポロジーがあるSplit Graphで表される広域の並列分散システムに対する最小ブロードキャスト時間問題について考察する．通信に要する時間が均一である同種並列分散システムにおいては，この問題が多項式時間で解けることを示す．また通信に要する時間が均一でない異種並列分散システムに対するヒューリスティックなアルゴリズムを提案し，シミュレーション結果から効率の良い配信手順が短時間で得られることを確認した．

キーワード　グラフ理論，最小ブロードキャスト時間問題，スケジューリングアルゴリズム，Split Graph，並列分散システム

1. まえがき

近年，クラスタシステムなどの並列分散システムに関する研究，及び開発が盛んである．このシステムは，単独で稼動できるコンピュータ（ノード）を通信リンクで結合したシステムであり，高性能な並列計算機が比較的安価に構築できるという特徴をもつ．また，最近では複数の並列分散システムをWANなどで結合した広域の並列分散システムも研究されるようになってきた[1], [2].

並列分散システムの重要な処理の一つにブロードキャストがある．ブロードキャストとは，システムを構成する全ノードにデータを配信する処理である．各ノード間の通信が1対1でしか行えない並列分散システムにおいては，ブロードキャストは1対1通信の組合せで実現する必要がある．本論文ではこの組合せをブロードキャストスケジュール（または単にスケジュール）と呼ぶことにする．ブロードキャストスケジュールには数多くのパターンが存在するが，高性能なシステムにおいては短時間で完了するスケジュールが望まれる．完了までの時間が最短のスケジュールを見つける問題は，最小ブロードキャスト時間問題（Minimum Broadcast Time Problem: MBT），またはブロードキャストスケジューリングとして知られ，これまでに数多く研究されてきた[2]～[11].

この問題に対する従来の研究は，ネットワークトポロジーと問題の複雑さとの関係について議論するか[3]～[7]，実際のクラスタへの適用を想定してトポロジーが完全グラフのシステムを対象とするもの[8]～[11]が多く，複数のクラスタを結合した広域並列分散システムを対象にしたものはあまり多くない．そこで，本論文では複数のクラスタを結合した広域並列分散システムに対するブロードキャストスケジューリングを検討する．特に，トポロジーがある種のSplit Graphで表されるシステムに限定して，ブロードキャストスケジューリングを検討する．

† 新潟工科大学，柏崎市
Niigata Institute of Technology, 1719 Fujihashi, Kashiwazaki-shi, 945–1195 Japan
†† 新潟大学工学部，新潟市
Faculty of Engineering, Niigata University, 2–8050 Ikarashi, Niigata-shi, 950–2181 Japan
††† 中央大学理工学部，東京都
Faculty of Science and Engineering, Chuo University, 1–13–27 Kasuga, Bunkyo-ku, Tokyo, 112–8551 Japan
* 現在，日本精機株式会社 R&D センター

本論文では，まず，データ通信に要する時間が均一である同種並列分散システムに対する最小ブロードキャスト時間問題を検討する．同種並列分散システムに対するこの問題は，トポロジーが完全グラフや木を除いて一般に NP 完全であることが知られている [3]. Split Graph に対しては，ブロードキャスト開始前にデータを所持するノード数が 2 以上の場合には NP 完全であるが，一つだけの場合は NP 完全か否かは不明である [5]．本論文では，点集合を分離したときに，独立集合に属するすべての点（以後，この点を独立点と呼ぶ）の次数が 1 である Split Graph に限定すると，最小ブロードキャスト時間問題が多項式時間で解けることを示す．これについては **3.** で述べる．

最近のブロードキャストスケジューリングに関する研究では，データ通信に要する時間が均一でない異種並列分散システムに注目が集まってきている [7]～[11]．そこで，異種並列分散システムのトポロジーを独立点の次数が 1 である Split Graph に限定した場合のブロードキャストスケジューリングについても検討する．**4.** では，異種並列分散システムに対するスケジュールを出力するヒューリスティックなアルゴリズムを提案し，計算機シミュレーションによりその効果を確認する．

2. 通信モデル

本論文では小規模の並列分散システムを WAN などで結合した広域並列分散システムを想定する．小規模のシステムはそのトポロジーがスター型であるとし，広域のシステムは小規模システムのスター型の中心に位置するノードを完全グラフで結合したものとする．想定するシステムのネットワークの例を図 1 に示す．また，このシステムにおいてはすべての通信が 1 対 1 のブロッキング通信で行われるとする．つまり，通信中のノード対はその通信が完了するまで第 3 のノードと通信できないものとする．

以降では，並列分散システムをグラフで表現し，抽象化して考える．つまり，ノードを点，通信リンクを辺としたグラフでシステムを表現する．点集合を V，辺集合を E とするグラフ $G=(V,E)$ において，点集合がクリーク V_C と独立集合 V_I に分離できるとき，グラフ G は Split Graph と呼ばれる [5]．上記システムにおいて，スター型の中心に位置するノードの集合はクリークであり，それ以外のノードの集合は独立集合である．つまり，本論文で想定する広域並列分散システムのトポロジーは，すべての独立点の次数が 1 の Split Graph である．

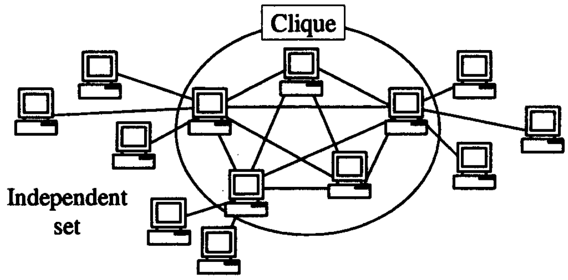


図 1　ネットワークトポロジーの例
Fig. 1　An example of the network topology.

3. 同種並列分散システムのブロードキャスト

3.1 同種並列分散システム

同種並列分散システムは，ノードや通信リンクなどシステムを構成する要素の能力がすべて同じシステムである．つまり，システム内でのあらゆる通信において，通信に要する時間が均一であるという特徴をもつ．いかなる通信においても 1 回の通信に要する時間が同じなので，いくつかの通信が同時に開始すれば，それらの通信は同時に終了する．そこで，同種並列分散システムにおける通信は常に同期がとれているものとする．そして，1 回の通信に要する期間をステップと呼ぶ．

3.2 ブロードキャストスケジュール

この節では，ブロードキャストスケジュールを定義する．ブロードキャストの開始前から配信データを所持する点を初期点と呼び，またその集合を初期点集合と称して V_0 $(V_0 \subseteq V)$ で表す．ブロードキャストを開始してから k ステップ目に通信を行う辺の集合を E_k $(E_k \subseteq E)$ で表し，k ステップ目完了時にデータを所持する点の集合を V_k $(V_k \subseteq V)$ で表す．ブロードキャストが K ステップで完了するとき，つまり $V_K = V$ であるとき，ブロードキャストスケジュールは点集合 V_k と辺集合 E_k が交互に現れる列 $S=(V_0,E_1,V_1,E_2,\cdots,E_K,V_K)$ で表記できる．スケジュール S において，V_k，E_k，及び各 V_k, E_k の任意の要素は，$1 \le k \le K$ のすべての k で次の条件を満足する必要がある [3]．

（1）　k ステップ目に通信を行う辺 e $(e \in E_k)$ の端点の一つは，必ず V_{k-1} の要素である．

（2）　任意の異なる 2 辺 e_i, e_j $(e_i, e_j \in E_k, e_i \neq$

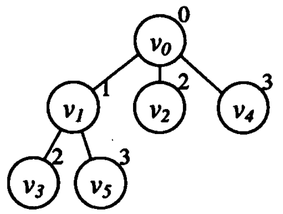


図 2 ブロードキャスト木の例
Fig. 2 An example of the broadcast tree.

e_j) において，e_i，e_j の端点はすべて異なる．

（3） $V_k = V_{k-1} \cup \{v_i | (v_i, v_j) \in E_k, v_j \in V_{k-1}\}$.

