Title	超並列・分散コンピュータネットワークにおける並列計算機モデルと設備配置に関する研究
Author(s)	當山, 孝義
Citation	
Issue Date	1998-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/862
Rights	
Description	Supervisor:堀口 進, 情報科学研究科, 博士

博 士 論 文

超並列・分散コンピュータネットワークにおける
並列計算機モデルと設備配置に関する研究

指導教官　堀口 進 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

當山 孝義

1998 年 1 月 16 日

要旨

近年，最先端科学技術分野では大規模計算が行われるようになり，多数の高速プロセッサからなる超並列計算機や分散ネットワーク上の多数の計算機を利用した超並列・分散処理が注目されている．しかし，並列計算機の物理的制約を十分に考慮した実用並列計算モデルがなく，各機種独自のプログラム開発が必要であり，超並列計算機は高速計算機の主流であるベクトル型計算機の代替としてそれを駆逐するまでには至っていない．また，並列計算機やネットワーク自体の性能およびコストパフォーマンス向上が要求されている．そこで本論文では，超並列・分散コンピュータネットワークにおける実用的な並列計算モデルと高性能でコストパフォーマンスが高いネットワークの設備配置法について議論する．

並列計算機モデルについては，Culler らの提案した並列計算モデル LogP に対して実用的側面からの検討を行い，通信路のバッファ動作を考慮した新しい実用並列計算モデル LogPQ を提案する．LogPQ モデルのエミュレータを構築し，各種の並列アルゴリズムの並列処理効率を詳細に解析する．また，LogPQ モデルの実用性を検証するために，実在する商用並列計算機である CM-5 上で並列アルゴリズムを実行し，LogPQ モデルにより実際の並列計算機の物理的制約を考慮した詳細な並列処理効率の解析ができることを示す．

一方，超並列・分散ネットワークに関しては，木構造ネットワークに注目し，高速通信設備の最適配置に関して検討を行う．先ず，設備の構築コストを低減する同一の小サイズの部品を用いて構築される設備や，複数の通信に対応した同一サイズの設備で構成される複数設備を提案し，木形状設備を適正に配置することにより高性能な木構造通信ネットワークが構築できることを明らかにする．次に，ネットワークの各二点間の平均通信コストを表す評価指標である全対距離和を定式化し，その設備配置方法を示す．また，設備内の通信コストを考慮した評価指標である全対コストを定式化し，設備の高速化率が一定の場合についてその設備配置方法を示す．その結果，高性能でコストパフォーマンスの高い通信ネットワークシステムが構築できることを明らかにする．最後に，より実用的な高性能ネットワーク構築法についての議論のため，ネットワークの各通信路に設備構築コストと通信コストを付与した実用設備配置問題を定式化し，その設備配置方法を提案する．

目 次

図 目 次

表 目 次

第 1 章

序論

1.1 研究の背景

1940 年代に最初の電子計算機 ENIAC が開発されて以来，計算機は，高性能化・高機能化を目指して様々な改良が加えられてきた．その結果，その用途は急速に広がり，より高度な処理を行うようになった．当初は集計処理や弾道計算といった単純な処理に用いられてきたが，性能向上と共に一般の事務処理や科学技術計算に用いられるようになり，現在では統合情報システムや人工知能，仮想現実などの高度な処理にも用いられている．今日の技術社会では，計算機は産業基盤としてなくてはならないものになっている．

現在まで，計算機の高性能化はデバイス技術の進展によるものが中心で，アーキテクチャ面ではノイマン型が守られてきた．デバイスは，当初の真空管や磁気メモリなどから，トランジスタ，IC，LSI，VLSI，ULSI へと速度と集積度を飛躍的に向上させ，計算機の処理速度や記憶容量の向上に寄与してきたが，アーキテクチャはほとんど変化せず，旧来の構造が用いられてきた．しかし，現在ではデバイス技術が技術的限界に近付き，逐次処理方式に基づく従来のノイマン型計算機では性能向上が望めなくなりつつある．

計算機技術の一つとして，1950 年代後半より並列計算機が研究されてきた．1960 年後半に 64 プロセッサからなる ILLIAC IV が開発されて以来，様々な並列計算機の研究および開発がなされてきた．近年、半導体集積回路技術の進歩により，多数のプロセッサを一つの計算機内に置くことが可能になってきた．WSI(Wafer Scale Integration) 技術の進展目覚ましく，数百〜数千のプロセッサを一つのウェハ内に，計算機全体では数万個のプロセッサを持つことができる．現在，膨大な数のプロセッサからなる超並列計算機は，ベクトル型

のスーパーコンピュータに代わるものとして，先端科学技術分野において期待されている．

近年，さまざまな超並列計算機が開発され実用化されている．しかし，現在の高速計算機の主流であるベクトル型計算機の代替としてそれを駆逐するまでには至っていない．この一因として，現在の超並列計算機は各機種独自のプログラム開発が必要なことが挙げられる．超並列計算機におけるプログラム開発は，各種のパラメータや物理的制約のために効率のよい並列プログラムを作ることが困難である．逐次的なプログラムの並列化という手法もあるが，元の問題の持っている並列性を一旦無にしたプログラムから並列性を再抽出するというのは無駄の多い手法であり，また，十分な並列性を得るのは困難である．現在，並列計算機の物理的制約を考慮した並列アルゴリズム開発のための新しい実用並列計算モデルが必要とされている．

一方，最先端科学技術分野では大規模計算を高速に行うため，並列計算機システムに必要とされる能力が著しく増大し，多数のプロセッサとネットワークから構成される超並列計算機自体の性能およびコストパフォーマンス向上が要求されている．また，ネットワーク上における多数の計算機による超並列・分散計算環境は，ネットワークコンピューティングとして非常に注目されている．これらの分野では，高性能でコストパフォーマンスの高い超分散ネットワークが必要とされている．

そこで本論文では，超並列・分散コンピュータネットワークにおける，実用的な並列計算モデルと高性能でコストパフォーマンスが高いネットワークの設備配置問題について議論する．

1.2 研究の目的

本研究の目的の一つは，並列計算機における一般性のある並列プログラム開発環境を構築するための基礎技術となる実用並列計算モデルを提案することである．並列計算機の登場以来，理論的な並列計算モデルである PRAM モデルに，現実性を考慮した様々な制約を付加する試みが行われてきた．Mehlhorn らは 1984 年に共有メモリを分割した MPC モデルを提案した．Kruskal らは 1986 年に共有メモリを除き局所メモリのみとした DCM モデルを提案した．Cole らは 1989 年に非同期のモデル化である APRAM モデルを提案した．Aggarwal らは 1989 年に共有メモリと局所メモリを持ち共有メモリに通信バンド幅を付加した LPRAM モデルや，共有メモリに通信遅延を付加した BPRAM モデルを提案した．Vaidyanathan らは 1992 年にワード長制限を加えた MPRAM モデルを提案した．他

にも BSP モデルなどさまざまな並列計算モデルが提案されてきた．しかし，これらはモデル能力の観点を重視して検討されており，現実の並列計算機動作を重視したものではなかった．

Culler らは 1993 年に実用的な並列計算モデルとして LogP モデルを提案した．これは，複数の RAM（Random Access Machine）とその間の通信路に物理的制約を反映した 4 パラメータ（L，o，g，P；それぞれ通信レイテンシ，通信オーバヘッド，通信ギャップ，プロセッサ数を表す）を付加したモデルである．しかし，LogP モデルに関する詳細な検討は十分にはなされていない．特に，並列計算機の効率的な並列プログラム開発という観点における LogP モデルの実用性については不明なところが多い．

本論文では，実用的側面から LogP モデルの詳細な検討を行い，LogP モデルでは並列アルゴリズムを記述するには不十分な点のあることを示す．この検討結果に基づいて，LogP モデルにおける通信路をキュー（Queue）の結合で表す新しい実用並列計算モデル（LogPQ）の提案を行い，LogPQ モデルの有用性について詳しく検討する．また，LogPQ モデルや商用並列計算機 CM5 上で並列アルゴリズムの実験的評価を行うことにより，LogPQ モデルにより並列アルゴリズムの詳細な性能評価が可能であり，LogPQ モデルの実用性が高いことを示す．

本研究のもう一つの目的は，高性能でコストパフォーマンスの高い通信ネットワークシステムを構築するための設備配置問題の定式化とその実用設備配置法を提案することである．ネットワーク上の設備配置問題に関する研究は，1960 年代より多くの研究者や技術者によって行われてきた．初期には，ネットワークに点形状の設備を配置する研究が行われた．木構造ネットワークに対する設備配置，一般のネットワークに対する設備配置，各種評価関数を用いた設備配置，複数個の設備の配置，階層的な設備の配置などのさまざまな研究が行われた．また，設備配置手法として，動的プログラム法などの基本的手法，branch-and-bound 法や焼きなまし法などのヒューリスティック手法，primal-dual 法などの近似手法を用いた方法が提案されてきた．

近年の超並列・分散計算機システムの実用化にともない，高性能でコストパフォーマンスの高い通信ネットワークシステムの構築技術として，ネットワークにパス形状や木形状の設備を配置する研究の重要性が指摘されている．木構造ネットワークにおけるパス形状や木形状の設備配置に関する研究は Slater により始められた．Morgan らは 1980 年に頂点数 n のネットワークの，各頂点と設備との距離の総和（距離和）が最小となるパス形状設備

を算出する $O(n)$ 時間のアルゴリズムを示した．Slater は 1981 年以降に木構造ネットワークにおける各種のパス形状や木形状の設備配置を行った．Minieka は 1985 年に距離和を最小とするサイズ指定のパス形状と木形状設備を求める各々$O(n^3)$ 時間，$O(n^2)$ 時間のアルゴリズムを提案した．Tamir らは 1992 年に一般化した評価関数を用いて p 個の木形状設備の配置を行う $O(n^3p^2)$ のアルゴリズムを提案した．Hakimi らは 1993 年に木構造や一般のネットワーク上の各種のパス形状や木形状設備の配置に必要な計算時間を示した．その他，多くの研究者や技術者によりさまざまな研究が行われている．しかし，従来の多くの研究は，製造工学・交通工学・経営工学的側面を重視したり，理論的側面に偏った研究であった．現在のインターネットやマルチプロセッサなどの，超並列・分散計算機システムの通信ネットワークシステム構築に重点を置いた研究は今まで十分になされていなかった．

本論文では，通信ネットワークとして木構造ネットワークに注目し，コストを考慮した高性能ネットワーク構築のための最適設備配置法について議論する．設備配置により通信ネットワークを高速化するには，通信ネットワークの大きさに合わせたサイズの設備が必要となる．通信ネットワークシステムの構築コストを低減するには，設備の構築コストを削減する必要がある．そこで，設備の構築コストを低減する，同一小サイズの部品を用いて構築される設備（等分割可能設備）を提案し，その構築方法について議論する．次に，並列・分散計算機システムでは複数の通信が同時に実行されることに注目し，同一サイズの複数個の設備からなる複数設備配置問題を定式化し，その最適設備配置方法について議論する．また，設備内の通信コストに注目し，二点間の通信の平均通信時間を表す評価指標である全対距離和を提案し，全対距離和が最小となる設備配置について議論する．更に，設備内の通信コストを重視した，設備を配置する辺の長さを削減するモデルを提案し，その最適設備配置問題について議論する．最後に，通信ネットワークの各通信路における設備の構築コストと通信コストが異なる実用設備配置を定式化し，その最適設備配置問題について議論する．

1.3 本論文の構成

本論文の構成は全 6 章より構成されている．

2 章では，実用並列計算モデル LogPQ の提案と評価を行う．先ず，超並列計算機に対する並列計算モデルの必要性を議論する．次に，Culler らの提案した並列計算モデル LogP の検討を行い，並列計算機の効率的なプログラム開発を行うには不十分な部分のあること

を示す．そこで，通信におけるバッファ動作，通信の集中，通信メッセージのサイズを考慮に入れた実用並列計算モデル LogPQ を提案する．また，LogP モデルと LogPQ モデルの関係について詳しく検討し，LogPQ モデルにより十分に効率的な並列プログラムを開発できることを明らかにする．

3 章では，実際の並列計算機上の並列アルゴリズムの動作解析を行い，LogPQ モデルの実用性と有用性を評価検討する．先ず，並列計算モデル LPRAM で並列整数 GCD アルゴリズムの動作解析を行い，実用性が不十分であることを明らかにする．次に，LogPQ モデルで並列多倍長 GCD アルゴリズムの動作解析を行い，実用性を重視した動作解析ができることを示す．また，超並列計算機 CM-5 上で並列乗算を行い，LogPQ モデルが LogP モデルより実際の性能を詳細に検討できる実用的なモデルであることを示す．

4 章では，超並列・分散計算機システムの通信ネットワークにおける設備配置について議論する．先ず，木構造ネットワーク上での設備の最適配置について具体例を用いて議論する．そして，木構造ネットワーク上での超並列・分散計算機システムの構築に適した，実用性を重視した設備配置法について議論する．次に，通信ネットワークシステムの構築コストの低減を目的とした，等分割可能な木形状設備の配置を提案し，その配置方法を示す．最後に，通信量の多い通信ネットワークシステムに対応した，複数個の同サイズ木形状設備の配置を提案し，その配置方法を示す．

5 章では，超並列・分散計算機システムの通信ネットワークにおける，設備内の通信コストを考慮した設備配置について議論する．先ず，木構造ネットワーク上での各二点間の通信の平均通信コストを表す評価指標である全対距離和を提案し，全対距離和が最小となる設備の配置方法を示す．次に，木構造ネットワーク上での設備内の通信コストを考慮した評価指標である全対コストを提案し，設備の高速化率が一定の場合の，全対コストが最小となる設備の配置方法を示す．最後に，木構造ネットワークの各辺に設備の構築コストと通信コストを付与する実用設備配置を提案し，評価関数が最小となる設備の配置方法を示す．

6 章では，本論文の結論を示す．

第 2 章

超並列計算機モデル

2.1　並列計算モデルの必要性

近年，様々な超並列計算機が開発され実用されている．しかし，現在の高速計算機の主流であるベクトル型計算機の代替としてそれを駆逐するまでには至っていない．この一因として，超並列計算機におけるプログラム開発は，各種のパラメータや物理的制約のために効率のよい並列プログラムを作ることが困難なこと，また各機種独自のプログラム開発が必要なことが挙げられる．そこで，並列計算機の物理的制約を考慮した並列計算モデルを用いた並列アルゴリズム開発の重要性が高まっている．

従来，一般に，Fortune らの提案した PRAM（Parallel Random Access Machine）モデル [1] が利用されてきた．PRAM モデルは，並列処理の一般性を最も有する並列計算モデルの一つである．しかし，実際の並列計算機の物理的制約をほとんど反映しておらず，その動作は実機と乖離しがちであった．そこで，各種の物理的制約を付加する試みが行われてきた．Mehlhorn らは共有メモリを分割した MPC モデル [2] を提案した．Kruskal らは共有メモリを除き局所メモリのみとした DCM モデル [3] を提案した．Aggarwal らは共有メモリと局所メモリを持ち共有メモリに通信バンド幅を付加した LPRAM モデル [4] を提案した．

一方，並列計算機のハードウェア構成を直接に反映した並列計算モデルが提案されてきた．nCUBE2 や CM-5 等の商用超並列計算機はメッセージパシング型計算機であり，これに適合した並列計算モデルとして，プロセッサ間通信を送信命令と受信命令の組で示した Send-Recieve モデルがある [5][6]．また J-Machine は，各プロセッサがメッセージ受信に対

し処理を行う Message Driven モデルを用いている [7]．Eicken らは，メッセージ受信に対して割込みによるハンドラ実行を行う Active Message モデルを提案した [8]．

近年，多くの商用機が登場し，超並列計算機が一般的に使用されるようになった．そして，実用化の時代を迎えた今，特定の計算機に特化しない一般性を持った，並列計算機アプリケーションの開発環境が必要とされている．そのためには，並列環境としての OS や言語のみならず，並列計算モデルの一般性を保持し実際の並列計算機の物理的制約を反映させることのできる実用的な並列計算モデルが必要とされている．

Culler らは，実用的な並列計算モデルとして LogP モデルを提案した [9]．これは，複数の RAM(Random Access Machine) とその間の通信路よりなるもので，物理的制約を反映した 4 パラメータを付加したものである．並列計算モデルの一般性を保持しつつ、実用性を重視している．しかしながら，LogP モデルに関する詳細な検討は行われておらず，その実用性については十分には明らかにされていない．そこで，次節で LogP モデルの実際的側面を詳細に検討する．

2.2 LogP モデル

2.2.1 LogP モデルの概要

LogP モデルは，分散メモリ型並列計算モデルであり，各プロセッサはメッセージ通信によりコミュニケーションを行う．LogP モデルは，メッセージ通信を通信遅延 L，通信オーバヘッド o，通信バンド幅 g，プロセッサ数 Pの 4 パラメータで特徴付ける．図 2.1にメッセージ通信動作を示す．ここでは LogPQ モデルとの混乱を避ける為，L^*, o^*, g^*と表わす．L^* は元プロセッサから先プロセッサへのメッセージ通信における遅延の上限である．o^*はプロセッサがメッセージの送受信処理の為に他命令を実行できない時間である．g^*はプロセッサが連続してメッセージを送受信できる最少間隔時間であり，プロセッサ当りの通信バンド幅に対応する．Pはプロセッサ数である．プロセッサは各局所命令を 1 単位時間（1 クロック）で実行する．L^*, o^*, g^*パラメータはクロックを単位として表わされる．

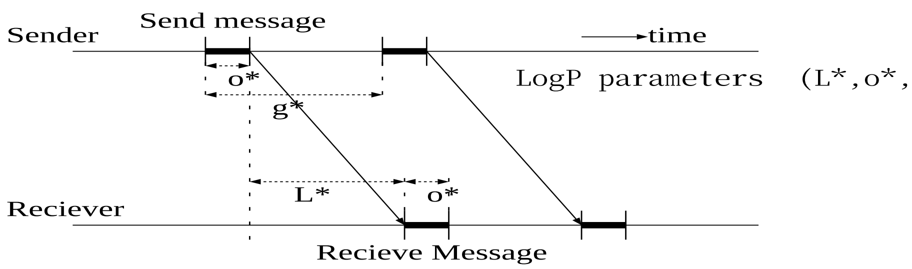


図 2.1: LogP モデルの通信動作

2.2.2 LogP モデルの問題点

Culler ら [9] は，今まで提案されてきた MPC，DCM，LPRAM モデルなどと比べて LogP モデルの実用性が高いことを指摘している．LogP モデルは，従来の並列計算モデルで不十分であった並列計算機の通信路の制約を 3 つのパラメータ L^*, o^*, g^*に限定し，並列アルゴリズム解析に必要なモデルの簡潔さを有している． また，実際の超並列計算機に対する LogP パラメータの対応性を検討し，いくつかのアルゴリズムの LogP モデル上での検討を行い LogP モデルの有用性を指摘した．しかし，LogP モデルには実用性の観点からは以下に示す問題点がある．

（1）バッファ動作

LogP モデルは，複数のメッセージ送受信を行うとき，その命令の不連続実行を前提としている．しかし，実際の超並列計算機はバッファを有し，連続実行を実現できるものが多い．

すなわち，各プロセッサの送受信部分の高度化により実質的なギャップ gの削減を実現する．$g = o$ になると，プログラム上は送信命令や受信命令の連続実行が可能となり，アルゴリズムのより柔軟なインプリメンテーションが可能になる．これは，計算機ハードウェアの非同期式演算パイプラインと同様である．パイプライン動作において，各段の間にバッファを関与させることにより，より柔軟にその動作検討が行える． すなわち，通信量が通信バンド幅に対して小さい場合，送信側プロセッサのプログラムは，他プロセッサが必要とするデータを作成してからそれを送信するまでの時間間隔に，通信バンド幅を考慮する必要がない．LogP モデルでは，送信プロセッサのプログラムは，送信するデータを作成

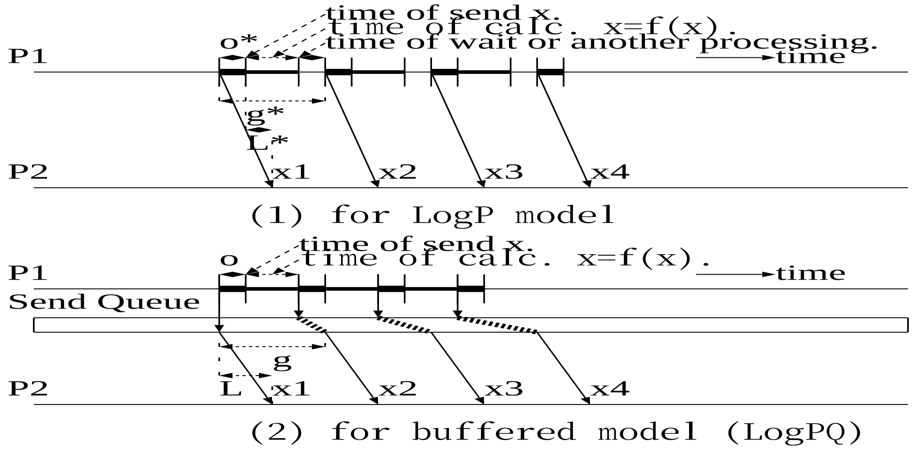


図 2.2: 繰り返し通信の動作

後，前回の送信から g以上の時間間隔を確保すべく，送信を後送りする必要がある．場合によっては，プログラムはそのデータを別領域にコピーしたり，もしくはプログラムの各 iteration において使用するデータ領域をインターリーブする必要がある．例えば，図 2.2 は，プロセッサ 1 がプロセッサ 2 に $x1, x2, ...(x_{i+1} := f(x_i))$ の送信を行う場合を示している．通信バッファを考慮したモデルでは $x := f(x)$ の計算と x の送信とを単純に交互に行えば良いが，LogP モデルは $x := f(x)$ の計算後，次の送信が可能となるまで待つ必要がある．この処理を効率的に行うには，プログラムは待ち時間に何らかの別処理を行う必要がある．この別処理の埋め込みは，処理間での独立性の確保を必要とし，プログラムを複雑にする．実用的モデルにおいて，実際に存在するこの柔軟性の使用は重要と考えられる．

（2）通信集中

一般に，同時に複数のプロセッサがあるプロセッサにメッセージ送信を行うとき，通信路にボトルネックが生じる．LogP モデルは，この制約を同時に同じプロセッサに到着するようなメッセージ送信の禁止で表現するため，nCUBE2 等の可能な超並列計算機の非同期動作をモデル化することが困難である．

（３）メッセージのデータ長

LogP モデルは，様々なデータ長のメッセージ通信を行う場合に，各データ長毎に L, o , gパラメータが必要である．従って，モデルのパラメータ数は増加し，通信データ長とモデル動作の関係が各データ長のパラメータ間の関係に隠蔽され，不明確になる．このため，LogP モデルでは並列アルゴリズムの詳細な動作解析が困難になる．

一方，メッセージのデータ長を一定値，たとえば 1 ワードと規定し，様々な大きさのデータの通信を複数回のメッセージ通信で行う場合，通信のオーバヘッドは主に gにより定まる．すなわち，m ワードのデータを通信する場合のオーバヘッドは $o^* = (m-1)g + o$ となり，m が大きい場合には gが主に影響する．これは，通信オーバヘッドのデータ長比例成分を，通信ギャップと独立に表せないことを意味する．文献 [10] における通信では，通信オーバヘッド o^*のデータ長比例成分は，通信路のギャップ gでなく通信インタフェースのオーバヘッド o である．LogP モデルは，このような通信路の制約を反映することができない．

次節では，これらの問題点を解決するため，通信路の制約にバッファ機能を有する LogPQ モデルを提案する．

2.3　LogPQ モデル

LogPQ モデルは，LogP モデルに対して通信路にキューパラメータを追加することで実用性を向上させたものである．図 2.3に，LogPQ モデルを示す．LogPQ モデルは，通信路端に送信キュー（Send Queue）および受信キュー（Recieve Queue），通信路の複数プロセッサからの集結点に転送キュー（Transfer Queue）を導入し，各キュー長の上限として SQ，RQ，TQ の 3 パラメータを追加する．図の transfer interval は通信路のバンド幅を，transfer latency および macro capacity は通信の集中がない場合の送受信プロセッサ間の遅延および通信路容量を示している．

LogPQ モデルは，通信路にキューを付加することで，実際の通信ネットワークのバッファ動作や通信集中の影響を反映することができる．また，通信パラメータを 1 ワードメッセージに対して構築することで，通信メッセージサイズの通信遅延に対する影響を反映する．更に，キューパラメータで与えられる制限による，通信路のバッファ動作の明示的なオーバーフロー制御により，実用的な通信処理の解析が可能である．

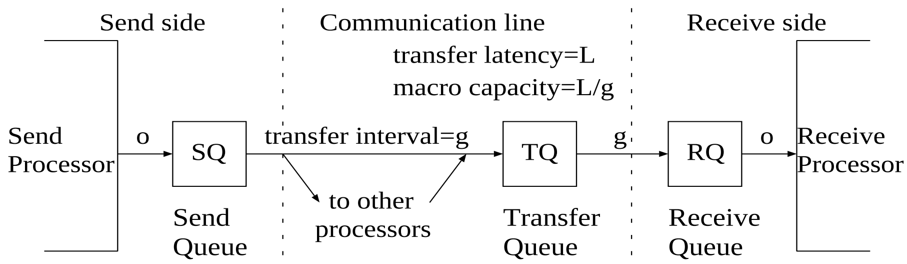


図 2.3: LogPQ モデルの構造

2.4 LogPQ モデルの評価

2.4.1 通信路との関係

LogPQ モデルは，LogP モデルと同様に，通信路の通信動作に密接に関連している．図 2.4は，LogPQ モデルの通信動作を示す．LogPQ モデルはワード単位の処理を基本とする．商用計算機でプロセッサの 1 ワード長と通信路の 1 ワード長が異なる場合がある．このとき，物理的な通信ワードの通信動作をパラメータ L_0, o_0, g_0で表わすと，LogPQ パラメータ L, o, gの値は以下のようになる．

$$
\begin{aligned}
L &= L_0 + o_0 + (r-1)\cdot g_0 \\
o &= o_0 + (r-1)\cdot g_0 \\
g &= r\cdot g_0
\end{aligned}
\tag{2.1}
$$

ただし，rはプロセッサのワード長 w_pを通信路のワード長 w_cで除したものである．L は送信命令開始時から受信可能になるまでの時間，o は通信命令の実行時間，gはデータ送信間隔を示す．L, o, gの単位はクロックである．LogPQ モデルでは，大きな（例えば m ワード）データの通信は，この 1 ワードのメッセージ通信の m 回繰り返しで実現される．パラメータ Pは，LogP モデルと同じくプロセッサ数を示す．

2.4.2 LogP モデルとの関係

LogP モデルでの m ワードのデータ長のメッセージの送信は，LogPQ モデルでは図 2.5に示すように m 回のメッセージ送信で表わすことができる．なお，図 2.5の L, o, gは LogPQ モデルのパラメータである．LogP モデルのメッセージ通信の開始/終了処理のオーバヘッ

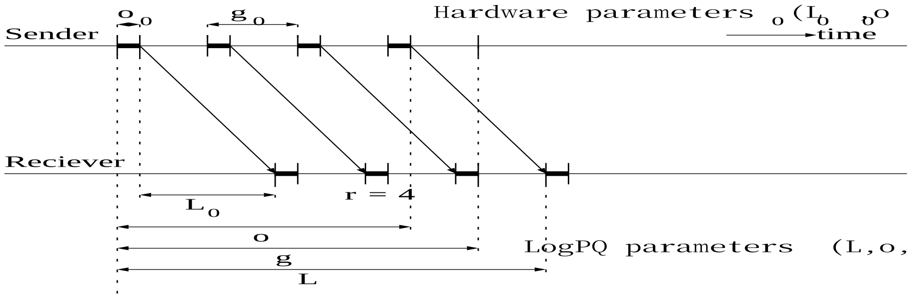


図 2.4: LogPQ モデルの通信動作

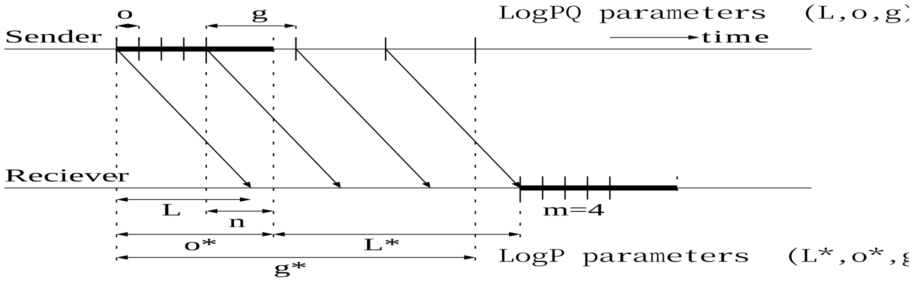


図 2.5: LogP モデルと LogPQ モデルの関係

ドは，図の n である．LogPQ モデルは，1 ワード通信動作を基本としているため，プログラム側で n のウエイトを付加する．このとき，LogP モデルのパラメータ L^*, o^*, g^*と LogPQ モデルのパラメータの間には以下の関係がある．

$$
\begin{aligned}
L &= L^* + o^* - g^*(m-1)/m \\
o &= (o^* - n)/m \\
g &= g^*/m
\end{aligned}
\tag{2.2}
$$

使用する通信メッセージのデータ長を一定値に固定した場合，メッセージ通信の開始/終了処理のオーバヘッドは，メッセージの各ワードに分散すると考えると，$n = 0$ となる．

LogP モデルの本質は，通信路を長さ L^*，開始レート g^*，端点のオーバヘッド o^*のパイプラインとして扱うところにある．従って通信路の有効利用は，g^*ごとに，すなわち時間的に一様な通信を行う必要がある．一般に，並列アルゴリズムは計算と通信の組合せで行われる．そして並列アルゴリズムには計算部分が一様でない場合がある．このときの通信

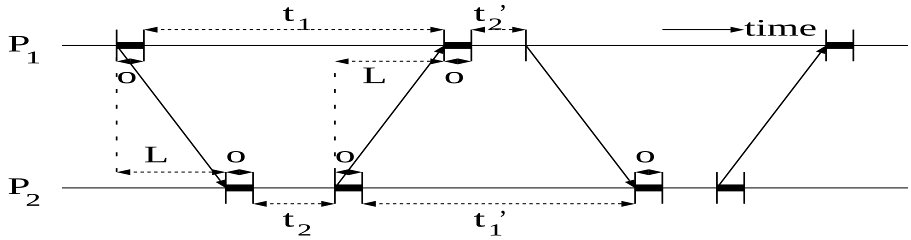


図 2.6: LogPQ モデルの繰り返し通信動作

路状況の表現は，並列アルゴリズムの効率化に重要である．

LogP モデルは，長いメッセージ通信に対して，モジュール内に計算用プロセッサと通信用プロセッサの 2 プロセッサを用いて，計算と通信のオーバラップが可能である．これにより，通信オーバヘッドはワード当り o^*で表わされる．しかし，2 プロセッサの存在はモデルを複雑にするし，送受信速度は最大 2 倍に制約される [9].