スケジュール S において，点 v が i 番目にデータを送信する点を $CH_S(v,i)$ ($CH_S(v,i) \in V$) と表記する．そして，点 v がデータを送信する点の総数を $NC_S(v)$ で表す．点 v と点 $CH_S(v,i)$ との間には親子の関係があり，点 $CH_S(v,i)$ は番号 i の順に兄弟の関係がある．兄弟関係を左から順に並べるものとして親子，兄弟関係を木構造で表現すると，スケジュール S は初期点を根とする順序木になる．この順序木をブロードキャスト木と呼ぶ．以降では，このブロードキャスト木をブロードキャストスケジュールと同義に扱う．図 2 はブロードキャスト木の一例である．図 2 において，各点の右上に添えられた数字はデータを受信したステップを表す．そして，データを受信したステップの最大値はシステム全体へのブロードキャストに要する時間を意味し，以降ではこの時間をブロードキャスト完了時間と呼ぶことにする．

3.3 限界時刻付き最小ブロードキャスト時間問題

想定するシステムのトポロジーにおいて，独立点の次数が 1 であることから，初期点以外の独立点はデータを受信しても他の点への送信は行わない．よって，次の定理が成立する．

［定理 1］ 独立点の次数が 1 である Split Graph $G = (V, E)$ に対するブロードキャストにおいて，クリーク側の点はその点に隣接する初期点以外の全独立点を最後にまとめて配信する最適なスケジュールが存在する．言い換えると，すべてのクリーク側の点 v ($v \in V_C$) において，以下の条件を満たす最適なスケジュール S が存在する．

$$\begin{cases} CH_S(v,i) \in V_C \\ \quad (1 \le i \le NC_S(v) - N_I(v)) \\ CH_S(v,i) \in V_I \\ \quad (NC_S(v) - N_I(v) + 1 \le i \le NC_S(v)) \end{cases} \tag{1}$$

なお，この式の $N_I(v)$ はクリーク側の点 v に隣接する初期点以外の独立点の個数を表す． □

この定理の証明は付録 **1.** に示す．

定理 1 は，初期点を除いた独立点を最後にまとめて配信するスケジュールを考慮すれば，最適なスケジュールが得られることを意味する．そこで，以降では定理 1 を満たすスケジュールだけを考えることにする．

定理 1 を満たすスケジュールが K ステップ以内に完了するためには，クリーク側の点 v は $K - N_I(v)$ ステップ目までに受信しなければならず，$K - N_I(v)$ ステップ目より後に他のクリーク側の点に送信してはならない．また，独立集合側初期点がそれに隣接するクリーク側の点へ第 2 ステップ以降に配信するスケジュールは，1 ステップ目に配信するスケジュールと比べてブロードキャスト完了時間が遅くなることはあっても早くなることはない．つまり，独立集合側初期点において，1 ステップ目より後でのデータの配信は不要である．そこで，クリーク側の未受信点との通信が行えなくなる時刻を限界時刻と呼び，次式で定義する(注1)．

$$e(v) = \begin{cases} K - N_I(v) & (v \in V_C) \\ 1 & (v \in V_I \cap V_0) \end{cases} \tag{2}$$

クリーク側の点において，その点の限界時刻までにデータを配信すれば，隣接するすべての独立点へも K ステップ数以内に配信できる．つまり，各点に限界時刻を付与すると，初期点以外の独立点は考慮する必要がない．そこで以降では，初期点以外の独立点を取り除いたグラフでブロードキャストを考えることにする．また，次の条件を満たす独立集合側初期点を除去しても，問題 MBT に影響を与えないのは明らかである．

- 隣接するクリーク側の点も初期点である独立集合側初期点は除去する．
- 同一のクリーク側の点に隣接する独立集合側初期点が二つ以上存在する場合は，一つを残して独立集合側初期点をすべて除去する．

限界時刻を付与して，余分な独立点を除去したグラフの例を図 3 に示す．

以上のように限界時刻の付与と余分な独立点の除去を行ったグラフに対する問題 MBT は以下のように定義される．

(注1)：限界時刻の意味より，初期点以外の独立点の限界時刻は $e(v) = K$ であるが，以降の議論において不要なため式 (2) の定義では省略する．

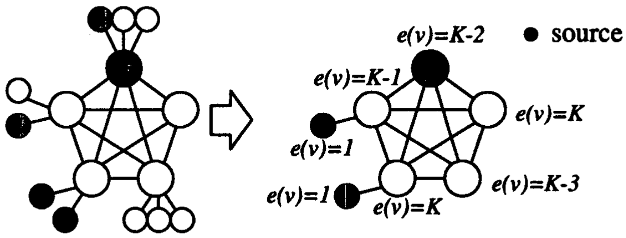


図 3　限界時刻の付与の例
Fig. 3　An example of installation of the boundary time at each vertex.

［定義 1］（MBT with boundary time: MBTWBT）

限界時刻を付与して変形したグラフ $G=(V,E)$，初期点集合 $V_0 \subseteq V$，各点 v の限界時刻 $e(v)$，及び正整数 K が与えられるとする．このとき，$1 \leq k \leq K$ において式 (3) を満足するブロードキャストスケジュール S が存在するか．

$$\forall (v_a, v_b) \in E_k,\ e(v_a) \geq k,\ e(v_b) \geq k \tag{3}$$

□

なお，余分な独立点の除去を行わず，すべての点の限界時刻を $e(v)=K$ とすると，この問題は一般的な MBT と等価になる．

3.4　ブロードキャストスケジュールの存在条件

問題 MBTWBT において，K ステップ以内にブロードキャストが完了するための条件について述べる．以降の説明において，独立集合側初期点に隣接するクリーク側の点の集合を V_{C1} と表記する．つまり，$V_{C1} = \{v|(u,v) \in E, u \in V_I \cap V_0\}$ とする．

K ステップでブロードキャストが完了するスケジュール S において，そのスケジュール S の要素 E_k は式 (3) を満たさなければならない．これは，$0 \leq k \leq K$ であるすべての k において，次式を満たすことと等価である．

$$0 \leq k \leq K,\ \ \{v|e(v) \leq k\} \subseteq V_k \tag{4}$$

ここで，式 (4) を満たさない，つまりある $k_0 (0 \leq k_0 \leq K)$ で，

$$\{v|e(v) \leq k_0\} \not\subseteq V_{k_0} \tag{5}$$

が成立すると仮定する．このとき，第 k_0 ステップ終了時の未受信点集合 $V \setminus V_{k_0}$ の中には，$e(v') \leq k_0$ を満たす点 v' が存在する．そして第 k_0 ステップより後に点 v' へは配信できないので，この場合はブロードキャストが完了しない．そこで式 (5) を満たすとき，k_0 ステップ目でブロードキャストが破綻すると呼ぶことにする．一方，$0 \leq k \leq k_0$ のすべての整数 k に対して式 (4) が成立する，つまり，

$$0 \leq k \leq k_0,\ \ \{v|e(v) \leq k\} \subseteq\ V_k \tag{6}$$

であるとき，k_0 ステップ目までブロードキャストが破綻しないと呼ぶことにする．

次に，第 k ステップ完了時におけるデータ受信済み点の総数の最大値を導出する．

［補題 1］　$k-1$ ステップ目までにブロードキャストが破綻しないとき，k ステップ目終了時におけるデータ受信済み点の総数の最大値 $A(k)$ は次式で表される．

$$A(k) = \begin{cases} |V_0| & (k=0) \\ 2A(k-1) - |\{v|e(v) \leq k-1\}| & (k \geq 1) \end{cases} \tag{7}$$

（証明）　$k=0$ の場合，$A(0)=|V_0|$ は自明である．

初期点がクリーク側と独立集合側とに混在していても，混在していなくても，1 ステップ目においては最大で $|V_0| - |\{v|e(v) \leq 0\}|$ 個の点が送信できる．つまり，$A(1) = 2|V_0| - |\{v|e(v) \leq 0\}|$ であり，$k=1$ のときは式 (7) が成立する．

k ステップ目終了時の受信済み点の総数の最大値が式 (7) で表されるとする．また，k ステップ目まで破綻しないならば，$\{v|e(v) \leq k\} \subseteq V_k$ である．$k+1$ ステップ目において $e(v) \leq k$ を満たす点はデータの送信ができない．よって，$k+1$ ステップ目で送信が可能な点は最大で $A(k) - |\{v|e(v) \leq k\}|$ 個存在する．この点すべてが他の未受信点へデータを配信すれば，$k+1$ ステップ目でデータ受信が完了する点は最大で，

$$\begin{aligned} &A(k) + \{A(k) - |\{v|e(v) \leq k\}|\} \\ &\quad = 2A(k) - |\{v|e(v) \leq k\}| \end{aligned}$$

個存在する．つまり，$k+1$ のときも式 (7) は成立する．

以上より，式 (7) は第 k ステップ完了時の受信済み点の最大総数を表す．　□

次に，途中で破綻しない限り，第 k ステップ完了時の受信済み点の個数が $A(k)$ であるスケジュールが存在することを示す．

［補題 2］　$k \geq 1$ とし，$A(k) < |V|$ であるとする．$k-1$ ステップ目までブロードキャストが破綻しなければ，$|V_k| = A(k)$ となるスケジュールが存在する．

（証明）　ブロードキャスト開始前から破綻せずに $A(1) < |V|$ である場合，明らかにすべての初期点が未受信の点に配信することが可能だから，$|V_1| = A(1)$

となるスケジュールが存在する．

あるスケジュール S において，$k-1$ ステップ目まで破綻せずに $|V_{k-1}| = A(k-1)$ であったとする．$k-1$ ステップ目まで破綻していないので，第 k ステップ開始時に送信可能な点は $A(k-1) - |\{v|e(v) \leq k-1\}|$ 個存在する．また，未受信点は $|V| - A(k-1)$ 個存在し，そのすべての点が $e(v) \geq k$ を満たす．$A(k) < |V|$ であるから，式 (7) より次式が成立する．

$$A(k-1) - |\{v|e(v) \leq k-1\}| < |V| - A(k-1)$$

未受信点は送信可能な点より多いので，$|V_k| = A(k)$ であるように配信することができる．

以上より，$k-1$ ステップ目まで破綻しない限り，$A(k) < |V|$ ならば $|V_k| = A(k)$ であるスケジュールが存在する． □

次に，$|V_k| = A(k)$ であるスケジュールが満たす条件を示す．

［補題 3］ $k_0 \geq 1$ とする．第 $k_0 - 1$ ステップまで破綻しないスケジュールが $|V_{k_0}| = A(k_0)$ であるとき，$1 \leq k \leq k_0$ のすべての k で $(V_0 \cup V_{C1}) \subseteq V_k$ を満たす．

（証明） 独立集合側の初期点がない場合は $V_{C1} = \emptyset$ であり，明らかに $(V_0 \cup V_{C1}) \subseteq V_k$ を満たす．そこで，以下では独立集合側の初期点がある場合を考える．

式 (7) より，$|V_{k_0}| = A(k_0)$ であるためには k_0 ステップ目までのすべてのステップにおいて配信可能な最大個数ずつ配信する必要がある．つまりスケジュール S において $|V_{k_0}| = A(k_0)$ ならば $|V_1| = A(1)$ を満たす．$|V_1| = A(1)$ となるには，すべての独立集合側の初期点がそれに隣接するクリーク側の点 $v \in V_{C1}$ へ配信しなければならない．逆に，独立集合側初期点が一つでも 1 ステップ目に配信しなければ，$|V_1| = A(1)$ とはならない．したがって，$|V_1| = A(1)$ ならば $(V_0 \cup V_{C1}) \subseteq V_1$ を満たす．$V_1 \subseteq V_2 \subseteq \cdots \subseteq V_k$ だから，$|V_{k_0}| = A(k_0)$ ならば $1 \leq k \leq k_0$ のすべての k で $(V_0 \cup V_{C1}) \subseteq V_k$ を満たす． □

次に，破綻しないスケジュールが存在するための必要十分条件を示す．

［定理 2］ $\{v|e(v) \leq 0\} \subseteq V_0$ であるとする．また，$k_0 \geq 1$ とし，$A(k_0) < |V|$ であるとする．k_0 ステップ目まで破綻しないスケジュールが存在するための必要十分条件は，$1 \leq k \leq k_0$ のすべての k において次式が成立することである．

$$A(k) - |V_0| - |V_{C1}| \geq |\{v|e(v) \leq k, v \in V \setminus (V_0 \cup V_{C1})\}| \tag{8}$$

（証明） まず，必要条件であることを示す．k_0 ステップ目まで破綻しないスケジュールは，$1 \leq k \leq k_0$ であるすべての k で式 (6) を満たす．このとき，補題 2 より $|V_k| = A(k)$ であるスケジュール S が存在する．更に補題 3 より，このスケジュール S は $1 \leq k \leq k_0$ で $(V_0 \cup V_{C1}) \subseteq V_k$ を満たす．また，

$$\{v|e(v) \leq k, v \in (V_0 \cup V_{C1})\} \subseteq (V_0 \cup V_{C1}) \tag{9}$$

は自明であるから，式 (6) と式 (9) より，$1 \leq k \leq k_0$ のすべての k で，

$$\{v|e(v) \leq k,\ v \in V \setminus (V_0 \cup V_{C1})\} \subseteq V_k \setminus (V_0 \cup V_{C1}) \tag{10}$$

が成立する．V_0 と V_{C1} は互いに素であるから，

$$|V_k| - |V_0| - |V_{C1}| \geq |\{v|e(v) \leq k,\ v \in V \setminus (V_0 \cup V_{C1})\}| \tag{11}$$