2.4.3 通信パラメータ

パラメータ L, o, gは，通信動作を以下のように規定する．LogPQ モデルは，L, gをプロセッサ間動作に関係するパラメータ，o をプロセッサ単独動作に関係するパラメータとする．また，L は通信路環境のパラメータ，gは通信路状況のパラメータである．

図 2.6は，繰り返し処理の動作に対する L, o, gパラメータの影響を示したものである．プロセッサ P_1と P_2は繰り返し処理を行っており，各繰り返しにおいて互いにメッセージ通信を行う．このとき，通信と計算のオーバラップは $t_1 = t_2 + 2L$ で最適となる．図の $t1', t2'$に関しても同様である．また，o は，両プロセッサの相対的な関係に影響を与えない．各プロセッサのプログラムが通信に要する総時間は，受信待ちがなければ，o に通信量を乗じたものとなる．最後に，gは，通信路のバンド幅に対応しており，各プロセッサの通信頻度に影響を与える．

2.4.4 キューパラメータ

パラメータ SQ, TQ, RQ は，通信動作を以下のように規定する．

LogP モデルでは，通信路容量は $\lfloor L^*/g^* \rfloor$ で，通信路の飽和は大きな負荷に対する遅延

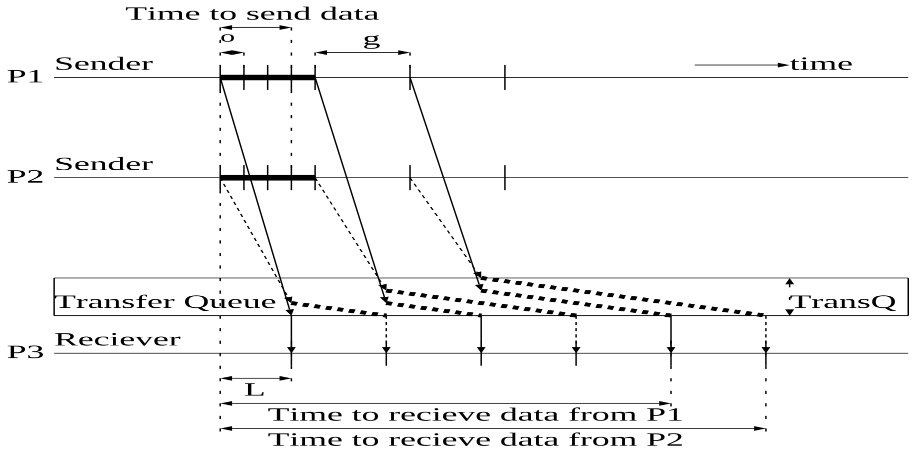


図 2.7: 2 プロセッサ ($P1, P2$) が同一プロセッサ ($P3$) にメッセージ送信した場合の通信動作

の急激な増加により起る．一方，LogPQ モデルは，通信路容量 $\lfloor L/g \rfloor$ を通信路の平均負荷に対するものとし，送受信キューの上限 SQ, RQ を負荷の偏差の許容量とする．なお，各キューの最大長が 1 の場合，通信路の飽和動作は LogP モデルに等しい．

通信路の平均負荷が小さい場合，送受信キューの上限 SQ, RQ は，連続して送/受信できるメッセージ数，例えば m ワードの大きさのデータを一度に送/受信したいと考えたときの m の上限を示す．またこれは，大きなデータを送/受信する場合の，長時間の通信路占有などの通信コスト上昇の反映を示す．データ長が SQ や RQ 以下の場合，プロセッサの通信オーバヘッドはワード当り o 時間であるが，それ以上の場合はワード当り g時間に増加し，通信コストは割高になる．また，RQ は，受信側プロセッサのプログラムにおける受信の後送りの制限を示す．送受信間隔を偏向させて一時的に送信間隔を受信間隔より小さくさせることは，プロセッサに待ち時間を生じさせないために良く用いられるテクニックではあるが，これを無制約に行うことはできない．例えば，非常に多くのメッセージを連続送信したり，受信処理を大きく後送りすることは，通信部分に非常に大きなバッファを必要となるため，一般に現実的でない．

LogPQ モデルは，多対多の通信の制約を通信キュー長で示している．各プロセッサは，小数のプロセッサからは多数のメッセージを連続して受信でき，多数のプロセッサからは小数のメッセージしか連続受信できない．なお，各プロセッサは各時点において，通信キュー長により通信遅延が定まる．

図 2.7は，2 つのプロセッサが同時に同じプロセッサに 3 ワードのデータを連続送信した場合の通信路の動作を示している．データ長が長くなるほど，通信キューに蓄積されるデータ数が増加し，通信キューからの出力に時間を要するようになり，送信されたデータを全て受信するまでの遅延が増加する．また，データ長が長くなると，通信キューのオーバフローが生じる．

LogPQ モデルは，メッセージ（データ長は 1 ワードである）の送受信に o 時間のオーバヘッドを必要とする．一方，Active Message モデルは，メッセージ受信に対して割込みハンドラを使用することにより，バッファを除去し，またハンドラが直接受信側のメモリをアクセスしてそれを送信側にリターンする即時リプライを可能にしている．また，通信用副プロセッサや DMA 等を用いることによる通信と計算の平行実行を行う場合，データ量に比例するオーバヘッドは発生しない． LogPQ モデルは，これらの動作を直接的に表現することはできない．しかし，LogPQ モデルにおいても，この動作をある程度考慮することができる．

通信部分と計算部分の平行実行は，以下のように考えることができる．送信側プロセッサから受信側プロセッサに通信されるメッセージは，当然ながら，送信側プロセッサがある時点で作成し，受信側プロセッサがある時点で使用するものである．もし，受信側プロセッサがそのメッセージを使用しないのなら，その通信は本来不必要なもので除去できる．また，受信側プロセッサがメッセージを他プロセッサへの送信にしか使用しないのならば，送信側プロセッサが直接それを他プロセッサに送信できるであろう．従って，送信側プロセッサが送信の為にデータをレジスタよりメモリにストアする代りにメッセージ送信を行い，受信側プロセッサが使用する為にデータをメモリよりレジスタにロードする代りにメッセージ受信を行うと考えると，通信と計算の平行実行は通信パラメータを $o = 1$ と設定することにより模倣できる．

また，バッファの除去は，キュー長が無制限になったものと考えることができる．

しかし，割込みハンドラからの即時リプライの考慮は困難である．即時リプライを，受信側プロセッサのプログラム上で実行する場合，プログラムは複雑になり，また即時のリターンは困難になる．

されど，メッセージ（m_2）を受信したプロセッサ（P_2）の受信ハンドラによる即時リプライは，m_2を送信したプロセッサ（P_1）が m_2を送信する直前に受信した P_2からのメッセージ（m_1）を P_2が送信した時点で，P_2 はハンドラが送信するメモリセルの内容を確定して

なければばならず，またその内容はリプライ実行まで保存されてなければばならない．この確実な確定の為に，両プロセッサ動作の同期が必要である．同期処理を含めて考慮すると，ハンドラからの即時リプライは，従来の方法で代替が可能な場合がある．

2.4.5　通信オーバヘッド

図 2.5の通信において，送受信プロセッサの通信オーバヘッドは $m \cdot o + n$，メッセージの送信開始からメッセージ全体が受信可能になるまでの時間は $L + (m-1)g$，送信プロセッサが送信処理に要する時間は $m \cdot g$となる．通信動作は，通信路の物理的制限のみにより規定される LogPQ パラメータおよびオーバヘッド n と，通信データ長 m で表わされる．

LogPQ モデルはオーバヘッドの送信データ長に比例しない部分を直接には考慮しない．この部分はプロセッサ間通信に固有なものではなく，通信インターフェース内部の問題である．すなわち，データ送信に必要なオーバヘッドは，送信データ長に比例した部分のみであり，それ以外はマルチスレッドやインテリジェントインターフェースの使用により避けることができる．事実，nCUBE2 や CM5 等の商用機に比べ，J-Machine 等の実験機では極力小さくなっている [9]．

2.4.6　通信動作例

図 2.8は，同期通信の様子を示す．同期通信は，通信完了まで送信プロセッサが次の処理を実行しないものである．LogP パラメータは 1 ワード通信を基準としている．LogPQ モデルは，同期動作を明示せず，その影響を L に内包した場合を示している．この場合，図のように，LogP パラメータに対する LogPQ パラメータが得られる．なお，$n = 2L^* + 5o^* + m(g^* - o^*)$ である．

プロトコル通信は，プロセッサ $P1$ から $P2$ に m ワードのデータ通信後，$P2$ から $P1$ に，そして $P1$ から $P2$ に同期のためのデータ通信を行うプロトコル動作（$P1$ は m ワード通信完了まで次の動作を実行せず，$P2$ は $P1$ より後に次の動作を実行する）である．図 2.9は，プロトコル通信におけるパラメータの関係を示す．LogP パラメータは 1 ワード通信を基準としている．LogPQ モデルはプロトコル動作をそのまま示している．この場合，図のように，LogP パラメータに対する LogPQ パラメータが得られる．

以上のように，LogP モデルにおける通信に対して LogPQ パラメータを定め，動作がほぼ一致する LogPQ モデルを構築することができる．

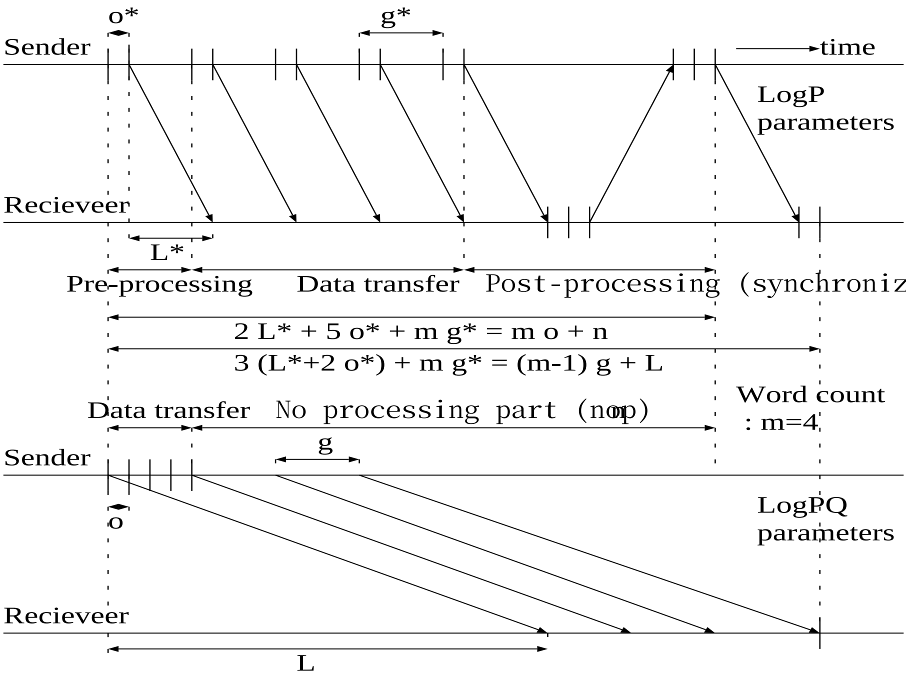


図 2.8: m ワード メッセージの同期通信動作 ($m = 4$)

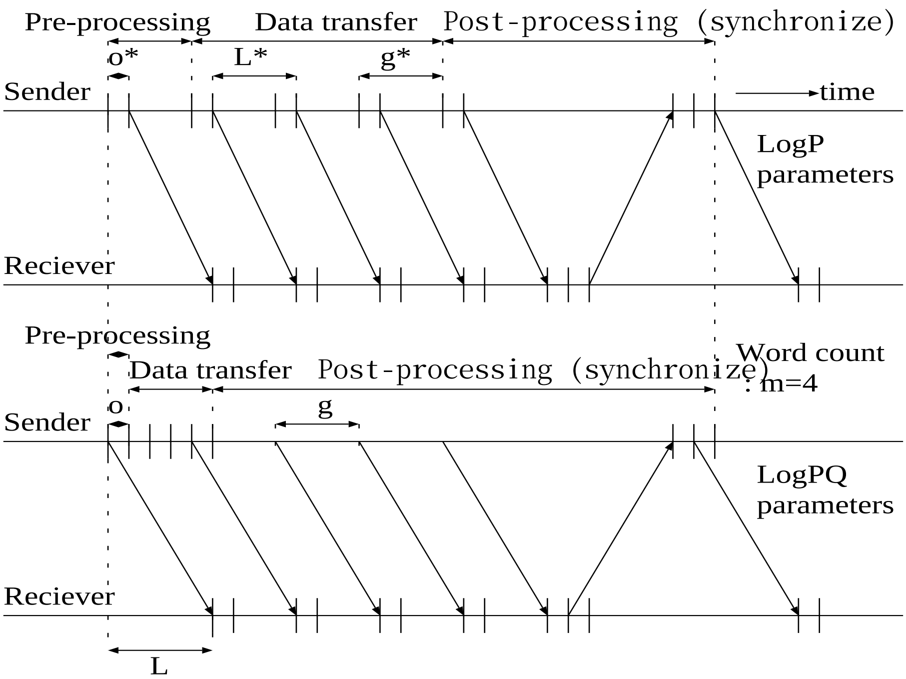


図 2.9: m ワード メッセージのプロトコル通信動作 ($m = 4$)

2.4.7 受信ハンドラ処理

LogPQ モデルでは，受信ハンドラ処理コストに関する検討を LogP モデルよりも詳細な形で行うことができる．受信ハンドラ処理コストを並列アルゴリズム検討の枠内に含める程度により，並列アルゴリズム検討の詳細度が定まる．

Active Message モデルは，通信ハンドラをアプリケーション側で指定することにより，そのアプリケーションの通信動作に最適な受信ハンドラ処理を行うことが可能である．例えば，あるプロセッサにメッセージ送信する送信元プロセッサが既に明らかならば，そのプロセッサの受信ハンドラ処理において送信元の調査等を省略でき，その処理削減による通信コスト削減が可能である．同じハンドラ処理をソフトウェアで行う場合，ハードウェアで行う場合に比べ処理時間が大きく通信コスト増となる．しかし，ハンドラ処理をハードウェアで行う場合，そのハンドラは様々な通信動作に対応する必要がある．ソフトウェアで行う場合，アプリケーションに合せたハンドラ処理により効率的なハンドラ実行が可能となり，システムを柔軟に設計できる [8].

本論文では，次に示す 3 種類の受信ハンドラ処理について検討する．

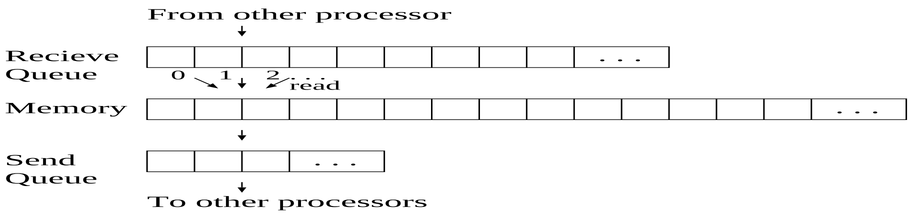
（1）単純通信

図 2.11（a）は，複数プロセッサからの受信メッセージをそのまま受信キューに蓄積し，受信命令で受信キューの先頭から順に受理する単純通信動作を示す．このとき，受信ハンドラ処理はアプリケーション側で実行され，ハンドラ処理コストはアプリケーション実行時間に包含される．

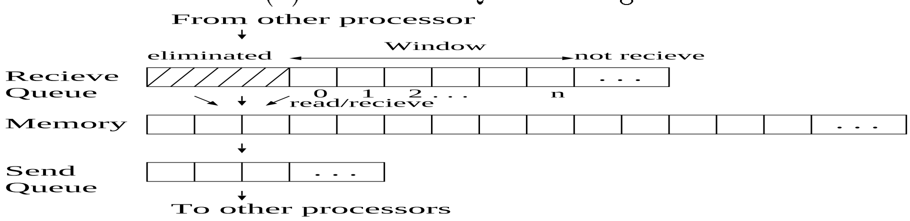
すなわち，乱雑な通信に対してはそのハンドラ処理は複雑になり，その処理をプログラムで明示的に行うことにより大きなコストが課される．単純な通信に対しては，ハンドラ処理のコストは小さい．なお，L, o, gパラメータにはハンドラ処理コストは付加されない為，各パラメータは小さな値になる．

（2）送信元プロセッサ指定

図 2.11（b）は，受信命令で受信メッセージの送信元プロセッサ指定を行う場合の動作を示す．これは,（単純な）受信ハンドラ処理をハードウェア（または通信インターフェース部）が行ない，そのコストを L, o, gパラメータ内に付加したものである．このとき，多種類の通信を受信する場合の受信ハンドラ処理コストの増加を考慮できないが，より単純に



(1) Without RQ restricting.



(2) With RQ restriction.

図 2.10: LogPQ モデルの RQ 制限

並列アルゴリズムを検討できる．

乱雑な通信の場合，そのハンドラ処理コストは大きく，L, o, gパラメータは大きな値になる．そのようなパラメータ値を基準として実用アルゴリズムの検討を行うことは不合理な場合がある．例えば，プロセッサ数が非常に多い場合，受信メッセージの送信元プロセッサ指定による検索は，大きなハードウェア使用を伴わない限り通信コストが大きくなる．実用性を考慮すると，通信の乱雑さに制限があることを前提とした L, o, gパラメータ，すなわち削減されたパラメータ値を用い，その条件下で実用アルゴリズムの検討を行うことが実用上適当である．

なお，受信キュー長の上限 SQ は受信メッセージの総数の上限である．SQ が各送信元プロセッサ毎のメッセージ数制限の場合，プロセッサ数が多いと各プロセッサの受信キュー総量も比例して大きくなり，必要なハードウェア資源が多大になることから現実的でない．

（３）タグ指定

・タグ指定の受信動作

タグ指定受信命令の使用により，比較的高度な通信インタフェース部を持つ超並列計算機における，並列アルゴリズムの検討が可能である．

図 2.11（c）は，送信元プロセッサ指定の受信命令で，受理するメッセージとして受信

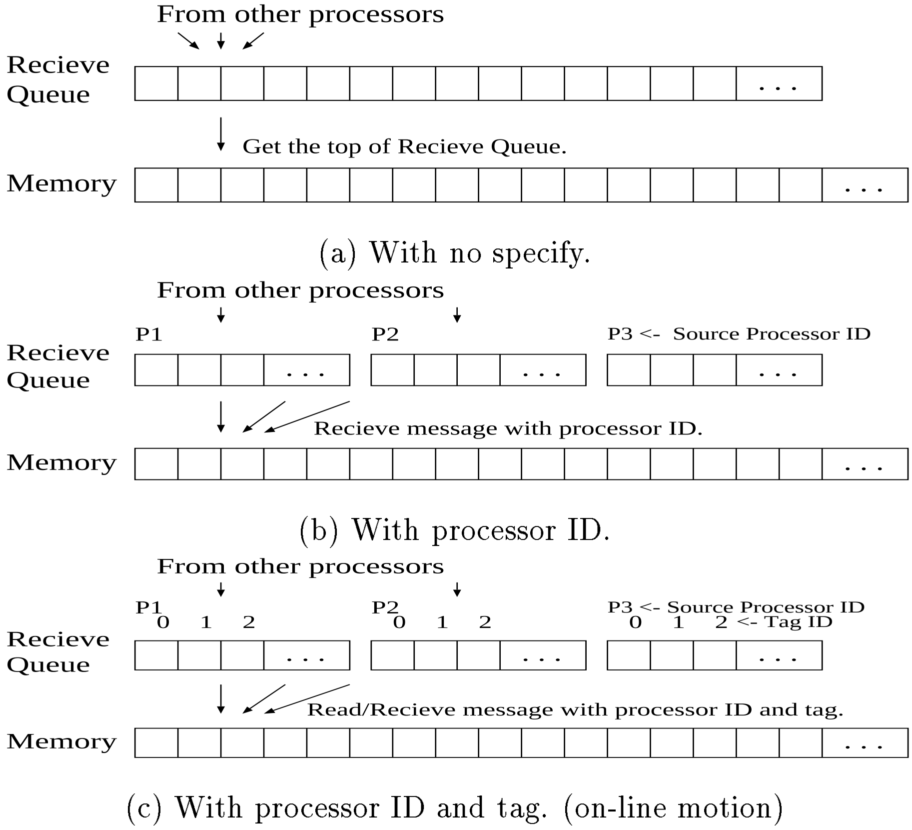


図 2.11: LogPQ モデルの受信部

キュー先頭のみでなく，その（受信キュー先頭からの相対的）データ位置をタグとして指定する場合の動作を示す．

これは，受信メッセージに一種の暗黙的なタグ名を与えた場合に相当する．すなわち，各プロセッサ間通信は，プログラム開始から何番目の通信データであるかを絶対番地とし，受信命令により受理した（そしてキューより除かれた）データ数との相対値を受信命令において指定する．これは，送受信プロセッサ間の送受信データ順序の入れ換えを受信ハンドラ処理に含めたものであり，最も単純なタグ付きメッセージ通信形態である．

受信命令として，recieve 命令と read 命令を用意する．recieve 命令は通常の受信命令であり，受理するデータを受信キューより除去する．read 命令は，受理するデータをそのまま受信キューに保持する．このとき，受信キューはバッファとしての性格が明確になり，受信キュー長制限の意味が明確になる．

図 2.10は，タグ指定受信に対する受信キュー長制限動作を示したものである．図 2.10（1）

は，受信キュー長が無制限の場合を示す．このとき，recieve 命令の実行は必要ない．受信キューは ROM（リードオンリメモリ）となり，タグ値は ROM アドレスを示す．図 2.10（2）は，受信キュー長に制限 RQ を付加した場合を示す．RQ は，受信キューに到達したデータと除去されたデータの差，すなわち図の Window 幅の上限として表わされる．この制限を満す為に適切な recieve 命令実行が必要となる．

すなわち，受信キュー長制限がない場合，recieve 命令の実行は必要ない．このとき，受信キューは ROM（リードオンリメモリ）となり，タグ値は ROM アドレスを示すものとして扱われる．そして RQ は，受信キューに到達したデータと recieve 命令により除去されたデータの間に設定される窓幅の上限として表わされる．アプリケーションは，この上限を越えないように，recieve 命令の実行と，送信元プロセッサとのコミュニケーションを必要とする．

・メモリキャッシュ動作の対応

タグ指定の通信動作は，遂次計算機におけるメモリキャッシュ動作との対応付けが可能である．これにより，より詳細な実用アルゴリズム検討が可能である．

現実の計算機において，パラメータ L, gは，ハードウェアの制約により削減には限界がある．一方，パラメータ o は，通信経路の影響が含まれないことから，その削減は比較的容易である．そして $o = 1$ の場合，メッセージ受信は通常の命令と同時間（1 クロック）で実行され，受信キューはプログラム上テンポラリな高速バッファとしての役割を果す．すなわち，遂次計算機におけるメモリキャッシュと同様の動作が含まれる．

遂次計算機において，メモリキャッシュの動作をもアプリケーションが指示する詳細な動作検討，これはハードウェアで自動的にメモリキャッシュ操作を行う場合に比べて不要な実メモリライトが行われず高速であるが，このような厳密な高速性の追求が考えられる．タグ付きメッセージ受信は，これを超並列計算機へ拡張したものである．

各プロセッサは，局所メモリアクセスに数クロックを必要とし，局所メモリのリード結果を受信キューに保持できるとする．すると，局所メモリアクセスは他プロセッサへの送受信と同様になる．更に，受信キューの自プロセッサ部のみはライト可能とすると，この部分はメモリキャッシュに直接に相当する．

一般的に，遂次計算機はメモリキャッシュ処理をハードウェアで自動実行している．並列計算機においてこれに対応するのは仮想共有メモリにおけるメモリキャッシュ処理である．しかし，仮想共有メモリ処理は複雑であり，多プロセッサの並列計算機においてはそ

のコストは大きい．多数のプロセッサを有する超並列計算機のコストを考慮すると，ソフトウェアによるキャッシュ動作実行は不要なメモリアクセス動作がなく必要なハードウェア性能が小さい．

2.4.8 LogGP モデルとの比較

現在，さまざまな実用並列計算モデルが提案されている．LogP モデルを拡張したものに，Alexandov の提案した LogGP モデル [11] がある．LogGP モデルは，2 つの通信ギャップパラメータを用いることで，サイズの小さいメッセージとともにサイズの大きいメッセージの通信に対応する．すなわち LogGP モデルは，LogP モデルと同じレイテンシ L'，オーバヘッド o'，プロセッサ数 P' に加え，メッセージ間のギャップ g' とメッセージのバイト当たりのギャップ G' を用いた 5 パラメータで表される．メッセージサイズが m バイトの場合，二つのメッセージ間の最小送受信間隔は $g' + (m-1)G'$ となり，通信遅延も同様に増加する．

キューパラメータ SQ，RQ，TQ が十分な大きさの場合の LogPQ モデルは，LogGP モデルに類似する．LogPQ モデルのパラメータ L，g，P，n は，LogGP モデルのパラメータ $L' + o'$，G'，P'，o' にそれぞれ等しい．両モデルの違いは，LogPQ モデルはギャップの固定成分 g' を考慮できず，LogGP モデルはオーバヘッドの比例成分 o を考慮できないことである．すなわち，サイズの大きいメッセージの通信において，LogPQ モデルはパケット通信を用いる場合に，LogGP モデルは DMA 等によるバースト通信を用いる場合に適する．

Alexandov らは，LogGP モデルの有用性に関する理論的検討を行うとともに，LogGP モデルの実用性を並列計算機 CS-2 を用いて実証した [11]．LogGP モデルの有用性のほとんどは，上記の両モデルの類似性により LogPQ モデルでも成り立つ．一方，LogPQ モデルは，Single Node Scatter 問題に対する Binomial Tree アルゴリズム [11] の通信手法の効率性を表現できない．しかし，Binomial Tree アルゴリズムの手法は並列計算機全体の通信量のネットを増大させるため，一般性に議論の余地があり，また複数プロセッサ間の all-to-all 通信に拡張することができない．したがって，LogPQ モデルにおいて，この問題の重要性は低いと考えられる．

2.5 むすび

本章では，Culler らの提案した並列計算モデル LogP に対し，実用的側面から詳細な検討を行い，並列アルゴリズムを記述するには不十分な点のあることを示した．すなわち，LogP モデルは，実際の通信ネットワークのバッファ動作や通信集中の影響や，通信メッセージサイズの通信遅延に対する影響を反映することができない．

そこで，LogP モデルにおける通信路をキュー（Queue）の結合で表す新しい実用並列計算モデル LogPQ を提案した．LogPQ モデルは，通信路にキューを付加することで，実際の通信ネットワークのバッファ動作や通信集中の影響を反映することができる．また，通信パラメータを 1 ワードメッセージに対して構築することで，通信メッセージサイズの通信遅延に対する影響を反映できる．更に，キューパラメータで与えられる制限による，通信路のバッファ動作の明示的なオーバーフロー制御により，実用的な通信処理の解析が可能である．

また，計算機アーキテクチャの観点から LogPQ モデルの構造に関する検討を行い，LogPQ モデルにより並列計算機上での並列アルゴリズムの詳細な検討が可能であることを明らかにした．先ず，LogPQ モデルと計算機ネットワークの通信路の関係を示した．次に，LogPQ モデルと LogP モデルの関係を示した．また，LogPQ モデルの通信パラメータやキューパラメータの通信動作に与える影響を示した．更に，LogPQ モデルと通信システムの受信ハンドラ処理の関係を示した．最後に，LogPQ モデルと LogGP モデルの関係を示した．

第 3 章

LogPQ モデルによる並列アルゴリズムの解析

3.1 LogPQ モデルの実用性

本章では，さまざまな並列アルゴリズムを実際の並列計算機やエミュレータで実行することにより，実用並列計算モデル LogPQ の有用性や実用性に対する実験的評価を行う．

先ず，変形 CG アルゴリズムの LPRAM モデル上での実験的評価により，同期処理や通信遅延が並列アルゴリズムの実行時間に大きな影響を与えることを示す．また，並列多倍長 GCD アルゴリズムの LogPQ モデル上での実験的評価により，並列計算機の通信性能の各成分がどのように並列アルゴリズムの性能に影響を与えるかを示し，LogPQ モデルが効率的な並列アルゴリズム構築に対し有用であることを明らかにする．更に，並列行列乗算アルゴリズムの並列計算機 CM5 上での実験的評価により，通信にバッファを用いることによりアルゴリズムの通信遅延をより隠蔽でき，並列アルゴリズムの効率を LogPQ モデルを用いて検討できること，従来の LogP モデルより LogPQ モデルが実用性が高いことを示す．最後に，結論を述べる．

3.2 並列整数 GCD アルゴリズム

最大公約数 (GCD) を求めるアルゴリズムは，数千年前の Euclid 法以来，様々な変形が提案されてきた．Euclid 法は，剰余演算を繰り返すことにより GCD を求める．これに対

し，除算をシフトで置き換えたバイナリアルゴリズムは，n ビットで表現される 2 整数 a, b の一方が偶数なら奇数になるまで 1/2 し，両方奇数になると，$a > b$ になるよう交換後，$a = a - b$ する．以上を $a = 0$ になるまで繰り返す．この改良として Brent-Kung(BK) アルゴリズム [12] がある．BK アルゴリズムは，a, b の大小比較を除いたもので，一回の処理に $O(n)$ 時間を必要とし，$O(n)$ 回繰り返すために，全体の実行時間 $O(n^2)$ になる．