が成立し，$A(k) \geq |V_k|$ より式 (8) が成り立つ．したがって，式 (8) は k_0 ステップ目まで破綻しないスケジュールが存在するための必要条件である．

次に，十分条件であることを示す．$A(k_0) < |V|$ であるとき，$1 \leq k \leq k_0$ のすべての k で式 (8) が成立しているとする．ある k_x $(1 \leq k_x \leq k_0)$ において，$k_x - 1$ ステップ目まで破綻しないスケジュールが存在すると仮定し，$k = k_x$ のときの破綻の有無を考える．$k_x - 1$ ステップ目まで破綻しないことと $A(k_x) < |V|$ であることから，補題 2 より $|V_{k_x}| = A(k_x)$ であるスケジュールが存在する．また補題 3 より，このスケジュールでは $(V_0 \cup V_{C1}) \subseteq V_{k_x}$ が成り立つ．このスケジュールにおいて式 (8) は，

$$|V_{k_x}| - |V_0| - |V_{C1}| \geq |\{v|e(v) \leq k_x,\ v \in V \setminus (V_0 \cup V_{C1})\}| \tag{12}$$

である．この式の左辺は V_{C1} を除いて k_x ステップ目までに受信可能な点数を表す．したがってこの式は，k_x ステップ目までの配信で集合 $V \setminus (V_0 \cup V_{C1})$ 中にある限界時刻が k_x 以下のすべての点へ配信が可能であることを意味する．つまり，次式を満たす V_{k_x} が存在する．

$$\{v|e(v) \leq k_x,\ v \in V \setminus (V_0 \cup V_{C1})\} \subseteq V_{k_x} \setminus (V_0 \cup V_{C1}) \tag{13}$$

$(V_0 \cup V_{C1}) \subseteq V_{k_x}$ であるから，

$$\{v|e(v) \leq k_x\} \subseteq V_{k_x}$$

が成立し，k_x ステップ目まで破綻しないスケジュールが存在する．

さて，前提条件より $\{v|e(v) \leq 0\} \subseteq V_0$ が成立するので，$k=1$ で式 (8) が成立すれば 1 ステップ目で破綻しないスケジュールが存在する．したがって，$A(k_0) < |V|$ であり，かつ $1 \leq k \leq k_0$ のすべての k で式 (8) が成立しているならば，k_0 ステップ目まで破綻しないスケジュールが存在する．つまり，式 (8) は破綻しないスケジュールが存在するための十分条件である． □

この定理は，点集合 $V \setminus (V_0 \cup V_{C1})$ に注目して，k ステップ目までに配信が可能な点の最大数が $e(v) \leq k$ を満たす点の個数以上ならば，破綻しないスケジュールが存在することを意味する．逆にいえば，式 (8) が成立しないならば，ブロードキャストが完了するスケジュールは存在しない．

次に，ブロードキャストが K ステップ以内に完了するための十分条件を示す．

［定理 3］ $\{v|e(v) \leq 0\} \subseteq V_0$ であるとする．$A(k_0) \geq |V|$ を満たす $k_0 (k_0 \leq K)$ が存在し，かつ $k \leq k_0 - 1$ であるすべての k において式 (8) を満たすならば，ブロードキャストは K ステップ以内に完了する．

この証明は付録 **2.** に示す．

3.5 アルゴリズム

この節では，問題 MBTWBT を解くアルゴリズムを述べる．式 (3) の条件より，k ステップ目において $e(v) < k$ である点はデータの送受信ができない．そこで，$e(v)$ の小さなノードへ優先的に配信すると，K ステップ以内にブロードキャストが完了するスケジュールを得ることができると考えられる．以下では，この考えに沿ったアルゴリズムを示す．

［定義 2］ (Boundary Time Ordering: BTO)

Input： 限界時刻を付与して余分な独立点を除去したグラフ $G = (V, E)$，ソースノード集合 $V_0 \subseteq V$，限界時刻 $e(v)$，正整数 K が与えられる．

Step 1: 初期点集合 V_0 を限界時刻 $e(v)$ の昇順にソートし，ソート後の点列を $v_1, v_2, \cdots, v_{|V_0|}$ とする．独立集合側初期点が隣接するクリーク側の点集合 V_{C1} を限界時刻 $e(v)$ の昇順にソートし，ソート後の点列を $v_{|V_0|+1}, v_{|V_0|+2}, \cdots, v_{|V_0|+|V_{C1}|}$ とする．更に，上記以外の点集合 $V \setminus \{V_0 \cup V_{C1}\}$ を限界時刻 $e(v)$ の昇順にソートし，ソート後の点列を $v_{|V_0|+|V_{C1}|+1}, v_{|V_0|+|V_{C1}|+2}, \cdots, v_{|V|}$ とする．

Step 2: $e(v_{|V_0|+1}) \leq 0$，または $e(v_{|V_0|+|V_{C1}|+1}) \leq 0$ ならば，No を出力して終了する．

Step 3: ステップ数を変数 k で，受信済み点数を変数 i で表記し，以下のように値を設定する．
$k \leftarrow 0$，$i \leftarrow |V_0|$．

Step 4: $i \geq |V|$ ならば，Yes を出力し終了する．

Step 5: $k = K$，または $e(v_{i+1}) \leq k$ ならば，No を出力し終了する．

Step 6: パラメータ k, i を以下の順に更新した後，Step 4 に戻る．
$k \leftarrow k+1$，$i \leftarrow 2i - |\{v|e(v) \leq k-1\}|$． □

アルゴリズム BTO の適用例を図 4 に示す．図 4 (a) のグラフが与えられたとする．このグラフに対して，$K=3$ として限界時刻を付与し，余分な独立点を除去したものが図 4 (b) である．図 4 (b) において，各点のラベルの右下に添えられた数字は，アルゴリズム BTO の Step 1 で得られた点列の順序を意味する．図 4 (c) は図 4 (b) のグラフに対してアルゴリズム BTO を適用したときの処理の様子であり，特に Step 6 に注目してアルゴリズムの挙動を示したものである．図 4 (c) 中の矢印は，k ステップ目の配信を意味する．図 4 (c) に見るように，このアルゴリズムはラベルに添えられた数字の昇順に配信する．そして，図 4 (d) のスケジュールがアルゴリズム BTO から容易に得られる．

次に，問題 MBTWBT がこのアルゴリズム BTO によって解けることを示す．

［補題 4］ 問題 MBTWBT はアルゴリズム BTO により解ける．

（証明） まず，アルゴリズム内の変数 i が第 k ステップ終了時の受信済み点の最大個数 $A(k)$ に等しいことを示す．Step 3 より $k=0$ のとき $i = |V_0|$ であり，これは $A(0) = |V_0|$ と一致する．次に Step 6 における変数 i の更新式に注目する．この更新式において，右辺の i は $A(k-1)$ を表しているので，左辺の i は式 (7) と一致する．したがって，変数 i は $A(k)$ に等しい．

次に，アルゴリズムの出力が No である場合に，問題の解と矛盾しないことを示す．アルゴリズムが No を出力するのは，Step 2 の条件を満たすときか，Step 5 の条件を満たすときである．まず，Step 2 で No を出力する場合を考える．点列 $v_{|V_0|+1}, \cdots, v_{|V_0|+|V_{C1}|}$ 及び点列 $v_{|V_0|+|V_{C1}|+1}, \cdots$ は限界時刻の昇順に並べ