Chor-Golereich(CG) アルゴリズム [13] は，BK アルゴリズムをビット単位で並列化したもので，$n^{1+\varepsilon}$プロセッサを用いて $O(n/\log n)$ 時間で実行することが知られている．

このアルゴリズムは現在最も速い並列アルゴリズムの一つであり，実用上重要である．しかし，理論的なモデルである PRAM モデル上で検討されているため，実際の計算機における実用性を十分示しているとは云えない．PRAM モデルは共有メモリ同期型並列計算モデルであり，通信コスト等の物理的制約を全く考慮していない．他方，実用性を考慮したモデルとして，共有メモリを除いた DCM モデル [3]，共有メモリをブロック分割しブロック内同時アクセスを禁止した MPC モデル [2]，共有メモリアクセスに遅延を付加した LPRAM モデル [4] などが存在する．LPRAM モデルは，通信コストの影響を最も直接的な簡潔な形で示しており，一般の超並列計算機に対応することができる．

本節では，CG アルゴリズムの実用上の有効性を詳細に検討するために，LPRAM エミュレータを開発し，通信遅延の影響について検討する．

先ず，CG アルゴリズムとその実装について説明する．次に，LPRAM モデルのエミュレータを用いて，CG アルゴリズムの性能評価を行う．最後に，結論を述べる．

3.2.1 並列 GCD アルゴリズム

3.2.1.1 CG アルゴリズム

並列 GCD アルゴリズムである CG アルゴリズムは，BK アルゴリズムに対し基本演算の並列化，予備計算およびデータ並列といった並列化法を取り入れている．基本演算では，加算乗算等の並列化を行い，加算を $O(1)$ 時間，乗算を $O(\log n)$ 時間で実行する．また，テーブル参照は 2^m個のエントリ (アドレス m ビット) のテーブル参照を $m2^m$プロセッサを用いて $O(1)$ 時間で実行する．予備計算では，始めに kビット乗算テーブルおよび，BK アルゴリズムの保存変換の k回分をパックした k 変換テーブルを作成し使用する．データ並列では，テーブル作成において各エントリを並列計算する．

BK アルゴリズムの保存変換は線型変換である．そして，BK 保存変換の k変換は以下の

---

1. $\delta := 0$
2. 以下の保存変換を $a = 0\ or\ b = 0$ になるまで実行する．
   - ・ $a : odd, b : even$ 時
     - ・ $\delta > 0$ の場合
       $gcd(a, b) := gcd(a, b/2), \delta := \delta - 1$
     - ・ $\delta = 0$ の場合
       $gcd(a, b) := gcd(b/2, a), \delta := \delta + 1$
   - ・ $a : even, b : odd$ 時
     $gcd(a, b) := gcd(a/2, b), \delta := \delta + 1$
   - ・$a, b : odd$ 時
     - ・ $(b + a)/2 : even$ の場合
       $gcd(a, b) := gcd(a, (b + a)/2)$
     - ・ $(b - a)/2 : even$ の場合
       $gcd(a, b) := gcd(a, (b - a)/2)$

3.GCD:=$|a|\ or\ |b|$

図 3.1: 変形 Chor-Goldreich アルゴリズム

式で表わされる．ここで，$-2^k \le c, d, e, f \le 2^k, \sigma = 1or - 1, -k \le g \le k$としている．なお，$\delta = \alpha - \beta$．ただし，$\alpha, \beta$は a, b のビット長の上限である．

$$\begin{aligned} [a, b] &= 2^{-k}[a, b]\begin{bmatrix} c & d \\ e & f \end{bmatrix} \\ \delta &= \sigma \cdot \delta + g \end{aligned} \tag{3.1}$$

ここで k変換テーブルのエントリは，2 進数で表わされた a, b の $(k+1)$LSB およびδのチェイン (符号，$|\delta| \le k$かどうかのフラグ，$|\delta| \le k$の場合の絶対値) で示され，エントリ数は $2^{2k+2}(4k+4)$ である．加減算は Chandra アルゴリズム [14] の使用により，乗算は乗算テーブルの使用により，2^{-k}乗算は右シフトにより各々$O(1)$ 時間で実行され，その合成である k 変換は $O(1)$ 時間で実行されるため，GCD 変換全体の実行時間は $O(n/k)$ である．使用プロセッサ数は $n2^{2k+1}$である．予備計算は，$k^3 2^{3k+6}$プロセッサを使用して $O(\log k)$ 時間で実行される．$k = \varepsilon \log n/2$ とすると，予備計算も含めた実行時間は，$n^{1+\varepsilon}$プロセッサを使用して，$O(n/\log n)$ である．なお，CG アルゴリズムはビット処理を基準としており，a が

奇数である条件が必要である．

3.2.1.2 CG アルゴリズムの変形

CG アルゴリズムの変形を行う．変形 CG アルゴリズムは，図 3.1のように BK 保存変換の実行順序を変更するとともに，k変換テーブルのエントリ数を削減し，使用プロセッサ数を削減したものである．図における$\delta = \beta - \alpha$である．CG アルゴリズムにおけるσを除去している．また，$\delta \geq 0$ である．δをチェイン (符号，$\delta < k$かのフラグ，$\delta < k$の場合の絶対値) で表わし，テーブルエントリを $0 \leq \delta \leq k$の場合のみと，エントリのビット長を 2 ビット減らし，テーブルアクセスに必要なプロセッサ数を 1/4 にすることができる．また，エントリ数も $0 \leq \delta \leq k$の場合のみになるために 1/4 に削減できる．一方，保存変換後の比較は，BK アルゴリズムは b のみ行えば良かったが変形 CG アルゴリズムでは a の比較も必要となる．従って，比較時に必要なプロセッサ数は倍になるが，これは他部分での使用プロセッサ数に比べて小さく，アルゴリズム全体に影響を与えない．なお，変形 CG アルゴリズムは，a または b が奇数である条件が必要である．

3.2.1.3 インプリメント

文献 [13] は CRCW(Concurrent Read Concurrent Write) と CREW(Concurrent Read Exclusive Write) のアルゴリズムを示しているが，本論文では CRCW のインプリメントを行った．実際の超並列計算機では CRCW アクセスは困難であるが，この影響は LPRAM の通信遅延で考慮できる．

インプリメントにあたり，加算部分を変更した．Chor-Goldreich は，Chandra の考案したほぼ線型数プロセッサ $O(1)$ 時間アルゴリズム [14] を採用しているが，本論文では，[14] で一般的な形で示されている n^3プロセッサ $O(1)$ 時間アルゴリズムを使用した．これは，Chandra アルゴリズムが複雑になること，および実行ステップ数が多い為である．Chandra アルゴリズムは，各プロセッサに別々の動作をさせる必要がある．つまり，n に依存する多数の条件分岐が必要である．従って，CM-5 などの超並列計算機に採用されている Single Program Multiple Data stream(SPMD) モデル [15] では $O(1)$ 処理時間の実現が困難になる．後者のアルゴリズムは，条件分岐が少なく (2 通りの動作)，4 ステップで処理を実行できる．

また，実用性を重視したインプリメントを行った．たとえば，$ab + cd$ の計算において，

時間複雑さ等の理論的議論ならば，乗算にα_1個のプロセッサ，加算にα_2個のプロセッサを用いる時には，必要なプロセッサ数は$\max(\alpha_1, \alpha_2)$となる．実用性を考えるとabとcdは同時に計算できるため，必要なプロセッサ数は$\max(2\alpha_1, \alpha_2)$となる．ただし，$a = a \cdot c + b \cdot e$と$\delta = \delta + g$といった別々の計算は同時には行わない．

詳細な動作解析を行うため，変形 CG アルゴリズムをkビット乗算テーブルの作成，k変換テーブル作成，GCD 処理としてk変換実行の 3 つの部分に分けてインプリメントした．

3.2.2　エミュレータを用いた性能評価

並列計算モデルのエミュレータを開発し，エミュレータ上で並列アルゴリズムを動作させることにより，オーダレベルでなく実際の実行時間に即した並列アルゴリズムの詳細な動作解析を行うことができる．

3.2.2.1　LPRAM モデル

変形 CG アルゴリズムの詳細な動作解析を行うために LPRAM エミュレータを開発した．超並列マシンのエミュレータは，複雑なアルゴリズムをインプリメントできなければならない．特に，超並列マシンの場合は，各プロセッサが別々のプログラムを実行する MIMD 処理は現実的でない．また，CM-5 のように SPMD を採用した商用並列計算機が実在していることを考えると，SPMD タイプのエミュレータが望ましい．次に，エミュレータが実行する機械語は，一般のプロセッサに共通した命令セットであることが必要である．アセンブラレベルでは複雑なプログラミングは困難なことから高級言語が必要である．アルゴリズムの詳細な動作解析をするエミュレータは，入出力や初期設定を除いたアルゴリズム本体の実行時間をステップ単位で詳細に測定できなければならない．

Hämäläinen 等 [16][17] は，PRAM モデルに基づいたエミュレータを開発した．PRAM エミュレータは，Modula2 ライクな高級言語のコンパイラ，基本命令のみを有する機械語のアセンブラ，アセンブルした機械語を実行するエミュレータにより構成されている．PRAM エミュレータの高級言語は，プロセッサ間の共有変数と局所変数が使用できる． 我々は，この PRAM エミュレータを基に，共有メモリのアクセスにlクロックを要する LPRAM エミュレータを開発した．

3.2.2.2 性能評価

変形 CG アルゴリズムに対して，(1) 局所変数を共有メモリ上に置き共有メモリのアクセスによりプロセッサ間の同期を取る通信同期を行う場合 (Common access,synchronize)，(2) 局所変数を局所メモリ上に置き通信同期を行う場合 (Localaccess,synchronize)，(3) 局所変数を局所メモリ上に置き nop 命令による同期を行う場合 (Local access,nop-synchronize) について比較検討を行った．

図 3.2(a) は，kビット乗算テーブル作成部分の実行時間である．横軸は共有メモリアクセスのレイテンシを示す．(1)(2)(3) の順に共有メモリのアクセス頻度が大きくなるが，その影響がはっきり表われている．すなわち，共有メモリの通信遅延の影響は大きいが，局所変数を局所メモリに割当てるとその影響はかなり小さくなる．また，同期処理には多くの通信を要し，通信遅延の影響を大きく受けていることが分かる．

図 3.2(b) は，k変換テーブル作成部分の実行時間である．通信遅延の影響はkビット乗算テーブル作成部分と同様の振舞を示す．なお，$k = 2$ の場合は，使用プロセッサ数が多いことから通信同期による遅延の影響は大きくなっている．

図 3.2(c) は，$60, 99$ の 2 整数が入力された場合の GCD 処理部分の実行時間である．この実行時間は BK 保存変換の繰り返し数 lに比例し，lは入力のビット長 n に対し $O(n/\log n)$ である [13] が，本入力では $l = 12$ となる．通信遅延の影響は図 3(a) と同様である．使用プロセッサ数が多いことから通信同期による遅延の影響が大きくなっている．

この入力において，$k = 2$ は $k = 1$ に比べて GCD 処理部分の実行時間は半減するが kビット乗算テーブル作成部分の実行時間は増加し，変形 CG アルゴリズム全体の実行時間はほぼ一致している．lが小さいときには，k増加による GCD 処理部分の実行時間減少よりも kビット乗算テーブル作成部分の実行時間増加のほうが大きく，アルゴリズム全体の実行時間は増加する．今回の場合，$k = 2$ にする意味があるのは lが 12 程度以上のときである．lが 4 倍になるごとに kを倍にすると実行時間は最適になるものと見込まれる．しかし，kを増すことにより使用プロセッサ数が膨大なものになるために実行時間は制約される．例えば $n = 32, k = 2$ の場合，使用プロセッサ数は 10^5程度となる．

変形 CG アルゴリズム処理時間は，lが小さい場合でも同期処理の実行時間に対する影響が大きい．すなわち，変形 CG アルゴリズムは小粒度 (fine grain) の並列性を用いているために同期を頻繁に必要とすることから，通信コストが低い場合においても同期のオーバヘッドは無視できない．

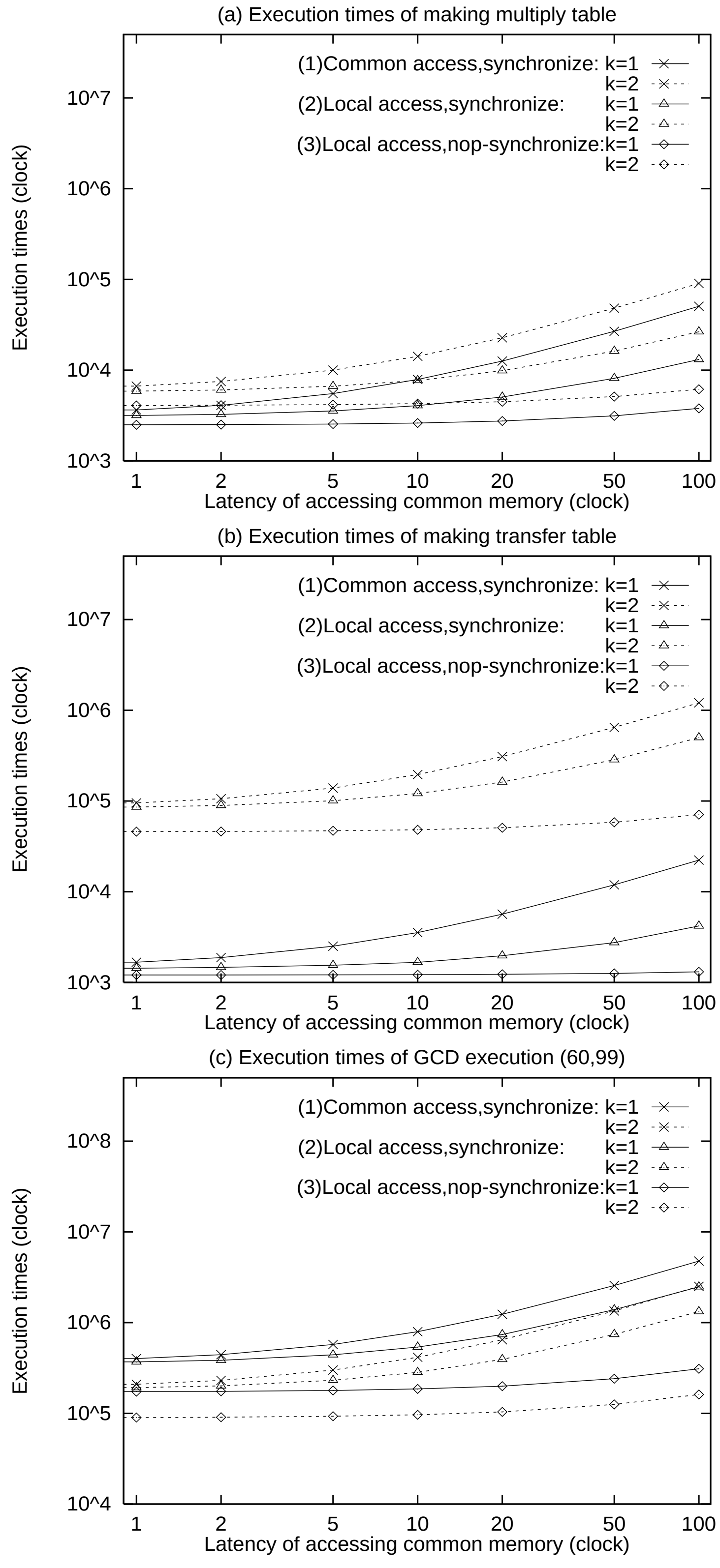


図 3.2: LPRAM モデル上での変形 Chor-Goldreich アルゴリズムの処理時間

また，変形 CG アルゴリズムは共有変数のアクセスが多く，共有メモリの通信遅延の影響は大きい．しかし，局所変数を共有メモリから局所メモリに割当てることにより通信遅延の影響はある程度削減できることが明らかになった．

変形 CG アルゴリズムは，同期型で通信遅延の小さい並列計算機においてはある程度実用性を確保できる可能性があるが，非同期型では遅延の影響が大きく実用上の問題となることを確認した．

3.2.3　まとめ

変形 CG アルゴリズムの LPRAM モデルの並列マシン上での動作解析を行った．その結果，変形 CG アルゴリズムに対して同期処理や通信遅延が実行時間に大きな影響を与えることや局所変数の割当て方法により処理時間を改善できることが明らかになった．

3.3　並列多倍長 GCD アルゴリズム

拡張 Signed Digit 表記を用いた多倍長整数演算による，整数 GCD 並列アルゴリズムについて提案する．また，並列計算機エミュレータを開発し，実際の並列計算機を対象とした GCD 並列アルゴリズムの性能評価を行う．

先ず，並列多倍長 GCD アルゴリズムを示す．次に，並列計算機における各プロセッサの同期方法について議論する．次に，LogP モデルのエミュレータを用いて，並列多倍長 GCD アルゴリズムの性能評価を行う．最後に，結論を述べる．

3.3.1　並列多倍長 GCD アルゴリズム

3.3.1.1　GCD アルゴリズム

図 3.3に実用性の高い逐次アルゴリズムである Brent-Kung(BK) アルゴリズム [12] を示す．入力を n ビットの 2 整数 A, Bとする．但し，A は奇数とする．BK アルゴリズムは，a, b の奇偶比較により，$gcd(a, b) = gcd(A, B)$ の関係を保ちながら a, b の変換を行う．この変換を保存変換と言い，$O(n)$ 回繰り返し GCD を求める．

```
{ INPUT A, B  :  A is odd, B ≠ 0, |A|, |B| ≤ 2^n }
a := A, b := B, δ := 0  { α := n, β := n }
while b ≠ 0
    { a is odd, |a| ≤ 2^α, |b| ≤ 2^β, δ = α − β }
    while b is even
        b := b/2, δ := δ + 1  { β := β − 1 }
    if δ > 0
        swap(a, b), δ := −δ  { swap(α, β) }
    if (b + a)/2 is even
        b := (b + a)/2
    else
        b := (b − a)/2
GCD:=|a|
```

図 3.3: Brent-Kung アルゴリズム

3.3.1.2 拡張 SD 法を用いた冗長表記

多ビットの整数計算を通常の計算機で行う方法に，固定ビット長のワードを複数用いた多倍長整数による多倍長計算がある．p 個のプロセッサよりなる並列計算機では，多倍長整数 $a = \sum_{i=0}^{p-1} a_i d^i$の各ワード a_iをプロセッサ P_iに割り当て並列実行することにより処理の高速化が可能である．P_iは P_{i-1}からの通信により動作を開始し P_{i+1}に通信を送る．キャリー処理は隣接プロセッサとの通信で行う．各プロセッサはプロセッサ P_0から P_{p-1}に向けて線型に同期して動作する．

並列多倍長演算は，バイナリ表記ではキャリー伝搬による通信遅延の影響が大きい．a_0に対する処理の結果が a_{p-1}に影響するため，キャリー処理の通信遅延の影響が $p-1$ 倍されて演算を遅延させる．これに対し，各ワードに冗長性を付加することによりキャリー伝搬が除去される．

冗長表記である carry save 法 [18] は，2 つの多倍長整数 $a = \sum_{i=0}^{p-1} a_i d^i$, $b = \sum_{i=0}^{p-1} b_i d^i$の和 $s = \sum_{i=0}^{p-1} s_i d^i$が，$0 \le a_i, b_i < 2d$ としたときに $s_i = w_i + c_{i-1}(w_i = (a_i + b_i)\ mod\ d$, $c_i = (a_i + b_i)\ div\ d)$ で求まる．しかし，carry save 法は負数を 2 の補数で表現した場合に，等号比較を効率よく実行できない制限がある．

一方 Signed Digit 表記 [19] は，$d \geq 3$ の場合，$-d < a_i, b_i < d$ としたときに $s_i = w_i + c_{i-1}$ で求まる．但し，$z_i = a_i + b_i$として，$z_i \leq -d+1$ では $w_i = z_i + d$, $c_i = -1$，また $-d+1 < z_i < d-1$ では $w_i = z_i, c_i = 0$，$z_i \geq d-1$ では $w_i = z_i - d, c_i = 1$ である．Signed Digit 表記は等号比較が容易である．$a = 0$ かどうかは，すべての $a_i = 0$ かどうかで判別できる．しかし，複数の演算を実行する場合，Signed Digit 表記は複数回のキャリー処理を必要とする．

そこで，Signed Digit 表記を拡張し，キャリー処理順序を変え，キャリー処理の回数を削減する拡張 SD 法を提案する．拡張 SD 法による多倍長整数は,

$$a = \sum_{i=0}^{p-1} a_i 2^{il}, h > l, -2^{h-1} \leq a_i < 2^{h-1} \tag{3.2}$$

上式のように冗長性を増やすことにより，一回のキャリー処理後に複数の演算を実行できる．例えば積和演算は，(1)a_i, b_iの上位 $h-l$ビットを冗長部 n_{a_i}, n_{b_i}として抜き取る，(2)a_i, b_iに下位ワードの冗長部 $n_{a_{i-1}}, n_{b_{i-1}}$を加える，(3)a_i, b_iの積和を計算する，の 3 段階で行う．また，正負反転 $a = -a$ は，上記 (1)(2) に加え，(3')$a_i = -a_i$を行う．正負反転を 2 度実行すると，$a = 0$ ならば $a_i = 0$ となり，等号比較できる．

3.3.1.3 並列多倍長 GCD アルゴリズム

拡張 SD 法による多倍長計算を用いて BK アルゴリズムを行う．並列多倍長 GCD アルゴリズムでは，多倍長整数 a, b の各ワード a_i, b_iをプロセッサ P_i に割り当てる．

BK アルゴリズムは，a, b の下位 2 ビットの値により保存変換を決定する．多倍長整数の保存変換は a_0, b_0にのみ依存する．従って，プロセッサ P_0 は通信の必要なく保存変換を算出できる．P_0は保存変換を各プロセッサ P_iに伝達し，P_iは a_i, b_iに対し保存変換を実行する．

保存変換は，a, b に対する積和演算と 1 ビット右シフトよりなる．多倍長整数を用いた場合，a_i, b_iは $a_{i+1}, b_{i+1}, a_{i-1}, b_{i-1}$とのキャリー処理が必要である．従って，$P_i$は P_{i+1}, P_{i-1}と双方向の通信を行う必要がある．もし P_0での保存変換算出時間が，P_1に対する双方向の通信時間より小さい場合，P_0に待ち時間が生じる．この通信遅延を隠蔽するため，保存変換 k回毎にキャリー処理を削減する手法として k変換を行う．

k回の保存変換の合成である k変換は，$k+2$ ビット符号付き整数 c, d, e, fにより，次式

で表される [13].

$$\begin{bmatrix} a & b \end{bmatrix} := \frac{1}{2^k} \begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} c & d \\ e & f \end{bmatrix} \tag{3.3}$$

k変換を a_i, b_iに対して実行することにより，キャリー処理は保存変換 k回をまとめた形で行われる．

並列多倍長 GCD アルゴリズムは，k変換と正負反転の合成を保存変換行列 $Y = \begin{bmatrix} c & d \\ e & f \end{bmatrix}$ として，この保存変換を実行する．$b = 0$ の場合，$Y = \begin{bmatrix} -2^k & 0 \\ 0 & -1 \end{bmatrix}$となり，$b$ に対して正負反転および右シフトとなる．この 2 回の実行により Z_{p-1}は真となる．なお，Z_jは $b_i = 0$ $(0 \leq i \leq j)$ を示す変数である．

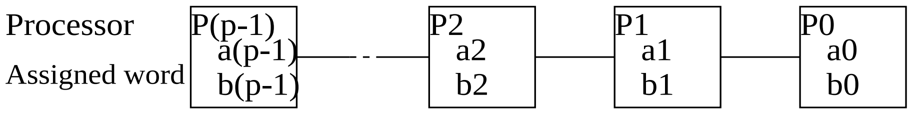
図 3.4に，p 個のプロセッサで実行される並列多倍長 GCD アルゴリズムの概要を示す．P_0は，先ず a_0, b_0より保存変換行列 Yを算出し保存変換を実行する．次に保存変換行列 Y，変数 Z_0，保存変換実行で発生した上キャリー n_{a_0}, n_{b_0} を P_1に送信する．最後に変数 Z_0を算出する．なお，P_1からの下キャリー r_{a_1}, r_{b_1}の処理は，通信遅延の隠蔽のため，次回の保存変換行列算出後に行う．P_j $(0 < j \leq p-1)$ は，先ず P_{j-1}から保存変換行列 Yと変数 Z_{j-1} を受信し，対応するものを P_{j+1}に送信する．次に a_i, b_iに対し保存変換を実行する．なお，P_{p-1}は，更に $b = 0$ (Z_{p-1}) を調べる．以上の処理を保存変換ステップとよぶ．保存変換ステップを，$b = 0$ が確認されるまで繰り返す．これを保存変換フェーズとよぶ．最後に a をバイナリ表記に変換し，その絶対値が GCD となる．これを終了処理フェーズとよぶ．

保存変換実行の詳細を示す．各プロセッサ $P_i(i > 0)$ は，先ず a_i, b_i の冗長部を上キャリー n_{a_i}, n_{b_i}として P_{i+1}に送出し，P_{i-1}より得たキャリー $n_{a_{i-1}}, n_{b_{i-1}}$を加算する．次に積和演算と右シフトを行う．最後に下キャリー r_{a_i}, r_{b_i}を P_{i-1} に送出し，P_{i+1}より得たキャリー $r_{a_{i+1}}, r_{b_{i+1}}$を加算する．また，変数 $Z_j = Z_{j-1} \wedge (b_i = 0)$ である．

保存変換ステップにおいて，$P_i(i > 0)$ の処理時間が P_0の処理時間より小さい場合，P_i $(i > 0)$ に待ち時間が生じる．そこで実際には，P_i $(0 < i < p-1)$ には s ワード，P_{p-1}には $s+1$ ワードと，複数のワードを配置する．以下では，s をワード複合数とよぶ．

3.3.1.4 実行遅延時間

実行時間の遅延部分は，保存変換フェーズでのキャリー処理で生じる遅延時間と，保存変換フェーズで $b = 0$ になってからそれを P_{p-1}で検出するまでの遅延時間 (Z_i伝搬遅延時



(a) 各プロセッサ P_iにワード a_i, b_iを配置

(b) P_0は，保存変換行列 Yを算出し，保存変換を実行

(c) 各 P_i $(i > 0)$ は，P_{i-1}から Yと Z_{i-1}を受け取り，P_{i+1} に Yと Z_iを送信する

(d) 各 P_i $(i > 0)$ は，P_{i-1}から符号付き整数 $n_{a_{i-1}}, n_{b_{i-1}}$ を受け取り，下記を実行

1. $n_{a_i} = (a_i$の左 $h - l$ビット$), n_{b_i} = (b_i$の左 $h - l$ビット$)$
2. $a_i = ($符号付き整数 $n_{a_{i-1}}) + (a_i$の右 lビット$)$
   $b_i = ($符号付き整数 $n_{b_{i-1}}) + (b_i$の右 lビット$)$
3. $[a_i \ \ b_i] = [a_i \ \ b_i] \cdot Y$

(e) 各 P_i $(i > 0)$ は，P_{i+1}から符号付き整数 $r_{a_{i+1}}, r_{b_{i+1}}$ を受け取り，下記を実行

1. $r_{a_i} = (a_i$の右 kビット$), r_{b_i} = (b_i$の右 kビット$)$
2. $a_i = (a_i$を kビット右シフト$)$
   $\quad +(r_{a_{i+1}}$を $l - k$ビット左シフト$)$
   $b_i = (b_i$を kビット右シフト$)$
   $\quad +(r_{b_{i+1}}$を $l - k$ビット左シフト$)$

(f) Z_{p-1}が真 (全 $b_i = 0$) になるまで (b)〜(e) を繰り返す

(g) a をバイナリ表記に変換しその絶対値を求める

図 3.4: 並列多倍長演算 GCD アルゴリズムの概要

間) と終了処理フェーズの実行時間の両時間の和の，二種類に分けられる．前者は保存変換の実行回数に比例するので定率遅延，後者は一定なので定量遅延とよぶ．

バイナリ表記では保存変換ステップが終了するまで n_a, n_b, Z_iを得られないが，拡張 SD 法では保存変換ステップ開始時に得られる．従って，Z_iの P_0から P_{p-1}への伝搬時間は，保存変換ステップの実行時間の $p-1$ 倍の時間だけ減少する．

3.3.2 並列多倍長 GCD アルゴリズムの同期方法

3.3.2.1 線型同期

3.3.1節では，各プロセッサは P_0から P_{p-1}に向けて線型に同期して動作する．各プロセッサ P_iはプロセッサ P_{i+1}, P_{i-1}としか通信処理を行わないため，通信処理が簡潔になり，保存変換ステップにおいて通信処理に要する時間が小さい．しかし，保存変換行列 Yの全プロセッサへの伝達や，各プロセッサの状態 $(b_i = 0)$ の論理積算出が逐次的に行われることから，定量遅延時間はプロセッサ数に比例する．使用するプロセッサ数が多い場合，この実行時間に与える影響は無視できない．

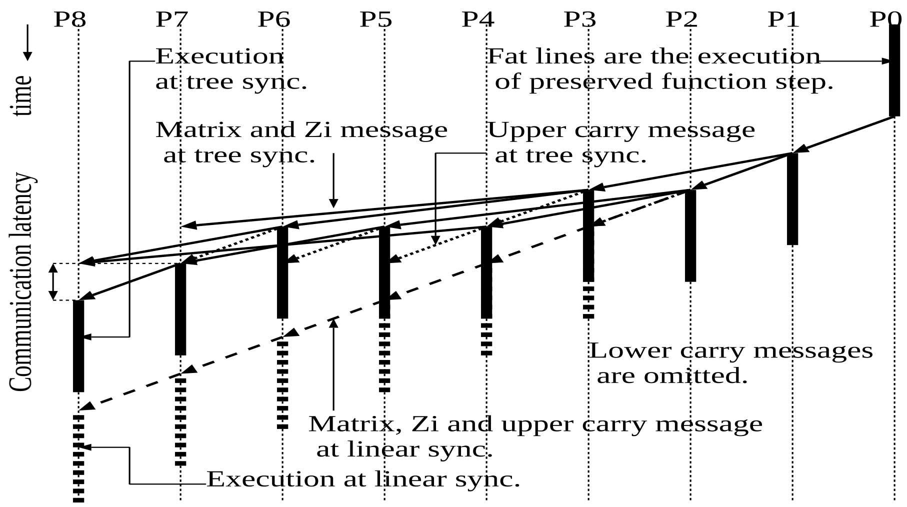
3.3.2.2 木状同期

拡張 SD 法はキャリー伝搬がないため，線型同期で動作する必要はない．そこで，図 3.5(a) に示すように，プロセッサ P_iは，プロセッサ P_{i+1}, P_{i-1}とのキャリー処理と非同期に，保存変換行列 Yの伝達と各 $(b_i = 0)$ の論理積算出を行う．木状同期により，プロセッサ間の伝達遅延を少なくし，定量遅延時間を削減できる．

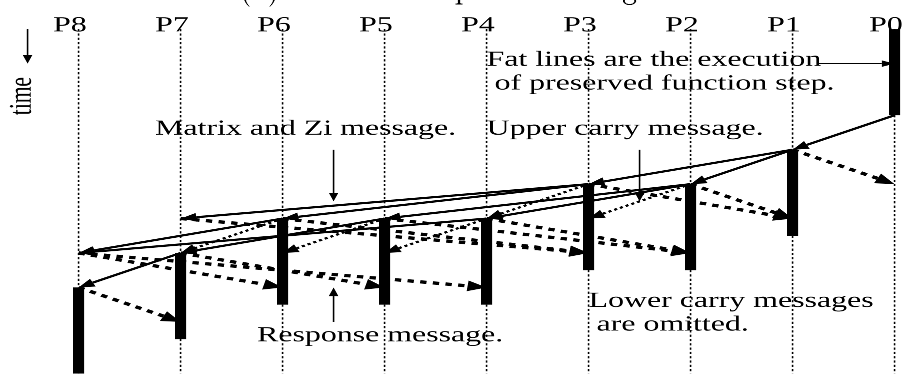
木状同期動作において，プロセッサ P_iは，保存変換行列 Yと変数 Z_iを上方向のプロセッサ P_{p-1}〜P_{i+1}のいずれかに送信する．一方，下方向への通信は P_{i-1}に対するキャリー通信のみである．P_jから P_i $(i > j+1)$ への通信は P_iの動作を待つことなく $i-j$回実行され得るため，P_iの通信受信部のバッファ長は通信 1 回の場合の $i-j$倍を必要とする．

3.3.2.3 双方向通信による木状同期

木状同期では，プロセッサ P_iの上方向の通信は，最遠で $P_{i+\lceil p/2 \rceil}$に送信される．よって，受信バッファ長は，上方向の通信メッセージ長の $\lceil p/2 \rceil$ 倍を必要とする．この軽減のために，上方向のプロセッサ P_{p-1}〜P_{i+2}への通信に対し逆方向の通信を付加する．



(a) Without response message.



(b) With response message.

図 3.5: 並列多倍長 GCD アルゴリズムの木状同期動作

図 3.5(b) に，この双方向木状同期の動作を示す．受信バッファ長は上方向の通信メッセージ長の 2 倍となる．木状同期は線型同期に比べ通信動作が複雑なので，一般には通信による遅延が増加する．しかし通信遅延がある程度以下なら，1 保存変換ステップの実行時間内に逆方向通信がリターンされ，遅延時間は増加しない．

3.3.2.4　並列計算機のネットワークとの対応

並列多倍長 GCD アルゴリズムは，並列計算機に通常用いられるネットワークトポロジー上に配置できる．線形同期の場合，アルゴリズムの各 P_i $(0 \leq i < p)$ をリング状に配置すれば，すべての通信は隣接プロセッサと行われる．

木状同期では，情報を木状に分配・収集する通信を行う．例えばハイパーキューブは，これを効率的に実行できる．また，次数固定であるシャッフルネットワークの一種の場合も効率的に実行できる．これは，各 P_i $(0 \leq i < p)$ に $P_{(i+1)\ mod\ p}$，$P_{(2i)\ mod\ (p+1)}$，$P_{(2i+1)\ mod\ (p+1)}$ (あて先が P_pの場合は P_0とする) とその逆を接続する次数 6 のネットワークであり，図 3.5 で示されたすべての通信は隣接プロセッサと行われる．

3.3.3　並列多倍長 GCD 計算の動作解析

3.3.3.1　LogP モデル

Culler 等 [9] が提案した実用並列計算モデルである LogP モデルは，並列計算機の物理的制約をメッセージ通信の通信遅延 L，通信オーバヘッド o，通信バンド幅 g，プロセッサ数 Pの 4 パラメータで表す（図 3.6）．

並列多倍長 GCD アルゴリズムは，並列計算機に通常用いられるネットワーク上に配置できる．従って，アルゴリズムの各通信の振舞いは，LogP モデルのパラメータで直接表される．並列アルゴリズムを LogP モデルを用いて動作解析することにより，実際の並列計算機上での性能評価ができる．

並列多倍長 GCD アルゴリズムの性能評価を行う為に，LogP モデルのエミュレータを開発した [20]．並列多倍長 GCD アルゴリズムの通信動作を詳しく検討するために，各プロセッサの送受信部と通信路の集結部にキューを付加した．LogP エミュレータでは，通信メッセージ長を 1 ワードに固定している．

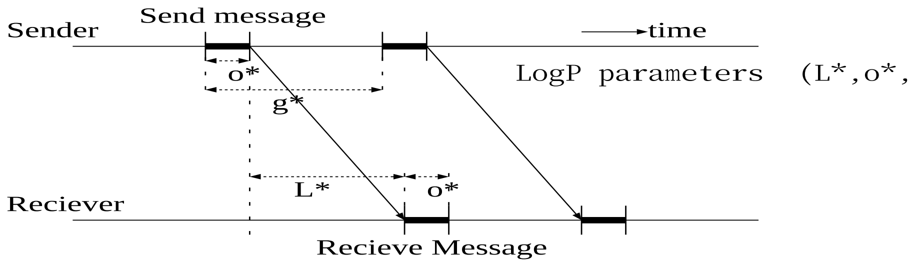


図 3.6: LogP モデルの通信動作

3.3.3.2 並列多倍長 GCD アルゴリズムの性能評価

実在する並列計算機に対応するパラメータを用いて，10^4ビット並列多倍長 GCD の性能評価を行った．

ここで，拡張 SD 表記の h, lを 23, 20 とし，変換複合数 kを 8 とした．また，入力は (1 , 2^{x-1}) を用いた．

パラメータは J-Machine[9] を参考に，L, o, gを各々32, 4, 8 として，並列アルゴリズムの動作解析を行った．

（1）使用プロセッサ数

並列多倍長アルゴリズムでは，ワード複合数 s により使用するプロセッサ数が定まる．プロセッサ数と実行時間との関係を調べ，最適なプロセッサ数について議論する．

図 3.7に，10^4ビット並列多倍長 GCD アルゴリズムの 10^4ビット逐次多倍長 GCD アルゴリズムに対する高速化率を示す．入力 $x = 10000$ においてはプロセッサ数増加により速度が向上するが，プロセッサ数 $P = 100$ 前後で頭打ちになる．$x = 1$ においてはプロセッサ数増加により単調に速度が低下している．前者は定率遅延の増加を，後者は定量遅延の増加を示している．ワード複合数 s に対する実験結果から，線型同期では $s = 6$，木状同期では $s = 5$ が最適となった．使用プロセッサ数は各々84, 101 個である．

（2）並列アルゴリズムの高速化率

並列多倍長 GCD アルゴリズムの実行時間を逐次アルゴリズムと比較し，並列アルゴリズムの性能評価を行う．

図 3.8に，入力 $(1, 2^{x-1})$ に対する並列多倍長 GCD アルゴリズムの高速化率を示す．Single-precision GCD は単倍長 GCD の逐次多倍長 GCD に対する高速化率であり，並列多倍長 GCD の高速化の上限を示す．Serial GCD は各入力に対して多倍長演算のビット数を小さ

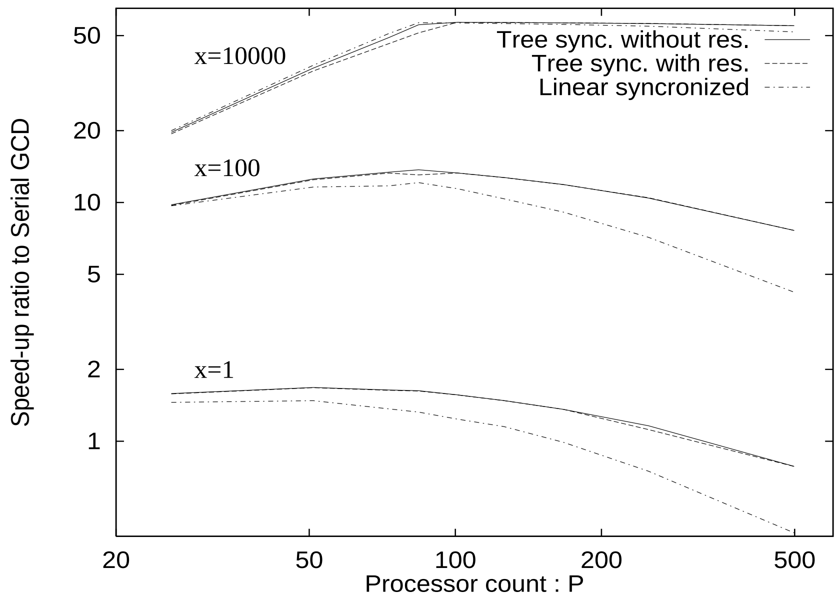


図 3.7: 使用プロセッサ数と並列多倍長 GCD アルゴリズムの高速化率 (L=32,o=4,g=8)

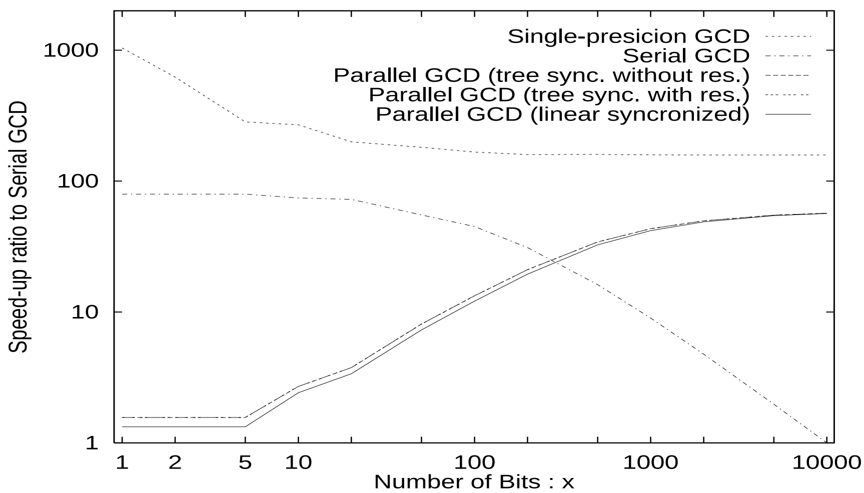


図 3.8: 並列多倍長 GCD アルゴリズムの高速化率 (L=32,o=4,g=8; 線型同期:P=84,s=6, 木状同期:P=101,s=5)

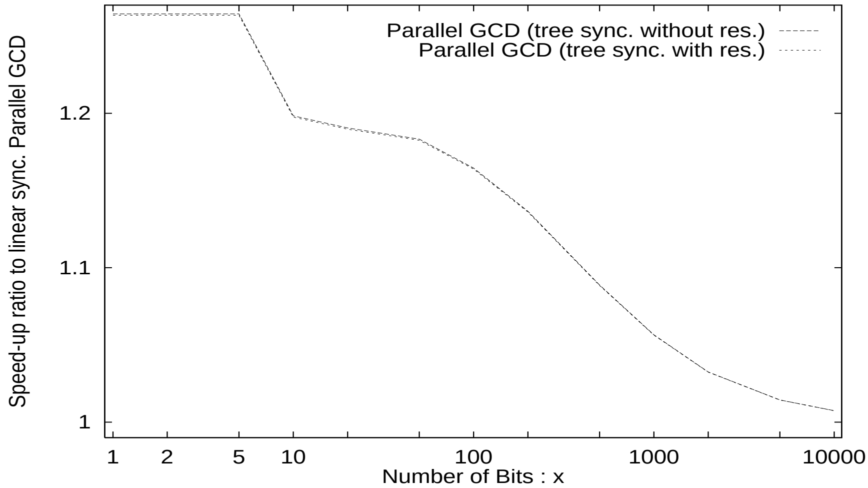


図 3.9: 並列多倍長 GCD アルゴリズムの木状同期の高速化率 (s=5)

く設定した小ビット逐次多倍長 GCD の高速化率であり，逐次多倍長 GCD の高速化の上限を示す．Parallel GCD は並列多倍長 GCD の高速化率であり，linear syncronized は線型同期，tree sync. は木状同期を，with res. は逆方向通信の付加を表わしている．並列多倍長 GCD は逐次多倍長 GCD より常に高速であり，$x = 10^4$では 56.6 倍の速度となった．これは，並列多倍長 GCD アルゴリズムの高速化上限である単精度 GCD の 1/2.8 であり，多ビット長の GCD を求めるには並列アルゴリズムが有効であることが分る．しかし，小ビット逐次多倍長 GCD とは $x = 300$ 前後でクロスしている．これは並列多倍長 GCD の定量遅延によるもので，$x = 500$ 未満の入力に対しては逐次計算の方が速くなることを示している．従って，定量遅延の削減は重要である．

（３）同期方法

次に，同期方法による並列計算の処理時間の違いについて検討する．図 3.9に，線型同期並列多倍長 GCD アルゴリズムに対する木状同期の高速化率を示す．比較の為，$s = 5$ に統一した．木状同期の場合は線型同期の場合より常に高速であることが分る．また，双方向木状同期による遅延増加はほとんどないために有効な同期方法と考えられる．

3.3.3.3 通信パラメータの影響

通信パラメータの実行時間に対する影響を調べる．

図 3.10に，レイテンシ L に対する線型同期並列多倍長 GCD アルゴリズムの実行時間の

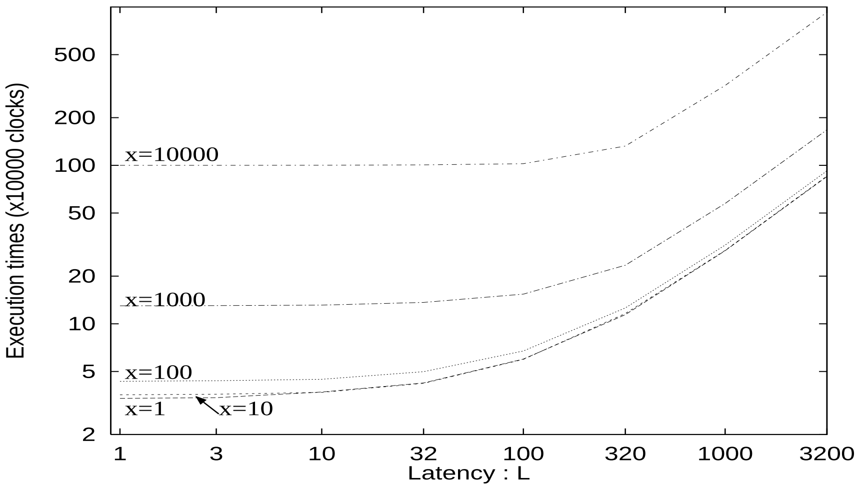


図 3.10: レイテンシ L と並列多倍長 GCD アルゴリズムの実行時間 (o=4,g=8,s=6)

変化を示す．$L \leq 100$ においては L の実行時間に対する影響は比較的小さく，x が小さいときのみである．$L > 100$ では全体的に影響するようになる．これは，$L = 100$ までのレイテンシが隠蔽されることを示している．従って，通信遅延の隠蔽が一定の範囲で確保される．

なお，L と同様に o は保存変換ステップに付加される定率遅延増加因子であり，gは L に付加される遅延増加因子である．

3.3.4 まとめ

本節では，実際の並列計算機の物理的制約を考慮した並列多倍長 GCD アルゴリズムを提案した．これはプロセッサ間同期や通信バッファの制約を考慮した並列アルゴリズムである．

また，並列計算モデル LogP のエミュレータを用いた性能評価により，並列アルゴリズムの有用性と並列計算機の通信遅延による性能への影響を検討した．

3.4 超並列計算機 CM-5 での行列乗算の性能解析

本節では，並列計算機 CM5 上で並列行列乗算アルゴリズムを実行し，LogP モデルと LogPQ モデルの実用性を比較検討する．

先ず，Cannon の並列行列乗算アルゴリズムについて説明する．次に，並列計算機 CM5 上で並列行列乗算アルゴリズムを実行し，通信バッファの有無による各種通信方式の効率について比較検討する．次に，並列行列乗算アルゴリズムを LogP モデルと LogPQ モデルを用いて解析し，実験結果と比較検討する．最後に，結論を述べる．

3.4.1 Cannon のアルゴリズム

Cannon アルゴリズム [21] は，行列 A と Bの積 $C = A \cdot B$を求める並列行列乗算アルゴリズムである．行列サイズを $N \times N$，プロセッサ数を $P = p^2$とする．なお，行列サイズ Nは p の倍数とする．先ず，プロセッサをサイズ $p \times p$ の行列に配置し，P_{ij} $(0 \leq i, j < p)$ と表す．そして，各プロセッサ P_{ij}に，部分行列 A_{ij}，B_{ij}，C_{ij}を割り当てる．各 A_{ij}，B_{ij}，C_{ij}は，それぞれ行列 A，B，Cの i 行目で j列目のブロックである．各ブロックのサイズは $l \times l$，但し $l = N/p$ である．積 Cは以下のように算出される．

1. 各プロセッサ P_{ij}は，A_{ij}を $P_{i((j-i)\ mod\ p)}$に，B_{ij}を $P_{((i-j)\ mod\ p)j}$に送信
2. 各プロセッサ P_{ij}は，C_{ij}に $(A_{ij} \cdot B_{ij})$ を加える
3. 以下を $(p-1)$ 回繰り返す．
   (a) 各プロセッサ P_{ij}は，A_{ij}を $P_{i((j-1)\ mod\ p)}$に，B_{ij}を $P_{((i-1)\ mod\ p)j}$に送信
   (b) 各プロセッサ P_{ij}は，C_{ij}に $(A_{ij} \cdot B_{ij})$ を加える

このアルゴリズムの計算時間は $O(N^3/P)$ であり，通信時間は $O(N^2/\sqrt{P})$ である．

3.4.2 通信バッファの通信遅延に対する影響の評価

3.4.2.1 プロセッサ間通信

Cannon の並列アルゴリズムを C 言語を用いて並列計算機 CM5 に実装した．プロセッサ間通信は，CM5 の通信ライブラリ CMMD[6] の非同期通信命令を用いる．非同期通信命令は，以下のように用いられる．

1. 非同期メッセージの送信/受信コマンドを呼び出す
   Message Control Block（MCB）が確保され，そのメッセージの状態が保存される [Send/Receive]
2. MCB をチェックし，メッセージ通信の完了を待つ [Wait until finishing]
3. MCB を解放する．[Release MCB]
4. 受信したメッセージを利用して計算処理を行う [Computation]

並列アルゴリズムにプロセッサ間通信方式が及ぼす影響を解析するため，単純通信方式と隠蔽通信方式を実装した．図 3.11に，以下の各種通信方式を用いた処理方式を示す．

(a) 単純通信方式

単純通信は，通信処理が終了するまで計算処理を実行しない．したがって，通信処理による遅延は，並列処理全体の実行時間に直接影響する．一方，通信処理がハンドシェイクで制御されるので，各プロセッサは $2r$ ワード長の送受信バッファを持てばよい．

(b) 隠蔽通信方式

隠蔽通信は，通信処理の実行中に計算処理を行い，通信遅延を隠蔽する．この場合もハンドシェイクにより，通信に必要な送受信バッファサイズは $2r$ ワードでよい．しかしこのとき，二つの遅延が発生する可能性がある．先ず，各プロセッサはメッセージ送信の前に以前の送信の完了を待たねばならない．また，各プロセッサは他プロセッサと非同期的に動作するので，受信側プロセッサの開始が遅ければ送信側プロセッサは送信を待たされる．

(c) 受信バッファ付加

隠蔽通信方式は，受信バッファの付加により，受信側プロセッサ開始遅れによる遅延を隠蔽できる．

(d) 送信バッファ付加

隠蔽通信方式は，送信バッファの付加により，以前の送信の完了待ちによる遅延を隠蔽できる．

(e) 送受信バッファ付加

隠蔽通信方式は，図 3.11(c)(d) を合わせた送受信バッファの付加により，上記の両方の遅延を隠蔽できる．

[Source processor]	[Destination Processor]
loop:Call Send	loop:Call Receive
Wait until finishing	Wait until receive finishing
Release MCB	Release MCB
Computation	Computation
goto loop	goto loop

(a)Simple communication.

[Source processor]	[Destination Processor]
loop:Call Send	loop:Call Receive
Computation	Computation
Wait until finishing	Wait until receive finishing
Release MCB	Release MCB
goto loop	goto loop

(b)Hidden communication.

[Source processor]	[Destination Processor]
loop:Send	Call plural Receives
Computation	loop:Computation
Wait until finishing	Wait until Receive finishing
Release MCB	Release MCB
goto loop	goto loop

(c)Hidden using the receive buffer.

[Source processor]	[Destination Processor]
loop:Call Send	loop:Call Receive
Computation	Computation
goto loop	Wait until Receive finishing
Wait until all Sends finishing	Release MCB
Release all MCBs	goto loop

(d)Hidden using the send buffer.

図 3.11: Cannon アルゴリズムの通信方式

なお，Cannon アルゴリズムの各部分行列のサイズは $r = l^2$ である．このとき，アルゴリズムにおける一回の繰り返し処理で，$2r$ ワードのデータが送受信される．

3.4.2.2 通信遅延と実行時間

図 3.12に，CM5 の 64 プロセッサを用いた Cannon の並列行列乗算アルゴリズムの実行時間と通信方式の関係を示す．図 3.12(a)，(b)，(c) は，それぞれ行列サイズ $N = 8$，64，1024 に対する，単純通信，隠蔽通信，受信バッファ付加，送信バッファ付加，送受信バッファ付加を用いた場合の実行時間を示す．図の波枠部は実行時間，実枠部は計算処理時間である．通信遅延時間は，この両時間の差で表される．

行列サイズが小さい場合は，通信遅延時間が計算処理時間より大きいため，通信は隠蔽できない．一方，行列サイズが大きい場合は以下のようになる．単純通信の通信遅延時間は，計算処理時間より大きく，並列処理性能は小さくなる．隠蔽通信は，遅延を少なくできるが，行列サイズが大きくなるに従い増加するため，その効果は限定される．隠蔽通信に受信バッファを付加すると，遅延を更に小さくすることができる．また，送信バッファを付加すると，行列サイズが大きな場合に遅延はほぼ隠蔽され，実行時間は計算処理時間にほぼ等しくなる．送受信双方のバッファを付加した場合，行列サイズが大きな場合に通信遅延は完全に隠蔽され，誤差の範囲内程度になる．

これらの実験結果から，$4r$ ワード長の送受信バッファがあれば，CM5 上で Cannon アルゴリズムを効率的に実行できることが分かる．

3.4.3 LogP と LogPQ による並列行列乗算アルゴリズムの動作解析

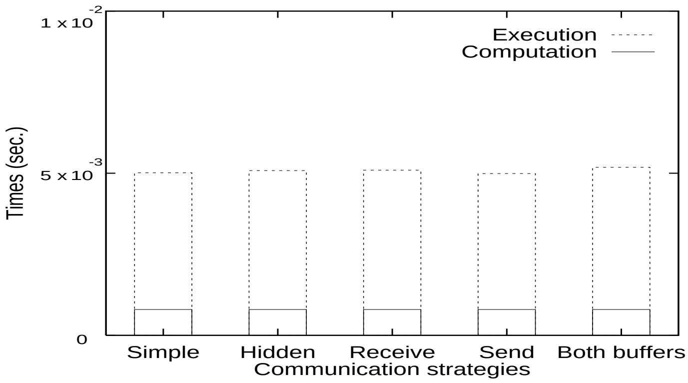
3.4.3.1 並列処理時間の解析

Cannon アルゴリズムの送受信バッファ付き隠蔽通信を用いて行う並列アルゴリズムを，LogP モデルと LogPQ モデルにより解析する．

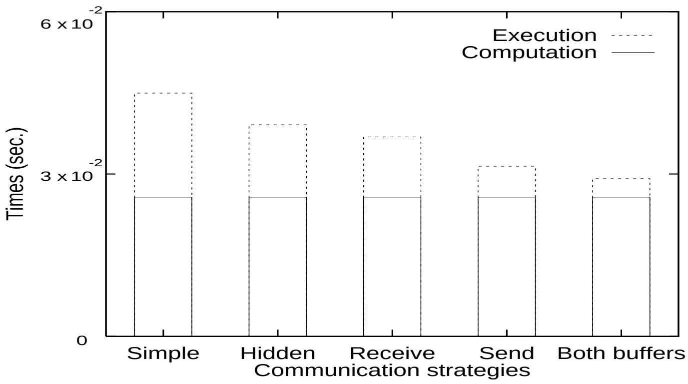
Cannon アルゴリズムの並列処理時間 T について考える．サイズ l の部分行列乗算一回の実行時間 t_M は，

$$t_M = t_f + t_s \cdot l^3 \tag{3.4}$$

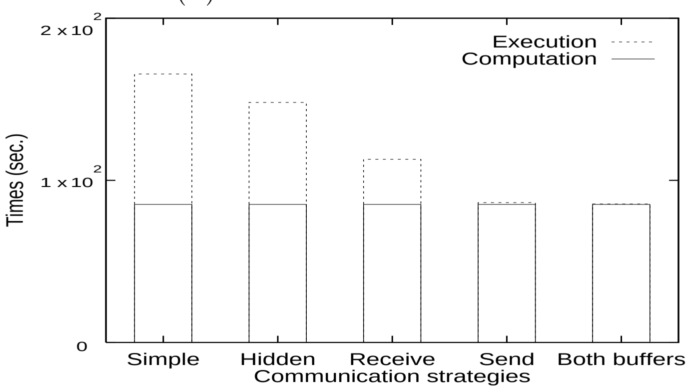
で与えられる．ここで，t_f は計算処理時間の固定部分，t_s は一回のスカラ乗算の（周辺処理



(a) Matrix size $N = 8$.



(b) Matrix size $N = 64$.



(c) Matrix size $N = 1024$.

図 3.12: 並列計算機 CM5 の 64 プロセッサによる Cannon アルゴリズムの実行時間と計算処理時間

を含めた）実行時間である．計算処理全体の実行時間 t_cは,

$$t_c = p \cdot t_M. \tag{3.5}$$

したがって，並列処理時間 Tは,

$$T = t_c + p \cdot t_{com} \tag{3.6}$$

となる．ここで，t_{com}は通信処理の実行時間である．

1 メッセージを基本とする LogP では，各プロセッサは各部分行列 A_{ij}，B_{ij}を 1 個の $r = l^2$ ワード長メッセージで送信する．このとき，通信処理の実行時間 t_{LogP_1}は，次式で表される．

$$t_{LogP_1} = \begin{cases} o^* + 3max(o^*, g^*) \\ \qquad\qquad\qquad\qquad (t_M \geq L^*) \\ (L^* - t_M) + (o^* + 3max(o^*, g^*)) \\ \qquad\qquad\qquad\qquad (\text{otherwise}). \end{cases} \tag{3.7}$$

1 ワード長メッセージを基本とする LogP では，各プロセッサが各部分行列を r個の 1 ワード長メッセージで送信する．このとき，通信処理の実行時間 t_{LogP_r}は，次式で表される．

$$t_{LogP_r} = \begin{cases} o_0 + max(o_0, g_0) \cdot (4r - 1) \\ \qquad\qquad\qquad\qquad (t_M \geq (L_0 - (2r-1) \cdot max(o_0, g_0))) \\ (L_0 - (2r-1) \cdot max(o_0, g_0) - t_M) + (o_0 + max(o_0, g_0) \cdot (4r - 1)) \\ \qquad\qquad\qquad\qquad (\text{otherwise}). \end{cases} \tag{3.8}$$

LogPQ モデルでは，各プロセッサは各部分行列を r個の 1 ワード長メッセージで送信する．このとき，通信処理の実行時間 t_{LogPQ}は，次式で表される．

$$t_{LogPQ} = \begin{cases} 4(o \cdot r + n) \\ \qquad\qquad\qquad\qquad (t_M \geq (L + (2r-1)g - 2r \cdot o - n)) \\ (L + (2r-1)g - 2r \cdot o - n - t_M) + 4(o \cdot r + n) \\ \qquad\qquad\qquad\qquad (\text{otherwise}). \end{cases} \tag{3.9}$$

LogP (1 messgae)	LogP (1-word length)	LogPQ
$L^* : 1.25 \cdot 10^{-4}$	$L_0 : 1.77 \cdot 10^{-4}$	$L : 2.44 \cdot 10^{-4}$
$o^* : 1.71 \cdot 10^{-4}$	$o_0 : 1.19 \cdot 10^{-5}$	$o : 4.59 \cdot 10^{-6}$
$g^* : 5.50 \cdot 10^{-5}$	$g_0 : 3.44 \cdot 10^{-6}$	$g : 3.44 \cdot 10^{-6}$
		$n : 9.75 \cdot 10^{-5}$

(sec.)