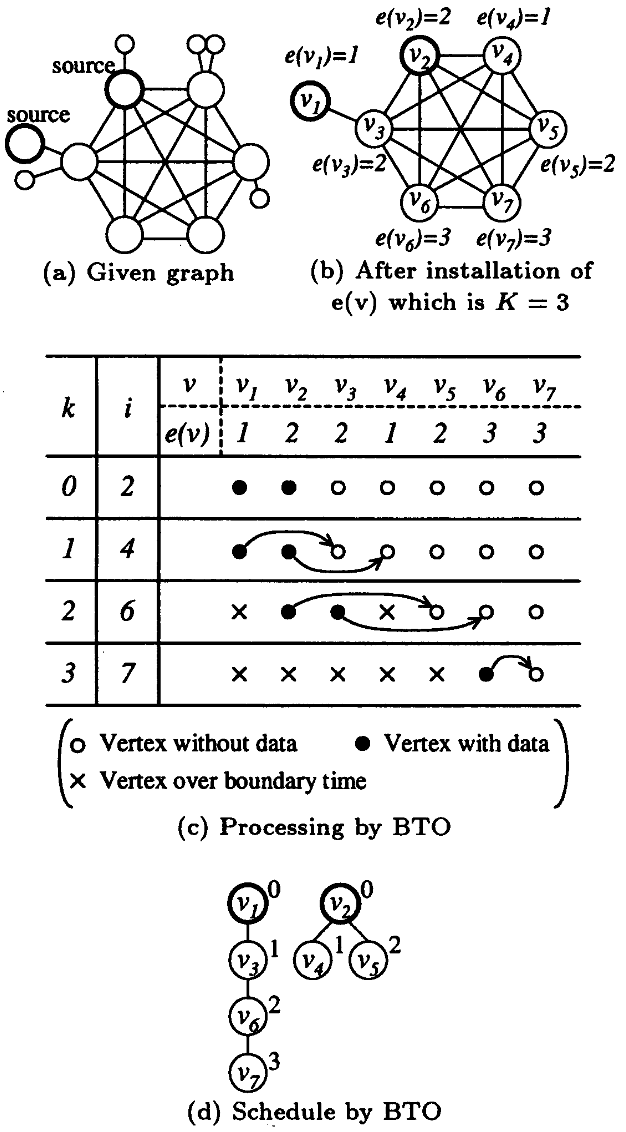


図 4　アルゴリズム BTO の適用例
Fig. 4　An example of use of the algorithm BTO.

られているので，点集合 V_{C1} または $V \setminus (V_0 \cup V_{C1})$ の中に $e(v) \leq 0$ である点 v が存在すれば，必ず Step 2 の条件を満たす．このとき，明らかに式 (5) が成立しているので，ブロードキャスト開始前から破綻している．よって，この場合は問題の解と矛盾しない．

次に，Step 5 で No を出力する場合を考える．まずは Step 5 の条件 $k = K$ を満たす場合を考える．Step 5 が実行されるのは $i < |V|$ の場合であり，このとき $k = K$ ならば明らかに $A(K) < |V|$ なので，ブロードキャストが K ステップで完了していないことを意味する．次に，Step 5 の後者の条件 $e(v_{i+1}) \leq k$ を満たす場合を考える．ここで，$i = i' + |V_0| + |V_{C1}|$ とすると，i' は $A(k) - |V_0| - |V_{C1}|$ を表す．つまり i' は，集合 $V \setminus (V_0 \cup V_{C1})$ 内の k ステップ目までに受信した点の個数を意味する．さて，$e(v_{i+1}) = e(v_{i'+|V_0|+|V_{C1}|+1}) \leq k$ であるということは，集合 $V \setminus (V_0 \cup V_{C1})$ の中に $e(v) \leq k$ である点が i' 個より多く存在するということを意味する．また，点列 $v_{|V_0|+|V_{C1}|+1}, \cdots, v_{|V|}$ は限界時刻の昇順に並べられているので，点集合 $V \setminus V_k$ の中に $e(v) \leq k$ である点 v が存在すれば，必ず Step 5 の条件 $e(v_{i+1}) \leq k$ を満たす，これは定理 2 の必要十分条件に反しており，ブロードキャストが破綻することを意味する．つまり，Step 5 で No を出力する場合は問題の解と矛盾しない．以上から，アルゴリズム BTO が No を出力する場合は問題の解と矛盾しない．

アルゴリズムの出力が Yes の場合に問題の解と矛盾しないことを示す．アルゴリズムが Yes を出力するのは，Step 4 の条件を満たした場合である．Step 5 の停止条件を踏まえると，$i \geq |V|$ を満たすときは必ず $k \leq K$ である．更に，Step 4 で $i \geq |V|$ を満たすということは，Step 5 の条件 $e(v_{i+1}) \leq k$ を 1 度も満たさないので，この場合はブロードキャストが破綻しない．したがって，Step 4 の条件を満たすときは定理 3 を満足しているので，アルゴリズム BTO の出力が Yes の場合も問題の解と矛盾しない．

以上より，アルゴリズム BTO の出力は問題 MBTWBT の解と矛盾しない．　□

アルゴリズム BTO の計算量について考える．Step 1 のソートと Step 4 から Step 6 までの繰返しの回数を考慮すると，計算量は $O(|V| \log |V| + K)$ である．しかし，K の上限が $|V| - 1$ であることを踏まえると，計算量は $O(|V| \log |V|)$ である．

3.3 でも触れたように，問題 MBTWBT は問題 MBT と等価である．そして，**3.3** で述べた余分な独立集合側の点の除去は多項式時間での処理が可能であることから，次の定理が成立する．

［定理 4］　独立集合側のすべての点の次数が 1 である Split Graph に対する最小ブロードキャスト時間問題は多項式時間で解ける．

4. 異種並列分散システムのブロードキャスト

4.1　異種並列分散システムの通信モデル

異種並列分散システムは，システムを構成するノード，及び通信リンクの性能が均一でないシステムである．

異種並列分散システムの通信モデルはいくつか提案

されている [7]～[9]．その一つに，Banikazemi らが提案したモデルがある [9]．これは，1 回の送受信に要する時間（以後，本論文ではこの時間を送信時間と呼ぶ）が送信側コンピュータの能力のみで決定されるという単純なモデルである．この章では，Banikazemi らが提案した通信モデルを適用し，**2.** で述べたトポロジーの異種並列分散システムに対するブロードキャストスケジューリングを検討する．ところで，異種並列分散システムに対しても定理 1 が成立することは容易に分かる．そこで同種並列分散システムの場合と同様に，異種並列分散システムにおいても初期点，及びクリーク側の点の配信順序だけを考えることにする．

4.2 アルゴリズム

ネットワークトポロジーが完全グラフで表される異種並列分散システムに対して，FNF（Fastest Node First）は効率の良いブロードキャストスケジュールを出力する [9]～[11]．まずは，独立点の次数が 1 である Split Graph をネットワークトポロジーとするシステムに対して FNF を適用した場合を考える．