表 3.1: 並列計算機 CM5 の LogP および LogPQ パラメータの値

3.4.3.2 LogP と LogPQ のパラメータ

並列計算機 CM5 上での各実行時間 t_s，t_fを測定した結果，$t_s = 5.98 \cdot 10^{-6}(sec)$，$t_f = 1.55 \cdot 10^{-4}(sec)$ となる．

また，CM5 に対する LogP モデルと LogPQ モデルのパラメータを表 3.1に示す．ここでは，通信パラメータをクロックでなく秒で表す．通信パラメータの基準点として，部分行列サイズ $l = 4$ の場合，すなわち 16 ワード長メッセージに対する実験により，LogP および LogPQ パラメータを求めた．

3.4.3.3 LogP と LogPQ の比較

図 3.13に，並列計算機 CM5 上で Cannon アルゴリズムを実行した並列処理時間と，LogP と LogPQ モデルにより求めた並列処理時間を示す．

LogP モデル（1 メッセージ）による解析結果は，$N < 32$ では実際より大きく，$N > 32$ ではその逆になる．LogP モデル（1 ワード長メッセージ）では，通信路のバッファリングとパラメータ n を考慮しないため，LogP モデル（1 メッセージ）と逆の結果になる．一方，LogPQ モデルは，LogP モデルに比べ実際の並列処理時間に近い性能を予測できることが分る．

3.4.4 まとめ

本節では，通信バッファを考慮した実用的な並列計算モデル LogPQ について議論した．並列行列乗算アルゴリズムを並列計算機 CM5 上で実行し，従来の LogP モデルより LogPQ

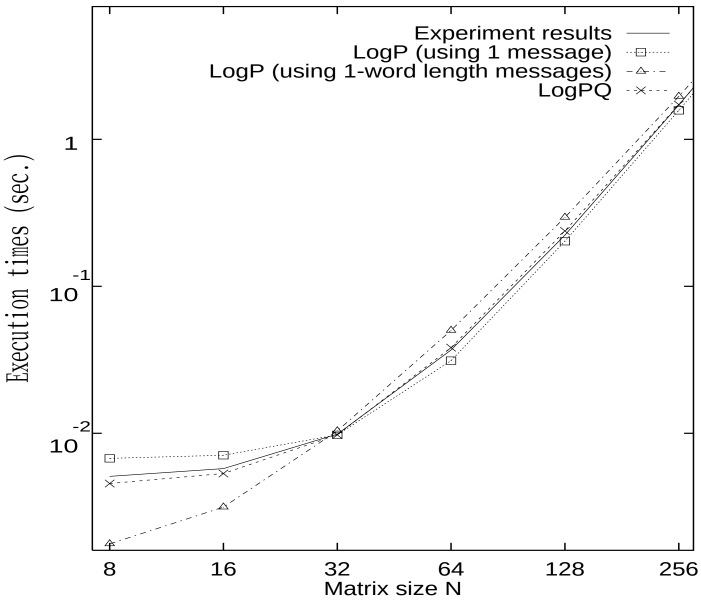


図 3.13: Cannon アルゴリズムの並列計算機 CM5 の 64 プロセッサによる実行時間と LogP および LogPQ モデルによる予測時間の比較

モデルが並列アルゴリズム解析に有用であることを示した．また，LogPQ モデルの通信バッファを用いて通信遅延をより隠蔽し，並列アルゴリズムの効率を改善できることを示した．

3.5 むすび

本章では，さまざまな並列アルゴリズムを実際の並列計算機やエミュレータで実行することにより，実用並列計算モデル LogPQ の有用性や実用性に対する実験的評価を行った．

変形 CG アルゴリズムの LPRAM モデル上での実験的評価により，同期処理や通信遅延が並列アルゴリズムの実行時間に大きな影響を与えることを確認した．

また，並列多倍長 GCD アルゴリズムの LogPQ モデル上での実験的評価により，並列計算機の通信性能の各成分がどのように並列アルゴリズムの性能に影響を与えるかを示し，LogPQ モデルが効率的な並列アルゴリズム構築に対し有用であることを明らかにした．

更に，並列行列乗算アルゴリズムの並列計算機 CM5 上での実験的評価により，通信にバッファを用いることによりアルゴリズムの通信遅延をより隠蔽でき，並列アルゴリズム

の効率を LogPQ モデルを用いて検討できることを示した．また，従来の LogP モデルより LogPQ モデルが実用性が高いことを示した．

第 4 章

超並列・分散ネットワークの設備配置

4.1　超分散システムにおける通信ネットワーク

近年，さまざまな超並列・分散計算システムが実用化され普及しつつある．そして，効率的な並列プログラム開発環境が必要とされている．2 章と 3 章では，超並列・分散計算システムの並列プログラム開発を重視した並列計算機モデルについて議論した．一方，先端科学技術分野などで用いる計算能力は年々増加し，計算能力に対する需要は供給を大きく上回っている．現在，より高性能な超並列・分散計算システムが必要とされている．超並列・分散計算システムは，複数個のプロセッサと，プロセッサ間を結ぶ通信ネットワークで構成される．超並列・分散計算システムの性能は，使用する通信ネットワークの性能に大きく依存する．そこで本章では，超並列・分散計算機システムの効率的な通信ネットワーク構築について議論する．

大規模分散システムは，さまざまな通信路を用いて通信ネットワークを構築する．一方，超並列・分散計算機システムは，プロセッサ間の距離が比較的短距離である．そして，プロセッサ間通信において，ネットワークの中継プロセッサ数（頂点数）が通信時間を支配する．

超並列・分散計算機システムの通信ネットワークとして，並列計算機 CM-5 等のように木構造のネットワークを用いる場合と，並列計算機 nCUBE2 等のように一般のネットワークを用いる場合がある．一般のネットワークの利点は，通信バンド幅が大きく，対故障性が高いことである．しかし，配線数が多く，また配線が複雑になるので，通信ネットワークシステムの構築コストが大きくなる．一方，木構造ネットワークは，通信バンド幅が小さ

く対故障性が低い．しかし，配線数が少なく，配線が単純な形状なので，通信ネットワークシステムの構築コストを小さくできる．一般に，LAN などの小規模ネットワークシステムから WAN などの大規模ネットワークシステムまで，木構造の計算機ネットワークを用いる場合が多い．超並列・分散計算機システムにおいても，木構造ネットワークを用いる場合が少なくない．

通信ネットワークシステムの高性能化の手段の一つに，全通信路の高速化がある．たとえば，速度 10Mbps の通信路を 100Mbps に変更した場合，システムの最大通信コストも平均通信コストも十分の一になる．しかし，通信システム全体を交換することから，その構築コストは非常に大きくなる．そこで，通信システムの一部のみを高速化することにより，通信ネットワークシステムの構築コストを低減でき，予算に合わせたシステムを構築できる．このとき，高速化する通信路の選択により，同じ構築コストの通信システムの通信コストが増減する．

ネットワーク上に設備を適正に配置することにより，効率的な通信ネットワークシステムを構築することができる．ネットワークはグラフで表され，グラフの各辺は通信路に，各頂点は通信路端の I/O チャネル部に対応する．設備は高速通信路に対応し，その付加により通信ネットワークシステムの性能が向上する．

ネットワークに対する設備配置は 1960 年代より盛んに研究開発され，さまざまな点形状設備が提案されてきた [22]．木構造ネットワークに対する設備配置，一般のネットワークに対する設備配置，各種評価関数を用いた設備配置，複数個の設備の配置，階層的な設備の配置等のさまざまな研究が行われた．また，設備配置手法として，動的プログラム法などの基本的手法，branch-and-bound 法 [23] や焼きなまし法 [24] などのヒューリスティック手法，primal-dual 法 [25] などの近似手法を用いた方法が提案されてきた．たとえば，Gavish ら [26] は，木構造ネットワークに距離和最小となる 2 個の点形状設備（2-median）を求める $O(n \log n)$ 時間のアルゴリズムを提案した．Tamir[27] は，木構造ネットワークに距離和最小となる p 個の点形状設備（p-median）を求める $O(pn^2)$ 時間のアルゴリズムを提案した．Chardaire ら [28] は，時間変化する需要に対応した動的な点形状設備を配置するヒューリスティック手法を提案した．Righini[24] は，一般のネットワークに対する p-median を求めるヒューリスティック手法を提案した．Krumke ら [29] は，一般のネットワークにα番目に近い設備までの偏差が最小となる p 個の点形状設備（p-center$^{(\alpha)}$）を求める近似手法を提案した．その他，さまざまな研究が行われた．

現在，文献 [30] で概説されたように，さまざまな構造の設備が提案されている．点形状設備は，クライアント・サーバシステムの構築に適する．ハブ [31] は，ネットワークの各点からの集結点の集まりであり，階層型ネットワークシステムの構築に適する．コアは，葉数指定の木形状設備であり，階層型キャッシュを用いるデータベースシステムの構築に適する．また，設備に対するさまざまな評価指標が提案されている．偏差は，ネットワークの各点と設備との距離の最大値であり，ネットワークの最大通信コストに対応する．距離和は，ネットワークの各点と設備との距離の総和であり，ネットワークの平均通信コストに対応する．他にもさまざまな評価指標が存在する．

設備の一つに，木形状やパス形状設備がある．木構造ネットワークにおけるパス形状や木形状の設備配置は，Slater により提案された [32][33]．Morgan ら [34] は，頂点数 n のネットワークの，各頂点との距離の総和（距離和）が最小となるパス形状設備を算出する $O(n)$ 時間のアルゴリズムを提案した．Minieka[35] は，距離和を最小とするサイズ指定のパス形状と木形状設備を求める各々$O(n^3)$ 時間，$O(n^2)$ 時間のアルゴリズムを提案した．Tamir ら [36] は，一般化した評価関数を用いて p 個の木形状設備の配置を行う $O(n^3p^2)$ のアルゴリズムを提案した．他にもさまざまな設備が提案され研究された [37]．

サイズ指定の木形状やパス形状設備は，十分に高速で通信コストの小さい高速通信路に対応する．その設備サイズは，高速通信路の配線数に対応し，高速通信路の構築コストの制限を表す．そして，この高速通信路は従来の通信路に並行に配置されるので，通信ネットワークシステムの構築・保守コストの低減が可能である．偏差最小の設備は，特定の構築コストで最大通信コストを最小にする設備である．距離和最小の設備は，特定の構築コストで平均通信コストを最小にする設備である．

本章および 5 章では，簡単化のため通信トラフィックは考慮せず，通信システムの性能を通信レイテンシで評価する．すなわち，各通信路の通信コストはその通信路のレイテンシであり，各通信の通信コストは通信に用いる通信路の通信コストの和となり，この通信コストを用いて通信システムの性能を評価する．たとえば各通信路の通信コストを通信路（各辺）の長さで表した場合，二点間の通信コストは二点間の通信路の長さの和となる．なお，設備サイズは設備に含まれる通信路（各辺）の長さの和で表される．

本章では，各通信路の構築コストと通信コストがそれぞれ等しい場合，すなわち各辺の長さを 1 とするネットワークについて議論する．また，対象とするネットワークとして，一般に使用されることの多い木構造ネットワークを用いる．大規模な通信ネットワークでは

Size of facility: p	0 (no ficility)	$\log n$	$\sqrt{n}$	$n/2$
Mean cost for communication	$\log n$	$\log n - \log\log n$	$\log n/2$	1
Maximum cost for communication	$2\log n$	$2(\log n - \log\log n)$	$\log n$	2

表 4.1: 頂点数 n の平衡二分木上に木形状設備を配置した場合の通信コスト

一般に低次ネットワークが用いられるので，各点の次数を制限した場合について議論する．

通信システムの構築コストは設備サイズで表し，サイズを指定した木形状設備の最適配置により通信システムを構築する．また，超並列・分散計算機システムの通信は多対多で行われる場合が多い．そこで，設備配置の評価指標に平均通信コストを用いる．本章では，木形状設備の構築コストを小さくするために小サイズの部分設備を用いた木形状設備の最適配置問題について検討する．具体的には，頂点数 n で次数が $O(\log n/\log\log n)$ である木構造ネットワーク上への等分割可能な木形状設備の配置について議論する．また，複数の通信に対応するために複数個の木形状設備を重ねずに配置する複数設備配置問題について検討する．具体的には，頂点数 n で次数が $O(\log n/\log\log n)$ であるバンド幅 1 の木構造ネットワーク上への複数個の木形状設備の配置について議論する．

次節では，木構造の通信ネットワークに対する木形状の高速通信路設備配置の有用性について議論する．

4.2 木構造ネットワーク

本節では，木構造ネットワークにおける木形状設備配置の具体例として，通信ネットワークが平衡二分木の場合について議論する．なお，ネットワークの各辺の長さは 1 とする．

4.2.1 設備サイズと通信コスト

木形状設備の設備サイズと通信コストの関係について議論する．通信ネットワークが平衡二分木の場合，距離和最小の木形状設備と偏差最小の木形状設備は等しく，これを最適

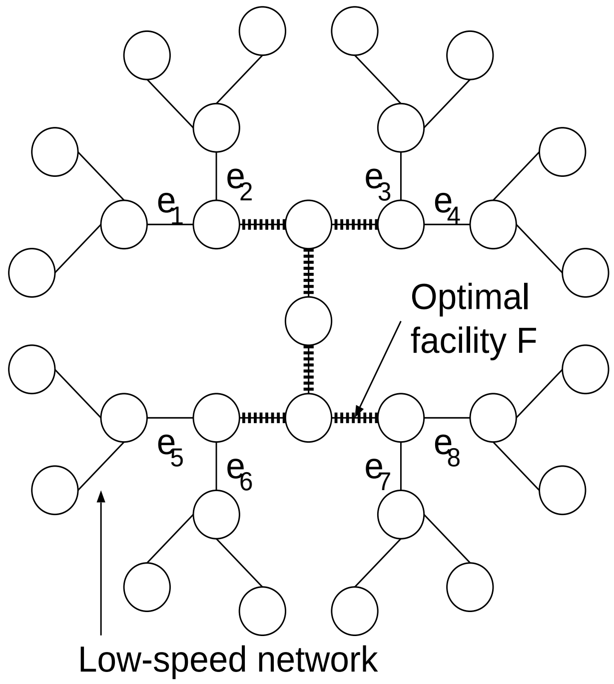


図 4.1: 平衡二分木上の最適な設備 F

な木形状設備とよぶ．平衡二分木に対する最適な設備は，根が中心となる平衡二分木である．このとき，通信ネットワークの平均および最大通信コストは設備の距離和および偏差のそれぞれほぼ倍となる．また，設備を配置しない場合の通信ネットワークの平均および最大通信コストは，サイズ 0 の最適な木形状設備を配置した場合の距離和および偏差のそれぞれほぼ倍となる．

図 4.1に，頂点数 31 の平衡二分木とサイズ 6 の最適な木形状設備 Fを示す．配置する設備のサイズを増加させることにより，通信ネットワークの通信コストを削減できる．たとえば，図 4.1に対するサイズ 14 の最適な木形状設備 F'は，図の Fに辺 e_1〜e_8を追加したものであるが，F'の距離和と偏差は Fの場合のほぼ半分になり，通信ネットワークの平均および最大通信コストは半減する．表 4.1に，頂点数 n の平衡二分木上にサイズ p の設備を配置した場合の通信ネットワークの平均および最大通信コストの概数を示す．たとえばサイズ$\sqrt{n}$の設備を用いることにより，設備のない場合に比べ，通信ネットワークの平均および最大通信コストを半減できる．

Size of facility: p	0 (no facility)	$\log n$	$\sqrt{n}$	$n/2$
Construction cost	$n-1$	$(n-1)+(\log n$ $+1)(\log n+2)/2$	$(n-1)+(\sqrt{n}$ $+1)(\sqrt{n}+2)/2$	$(n-1)+(n/2$ $+1)(n/2+2)/2$

表 4.2: 頂点数 n の平衡二分木上に完全ネットワークによる設備を配置した場合の構築コスト

4.2.2 設備サイズと構築コスト

設備サイズと構築コストとの関係について議論する．

図 4.2に，設備 Fの完全ネットワークを用いた実装を示す．設備を完全ネットワークを用いて実装する場合，設備内の通信は設備の各二点間でダイレクトに行うことから，低コストの通信路を用いることができる．したがって，設備のサイズ単位あたりの構築コストは低い．しかし，サイズ p の設備の構築に${}_{p+1}C_2$個の通信路が必要なので，設備サイズに対する構築コスト増加は $O(p^2)$ と大きい．一方，設備を一つのブロードキャスト通信路を用いて実装する場合，多対多通信やコリジョン対策が必要なことから，高コストの通信路が必要である．したがって，設備のサイズ単位あたりの構築コストは高い．しかし，設備サイズに対する構築コスト増加は $O(p)$ と小さい．

低速通信路を用いた完全ネットワークにより構築するサイズ p の設備の構築コストは $(p+1)(p+2)/2$ である．すなわち，配置する設備のサイズ増加により，通信ネットワークの構築コストは増加する．表 4.2に，サイズ p の設備の構築コストの概数を示す．たとえばサイズ$\sqrt{n}$の設備の構築コストが，およそ $n/2$ になることがわかる．

なお，ブロードキャスト通信路のサイズ単位あたりのコストを$\sqrt{n}/2$ 以下に抑えることができれば，サイズ$\sqrt{n}$の設備の構築コストを更に低減である．

4.2.3 実用的な設備の配置

頂点数 n の平衡二分木ネットワークにサイズ$\sqrt{n}$の設備を付加することにより，従来の 1.5 倍の構築コストで通信コストを半減できることが分る．

しかし，実用性の観点では，実際の通信ネットワークシステム構築において改良すべき点がある．以下に，木形状設備の問題点と，その対策を示す．

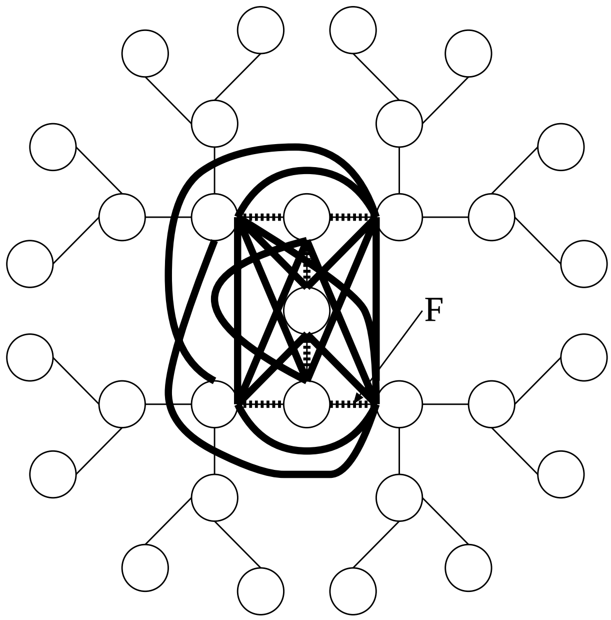


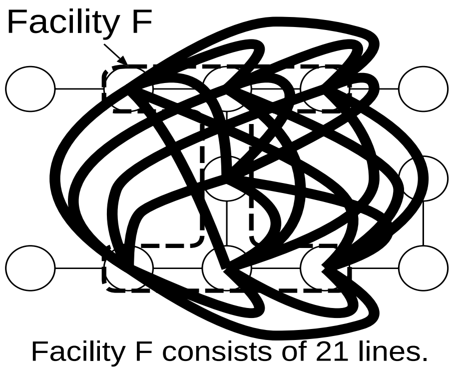
図 4.2: 完全ネットワークによる設備を配置した平衡二分木ネットワーク

（１）設備の構築コストの低減

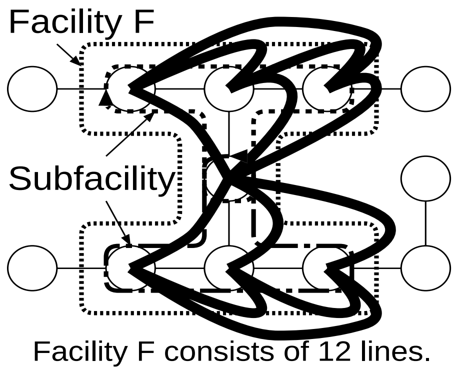
一般に，高速通信路設備を用いて通信ネットワークの高速化を行うには，膨大なコストが必要になる．そこで，小サイズの高速通信路設備を複数個用いて通信ネットワークを構築することにより，コストを抑えつつ高速化を行うことが期待されている．また，量産効果を考慮すると，同一サイズの高速通信路設備のみを用いてネットワークを構築することは，コスト低減に有用である．

たとえば，サイズ $p/2$ の部分設備を完全ネットワークで構築すると，その構築コストは $(p/2+1)(p/2+2)/2$ となる．よって，サイズ $p/2$ の部分設備を二つ用いて設備を構築した場合，設備の構築コストは $(p/2+1)(p/2+2)$ となる．これは，完全ネットワークによるサイズ p の設備の構築コスト $(p+1)(p+2)/2$ のおよそ半分である．この例として図 4.3を示す．図 4.3(a) の設備サイズは 6 なので，設備の構築コストは 21 である．図 4.3(b) のようにサイズ 3 の部分設備を二つ用いて設備を構築した場合，設備の構築コストは 12 とほぼ半減する．

しかし，同一サイズの設備を木構造ネットワーク上に配置することは，今まで十分に研



(a) Facility consists of complete connections.



(b) Facility consists of two subfacilities.

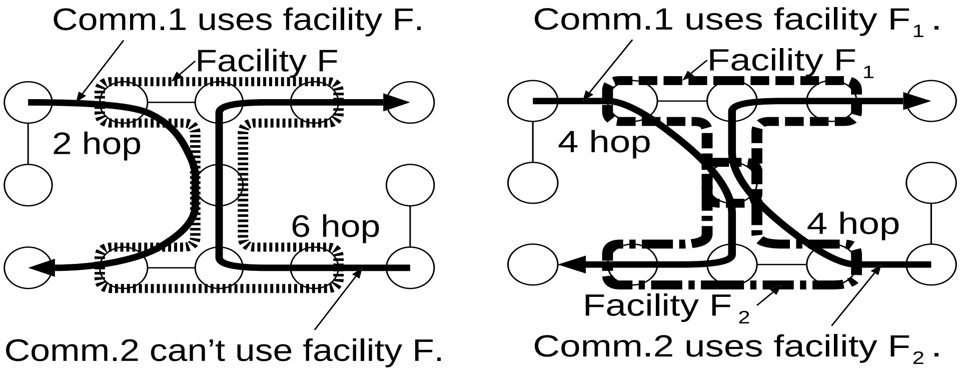
図 4.3: 部分設備による構築コスト削減

究されていない．

（２）複数の通信要求への対応

一つの高速通信路設備を配置した通信ネットワークは，同時に一つの通信しか高速化できない．一般に，複数の通信を同時に行う場合が多いので，一つの高速通信路設備を用いたシステムの性能向上は不十分な場合がある．そこで，複数個の高速通信路設備を配置し，一つの通信が一つの設備のみ使用するようルーティングすることにより，通信量の多いシステムにおいても十分な性能向上が期待できる．また，量産効果を考慮すると，同一サイズの高速通信路設備のみを用いてネットワークを構築することは，コスト低減に有用である．この例を図 4.4に示す．図 4.4(a) は一つの設備を配置した場合である．通信 1 と通信 2 が同時に行われるとき，通信 1 の通信コストは 2 になるが，通信 2 は設備を使用できず通信コスト 6 になる．そして，この通信 2 の通信コストがシステム全体の実行時間に影響する場合がある．図 4.4(b) のように設備二つを用いることにより，通信 1 と通信 2 の通信コストをともに 4 にすることができる．

しかし，同一サイズの設備を木構造ネットワーク上に複数個配置することは，今まで十分に研究されていない．



(a) Communications using facility. (b) Communications using two facilies.

図 4.4: 複数個の設備による通信コスト削減

4.2.4 まとめ

本節では，平衡二分木における設備配置について議論し，木形状設備を配置することにより低速通信路と高速通信路の混在する低コストで高性能な木構造通信ネットワークシステムを構築できることを明らかにした．実用性の観点から，実際に通信ネットワークシステムを構築する問題点について検討し，同一サイズの部分設備で構成される設備配置問題や，同一サイズの複数個の設備の配置問題が重要であることを示した．

以下の節では，通信ネットワークにこれらの実用的設備を配置する方法について議論する．

4.3 等分割可能な木形状設備の配置

本節では，頂点数 n で次数が $O(\log n/\log\log n)$ である木構造ネットワーク上への等分割可能な木形状設備配置問題について議論する．なお，プロセッサ間通信では，ネットワークの中継プロセッサ数（頂点数）が通信時間を支配するため，通信路（各辺）の長さが 1 の場合について議論する．

先ず，木構造ネットワーク上の等分割可能な木形状設備問題およびネットワーク上の設備配置に対する評価指標である距離和について説明する．次に，最適配置問題を解くアルゴリズムで用いる表記を述べる．また，等分割可能な木形状設備の性質をまとめ最適配置問題を解くアルゴリズムを提案する．更に，その実行時間について議論する．

4.3.1　ネットワーク上の設備配置

木構造ネットワークは木 Tで表される．木 Tの頂点集合を $V(T)$，辺集合を $E(T)$ と表す．木 Tの各辺 $(v, w) \in E(T)$ は，2 頂点 v, $w \in V(T)$ を長さ $d(v, w)$ でつないでいる．本章では，ネットワークの中継点数を重要と考え，長さ $d(v, w)$ は 1 とする．木 T'の頂点数を $|T'|$ と表す．n は，木 Tの頂点数 $|T|$ である．木 T'の部分木 T''は，$V(T'') \subset V(T')$, $E(T'') \subset E(T')$ なる木である．なお，頂点 vのみの部分木を$\tilde{v}$と表す．$V(\tilde{v}) = \{v\}$，$E(\tilde{v}) = \phi$である．

木構造ネットワーク上への設備の配置を考える．設備 Fは木 Tの部分木である．設備 Fのサイズ lは，設備の辺の長さの総和である．本章では各辺の距離を 1 とするため，lは設備の辺の数 $|E(F)|$ となる．

設備 Fが部分木 S_i $(1 \leq i \leq t)$ で構成されるとは，次式が成立することである．

$$\begin{aligned} V(F) &= \cup_{i=1}^{t} V(S_i), \\ E(F) &= \cup_{i=1}^{t} E(S_i), \\ E(S_i) \cap E(S_j) &= \phi \quad (i \neq j). \end{aligned} \tag{4.1}$$

設備 Fの部分設備は，Fを構成する部分木である．部分設備のサイズは，部分設備の辺の長さの総和である．設備 Fが部分設備 S_i $(1 \leq i \leq t)$ に分割できるとは，Fが S_i $(1 \leq i \leq t)$ で構成されることである．t 個に等分割可能な設備とは，t 個の同一サイズの部分設備に分割できる設備である．

図 4.5に，木構造ネットワークにおける等分割可能な木形状設備配置の例を示す．設備 Fは，サイズ 4 の 3 個の部分設備 S_1，S_2，S_3に分割できる．

ネットワーク上への設備配置問題の評価指標として距離和がある．2 頂点 v，w間の距離 $d(v, w)$ は，頂点 v，w間の経路 (パス) の辺の長さの総和である．頂点 vと設備 Fとの距離 $d(v, F)$ は，vと Fとの最少距離

$$d(v, F) = \min_{w \in V(F)} d(v, w). \tag{4.2}$$

である．設備 Fの距離和 $d(F)$ は，Fと Tの各頂点 u との距離の総和で，次式で与えられる．

$$d(F) = \sum_{v \in V(T)} d(v, F). \tag{4.3}$$

図 4.5の場合，Fまでの距離が 1 である頂点数が 13，Fに含まれる頂点数が 10 なので，$d(F) = 13$ となる．

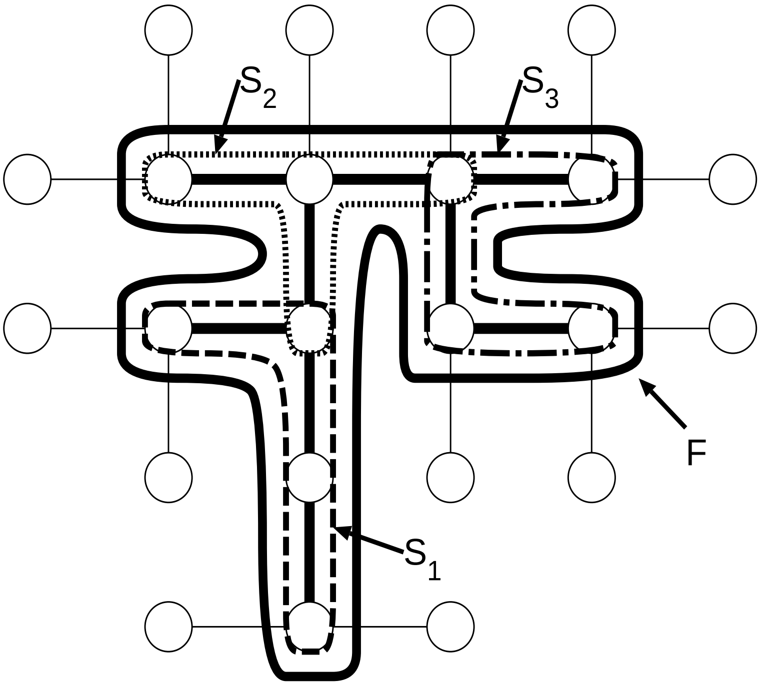


図 4.5: 木構造ネットワーク上における等分割可能な木形状設備（設備 Fは 3 つの同一サイズの木形状設備 S_1，S_2，S_3で構成される）

本章では，距離和 $d(F)$ が最小となる設備を最適な設備と定義する．図 4.6に，最適な等分割可能な設備の例を示す．図 4.6の太枠線で示される部分木 Fは，図の枠線で囲まれた 2 個の部分設備 S_1，S_2に等分割可能な設備である．Fの距離和は，$d(F) = 17$ となる．設備 Fは，2 個に等分割可能なサイズ 8 の設備のなかで最適である．一方，図の太破枠線で示される部分木 F'は，サイズ 8 の最適な設備で，距離和は $d(F') = 15$ である．一般に，等分割可能という条件の付加により，最適な設備の距離和は増加する．

本章では，木構造ネットワーク Tに対し，設備サイズ lおよび分割数 t を指定した場合の，最適な等分割可能な設備 Fを求めるアルゴリズムについて議論する．但し，Tの各頂点の次数は，$\alpha \log n / \log\log n$ 以下と仮定する．なお，αは定数である．

この次数制限は $O(\log n)$ より強いが，現実のマルチプロセッサを構成する疎結合ネットワークで本制限を満すものも多い．たとえば，k-ary d-cube[38] において，$k = \log n$，$d = \log n / \log\log n$ の場合，各頂点の次数は $2 \log n / \log\log n$ となる．

なお，辺の長さが任意である一般的な木構造ネットワークの場合は，設備の分割数が 1

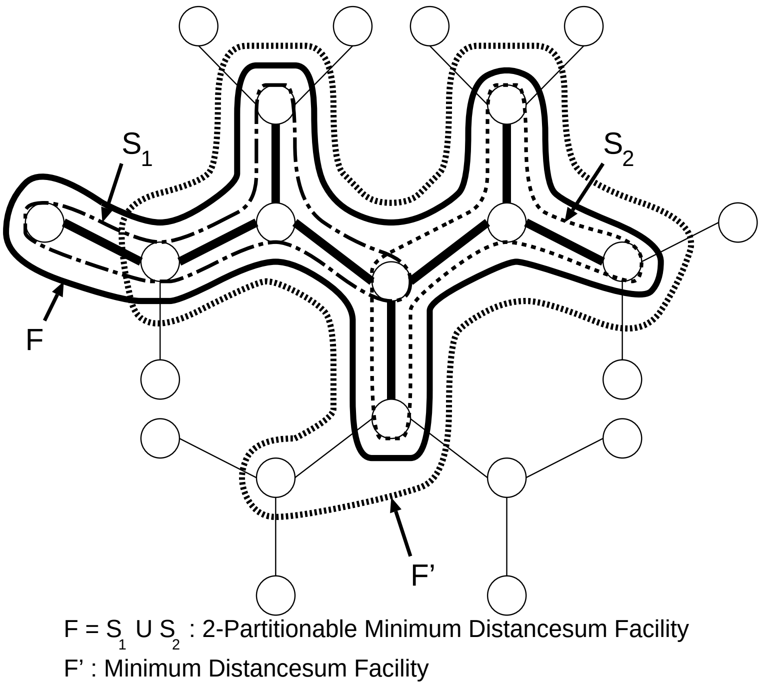


図 4.6: 最適な等分割可能な木形状設備と最適な木形状設備

であっても，本問題は NP-困難であることが知られている [37].

4.3.2 表記

木構造ネットワーク T の，t 個に等分割可能なサイズ l の設備を F とする．F を構成する部分設備のサイズ l' は，次式で与えられる．

$$l' = l/t \tag{4.4}$$

なお，l' が整数でない場合，等分割可能な設備はない．

部分木 T' に対する設備 F の距離和 $d(T';F)$ は，以下の式で表される．

$$d(T';F) = \sum_{v \in V(T')} d(v,F). \tag{4.5}$$

T の中点（median）q は，距離和 $d(\tilde{v})$ が最少となる頂点 v である．T に対し中点 q を根（root）とする有向木を T_q と表す．図 4.7に T_q の例を示す．頂点 v の子頂点の数は d_v と表す．v の子頂点は v_i（$1 \leq i \leq d_v$）と順序付けて表す．なお，中点 q は重心（centroid）である．

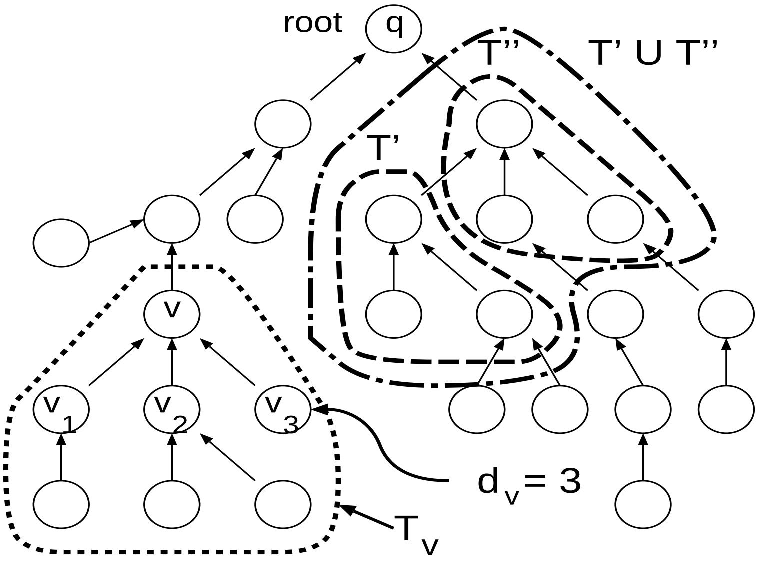


図 4.7: 根が中点 qである有向木 T_q

すなわち，qを含まない Tの部分木の頂点数が $|T|/2$ 以下になることが知られている [33]．本章では，子頂点数 d_vは次式のように制限する．

$$d_v \leq \Delta = \alpha \log n / \log\log n. \tag{4.6}$$

部分木の結合を $T' \cup T''$とすると，次式を満たす．

$$\begin{aligned} V(T' \cup T'') &= V(T') \cup V(T''), \\ E(T' \cup T'') &= \textstyle\bigcup_{v,w \in V(T' \cup T''),(v,w) \in E(T)} \{(v,w)\}. \end{aligned} \tag{4.7}$$

また，部分木の削除を T'/T''とすると，

$$\begin{aligned} V(T'/T'') &= V(T')/V(T''), \\ E(T'/T'') &= \textstyle\bigcup_{v,w \in V(T'/T''),(v,w) \in E(T)} \{(v,w)\}. \end{aligned} \tag{4.8}$$

が与えられる．

部分木 T'の頂点 vで，T_qの根 qであるか，その親頂点が $V(T')$ に含まれないものを，部分木 T'の根とよぶ．頂点 vを根とする最大の部分木を T_vと表す．

また，i 個の要素の置換を表す関数の集まりを$\pi[i]$ と表し，その数を $|\pi[i]|$ と表す．各々の置換関数を$\pi[i,j]$（$1 \leq j \leq |\pi[i]|$）と表す．$\pi[i,j](k)$（$1 \leq k \leq i$）は，入力 kに対する関数の

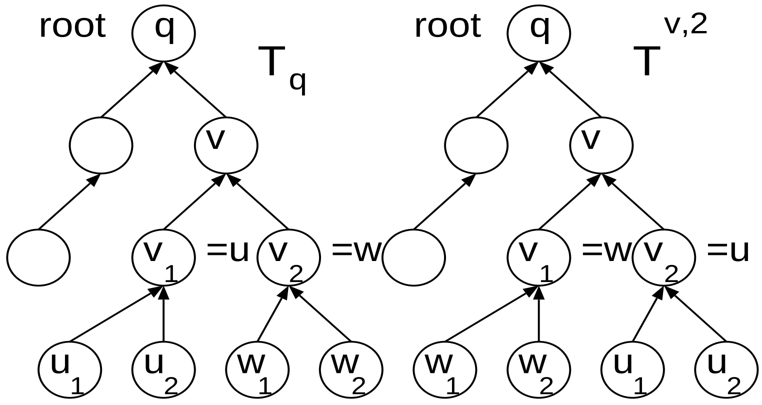


図 4.8: 有向木 T_qと $T^{v,2}$

出力である．たとえば，2 要素の置換関数は，$\pi[2,1](1)=1$，$\pi[2,1](2)=2$ と$\pi[2,2](1)=2$，$\pi[2,2](2)=1$ の 2 つである．

頂点 $v_{\pi[d_v,j](k)}$は，vの$\pi[d_v,j](k)$ 番目の子頂点である．有向木 T_qに対し，頂点 vの各子頂点 v_k（$1\le k\le d_v$）を $v_{\pi[d_v,j](k)}$に置き換えたものを，有向木 $T^{v,j}$と表す．図 4.8に $T^{v,j}$の例を示す．$\pi[2,2]$ が 2 要素の反転なので，$T^{v,2}$は T_qに対し頂点 vの子頂点 u，wの順序を反転したものとなる．

$T^{v,j}$における頂点 vと，vの子頂点 v_k（$1\le k\le i$）を根とする部分木 $(T^{v,j})_{v_k}$との結合を $T_v^j(i)$ とし，次式で表す．但し，$i\le d_v$である．

$$T_v^j(i)=v\cup\bigcup_{k=1}^{i}(T^{v,j})_{v_k}. \tag{4.9}$$

次に，(4.6) 式で定義されるΔは次式の関係を満足する．

$$\begin{aligned}\Delta! &\le 2^{\Delta\log\Delta}\\ &= 2^{(\alpha\log n/\log\log n)\log(\alpha\log n/\log\log n)}\\ &< n^{\alpha(1+\log\alpha)}.\end{aligned} \tag{4.10}$$

(4.10) 式より，$i\le\Delta$の場合，i 個の要素の置換関数の総数 $|\pi[i]|$ は，次式のように求まる．

$$|\pi[i]|<n^{\alpha(1+\log\alpha)}. \tag{4.11}$$

4.3.3　等分割可能な設備配置アルゴリズム

4.3.3.1　アルゴリズムの概要

Tamir 等 [36] は，動的プログラミング法を用いた p 個の木形状設備の配置アルゴリズムを提案した．ここでは，Tamir のアルゴリズムを用いて，木構造ネットワーク T における最適な等分割可能な設備を求める．設備配置アルゴリズムは，T に対し中点 q を根とする有向木を生成し，葉から根に向けて順に頂点 v を選択することにより，T_v に対する最適な設備を求める．そして，T_q 上の設備を用いて，T の最適な等分割可能な設備を導出する．次節では，T_v に対する最適な設備の算出に用いる，等分割可能な設備の性質を示す．

4.3.3.2　等分割可能な設備の性質

木構造ネットワーク T において最適な等分割可能な設備 F_{opt} は，T の中点 q を根とする有向木 T_q に対して，以下に示す性質を有する．

性質 1　設備 F が t 個に等分割可能であるとき，F に含まれる頂点を v，v に隣接し F に含まれない頂点を v_0 とする．ある部分木 $F \cup \tilde{v}_0/\tilde{v}_1$ は，t 個に等分割可能な設備である．但し，v_1 は F に含まれる頂点である．

証明　等分割可能な設備 F の例を図 4.9 に示す．F を構成するサイズ l' の t 個の部分設備を $S_j (1 \le j \le t)$ と表す．但し，S_1 は頂点 v を含むとする．S_1 の辺を含まない F の部分木のうち極大なものを残余部分木と定義し，$F_i (1 \le i \le m)$ で表す．各 F_i は，部分設備 S_j のいくつかで構成される．F_i を構成する部分設備の数を t_i と表す．

帰納法を用いて証明を行う．$t = 1$ の場合，F の辺 $e_0 = (u, w)$ を選択する．但し，w は F の葉で v 以外とする．部分木 $F \cup \tilde{v}_0/\tilde{w}$ はサイズ l' の部分設備となる．$t = t_0$ の場合，S_1 の辺 $e_0 = (u, w)$ を選択する．但し，w は S_1 の葉で v 以外とする．w が F の葉ならば $t = 1$ と同様．さもなくば，w は図 4.9 のように残余部分木 F_i の根である．$t_i \le t_0 - 1$ なので，仮定より，ある v_1 に対し，$F_i \cup \tilde{u}/\tilde{v}_1$ は t_i 個に等分割可能な設備である．よって，これと部分設備 $S_1 \cup \tilde{v}_0/\tilde{w}$ で構成される部分木 $F \cup \tilde{v}_0/\tilde{v}_1$ は，t 個に等分割可能な設備である．∎

性質 2　最適な等分割可能な設備 F_{opt} は中点 q を含む．

証明　F_{opt} が q を含まないと仮定する．F_{opt} の根を v とし，v の親頂点を v_0 とする．性質 1 より，等分割可能な設備 $F = F_{opt} \cup \tilde{v}_0/\tilde{w}$ がある．また，v_0 を含み F_{opt} の頂点を含まな

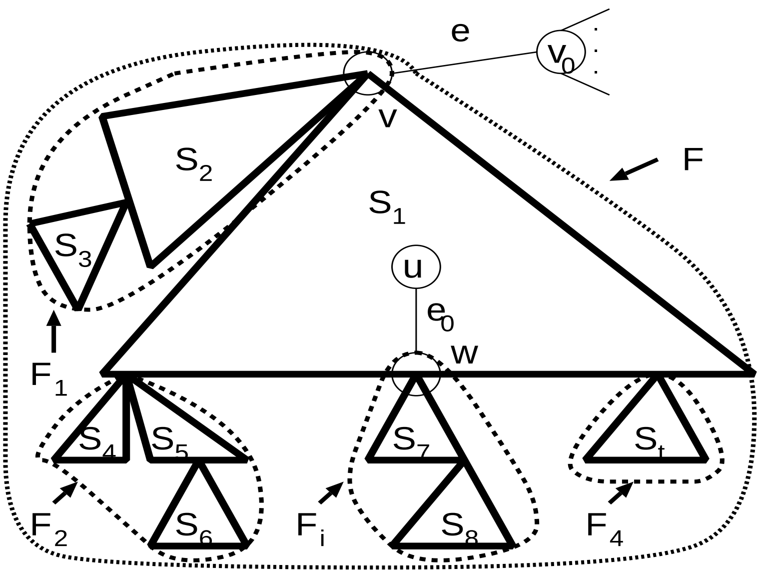


図 4.9: 等分割可能設備の部分設備と残余部分木

い最大の部分木を T'，wを含み Fの頂点を含まない最大の部分木を T''とする．距離和の差分 $d(F_{opt}) - d(F) = |T'| - |T''|$ であり，中点 qは重心であるから $|T'| > |T''|$ なので，$d(F_{opt}) > d(F)$ となり，F_{opt}は最適な等分割可能な設備ではない．∎

また，設備 F'，F''の距離和と設備 $F' \cup F''$の距離和との間に，性質 3の関係が成り立つ．

性質 3 同じ頂点を有しない部分木 T'，T''を考える．但し，T'の頂点 v_1と T''の頂点 v_2を結ぶ T_qの辺がある．また，T'に設備 F'があり，v_1は F'に含まれる．T''に設備 F''があり，v_2は F''に含まれる場合，部分木 $T' \cup T''$に対する設備 $F' \cup F''$の距離和 $d(T' \cup T''; F' \cup F'')$ は，距離和の定義より次式で表される．

$$d(T' \cup T''; F' \cup F'') = d(T'; F') + d(T''; F''). \tag{4.12}$$

T''に設備がない場合，距離和 $d(T' \cup T''; F')$ は，次式で表される．

$$d(T' \cup T''; F') = d(T'; F') + d(T''; \tilde{v_2}) + |T''|. \tag{4.13}$$

4.3.3.3 前処理

等分割可能な設備を求めるために，動的プログラミング法を用いる．ここで前処理として，有向木 T_q，各頂点 vに対する $|T_v|$ と，置換関数$\pi[i, j]$ を求める必要がある．

中点 qを根とする有向木 T_qの作成と，各頂点 vに対する $|T_v|$ の算出は，文献 [39] で証明されている各々$O(n)$ 時間のアルゴリズムを用いる．

また，$i = 1$〜Δでの，i 個の要素の置換関数$\pi[i,j]$ の数は，$\sum_{i=1}^{\Delta} |\pi[i]|$ である．また，各置換関数$\pi[i,j]$ は $O(\Delta)$ 時間で作成できる．従って，置換関数全体の作成は，

$$\begin{aligned} &\Delta \textstyle\sum_{i=1}^{\Delta} |\pi[i]| \leq \Delta^2 |\pi[\Delta]| \\ &\quad \leq (\alpha \log n / \log\log n)^2 n^{\alpha(1+\log\alpha)} \\ &\quad < O(n^{\alpha(1+\log\alpha)+1}) \end{aligned} \tag{4.14}$$

より，$O(n^{\alpha(1+\log\alpha)+1})$ 時間となる．

4.3.3.4　主処理

木構造ネットワークの頂点 vに対する部分木 $T_v^j(i)$ の制約設備および弱制約設備を求める．この繰り返しにより T_qの最適な弱制約設備を求め，Tの最適な等分割可能な設備配置を得ることができる．以下に，制約設備の定義と性質について説明する．

（１）弱制約設備

部分木 T'の部分木である設備のうち，サイズ l'の部分設備と，l'より小さいサイズで T'の根を含む部分設備が多くとも 1 つで構成され，T'の根を含むものを，部分木 T'の弱制約設備と定義する．この l'より小サイズの部分設備を小設備とよぶ．

サイズ l'の部分設備数と小設備のサイズが等しい部分木 T'の弱制約設備のうち，T'に対する距離和が最小となるものを，T'の最適な弱制約設備とよぶ．サイズ l'の部分設備 k個とサイズ l_0の小設備で構成される，部分木 T_vの最適な弱制約設備を，$F(v,k,l_0)$ と表す．小設備がない時，$l_0 = 0$ で表す．また $F(v,0,0)$ は，サイズ 0 の設備$\tilde{v}$とする．T_vに対する $F(v,k,l_0)$ の距離和を $d(v,k,l_0)$ と表す．なお，弱制約設備が存在しない状態を，次式で表現する．

$$\begin{aligned} F(v,k,l_0) &= \textbf{none}, \\ d(v,k,l_0) &= \infty. \end{aligned} \tag{4.15}$$

性質 2より，最適な等分割可能な設備 F_{opt}は，次式で得られる．

$$\begin{aligned} F_{opt} &= F(q,t,0), \\ d(F_{opt}) &= d(q,t,0). \end{aligned} \tag{4.16}$$

（２）制約設備

部分木 T' の根を v とする．サイズ l_0 の小設備 S_1 とサイズ l' の部分設備 S_i で構成される，T' の弱制約設備 F' は，次式で示される．

$$F' = S_1 \cup \bigcup_{i=2}^{k+1} S_i,\, v \in V(S_1). \tag{4.17}$$

部分木 T' の弱制約設備のうち，任意の $i < j$ に対して次の条件 (a)(b) を満すものを，部分木 T' の制約設備と定義する．

(a) 根の子頂点 v_i，v_j が同一部分設備 S に含まれるならば，$i < r < j$ なる子頂点 v_r は S に含まれるか，どの部分設備にも含まれない．

(b) 根の子頂点 v_i が小設備に含まれるならば，v_j は小設備に含まれるかどの部分設備にも含まれない．

サイズ l' の部分設備数と小設備のサイズが等しい部分木 T' の制約設備のうち，T' に対する距離和が最小となるものを，T' の最適な制約設備とよぶ．サイズ l' の部分設備 k 個とサイズ l_0 の小設備で構成される，部分木 $T_v^j(i)$ の最適な制約設備を，$F(v,k,l_0;j,i)$ と表す．$T_v^j(i)$ に対する $F(v,k,l_0;j,i)$ の距離和を，$d(v,k,l_0;j,i)$ と表す．なお，制約設備が存在しない状態を，次式で表現する．

$$\begin{aligned} F(v,k,l_0;j,i) &= \mathbf{none} \\ d(v,k,l_0;j,i) &= \infty \end{aligned} \tag{4.18}$$

（３）制約設備と弱制約設備の性質

制約設備と弱制約設備に対して，以下に示す性質 4，5が得られる．

性質 4 部分木 T_v の最適な弱制約設備 $F(v,k,l_0)$ と，部分木 $T_v^j(d_v)$ の最適な制約設備 $F(v,k,l_0;j,d_v)$ の間には，$1 \leq j \leq |\pi(d_v)|$ を満たすある j に対して以下の関係がある．

$$F(v,k,l_0) = F(v,k,l_0;j,d_v). \tag{4.19}$$

証明　最適な弱制約設備 $F(v,k,l_0)$ を構成する部分設備を $S_i (i \geq 1)$ と表す．但し，S_1 を小設備，S_0 を設備なしを示す記号とする．子頂点 v_i を含む部分設備を $S_{g(i)}$ と表し，$g(i)$ を降順にソートする置換を $\pi[d_v, j]$ と表すと，$F(v,k,l_0)$ は $T_v^j(d_v)$ の制約設備である．一方，$T_v^j(d_v)$ の制約設備は T_v の弱制約設備である．よって，$d(v,k,l_0) = d(v,k,l_0;j,d_v)$，$F(v,k,l_0) = F(v,k,l_0;j,d_v)$．∎

―――――――――――――――――――――――――――――――――――――

入力： 頂点 v，変数 k，l_0，j，i，

$T_v^j(i-1)$ の最適な制約設備

$d(v,*,*;j,i-1)$，$F(v,*,*;j,i-1)$，

$T_{v_{\pi[d_v,j](i)}}$ の最適な弱制約設備

$d(v_{\pi[d_v,j](i)},*,*)$，$F(v_{\pi[d_v,j](i)},*,*)$

出力： $T_v^j(i)$ の最適な制約設備

$d(v,k,l_0;j,i)$，$F(v,k,l_0;j,i)$

1. $i=1,l_0=0,k=0$ の場合

$d(v,k,l_0;j,i)=d(v_{\pi[d_v,j](i)},0,0)+|T_{v_i}|$

$F(v,k,l_0;j,i)=\tilde{v}$

2. $i=1,l_0=0,k>0$ の場合

$d(v,k,l_0;j,i)=d(v_{\pi[d_v,j](i)},k-1,l'-1)$

$F(v,k,l_0;j,i)=F(v_{\pi[d_v,j](i)},k-1,l'-1)\cup\tilde{v}$

3. $i=1,l_0>0$ の場合

$d(v,k,l_0;j,i)=d(v_{\pi[d_v,j](i)},k,l_0-1)$

$F(v,k,l_0;j,i)=F(v_{\pi[d_v,j](i)},k,l_0-1)\cup\tilde{v}$

4. $i>1,l_0=0,k=0$ の場合

$$d(v,k,l_0;j,i)=d(v,k,l_0;j,i-1)$$
$$+d(v_{\pi[d_v,j](i)},0,0)+|T_{v_{\pi[d_v,j](i)}}|$$

$F(v,k,l_0;j,i)=F(v,k,l_0;j,i-1)$

5. $i>1,l_0=0,k>0$ の場合

次の各 k_1,l_1の場合の d を比較し，最少となる場合に対応する設備 Fを選択し $F(v,k,l_0;j,i)$ とする

1) $0\le k_1<k, 0\le l_1<l'$の場合

$$d=d(v,k_1,l_1;j,i-1)$$
$$+d(v_{\pi[d_v,j](i)},k-k_1-1,l'-l_1-1)$$
$$F=F(v,k_1,l_1;j,i-1)$$
$$\cup F(v_{\pi[d_v,j](i)},k-k_1-1,l'-l_1-1)$$

2) $k_1 = k, l_1 = 0$ の場合

$$d = d(v, k_1, l_1; j, i-1)$$
$$+d(v_{\pi[d_v,j](i)}, k-k_1, l_0-l_1) + |T_{v_{\pi[d_v,j](i)}}|$$
$$F = F(v, k_1, l_1; j, i-1)$$

6. $l > 1, l_0 \geq 0, k = 0$ の場合

次の各 k_1, l_1の場合の d を比較し，最少となる場合に

対応する設備 Fを選択し $F(v, k, l_0; j, i)$ とする

1) $k_1 = 0, 0 \leq l_1 < l_0$の場合

$$d = d(v, k_1, l_1; j, i-1)$$
$$+d(v_{\pi[d_v,j](i)}, k-k_1, l_0-l_1-1)$$
$$F = F(v, k_1, l_1; j, i-1)$$
$$\cup F(v_{\pi[d_v,j](i)}, k-k_1, l_0-l_1-1)$$

2) $k_1 = 0, l_1 = l_0$の場合

$$d = d(v, k_1, l_1; j, i-1)$$
$$+d(v_{\pi[d_v,j](i)}, k-k_1, l_0-l_1) + |T_{v_{\pi[d_v,j](i)}}|$$
$$F = F(v, k_1, l_1; j, i-1)$$

7. $l > 1, l_0 > 0, k > 0$ の場合

次の各 k_1, l_1の場合の d を比較し，最少となる場合に

対応する設備 Fを選択し $F(v, k, l_0; j, i)$ とする

1) $0 \leq k_1 \leq k, 0 \leq l_1 < l_0$の場合

$$d = d(v, k_1, l_1; j, i-1)$$
$$+d(v_{\pi[d_v,j](i)}, k-k_1, l_0-l_1-1)$$
$$F = F(v, k_1, l_1; j, i-1)$$
$$\cup F(v_{\pi[d_v,j](i)}, k-k_1, l_0-l_1-1)$$

2) $k_1 = k, l_1 = l_0$の場合

$$d = d(v, k_1, l_1; j, i-1)$$
$$+d(v_{\pi[d_v,j](i)}, k-k_1, l_0-l_1) + |T_{v_{\pi[d_v,j](i)}}|$$
$$F = F(v, k_1, l_1; j, i-1)$$

図 4.10: 部分木の最適な制約設備を求めるアルゴリズム

性質 5 同じ頂点を有しない部分木 T'，T''を考える．但し，T''の根 v_kは，T'の根 vの子頂点とする．また，T'に含まれる vの子頂点 v_iは，$i < k$を満たすとする．部分木 $T' \cup T''$の最適な制約設備は，部分木 T'の最適な制約設備であるか，部分木 T'の最適な制約設備と部分木 T''の最適な弱制約設備の結合で表される．

証明　$T' \cup T''$の最適な制約設備を Fとする．$T' \cup T''$の根は vである．Fが T'の部分木の場合，Fは T'の制約設備である．式 (4.13) より，Fは T'の最適な制約設備でなければならない．Fが T'の部分木でない場合，$V(F') = V(F) \cap V(T')$ なる部分木を F'とし，$F'' = F/F'$とする．$F = F' \cup F''$であり，F'は T'の制約設備，F''は T''の弱制約設備である．式 (4.12) より，F'は T'の最適な制約設備であり，F''は T''の最適な弱制約設備でなければならない．

∎

なお，性質 5より，部分木 T'に制約設備が存在しない場合，部分木 $T' \cup T''$に制約設備は存在しない．また T''に弱制約設備が存在しない場合，部分木 $T' \cup T''$の制約設備は T'の制約設備になる．すなわち，T'の制約設備を F'，T''の弱制約設備を F''，$T' \cup T''$の制約設備を Fとすると，次式が示される．

$$\begin{aligned} &F = \mathbf{none} && (F' = \mathbf{none}), \\ &F = F' && (F'' = \mathbf{none}). \end{aligned} \tag{4.20}$$

（4）設備の導出

vが葉頂点の場合，最適な制約設備 $F(v, k, l_0; j, i)$ と距離和 $d(v, k, l_0; j, i)$ は，次式で表される．

$$\begin{aligned} F(v, 0, 0; j, i) &= \tilde{v}, \\ d(v, 0, 0; j, i) &= 0. \\ F(v, k, l_0; j, i) &= \mathbf{none}, \\ d(v, k, l_0; j, i) &= \infty \qquad (k > 0 \ or \ l_0 > 0). \end{aligned} \tag{4.21}$$

すなわち，$k = 0$，$l_0 = 0$ の場合のみ設備が存在する．

次に，頂点 vが内部頂点の場合は，次の補題が得られる．

補題 1 内部頂点 vの部分木 $T_v^j(i)$ の最適な制約設備 $F(v, k, l_0; j, i)$ は，図 4.10に示すアルゴリズムにより求まる．

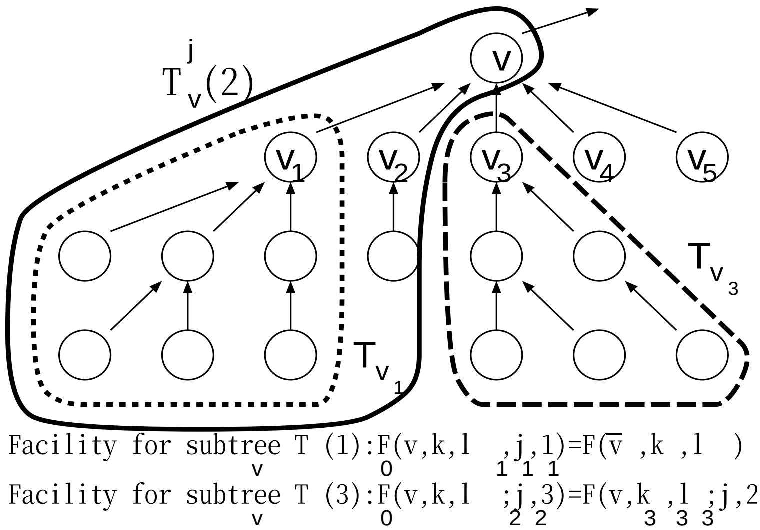


図 4.11: 部分木と最適な制約設備

証明　部分木 $T_v^j(i)$ の最適な制約設備 $F(v,k,l_0;j,i)$ は，性質 5より，図 4.11に示すように 2 つの設備の結合で表すことができる．$i=1$ の場合は，部分木 $T_{v_{\pi[d_v,j](1)}}$ における最適な弱制約設備と部分設備$\bar{v}$との結合となり，次式が成り立つ．

$$F(v,k,l_0;j,i) = F(v_{\pi[d_v,j](1)},k_1,l_1) \cup v. \tag{4.22}$$

但し，k_1，l_1は k，l_0により定まる値で，図 4.10における場合 1.〜3. の代入式右辺で与えられる．

$i>1$ の場合は，以下の (1)(2) のいずれかになる．

(1) 部分木 $T_v^j(i-1)$ の最適な制約設備 $F(v,k,l_0;j,i-1)$．

(2) 部分木 $T_v^j(i-1)$ の最適な制約設備 $F(v,k_1,l_1;j,i-1)$ と部分木 $T_{v_{\pi[d_v,j](i)}}$の最適な弱制約設備 $F(v_{\pi[d_v,j](i)},k_2,l_2)$ の結合．

但し，k_1，k_2，l_1，l_2は k，l_0により定まる組合せであり，k_2，l_2は k_1，l_1が定まると一意に決まる．

従って，$i > 1$ の場合，次式が成り立つ．

$$F(v,k,l_0;j,i) = \begin{cases} \quad F(v,k,l_0;j,i-1) \\ or \\ \quad F(v,k_1,l_1;j,i-1) \cup F(v_{\pi[d_v,j](i)},k_2,l_2). \end{cases} \tag{4.23}$$