FNF は次の手順でブロードキャスト木を生成する [9]．

（1） 受信済み点のうち，次のデータ送信が最も早く完了する点を選ぶ．この点を v_s とする．

（2） 未受信点のうち，送信時間が最短の点を選ぶ．この点を v_r とする．

（3） v_s を送信点，v_r を v_s が最後に送信する点としてブロードキャスト木に追加する．

（4） 未受信点がなくなるまで（1）～（3）を繰り返す．

トポロジーが Split Graph であるネットワークに対して FNF を適用した例を図 5 に示す．なお図 5 において，丸内の数字は各点の送信時間を表す．また，ブロードキャスト木において，各点の右上に添えられた数字はデータを受信した時刻（ブロードキャスト開始からの経過時間）を表す．図 5 (c) は，図 5 (a) のシステムに対して FNF を適用して得られたスケジュールである．一方，図 5 (b) は最適なスケジュールの一つである．FNF は初期点の送信速度がシステム内で最小であり，かつ送信速度が 2 種類の場合には最適スケジュールを出力することが知られている [10]．図 5 (a) は FNF が最適スケジュールを出力する条件を満たしているが，図 5 より明らかなように FNF は最適なスケジュールを出力しない場合がある．つまり，トポロジーが Split Graph の場合には，FNF に関する結果

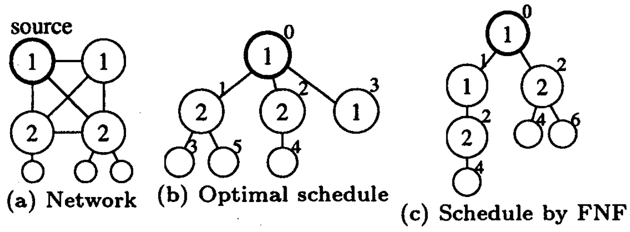


図 5　FNF 適用の問題点
Fig. 5　A problem of applying the FNF.

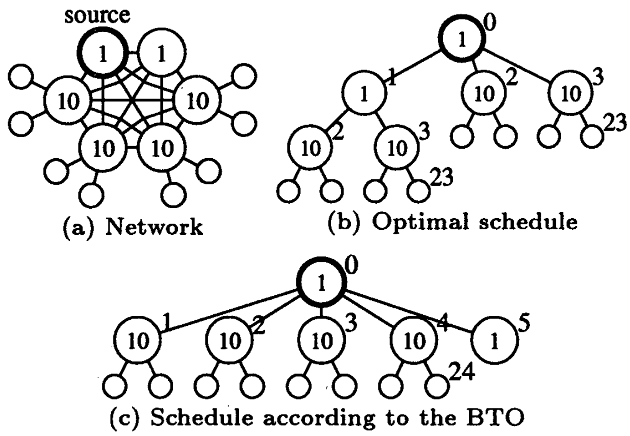


図 6　アルゴリズム BTO に沿った配信の問題点
Fig. 6　A problem of schedule according to the BTO.

をそのまま適用することはできない．

図 5 (b) のスケジュールは，独立集合側への配信に要する時間が多い順に配信している．つまり，異種並列分散システムに対してもアルゴリズム BTO と同様に，独立集合側への配信に要する時間の多いクリーク側の点へ優先的に配信すれば，良いスケジュールが得られると考えられる．ところが，この考えに沿っても良いスケジュールが得られない場合がある．その一例を図 6 に示す．図 6 (a) のネットワークに対し，アルゴリズム BTO の考えに沿うと図 6 (c) のスケジュールが得られる．しかし，図 6 (a) のネットワークに対する最適なスケジュールは図 6 (b) である．図 6 (b) に注目すると，FNF と同様に送信時間が短い点を利用してクリーク側の点への配信を促進している．そこで，異種並列分散システムに対しては，FNF と BTO を併用してブロードキャストスケジュールを得ることにする．

異種並列分散システムに対するアルゴリズムを以下に示す．

［定義 3］（Independent Verticies Delivery Time Ordering: IVDTO）

Input： **3.3** で述べた方法により余分な独立点を除去

したグラフ $G=(V,E)$，初期点集合 $V_0(V_0 \subseteq V)$，各点 $v(v \in V)$ の送信時間 $t(v)$，各点 $v(v \in V)$ に隣接する初期点以外の独立点の個数 $N_I(v)$ が与えられる．

Step 1: 送信可能な点の集合を V_S で表し，$V_S \leftarrow V_0$ とする．未受信点の集合を V_r で表し，$V_r \leftarrow V \setminus V_0$ とする．スケジュール S は初期点 $v(v \in V_0)$ が根の林とする．

Step 2: $V_r \neq \emptyset$ の間，Step 2-1 から Step 2-4 を繰り返す．

Step 2-1: 未受信点集合 V_r の中で，隣接する独立点への配信に要する時間 $t(v)N_I(v)$ が最大の点 v を受信点 v_r とする．

Step 2-2: 送信可能点 $v(v \in V_S)$ において，v が配信するクリーク側の最後の点として点 v_r をブロードキャスト木に追加した場合（図 7 参照），ブロードキャスト完了時間が最小となる点を集合 V_S の中から見つけ出す．見つけ出した点を第 1 の送信点候補 v_{s1} とし，そのときのブロードキャスト完了時間を T_1 とする．

Step 2-3: 集合 $V_r \setminus \{v_r\}$ のうち，送信時間 $t(v)$ が最小の点を中継点 v_i とする．送信可能点 $v(v \in V_S)$ において，v が最初に配信する点として v_i を，中継点 v_i が最初に配信する点として v_r をブロードキャスト木に追加した場合（図 8 参照），ブロードキャスト完了時間が最小となる点を集合 V_S の中から見つけ出す．見つけ出した点を第 2 の送信点候補 v_{s2} とし，そのときのブロードキャスト完了時間を T_2 とする．

Step 2-4: $T_1 < T_2$ ならば Step 2-4-1 へ，$T_1 \geq T_2$ ならば Step 2-4-2 へ進む．

Step 2-4-1: スケジュール S において v_{s1} が配信するクリーク側の最後の点として v_r を追加する．$V_r \leftarrow V_r \setminus \{v_r\}$，$V_S \leftarrow V_S \cup \{v_r\}$ として，Step 2 に戻る．

Step 2-4-2: スケジュール S において v_{s1} が配信する最初の点として v_i を，v_i が配信する最初の点として v_r を追加する．$V_r \leftarrow V_r \setminus \{v_r, v_i\}$，$V_S \leftarrow V_S \cup \{v_r, v_i\}$ として，Step 2 に戻る．

Step 3: スケジュール S を出力する． □

このアルゴリズムにおいて，Step 2-1 から Step 2-4 までの繰返し回数はたかだか $|V|-|V_0|$ である．更に，Step 2-2，2-3 において，点 v_{s1}, v_{s2} を見つけ出す処理の計算量は各 Step で $O(|V|)$ である．以上より，このアルゴリズムの計算量は $O(|V|^2)$ である．

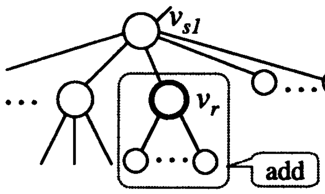


図 7 BTO に基づく点の追加
Fig. 7 Addition method based on BTO.