ここで，$l_1 \le l_0$または$l_0 = 0$である．また$l_0 > 0$のとき，制約設備の条件 (b) より，$k_1 < k$ ならば$l_1 < l_0$となる．したがって，各i，k，l_0に対し，k_1，l_1として図 4.10に示された場合のみを調べれば良い．たとえば，$i > 1$，$k = 0$ の場合，図 4.10の場合 6 に示すように，$k_1 = 0$ かつ $0 \le l_1 \le l_0$で，$l_0 + 1$ 個の最適な制約設備の候補 Fが比較され，d が最小となる Fが最適な制約設備 $F(v,k,l_0;j,i)$ になる．∎

補題 2 中点 qを根とし，各頂点の次数が$\alpha \log n / \log\log n$ 以下である有向木 T_qに対し，最適な等分割可能な設備 F_{opt}を求めるアルゴリズムは $O(n^{\alpha(1+\log\alpha)+3})$ 時間で実行できる．

証明　$F(v,k,l_0;j,i)$ において，vと i の組合せの数は，各 vの子頂点 v_iの総和 $n-1$ である．kと l_0の組合せの数は，$(t+1)\cdot l' \le 2n$ である．$\pi[d_v,j]$ の組合せの数は，$|\pi[d_v]| \le n^{\alpha(1+\log\alpha)}$ である．したがって，組合せの総数は $O(n^{\alpha(1+\log\alpha)+2})$ となる．また，図 4.10における設備の結合 fと距離和 d は $O(1)$ 時間で算出できる．そして，距離和の最小値を求めるための比較回数は $(k+1)\cdot l' \le 2n$ である．したがって，各 $F(v,k,l_0;j,i)$ は $O(n)$ 時間で算出でき，組合せ総数から全 $F(v,k,l_0;j,i)$ は $O(n^{\alpha(1+\log\alpha)+3})$ 時間で算出できる．

同様に，$F(v,k,l_0)$ における組合せの総数は $O(n^2)$ であり，性質 4より各 $F(v,k,l_0)$ は $O(n^{\alpha(1+\log\alpha)})$ 時間で算出できるので，全 $F(v,k,l_0)$ は $O(n^{\alpha(1+\log\alpha)+2})$ 時間で算出できる．したがって，各 $F(v,k,l_0;j,i)$，$F(v,k,l_0)$ を葉から根の順に算出することにより，最適な等分割可能な設備 F_{opt}は $O(n^{\alpha(1+\log\alpha)+3})$ 時間で算出できる．∎

4.3.3.5 配置アルゴリズムと実行時間

補題 1より，木構造ネットワークに等分割可能な設備を配置するアルゴリズムは図 4.12 のようになる．配置アルゴリズムは，有向木 T_qの作成，部分木のサイズ $|T_v|$ 算出および置換関数$\pi[i,j]$ の表を作成する前処理と，部分木の最適な制約および弱制約設備算出の繰り返しによる主処理部分よりなる．

なお，木 Tに等分割可能な設備が存在しない場合，$F_{opt} = \textbf{none}$ となる．

入力： 木 T，設備のサイズ l，分割数 t

出力： 最適な等分割可能な設備 F_{opt}

1. 前処理

   中心 qを根とする有向木 T_q作成

   各頂点 vに対する $|T_v|$ 算出

   各置換関数$\pi[i,j]$ の表を作成

2. 主処理

   $S = \{T_q$の頂点リストで，各頂点 vはそのすべての子頂点 v_kの後方に置かれるもの $\}$

   while $S \neq \phi$ {

   　Sの先頭から頂点 vを取り出す

   　for $j = 1$ to $|\pi[d_v]|$

   　　for $i = 1$ to d_v

   　　　for $k = 0$ to t

   　　　　for $l_0 = 0$ to $l' - 1$

   　　　　　$d(v, k, l_0; j, i)$ および

   　　　　　$F(v, k, l_0; j, i)$ の算出

   　for $k = 0$ to p

   　　for $l_0 = 0$ to $l' - 1$

   　　　$F(v, k, l_0)$, $d(v, k, l_0)$ の算出

   }

   $F_{opt} = F(q, p, 0)$

図 4.12: 木構造ネットワーク上への等分割可能な木形状設備の配置アルゴリズム

このアルゴリズムの実行時間は定理1により求まる．

定理 1 各頂点の次数が$\alpha \log n / \log \log n$以下である木構造ネットワークに対する最適な等分割可能な設備F_{opt}は，$O(n^{\alpha(1+\log \alpha)+3})$時間で算出できる．

証明　前処理の実行時間は4.3.3節より，主処理の実行時間は補題2であるから，全体の実行時間は$O(n^{\alpha(1+\log \alpha)+3})$となる．∎

各頂点の次数が$\log n / \log \log n$以下，すなわち$\alpha = 1$の場合の配置アルゴリズムの実行時間は$O(n^4)$である．

4.3.4　まとめ

本節では，低速通信路や高速通信路が混在する非均質な木構造通信ネットワークを，同一サイズの部品からなる高速通信路を用いて構築する問題について議論した．具体的には，木構造ネットワークに対する，距離和が最少となる等分割可能な木形状設備配置問題を示し，その配置アルゴリズムを提案した．そして，この設備配置アルゴリズムは各頂点の次数が$\alpha \log n / \log \log n$以下の木構造ネットワークにおいて$O(n^{\alpha(1+\log \alpha)+3})$時間で実行できることを示した．これにより，大規模な木構造ネットワークに等しいサイズの部品からなる設備を配置した，高性能で従来よりコストパフォーマンスの高い通信ネットワークシステムを構築することができる．

4.4　複数個の同サイズ木形状設備の配置

本節では，頂点数nで次数が$O(\log n / \log \log n)$である木構造ネットワーク上に，互いに重ならないように，複数個の同一サイズの木形状設備を配置する．すなわち，サイズlの設備p個を，各設備の距離和の総和が最少となるよう配置する．但し，プロセッサ間通信では，ネットワークの中継プロセッサ数（頂点数）が通信時間を支配するため，通信路（各辺）の長さを1として考える．これにより，高速通信路と低速通信路を組み合わせた効率的な木構造通信ネットワークが構築できる．

複数個の設備配置において，設備を重ねることにより，さらなる性能向上が期待できる．すなわち，複数個の設備が配置された辺は，同時に複数の通信が可能となる．これにより，

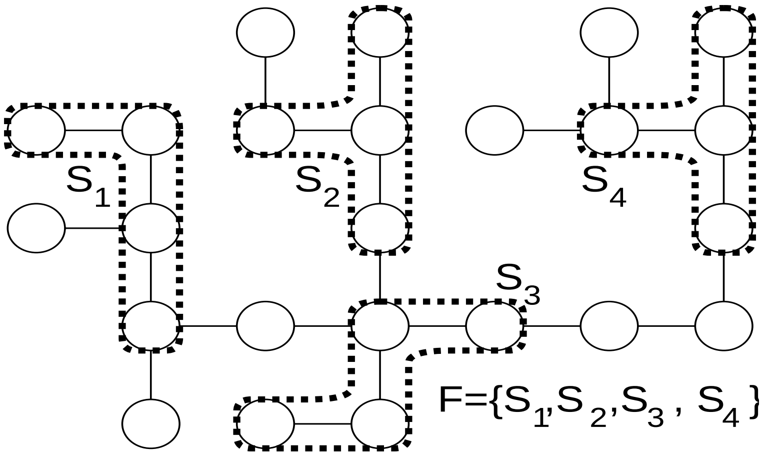


図 4.13: 木構造ネットワークの複数設備配置

通信ネットワークシステム全体の性能が向上する．そこで，通信コストの評価指標を，設備が重なる場合にも利用できるよう一般化する．

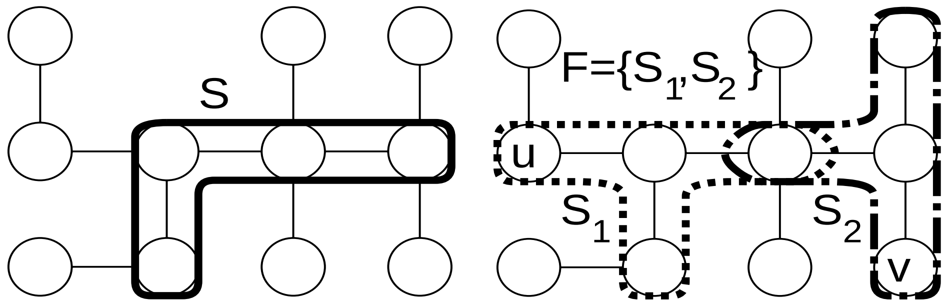
初めに，木構造ネットワーク上の複数設備の配置を説明し，ネットワーク上の設備配置に対する評価指標である距離和の総和を説明する．次に，複数設備の配置で用いる表記と性質を説明する．更に，互いに重ならない設備よりなる複数設備の配置を求めるアルゴリズムと実行時間について議論する．

4.4.1 木構造ネットワーク上の設備配置

4.4.1.1 複数設備

木構造ネットワークは木 Tで表される．木 Tの頂点集合を $V(T)$，辺集合を $E(T)$ と表す．木 Tの各辺 $(v,w) \in E(T)$ は，2 頂点 $v, w \in V(T)$ を長さ $d(v,w)$ でつないでいる．本章では，ネットワークの中継点数を重要と考え，長さ $d(v,w)$ は 1 とする．木 T'の頂点数を $|T'|$ と表す．n は木 Tの頂点数 $|T|$ である．b は木 Tのバンド幅である．木 T'の部分木 T''は，$V(T'') \subset V(T')$，$E(T'') \subset E(T')$ なる木である．なお，頂点 vのみの部分木を$\tilde{v}$と表す．$V(\tilde{v}) = \{v\}$，$E(\tilde{v}) = \phi$である．

木構造ネットワーク上への複数設備の配置を考える．設備 Sは木 Tの部分木である．設備 Sのサイズ lは，設備の辺の長さの総和である．本章では各辺の距離を 1 とするため，l は設備の辺の数 $|E(S)|$ となる．複数設備 Fは，p 個の同一サイズの設備 S_i（$i = 1, ..., p$）



(a) Optimal facility. (b) Optimal facilities.

図 4.14: 最適な複数設備

の集まりとし，次式で表す.

$$F = \{S_1, S_2, ..., S_p\} \tag{4.24}$$

図 4.13に，木構造ネットワークにおける複数設備の例を示す．複数設備 Fは，サイズ 3 の 4 個の設備 S_1，S_2，S_3，S_4で構成される．

4.4.1.2 距離和の総和

ネットワーク上への設備配置問題の評価指標として距離和の総和がある．2 頂点 v，w間の距離 $d(v, w)$ は，頂点 v，w間の経路 (パス) の辺の長さの総和である．頂点 vと設備 Sとの距離 $d(v, S)$ は，vと Sとの最少距離

$$d(v, S) = \min_{w \in V(S)} d(v, w). \tag{4.25}$$

である．設備 Sの距離和 $d(S)$ は，Sと Tの各頂点 u との距離の総和で，次式で与えられる．

$$d(S) = \sum_{v \in V(T)} d(v, S). \tag{4.26}$$

複数設備 Fの距離和の総和 $D(F)$ は，複数設備 Fを構成する各設備の距離和の総和であり，次式で表す.

$$D(F) = \sum_{S_i \in F} d(S_i) \tag{4.27}$$

図 4.14(a) に，木構造ネットワークにおける設備の例を示す．設備 Sのサイズは $l = 3$ である．また，設備 Sに対する距離 0，1，2 の頂点が各々4，6，1 個なので，式 (4.26) より，

設備 Sの距離和は $d(S) = 8$ である．この値は，サイズ 3 の設備のうちで最小である．図 4.14(b) に，木構造ネットワークにおける複数設備の例を示す．複数設備 Fは，設備 S_1，S_2 の集まりである．各設備の距離和は $d(S_1) = 9$，$d(S_2) = 13$ なので，式 (4.27) より，複数設備 Fの距離和の総和は $D(F) = 22$ となる．この値は，互いに重ならないサイズ 3 の設備 2 個で構成される複数設備のうちで最小である．

4.4.1.3 複数設備の評価指標

本節では，距離和の総和 $D(F)$ が最小となる複数設備を最適な複数設備と定義する．Hakimi 等 [37] は，設備全体に対する距離和や偏差を評価指標とした場合を検討した．しかし，通信量の多いシステムでは，一つの通信にすべての設備を占有させた場合，システム全体の性能向上が不十分な場合がある．たとえば，図 4.14(b) で，点 u と v間で双方向の通信が同時に行われるとする．サイズ 6 の設備 $S_1 \cup S_2$を配置する場合，一方の通信では，設備を用いることにより，通信コストは 0 になる．しかし，他方の通信は設備を使用できず，通信コストは 4 となる．一方，それぞれの通信が設備 S_1，S_2の一つを用いる場合，通信コストはともに 2 となる．通信コスト最大の通信が，システム全体の性能を決定する場合も多い．このとき，複数設備を用いることで，システム全体の性能向上が期待できる．そこで，本節では，各通信にランダムに選択した一つの設備を使用させる場合を議論する．そして，平均通信コストを最小とする複数設備を求める．これは，距離和の総和 $D(F)$ を最小とする複数設備 Fである．

また，複数設備の各設備を重ねない場合，重ねた場合に比べて平均通信コストは増大する．たとえば，図 4.14(a) の設備 Sを二重に重ねた場合の距離和の総和は 16 であるのに対し，図 4.14(b) の複数設備 Fの距離和の総和は 22 である．しかし，各設備を重ねない場合，各設備をネットワーク上に個別に配置できるので，通信システムの構築・保守コストの低減が期待できる．そこで，本節では，複数設備の各設備を重ねない．

本節では，木構造ネットワーク Tに対し，設備サイズ lおよび設備数 p を指定した場合の，互いに重ならない設備よりなる最適な複数設備 F_{opt}を求めるアルゴリズムについて議論する．但し，Tの各頂点の次数は，$\alpha \log n / \log\log n$ 以下と仮定する．なお，αは定数である．

4.4.2 表記と性質

4.4.2.1 表記

木 T の中点（median）q は，距離和 $d(\tilde{v})$ が最少となる頂点 v である．T に対し中点 q を根（root）とする有向木を T_q と表す．図 4.15に T_q の例を示す．頂点 v の子頂点の数は d_v と表す．v の子頂点は v_i（$1 \leq i \leq d_v$）と順序付けて表す．なお，中点 q は重心（centroid）である．すなわち，q を有しない T の部分木の頂点数が $|T|/2$ 以下になることが知られている [33]．本章では，子頂点数 d_v は次式のように制限する．

$$d_v \leq \Delta = \alpha \log n / \log\log n. \tag{4.28}$$

木 T のバンド幅を b とする．すなわち，バンド幅 b は，同じ辺 $e \in E(T)$ で重なることのできる設備数の上限である．なお，設備 S_1 と S_2 が重なるとは，S_1 と S_2 が同じ辺 e を有することをいう.

部分木の結合を $T' \cup T''$ とすると，次式を満たす.

$$\begin{aligned} V(T' \cup T'') &= V(T') \cup V(T''), \\ E(T' \cup T'') &= \textstyle\bigcup_{v,w \in V(T' \cup T''),(v,w) \in E(T)} \{(v,w)\}. \end{aligned} \tag{4.29}$$

また，部分木の削除を T'/T'' とすると，

$$\begin{aligned} V(T'/T'') &= V(T')/V(T''), \\ E(T'/T'') &= \textstyle\bigcup_{v,w \in V(T'/T''),(v,w) \in E(T)} \{(v,w)\}. \end{aligned} \tag{4.30}$$

が与えられる.

部分木 T' の頂点 v で，T_q の根 q であるか，その親頂点が $V(T')$ に含まれないものを，部分木 T' の根とよぶ．部分木 T' の根を，$root(T')$ と表す．また，頂点 v を根とする最大の部分木を，T_v と表す.

設備 S_i（$i = 1, ..., p$）で構成される設備集合 F' は，次式で表す.

$$F' = \{S_1, S_2, ..., S_p\}. \tag{4.31}$$

設備集合 F' と設備集合 F'' の結合 $F'[S'] + F''[S'']$ は，次式で表す．但し，設備 $S' \in F'$, $S'' \in F''$ とする.

$$F'[S'] + F''[S''] = (F'/\{S'\}) \cup (F''/\{S''\}) \cup \{S' \cup S''\}. \tag{4.32}$$

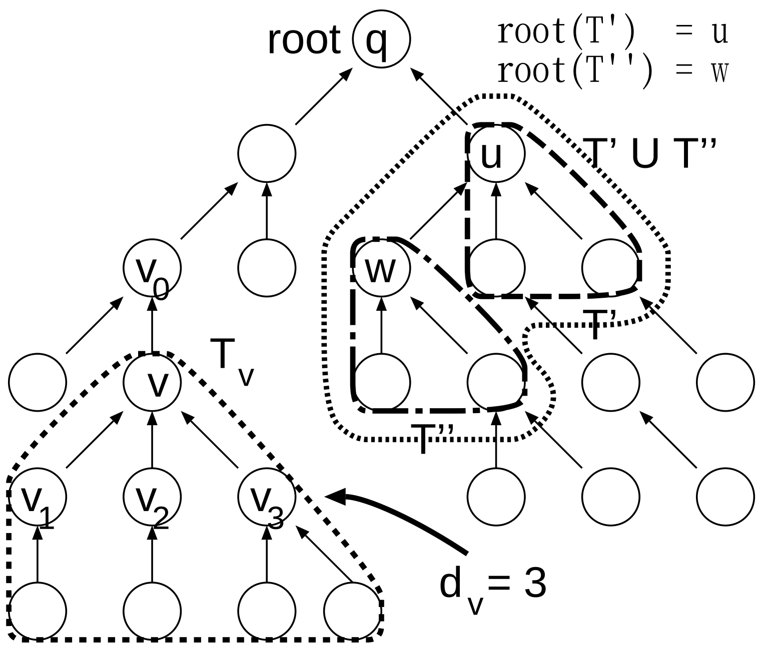


図 4.15: 根が中点 qである有向木 T_q

設備集合 F'において，頂点 vを有する設備数 $N_{F'}(v)$，辺 e を有する設備数を $B_{F'}(e)$ は，次式で表す.

$$
\begin{aligned}
N_{F'}(v) &= |\{S_i|v \in V(S_i), S_i \in F'\}|,\\
B_{F'}(e) &= |\{S_i|e \in E(S_i), S_i \in F'\}|.
\end{aligned}
\tag{4.33}
$$

また，頂点 vとその親頂点 v'との間の辺 (v, v') を有する設備数 $B_{F'}((v, v'))$ は，$B_{F'}(v)$ と略して表す．但し，$B_{F'}(q) = 0$ とする．

木 Tの互いに同じ頂点を有しない部分木を T^j（$j = 1, 2, ...$）と表したときに，$V(T'') = \cup_j V(T^j)$，$E(T'') = \cup_j E(T^j)$ で表されるものを，森 T''とよぶ．このとき，各 T^jを T''の断片とよぶ.

設備集合 F'に対し，バンド i の森 $G^i_{F'}$（$1 \leq i \leq b$）は，次式で表される．

$$
\begin{aligned}
E(G^i_{F'}) &= \{e|B_{F'}(e) \geq i\},\\
V(G^i_{F'}) &= \{v|(v, u) \in E(G^i_{F'})\}.
\end{aligned}
\tag{4.34}
$$

なお，$i < j$に対し，森 $G^j_{F'}$の各断片は，$G^i_{F'}$の対応する断片の部分木となる．

設備集合 F'に対し，バンド i の森 $G^i_{F'}$が存在する最大の i を，F'のバンド $M_{F'}$と表す.

すなわち，次式で表される．

$$\begin{aligned} &M_{F'} = i \\ &\quad \text{where } E(G_{F'}^{i+1}) = \phi \text{ and } E(G_{F'}^{i}) \neq \phi. \end{aligned} \tag{4.35}$$

但し，$G_{F'}^1 = \phi$の場合，$M_{F'} = 0$ とする．

複数設備 Fは，Fの各設備 S_jのサイズが lであり，木 Tの各辺 e に対する設備数 $B_F(e)$ が Tのバンド幅 b 以下である設備集合である．すなわち，次式で表される．

$$\begin{aligned} &|E(S_j)| = l \text{ forall } S_j \in F, \\ &B_F(e) \leq b \text{ forall } e \in E(T). \end{aligned} \tag{4.36}$$

また，複数設備 Fを構成する設備の数 p は，次式を満足する．

$$p \leq b(n-1)/l. \tag{4.37}$$

部分木 T'における森 G の距離和 $d(T';G)$ は，次式で表す．

$$d(T';G) = \sum_{v \in V(T')} d(v,G) \tag{4.38}$$

但し，$d(v,G)$ は，森 G の各頂点と頂点 vとの距離の最小値である．

また，部分木 T'における設備集合 F'の部分距離和 $D(T';F')$ は，T'における F'のバンド i の森 $G_{F'}^i$の距離和と，T_qにおけるサイズ l以上の各設備の根の距離和の総和とする．すなわち，次式で表される．

$$D(T';F') = \textstyle\sum_{i=1}^{M_{F'}} d(T';G_{F'}^i) + \sum_{S_j \in F', |S_j| \geq l} d(r\tilde{oo}t(S_j)). \tag{4.39}$$

木 Tのバンド幅 b が 1 の場合，$D(T';F')$ は次式で表される．

$$D(T';F') = d(T'; \textstyle\bigcup_{S_j \in F'} S_j) + \sum_{S_j \in F', |S_j| \geq l} d(r\tilde{oo}t(S_j)). \tag{4.40}$$

4.4.2.2 複数設備の性質

木構造ネットワーク Tに対する最適な複数設備 Fは，Tの中点 qを根とする有向木 T_qに対して，以下に示す性質を有する．

性質 6 木 T_qの頂点を v，vの親頂点を v'とする．但し，v'は根でないとする．最適な複数設備 Fにおいて，辺 (v,v') を有する設備数 $B_F(v) \leq B_F(v')$ である．

証明　頂点vにおいて，$B_F(v) > B_F(v')$とする．このとき，頂点v'を根とする設備Sが存在する．頂点$u \in V(S)$は，T_vの葉であるか，uの全ての子頂点u_iが$u_i \notin V(S)$となる頂点とする．設備$S' = S \cup \tilde{v}'/\tilde{u}$とする．中点$q$の定義より$|T_u| < |T_v| \leq |T| - |T_v|$であるから，$d(S') = d(S) - (|T| - |T_v|) + |T_u| < d(S)$．設備集合$F' = F \cup \{S'\}/\{S\}$とする．明らかに，$F'$は複数設備である．$d(F') < d(F)$なので，$F$は最適な複数設備でない．∎

性質6より，最適な複数設備Fに対するバンドiの森G_i（$1 \leq i \leq b$）は，部分木となる．また，性質6と同じ理由により，各G_iは頂点qを有する．

また，次の性質7，8が成り立つ．

性質 7 最適な複数設備Fの距離和総和$D(F)$は，Fに対する各バンドiの部分木の距離和$d(G_F^i)$に，各設備S_jの根の距離和$d(\tilde{root}(S_j))$を加えた値から，中点の距離和$d(\tilde{q})$を減じたものに等しい．

証明　図4.16に，複数設備Fが配置されたときの木T_qの頂点vを示す．$N_F(v)$と$B_F(v)$は，頂点vに対応する設備数である．T_v^iは，部分木$\tilde{v}$と，vの子頂点v_jのうち$B_F(v_j) = i$であるものに対する部分木T_{v_j}を結合したものである．T_v^rは，$T/T_v \cup \tilde{v}$とする．このとき，距離和の総和$D(F)$の，根q以外の頂点vで与えられる部分$D_v(F)$は，次式で表される．

$$\begin{aligned} D_v(F) &= (N_F(v) - B_F(v))d(T_v^r; \tilde{v}) + \textstyle\sum_{i=0}^{B_F(v)} (N_F(v) - i) d(T_v^i; \tilde{v}) \\ &= (N_F(v) - B_F(v))d(\tilde{v}) + \textstyle\sum_{i=0}^{B_F(v)-1} d(\cup_{j=0}^{i} T_v^i; \tilde{v}). \end{aligned} \tag{4.41}$$

同様に，$D(F)$の根qで与えられる部分$D_q(F)$は，次式で表される．

$$D_q(F) = (N_F(q) - 1)d(\tilde{q}) + \textstyle\sum_{i=0}^{N_F(q)-1} d(\cup_{j=0}^{i} T_q^i; \tilde{q}). \tag{4.42}$$

ここで，$N_F(v) - B_F(v)$は，頂点vを根とする設備S_jの数となる．従って，距離和の総和$D(F)$は，次式で表される．

$$\begin{aligned} D(F) &= \textstyle\sum_{v \in V(G_F^1)} D_v(F) \\ &= -d(\tilde{q}) + \textstyle\sum_{i=1}^{M_F} d(G_F^i) + \textstyle\sum_{v \in V(G_F^1)} (N_F(v) - B_F(v)) d(\tilde{v}) \\ &= -d(\tilde{q}) + \textstyle\sum_{i=1}^{M_F} d(G_F^i) + \textstyle\sum_{S_j \in F} d(\tilde{root}(S_j)). \blacksquare \end{aligned} \tag{4.43}$$

性質 8 最適な複数設備F_{opt}は，部分距離和$D(T_q; F)$が最小となる複数設備Fである．

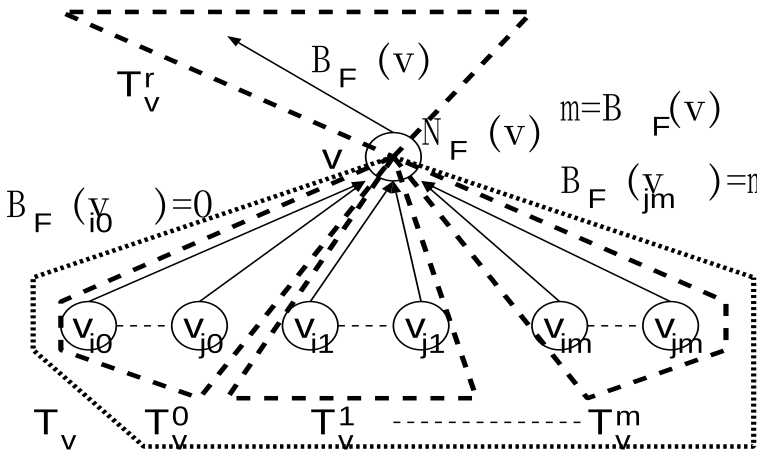


図 4.16: 複数設備 Fが配置されたときの木 T_qの頂点 v

証明　性質 7より，距離和の総和 $D(F)$ と部分距離和 $D(T_q;F)$ の間に，次の関係が示される．

$$D(F) = D(T_q;F) - d(\tilde{q}). \tag{4.44}$$

よって，最適な複数設備 F_{opt}は，部分距離和 $D(T_q;F)$ が最小となる Fである．∎

設備集合 F'，F''の部分距離和と設備集合 $F' \cup F''$の部分距離和の間に，性質 9の関係が成り立つ．

性質 9 同じ頂点を有しない部分木 T'，T''を考える．T'の頂点 v_1は，T''の頂点 v_2の親頂点とする．T'に設備集合 F'があり，その設備の一つ S'は v_1を有する．T''に設備集合 F''がある場合，その設備の一つ S''は v_2を有する．F'のバンド $H_{F'}$は，F''のバンド $H_{F''}$より小さくないとする．また，設備集合 F'，F''の最大バンドの部分木 $G_{F'}^{M_{F'}}$，$G_{F''}^{M_{F''}}$は，各々頂点 v_1，v_2を有するとする．このとき，部分木 $T' \cup T''$に対する設備集合 $F'[S'] + F''[S'']$ の部分距離和 $D(T' \cup T''; F'[S'] + F''[S''])$ は，次式で表される．

$$D(T' \cup T''; F'[S'] + F''[S'']) = \begin{cases} D(T';F') + \beta \\ \qquad\qquad\qquad (F'' = \phi) \\ D(T';F') + D(T'';F'') + \beta + \gamma \\ \qquad\qquad\qquad (F'' \neq \phi). \end{cases} \tag{4.45}$$

但し，β，γは，次式で表される．

$$
\begin{aligned}
&\beta = (M_{F'} - M_{F''})(d(T'';\tilde{v_2}) + |T''|),\\
&\gamma = \\
&\qquad \begin{cases}
0 \\
\qquad\qquad\qquad (|S'| + |S''| + d(v_1, v_2) < l \text{ or } |S'| \geq l \text{ and } |S''| < l) \\
d(\tilde{v_1}) \\
\qquad\qquad\qquad (|S'| + |S''| + d(v_1, v_2) \geq l \text{ and } |S'|, |S''| < l) \\
d(\tilde{v_1}) - d(\tilde{v_2}) \\
\qquad\qquad\qquad (|S'| < l \text{ and } |S''| \geq l) \\
-d(\tilde{v_2}) \\
\qquad\qquad\qquad (|S'|, |S''| \geq l).
\end{cases}
\end{aligned}
\tag{4.46}
$$

証明　F''が存在しない場合は，次式が成立する．

$$
\begin{aligned}
&D(T' \cup T''; F'[S'] + F''[S''])\\
&= \textstyle\sum_{i=1}^{M_{F'}} d(T' \cup T''; G^i_{F'[S']+F''[S'']}) + \sum_{S_j \in F'[S']+F''[S''], |S_j| \geq l} d(\tilde{root}(S_j))\\
&= \textstyle\sum_{i=1}^{M_{F'}} (d(T'; G^i_{F'}) + d(T''; \tilde{v_2}) + |T''|) + \sum_{S_j \in F', |S_j| \geq l} d(\tilde{root}(S_j))\\
&= D(T'; F') + \beta. \blacksquare
\end{aligned}
\tag{4.47}
$$