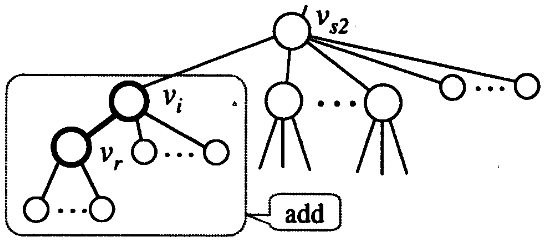


図 8 FNF も加味した点の追加
Fig. 8 Addition method which considered FNF.

4.3 シミュレーション

4.3.1 シミュレーション方法

アルゴリズム IVDTO の性能を評価する．ここでは初期点がクリーク側に 1 個だけ存在する単純なモデルで評価を行う．

ネットワークデータは，3 種類のパラメータ (1) クリーク側の点の個数 $|V_C|$，(2) 各クリーク側の点の送信時間 $t(v)$，(3) 各クリーク側の点に隣接する独立点の個数 $N_I(v)$ により一意に決まる．シミュレーションに用いたネットワークデータは次の手順で作成した．まず，$|V_C|$ を指定する．次に初期点 v_0 の送信時間を $t(v_0)=1$ とし，隣接独立点数を $N_I(v_0)=0$ とする．残りのクリーク側の点に対しては，各点 v の送信時間 $t(v)$ を整数値 1～10 の中からランダムに決定し，隣接する独立点の個数 $N_I(v)$ を整数値 0～10 の中からランダムに決定した．ただし，各点の送信時間の決定の際，すべての点の送信時間の種類が 2～5 種類となるように制限をつけ，この制限のもとにネットワークデータを作成した．以上のように作成したネットワークデータに対してシミュレーションを行った．なおシミュレーションは，Pentium4 2.4 GHz，512 MByte の RAM を搭載した PC 上で行った．

4.3.2 最適解との比較

クリーク側の点数を $|V_C| = 3, 4, \cdots, 9$ (注2)

(注2)：$|V_C| \geq 10$ については，ブロードキャスト木のパターン数がぼう大であるため，今回使用したシミュレーション環境では解を得るのが非常に困難であった．$|V_C|$ が多い場合に対しては，最適なスケジュールが満たす条件を利用して探索する解空間を制限する必要がある．

表 1 IVDTO の解の非最適解数（試行回数 200）
Table 1 A number of the un-optimal schedule by IVDTO. (200 trial)

$\|V_C\|$	3	4	5	6	7	8	9
非最適解数	0	0	0	0	1	0	0

として作成したネットワークデータに対して，最適ブロードキャスト木と IVDTO の解との比較を行った．なお，ネットワークデータは，各クリーク側の点数 $|V_C|$ に対して 200 個ずつ(注3)作成した．

IVDTO が出力するスケジュールのブロードキャスト完了時間と，最適解のブロードキャスト完了時間とが異なっていた解の個数を表 1 に示す．IVDTO で得られた解のほとんどは最適であったが，クリーク側の点数 $|V_C| = 7$ の場合に 1 件だけ最適解が得られなかった．しかし，クリーク側の点数が少ない場合には，かなり高確率で最適解が得られていることが分かる．

4.3.3 ランダム検索との比較

クリーク側の点数を $|V_C| = 10, 20, \cdots, 100$ として作成したネットワークデータに対して，ランダム検索と IVDTO との比較を行った．ネットワークデータは，各クリーク側の点数 $|V_C|$ に対して 400 個ずつ(注4)作成し，シミュレーションを行った．

ランダム検索は以下の操作で行った．まずはじめに，初期点だけからなる木を生成する．木の中からランダムに 1 点を選択し，これを送信点 v_s とする．次に未受信点集合の中からランダムに 1 点選択し，これを受信点 v_r とする．送信点 v_s が配信するクリーク側の最後の点として受信点 v_r を木に追加する．以上を未受信点がなくなるまで繰り返すと，一つのブロードキャスト木ができる．上記の手順によるブロードキャスト木を 10 万個生成し，10 万個のブロードキャスト木の中からブロードキャスト完了時間が最小の木をランダム検索の解とした．

ランダム検索に対する IVDTO の解の良さと，それぞれの方法の計算時間を表 2 に示す．この表の *ratio* は，ランダム検索解のブロードキャスト完了時間に対する IVDTO のブロードキャスト完了時間の比を表す．*ratio* は *ratio* < 1 ならば IVDTO が優れており，逆に *ratio* > 1 ならばランダム検索が優れていることを意味する．

表 2 ランダム検索と IVDTO との比較（試行回数 400）
Table 2 A comparison between a random search and the IVDTO. (400 trial)

$\|V_C\|$	*ratio* 平均値	*ratio* 最大値	平均計算時間 (ms) ランダム検索	平均計算時間 (ms) IVDTO
10	1.000	1.000	168	1 未満
20	0.996	1.000	439	1 未満
30	0.986	1.000	836	1
40	0.970	1.000	1336	1
50	0.959	1.000	1946	1
60	0.946	1.000	2720	1
70	0.929	1.000	3610	1
80	0.916	1.000	4534	2
90	0.902	0.990	5636	2
100	0.888	1.000	6811	3

表 2 において，*ratio* の最大値に注目すると，ランダム検索に対する IVDTO のブロードキャスト完了時間の比が 1 を超えることはなかった．つまり IVDTO は，ランダム検索と同等かそれより良い解を得ている．*ratio* の平均値に注目すると，クリーク側の点数 $|V_C|$ が増加するにつれ *ratio* は減少している．これは，解空間が広がっても IVDTO はランダム検索に比べて良い解が得られることを意味する．また，表 2 の計算時間に注目すると，IVDTO はランダム検索に比べて非常に短時間で計算でき，計算効率の良いアルゴリズムである．

5. むすび

小規模な並列分散システムを WAN で結合した広域並列分散システムのブロードキャストスケジューリングを目的として，独立点の次数が 1 である Split Graph に対する最小ブロードキャスト時間問題を検討した．まず，送受信に要する時間が均一である同種並列分散システムに対するアルゴリズム BTO を提案した．そしてこのアルゴリズム BTO により，独立点の次数が 1 である Split Graph に対する最小ブロードキャスト時間問題が多項式時間で解けることを示した．

また，送受信に要する時間が均一でない異種並列分散システムに対するブロードキャストスケジューリングを検討し，アルゴリズム BTO を応用したヒューリスティックなアルゴリズム IVDTO を提案した．計算機シミュレーションより，アルゴリズム IVDTO は効率の良いスケジュールを短時間で出力することを確認した．

本論文では，異種並列分散システムに対するアルゴリズム IVDTO を提案したが，このアルゴリズムが最

(注3)：作成した 200 個のネットワークデータの内訳は，送信時間が 2,3,4,5 種類のそれぞれの場合において各 50 個ずつである．
(注4)：作成した 400 個のネットワークデータの内訳は，送信時間が 2,3,4,5 種類のそれぞれの場合において各 100 個ずつである．

適解を出力するための条件やこのアルゴリズムの近似比などは未知である．今後はこれらについて検討する予定である．また，本論文では広域並列分散システムのトポロジーを独立点の次数が1である Split Graph である場合に限定して検討したが，広域並列分散システムとしては様々なトポロジーをもつシステムが考えられる．今後は，様々なトポロジーで表される広域並列分散システムについても検討する予定である．

文　献

[1] 伊藤　智，関口智嗣，"グリッドコンピューティングの現状と将来，" 第16回回路とシステム（軽井沢）ワークショップ，pp.483–488, April 2003.

[2] T. Kielmann, H.E. Bal, and S. Gorlatch, "Bandwidth-efficient collective communication for clustered wide area systems," Int. Parallel and Distributed Processing Symposium, pp.492–499, May 2000.

[3] M.R. Garey and D.S. Johnson, Computer and Intractability: A guide to the theory of NP-completeness, W.H.Freeman, 1979.

[4] S.M. Hedetniemi, S.T. Hedetniemi, and A.L. Liestman, "A survey of gossiping and broadcasting in communication networks," Networks, vol.18, pp.319–349, 1998.

[5] K. Jansen and H. Muller, "The minimum broadcast time problem for several processor networks," Theor. Comput. Sci., vol.147, no.1&2, pp.69–85, Oct. 1995.

[6] G. Kortsarz and D. Peleg, "Approximation algorithms for minimum time broadcast," SIAM J. Discrete Methods, vol.8, no.3, pp.401–427, 1995.

[7] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber, "Message multicasting in heterogeneous networks," SIAM J. Computing, vol.30, no.2, pp.347–358, 2000.

[8] P.B. Bhat, C.S. Raghavendra, and V.K. Prasanna, "Efficient collective communication in distributed heterogeneous systems," Proc. 19th IEEE Int. Conf. on Distributed Computing Systems, pp.15–24, 1999.

[9] M. Banikazemi, V. Moorthy, and D. Panda, "Efficient collective communication on heterogeneous networks of workstations," Proc. Intl. Conf. on Parallel Processing, pp.460–467, 1998.

[10] P. Liu and T-H. Sheng, "Broadcast scheduling optimization for heterogeneous cluster systems," Proc. 12th Ann. Symposium on Parallel Algorithms and Architectures, pp.129–136, 2000.

[11] 大下福仁，松前　進，増澤利光，都倉信樹，"異種並列計算環境におけるブロードキャストスケジューリング，" 信学論 (D-I), vol.J86-D-I, no.2, pp.88–98, Feb. 2003.

付　　録

1. 定理1の証明

（証明） 式 (1) を満たさない最適ブロードキャスト木があるとする．この最適ブロードキャスト木は，図 A·1 の左側のようにクリーク側の点 v_x がそれに隣接する a 個 $(a > 0)$ の独立点へ送信した後に別のクリーク側の点 v_y へ送信するような部分木を含む．

図 A·1 に示すように，上記の部分木に対して点 v_y へ送信してから a 個の独立点へ送信するように，点の配信順序を入れ換える操作を施す．この操作により，点 v_y は入換操作前に比べて a ステップ早くデータを受信する．更に，点 v_y の子孫であるすべての点も，入換操作前に比べて a ステップ早く受信する．一方，a 個の独立点はそれぞれ 1 ステップずつ遅くデータを受信することになる．しかし，点 v_x が送信する点数は $NC_S(v_x)$ 個であり，入換操作の前後で変わらない．そのため，点 v_x から受信する最後の点 $CH_S(v_x, NC_S(v_x))$ の受信タイミングは変わらない．したがって，入換操作を施したブロードキャスト木も最適である．

図 A·1 の左側のような部分木がなくなるまで入換操作を繰り返せば，すべてのクリーク側の点において式 (1) を満たす最適ブロードキャスト木を得る．□

2. 定理3の証明

（証明） k_0 が $A(k) \geq |V|$ となる最初の k であるとする．つまり $A(k_0-1) < |V|$ とする．$k \leq k_0 - 1$ であるすべての k で式 (8) を満たすから，定理2より $k_0 - 1$ ステップ目まで破綻しないスケジュールが存在する．更に，補題2より $|V_{k_0-1}| = A(k_0-1)$ となるスケジュール S が存在する．このスケジュール S において，k_0 ステップ目の配信を考える．$k_0 - 1$ ステップ目まで破綻しないから，k_0 ステップ目開始時にデータ送信が可能な点は最大で $A(k_0-1) - |\{v|e(v) \leq k_0-1\}|$ 個存在する．また，k_0 ステップ目開始時に未受信の点は $|V| - A(k_0-1)$ 個存在し，そのすべてが $e(v) \geq k$ を満たす．$A(k_0) \geq |V|$ であることから，式 (7) より

$$A(k_0-1) - |\{v|e(v) \leq k_0-1\}| \geq |V| - A(k_0-1)$$

が成り立つ．つまり，未受信の点の個数は送信可能な点の個数以下である．したがって，k_0 ステップ目にす

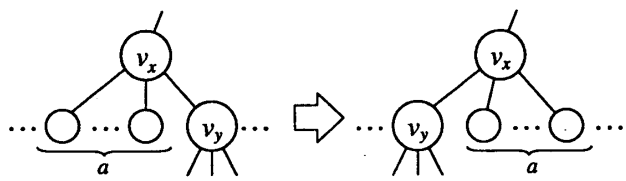


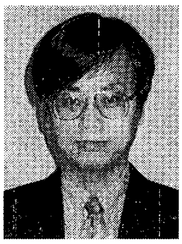
図 A·1　点の入換操作
Fig. A·1　The exchange operation between a vertex in clique and independent vertices.

べての未受信点へ配信することができるので，ブロードキャストは K ステップ以内に完了できる． □

（平成 15 年 6 月 5 日受付，10 月 24 日再受付，
16 年 1 月 15 日最終原稿受付）

田崎　太　（正員）

平 6 新潟大・工・情報卒．平 8 同大大学院工学研究科修士課程了．平 16 新潟工科大大学院博士課程了．工博．現在，日本精機（株）R&D センター勤務．グラフ理論とその応用の研究に従事．



田村　裕　（正員）

昭 57 新潟大・教育卒．平 2 同大大学院自然科学研究科博士課程了．学術博．同年同大学院自然科学研究科助手．現在，新潟工科大学教授．グラフ理論とその応用の研究に従事．平 3，7，9 年度本会論文賞．IEEE，情報処理学会各会員．

仙石　正和　（正員：フェロー）

昭 42 新潟大・工・電気卒．昭 47 北大大学院博士課程了．工博．同年北大・工・電子助手．新潟大・工・助教授を経て，現在，同教授．回路網理論，グラフネットワーク理論，情報伝送特に移動通信の研究に従事．平 3，7，8，9 年度本会論文賞，IEEE ICNNSP'95 最優秀論文賞．著書「演習グラフ理論」（共著）．IEEE Fellow，情報処理学会会員．

篠田　庄司　（正員：フェロー）

昭 39 中大・理工・電気卒．昭 48 同大大学院理工学研究科電気工学専攻了．工博．現在，同大学理工学部電気電子情報通信工学科教授．回路，ネットワーク，システムの解析，診断，制御の研究に従事．平 3，8，9 年度本会論文賞，IEEE ICNNSP'95 最優秀論文賞．著書「回路論入門（1）」ほか．IEEE Fellow．