F''が存在する場合，性質 6より，次式が成立する．

$$
\begin{aligned}
&D(T' \cup T''; F'[S'] + F''[S''])\\
&= \textstyle\sum_{i=1}^{M_{F'}} d(T' \cup T''; G^i_{F'[S']+F''[S'']}) + \sum_{S_j \in F'[S']+F''[S''], |S_j| \geq l} d(\tilde{root}(S_j))\\
&= \textstyle\sum_{i=1}^{M_{F''}} (d(T'; G^i_{F'}) + d(T''; G^i_{F''})) + \sum_{i=M_{F''}+1}^{M_{F'}} (d(T'; G^i_{F'}) + d(T''; \tilde{v_2}) + |T''|)\\
&\qquad + \textstyle\sum_{S_j \in F'[S']+F''[S''], |S_j| \geq l} d(\tilde{root}(S_j))\\
&= \beta + \textstyle\sum_{i=1}^{M_{F'}} d(T'; G^i_{F'}) + \sum_{i=1}^{M_{F''}} d(T''; G^i_{F''}) + \sum_{S_j \in F'[S']+F''[S''], |S_j| \geq l} d(\tilde{root}(S_j))\\
&= D(T'; F') + D(T''; F'') + \beta + \gamma.
\end{aligned}
\tag{4.48}
$$

4.4.2.3　各設備を重ねない場合の性質

各設備を重ねない場合の複数設備は，木Tのバンド幅bが 1 の場合の複数設備に等しい．各設備を重ねない場合の，性質 6，7，9に対応する性質は，以下の性質 10，11，12で示される．

木 T のバンド幅 b が 1 の場合，根 q 以外の各頂点 $v \in V(G_F^1)$ に対して，$B_F(v) = 1$ である．すなわち，最適な複数設備 F に対するバンド 1 の森 G_F^1 は，F の各設備 S_j の結合 $\cup S_j$ に等しい．したがって，性質 6は性質 10に，性質 7は性質 11になる．

性質 10 各設備を重ねずに配置する場合，最適な複数設備 F の各設備 S_j の結合 $\cup S_j$ は，q を有するサイズ $l \cdot p$ の等分割可能な設備である．

性質 11 各設備を重ねずに配置する場合，最適な複数設備 F の距離和の総和 $D(F)$ は，F の各設備の結合 $\cup_{S_j \in F_{opt}} S_j$ の距離和 $d(\cup S_j)$ に，F の各設備 S_j の根の距離和 $d(r\tilde{oo}t(S_j))$ を加えた値から，中点の距離和 $d(\tilde{q})$ を減じたものに等しい．

また，$M_{F'} = 1$，F'' が存在する場合は $M_{F''} = 1$ であることから，性質 9は性質 12になる．

性質 12 同じ頂点を有しない部分木 T'，T'' を考える．T' の頂点 v_1 は，T'' の頂点 v_2 の親頂点とする．T' に設備集合 F' があり，その設備の一つ S' は v_1 を有する．T'' に設備集合 F'' がある場合，その設備の一つ S'' は v_2 を有する．各設備を重ねずに配置する場合，部分木 $T' \cup T''$ に対する設備集合 $F'[S'] + F''[S'']$ の部分距離和 $D(T' \cup T''; F'[S'] + F''[S''])$ は，次式で表される．

$$
D(T' \cup T''; F'[S'] + F''[S'']) =
\begin{cases}
D(T'; F') + d(T''; \tilde{v_2}) + |T''| \\
\qquad\qquad\qquad (F'' = \phi) \\
D(T'; F') + D(T''; F'') + \gamma \\
\qquad\qquad\qquad (F'' \neq \phi).
\end{cases}
\tag{4.49}
$$

但し，γ は，性質 9と同じである．

4.4.3 複数個の木形状設備配置アルゴリズム

Tamir 等 [36] は，動的プログラミング法を用いた木設備の配置アルゴリズムを示している．性質 10より，4.3章のアルゴリズムの評価指標を性質 11に変更することで，木構造ネットワーク T の，互いに重ならない設備よりなる複数設備の最適な配置を求めることができる．また，性質 11の評価指標算出に必要な $|T_v|$ と $d(\tilde{v})$ は，前処理として，文献 [39] で証明されているそれぞれ $O(n)$ 時間のアルゴリズムを用いることができる．したがって，4.3章と同じく，次の定理 2が成り立つ．

定理 2 各頂点の次数が$\alpha \log n / \log\log n$ 以下である木構造ネットワークに対する，互いに重ならない設備よりなる最適な複数設備 F_{opt} の算出は，$O(n^{\alpha(1+\log\alpha)+3})$ 時間で実行できる．

4.4.4 まとめ

本節では，低速通信路や高速通信路が混在する非均質な木構造通信ネットワークを，同一サイズの高速通信路を複数個用いて構築する問題について議論した．具体的には，木構造ネットワークに対する，距離和の総和が最小となる複数個の同一サイズの木形状設備の配置問題を示し，そのアルゴリズムを提案した．そして，この配置アルゴリズムは各頂点の次数が$\alpha \log n / \log\log n$ 以下の木構造ネットワークにおいて $O(n^{\alpha(1+\log\alpha)+3})$ 時間で実行できることを示した．これにより，大規模な木構造ネットワークに等しいサイズの設備を複数個配置した，通信量の多い場合に高性能でコストパフォーマンスの高い通信ネットワークシステムを構築することができる．

4.5 むすび

本章では，木構造の通信ネットワークに対する，構築コストを重視した木形状の高速通信路設備の配置について議論した．

先ず，平衡二分木における設備配置について議論し，木形状設備を配置することにより低速通信路と高速通信路の混在する低コストで高性能な木構造通信ネットワークシステムを構築できることを明らかにした．そして，実用性の観点から，実際に通信ネットワークシステムを構築する問題点について検討し，同一サイズの部分設備で構成される設備配置問題や，同一サイズの複数個の設備の配置問題が重要であることを示した．

次に，低速通信路や高速通信路が混在する非均質な木構造通信ネットワークを，同一サイズの部品からなる高速通信路を用いて構築する問題について議論した．具体的には，木構造ネットワークに対する，距離和が最少となる等分割可能な木形状設備配置問題を示し，そのアルゴリズムを提案した．そして，この設備配置アルゴリズムは各頂点の次数が$\alpha \log n / \log\log n$ 以下の木構造ネットワークにおいて $O(n^{\alpha(1+\log\alpha)+3})$ 時間で実行できることを示した．

更に，低速通信路や高速通信路が混在する非均質な木構造通信ネットワークを，同一サイズの高速通信路を複数個用いて構築する問題について議論した．具体的には，木構造ネッ

トワークに対する，距離和の総和が最小となる複数個の同一サイズの木形状設備配置問題を示し，そのアルゴリズムを提案した．そして，この配置アルゴリズムは各頂点の次数が $\alpha \log n/\log\log n$ 以下の木構造ネットワークにおいて $O(n^{\alpha(1+\log\alpha)+3})$ 時間で実行できることを示した．

本章の議論により，大規模な木構造ネットワークに等しいサイズの部品からなる設備を配置することによる従来よりコストパフォーマンスの高い通信ネットワークシステムや，等しいサイズの設備を複数個配置することによる通信量の多い場合に高性能でコストパフォーマンスの高い通信ネットワークシステムを構築する方法が示された．

第 5 章

設備内の通信コストを重視した設備配置

5.1　ネットワークにおける設備内の通信コスト

4 章では，木構造の通信ネットワークに対する，構築コストを重視した木形状の高速通信路設備配置について議論した．先ず，木形状設備の構築コストを重視し，同一小サイズの部分設備で木形状設備を構築した場合の通信ネットワーク設備配置問題を示し，その最適配置を求めるアルゴリズムを提案した．また，同様に複数個の木形状設備を重ねずに配置する通信ネットワーク設備配置問題について詳しく議論した．このときの評価指標として距離和やその総和を用いた．この評価指標は高速通信設備の性能を一種類とした場合に有効である．たとえば 10Mbps の低速通信路に 100Mbps の高速通信路を追加配置する場合に利用できる．

しかし，実際には 100Mbps と 1Gbps の高速通信路を併用するなどの，さまざまな通信ネットワークシステムが存在する．多種類の高速通信設備を用いる複雑なネットワークシステムを構築するには，高速通信設備内の通信コストを考慮することが必要である．そこで本章では，多種類の高速通信設備を用いるネットワークシステムを構築する指標として，木構造ネットワークに対する二点間の平均距離が最小となるサイズ l の木形状設備やパス形状設備の配置について議論する．

なお，4 章と同様に，簡単化のため通信トラフィックは考慮せず，通信システムの性能を通信レイテンシで評価する．すなわち，各通信路の通信コストはその通信路のレイテンシである．各通信路の通信コストを通信路（各辺）の長さで表した場合，二点間の通信コストは二点間の通信路の長さの和となる．また，設備サイズは設備に含まれる通信路（各辺）

の長さの和で表される．本章では，実用的な木構造ネットワーク上での設備配置を考えるために，各辺の長さが 1 以外の場合も議論する．

先ず，全対距離和を提案し，全対距離和が最小となる設備の配置について議論する．そして，各辺の長さが 1 である木構造ネットワークに対する全対距離和最小の木形状設備が $O(n)$ 時間で求まることを示す．

次に，木構造ネットワークの各辺の長さを d としたとき，通信コスト低減関数 $f(d)$ を用いて設備内の各辺の長さを d から $f(d)$ に削減する設備配置を提案する．そして，設備による高速化率が一定（$f(d) = \alpha \cdot d$，αは定数）の場合の最適な設備が，全対距離和を最小とする設備に等しいことを示す．これにより，たとえば各辺の長さが 1 である木構造ネットワークに対する最適な設備が $O(n)$ 時間で求まることを明らかにする．

更に，各辺のコストを考慮した実用的な最適設備配置問題を検討する．従来のサイズ指定の設備配置問題は，各辺の長さ a_iを設備サイズの制約と設備の評価の両方に用いてきた．ここでは，設備サイズの制約に対しては辺の長さ a_i，設備の評価に対しては辺のコスト c_i としたより実用的な設備配置問題を取り扱い，サイズ指定の木形状やパス形状設備の配置について議論する．そして，最適な離散木形状設備を配置する疑似多項式時間アルゴリズムを示す．特に，通信ネットワークの各辺の長さが 1 の場合には，このアルゴリズムにより最適な離散木形状設備が $O(n^3)$ 時間で求まることを明らかにする．また，最適な連続および離散パス形状設備が $O(n^3)$ 時間で求まることを示す．更に，従来の距離和を用いた設備や，全対距離和や全対コストを用いた木形状やパス形状の設備配置を，距離和を用いた実用設備配置により求めることができることを示す．

5.2 設備による通信コストの削減

超並列・分散計算システムの通信ネットワークシステムに配置された高速通信路は，その通信路を通過する通信の通信コストを削減することができる．一般に超並列・分散計算システムの各プロセッサは多対多で通信を行う．そこで，超並列・分散計算システムにおける設備配置では，二点間の通信の平均通信コストを重視する必要がある．二点間の平均通信コストは，ネットワークの頂点数と，その全ての二点間の距離の総和により定まる．以下では，この二点間の距離の総和を，全対距離和とよぶ．効率的な通信ネットワークシステムを構築するには，全対距離和が最小となる設備を求める必要がある．

しかし，従来の木形状やパス形状の設備配置の研究は，全対距離和を評価指標として重

視していない．従来の研究では，評価指標として (1) 偏差，(2) 距離和，(3) 分岐重み，(4) 頂点数やカバー領域，などやその組み合わせを用いることが多い [30]．Tamir らは，一般化 p 森を提案し，木構造ネットワーク上に配置する $O(n^3p^2)$ アルゴリズムを提案した [36]．一般化 p 森は，p 個の木形状設備の集合である．そして，評価指標として，(1) 設備のサイズにより定まるコスト，(2) 各設備を構成する辺の使用コスト，(3) ネットワークの各点から最も近接する設備までの距離和，(4) 各設備間の通信が設備内の辺を通過するのに必要な通信コスト，(5) 各設備間の通信が設備間を繋ぐ辺を通過するのに必要な通信コスト，(6) 設備内の一点をサーバとして使用するためのコスト，の総和が最小になるものとした．一般化 p 森は，p メディアン，単純プラント配置，最小コスト最大カバー，固定コスト網羅森などの，さまざまな設備の一般化であることが示されている [36]．しかし，point-to-point な通信コストとしては設備間の通信のみが用いられており，全対距離和を考慮できない．一方，ハブ配置は評価指標として全対距離和を用いている [31]．しかし，ハブは点形状設備の集合であり，サイズ指定の木形状やパス形状に対応することができない．

そこで本節では，各辺の長さが任意である木構造の通信ネットワークに対する，全対距離和が最小となるサイズ上限指定の木形状やパス形状設備について議論する．

本節の内容を以下に述べる．先ず，全対距離和を用いた設備配置を提案し，全対距離和の性質を示す．次に，全対距離和最小の設備の性質について，木形状設備とパス形状設備，連続設備と離散設備の両方の場合を議論し，その性質や配置方法を明らかにする．最後に，全対距離和最小の離散木形状設備の算出について議論し，離散木形状設備の算出は NP 困難であることを明らかにする．また，最適な離散木形状設備を求める疑似多項式時間アルゴリズムや，完全多項式近似な離散木形状設備を求めるアルゴリズムを示す．

5.2.1 設備配置の評価関数

本章では，設備に部分辺を用いる場合も考慮するため，文献 [40] の定義を使用する．すなわち，無向の木構造ネットワークを $T = (V, E)$ とする．ただし，頂点集合 $V = \{v_1, v_2, ..., v_n\}$，辺集合 Eとする．各辺は単間隔で表され，内点を使用可能とする．また，$A(T)$ を Tの辺上にある点の集合とする．$A(T)$ は，$n-1$ 個の単間隔の接続で表される，接続され閉じた集合である．$P(v_i, v_j)$ を，$A(T)$ に含まれる，点 v_iと v_jを接続するパスとする．v_1を Tの根とする．$i = 2, ..., n$ に対し，$f(v_i)$ を v_iの親，すなわち $P(v_1, v_i)$ 上にあり，v_i以外で最も近くにある $v \in V$の点である．集合 $E = \{e_2, e_3, ..., e_n\}$ は，v_jと $f(v_j)$ を接続する辺 e_jの集合で

ある．点 (x, j)（$0 \leq x \leq 1$）は，e_j上で，$f(v_j)$ からの距離が x である点である．

そして，$T' \subseteq A(T)$ で，接続され閉じているものを部分木とする．部分木 T'は，どの二辺の交点も空であるか Vに含まれる点である部分辺の集合で表される．$E(T')$ をこの部分辺の集合とし，$V(T')$ を Vに含まれる $A(T')$ の点の集合とする．部分木 T'が離散であるとは，$E(T')$ の要素すべての端点が Vに含まれる点であることである．

各辺 e_jの長さを $l(e_j)$ で表し，点 (x, j) と (y, j) を接続する部分辺の長さを $l(e_j)|x-y|$ とする．そして，$A(T)$ の距離関数は $a_2, ..., a_n$で与えられ，点 (x, i) と (y, j) の距離 $d((x, i), (y, j))$ は，そのパスに含まれる部分辺の長さの和となる．また，T'のサイズ $L(T')$ は $E(T')$ の各部分辺の長さの和とする．部分木 $T' \in A(T)$ と点 $v_j \in V$との距離 $d(v_j, T')$ は，v_jから最も近い T'上の点までの距離とする．

各頂点 v_iの重みを w_iで表す．また，部分木 T'の重み $w^{T'}$は，T'の各頂点の重みの和とする．すなわち，次式で表される．

$$w^{T'} = \textstyle\sum_{v_i \in V(T')} w_i. \tag{5.1}$$

Tの（重み付き）重心 qは，v_iを含まないどの Tの離散部分木 T'も $w^{T'} \leq w^T/2$ となる頂点 v_iとする．

部分木 T'，T''の結合 $T' \cup T''$は，$v_i \in T'$または $v_i \in T''$なる点 v_iの集合とする．部分木 T'，T''の積 $T' \cap T''$は，$v_i \in T'$かつ $v_i \in T''$ なる点 v_iの集合とする．部分木 T'から T''の削除 T'/T''は，$v_i \in T'$かつ $v_i \notin T''$または $v_i \in T' \cap T''$なる点 v_iの集合とする．

設備 Fは，サイズ $L(F) \leq l$である部分木である．特に離散部分木である設備を離散設備とよぶ．離散設備と区別するとき，部分木である設備を連続設備とよぶ．

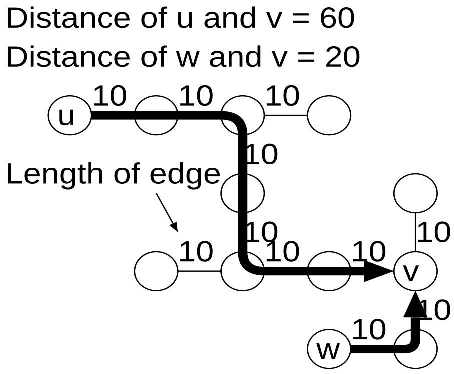
設備 Fの（重み付き）距離和 $D(F)$ は，次式で表される．

$$D(F) = \textstyle\sum_{v_i \in V} \{w_i \cdot d(v_i, F)\}. \tag{5.2}$$

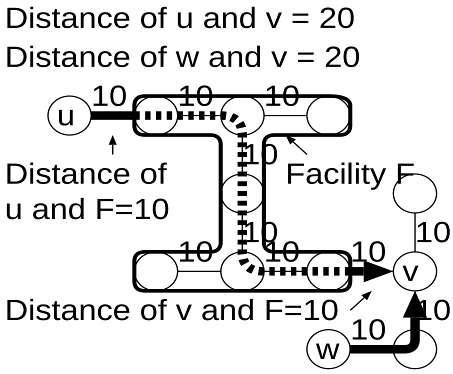
設備 Fの（重み付き）全対距離和 $P(F)$ は，ネットワークの二点の Fまでの距離の和と二点間の距離の短い方に二点の重みの積を乗じたものの総和であり，次式で表される．

$$P(F) = \textstyle\sum_{v_i, v_j \in V} \{w_i w_j \min(d(v_i, F) + d(v_j, F), d(v_i, v_j))\}. \tag{5.3}$$

図 5.1に設備と全対距離和の例を示す．各点の重み w_jは 1 とする．図 5.1(a) のネットワークにおいて，頂点 u，v間の距離は 60，頂点 w，v間の距離は 20 である．図 5.1(b) のよう



(a) Network without facility.



(b) Network with facility F.

図 5.1: 設備配置と全対距離和

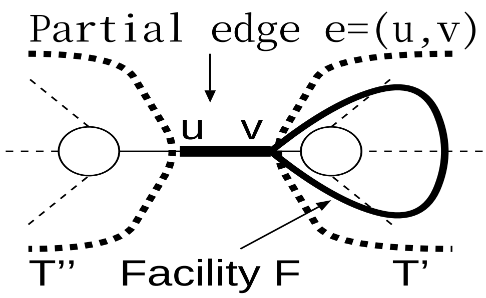


図 5.2: 木 Tの設備 Fと隣接する部分辺 e

に設備 Fを配置することにより，頂点 u，v間の距離は両者の設備 Fまでの距離の和 20 になり，頂点 w，v間の距離は 20 のままとなる．そして，ネットワークの各二点間の距離の総和 $P(F) = 850$ となる．

本節では，全対距離和 $P(F)$ が最小となる設備を最適な設備とする．

5.2.2 全対距離和の性質

全対距離和が満たす性質について議論する．なお，この性質は，設備が木形状とパス形状のいずれにおいても，また連続設備と離散設備のいずれにおいても成立する．

木構造ネットワーク Tに対して，文献 [41] のように距離和は性質 13，14を満たす．

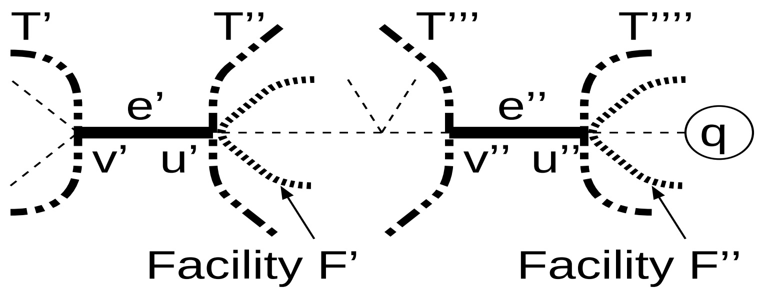


図 5.3: 木 Tの設備 F'，F''と隣接する長さϵの部分辺 e'，e''

性質 13 木構造ネットワークを T，設備を F，Fに隣接する部分辺を $e=(u,v)$ とする．ただし，$F\cap e=v$である．また，$T'\cap e=v$なる最大の部分木を T'，$T''\cap e=u$ なる最大の部分木を T''とする（図 5.2）．このとき，設備 Fに部分辺 e を追加することによる距離和の削減量 $DD(e)$ は，次式で表される．

$$DD(e)=D(F)-D(F\cup e)=l(e)w^{T''}. \tag{5.4}$$

性質 14 木構造ネットワークを T，部分辺を e_jとする．設備 Fが e_jに含まれるならば，e_jの端点を含み e_jに含まれる設備で，Fより距離和の大きくないものがある．

証明　性質 13による．∎

また，距離和の削減量は性質 15，16を満たす．

性質 15 木構造ネットワークを T，ある点 q，長さϵ（>0）の部分辺を e'，e'上の点を v，パス $P(q,v)$ 上の長さϵの部分辺を e''とする．部分辺 e'，e''上の最も qに近い点をそれぞれ u'，u''とする．そして，$e'\cap F'=u'$，$e''\cap F''=u''$なる設備を F'，F''とする．また，部分辺 e'，e''の最も qより遠い点をそれぞれ v'，v''とする．更に，$T'\cap e'=v'$なる最大の部分木を T'，$T''\cap e'=u'$なる最大の部分木を T''，$T'''\cap e''=v''$なる最大の部分木を T'''とする（図 5.3）．このとき，設備 F'，F''に対する部分辺 e'，e''の距離和削減量 $DD(e')$，$DD(e'')$は，次式を満たす．

$$\begin{cases} DD(e')=DD(e'') & (w^{T''\cap T'''}=0) \\ DD(e')<DD(e'') & (\text{otherwise}). \end{cases} \tag{5.5}$$

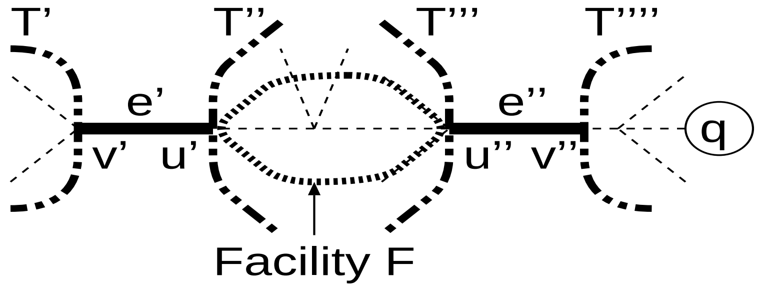


図 5.4: 木 T の設備 F と隣接する長さ ϵ の部分辺 e'，e''（重心 $q \in V(T''')$）

証明　$w^{T'' \cap T'''} = 0$ ならば $w^{T'} = w^{T'''}$ なので，性質 13より $DD(e') = DD(e'')$．さもなくば $w^{T'} < w^{T'''}$ なので，$DD(e') < DD(e'')$．∎

性質 16 木構造ネットワークを T，T の重心 q，設備 F，F に隣接する長さ ϵ の部分辺を $e' = (u', v')$，$e'' = (u'', v'')$ とする．ただし，$e' \cap F = u'$，$e'' \cap F'' = u''$ である．また，部分辺 e'，e'' の最も F より遠い点をそれぞれ v'，v'' と表す．更に，$T' \cap e' = v'$ なる最大の部分木を T'，$T'' \cap e' = u'$ なる最大の部分木を T''，$T''' \cap e'' = u''$ なる最大の部分木を T'''，$T'''' \cap e'' = v''$ なる最大の部分木を T'''' とする（図 5.4）．このとき，設備 F に対する部分辺 e'，e'' の距離和削減量 $DD(e')$，$DD(e'')$ は，次式を満たす．

$$\begin{aligned} &DD(e') = DD(e'') && (w^{T''''} = w^T/2 \text{ かつ } w^{T'' \cap T'''} = 0) \\ &DD(e') < DD(e'') && (\text{otherwise}). \end{aligned} \tag{5.6}$$

証明　$w^{T''''} = w^T/2$ かつ $w^{T'' \cap T'''} = 0$ ならば $w^{T'} = w^{T'''}$ なので，性質 13より $DD(e') = DD(e'')$．さもなくば $w^{T'} < w^{T'''}$ なので，$DD(e') < DD(e'')$．∎

一方，全対距離和は次の性質 17，18 を満たす．

性質 17 木構造ネットワークを T，設備を F，F に隣接する部分辺を $e = (u, v)$ とする．ただし，$F \cap e = v$ とする．また，$T' \cap e = v$ なる最大の部分木を T'，$T'' \cap e = u$ なる最大の部分木を T'' とする（図 5.2）．このとき，設備 F に部分辺 e を追加することによる全対距離和の削減量 $DP(e)$ は，次式で表される．

$$DP(e) = P(F) - P(F \cup e) = l(e) w^{T'} w^{T''}. \tag{5.7}$$

証明　辺 e の長さ $l(e)=0$ の場合は，$F\cup e=F$であることから全対距離和の削減量 $DP(e)=0$ である．辺 e の長さが 0 以外の場合は，Tの頂点集合 Vは，$V(T')$，$V(T'')$ に分割できる．設備 Fに辺 e を追加したとき，二点 v_i，v_jの距離は，$v_i,v_j\in V(T')$ と $v_i,v_j\in V(T'')$ の場合は変化せず，$v_i\in V(T')$，$v_j\in V(T'')$ の場合は $l(e)w_iw_j$減少する．したがって，全対距離和は $l(e)w^{T'}w^{T''}$減少する．∎

性質 18 木構造ネットワークを T，部分辺を e_jとする．同じサイズの設備 F'と F''がともに e_jに含まれるならば，F'と F'' の全対距離和は等しい．

証明　性質 17による．∎

また，全対距離和の削減量は性質 19，20を満たす．

性質 19 木構造ネットワークを T，Tの重心を qとする．また，性質 15と同様に e'，e''，F'，F''，T'，T''，T'''，T''''を表す（図 5.3）．このとき，設備 F'，F''に対する部分辺 e'，e''の全対距離和削減量 $DP(e')$，$DP(e'')$ は，次式を満たす．

$$\begin{aligned} &DP(e')=DP(e'') && (w^{T''\cap T'''}=0)\\ &DP(e')<DP(e'') && (\text{otherwise}). \end{aligned} \tag{5.8}$$

証明　$w^{T'},w^{T'''}\le w^T/2$，$w^{T''}=w^T-w^{T'}$，$w^{T''''}=w^T-w^{T'''}$である．したがって，$w^{T''\cap T'''}=0$ ならば $w^{T'}=w^{T'''}$なので，性質 17より $DP(e')=DP(e'')$．さもなくば $w^{T'}<w^{T'''}$なので，$DP(e')<DP(e'')$．∎

性質 20 性質 16と同様に T，q，e'，e''，F，T'，T''，T'''，T''''を表す（図 5.4）．このとき，設備 Fに対する部分辺 e'，e''の全対距離和削減量 $DP(e')$，$DP(e'')$ は，次式を満たす．

$$\begin{aligned} &DP(e')=DP(e'') && (w^{T''\cap T'''}=0)\\ &DP(e')<DP(e'') && (\text{otherwise}). \end{aligned} \tag{5.9}$$

証明　$w^{T'},w^{T'''}\le w^T/2$，$w^{T''}=w^T-w^{T'}$，$w^{T''''}=w^T-w^{T'''}$である．したがって，性質 19と同様に，$w^{T''\cap T'''}=0$ ならば $DP(e')=DP(e'')$，さもなくば $DP(e')<DP(e'')$．∎

よって，距離和と全対距離和は次の性質 21を満たす．

性質 21 木構造ネットワークを T, Tの重心 qを含む設備を F，Fに隣接する長さϵの部分辺を e'，e''とする．このとき，Fに対する部分辺 e'，e''の距離和と全対距離和の関係は，$DD(e')<DD(e'')$ ならば $DP(e')<DP(e'')$ であり，逆も成り立つ．また，$DD(e')=DD(e'')$ ならば $DP(e')=DP(e'')$ であり，逆も成り立つ．

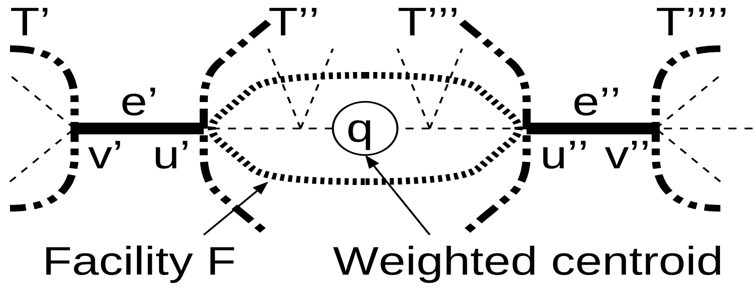


図 5.5: 木 Tの設備 Fと隣接する長さϵの部分辺 e'，e''（重心 $q \in V(T'' \cap T''''))$）

証明　長さ$\epsilon > 0$ とする．部分辺 e'，e''上の最も Fに近い点をそれぞれ u'，u''，最も遠い点をそれぞれ v'，v''とする．また，$T' \cap e' = v'$なる最大の部分木を T'，$T'' \cap e' = u'$なる最大の部分木を T''，$T''' \cap e'' = u''$なる最大の部分木を T'''，$T'''' \cap e'' = v''$なる最大の部分木を T''''とする．重心 qが T''''にも含まれる場合は，性質 16，20より，$DD(e') < DD(e'')$ かつ $DP(e') < DP(e'')$ である（図 5.4）．重心 qが T'にも含まれる場合も同様．一方，重心 qが $T'' \cap T'''$のみ含まれる場合を以下に示す（図 5.5）．$DD(e') < DD(e'')$ ならば，性質 13より $w^{T'} < w^{T''''}$．また，$w^{T''''} \leq w^T/2$．したがって，性質 17より $DP(e') < DP(e'')$ である．同様に，$DD(e') = DD(e'')$ ならば $DP(e') = DP(e'')$，$DD(e') > DD(e'')$ ならば $DP(e') > DP(e'')$ である．∎

5.2.3　全対距離和を指標とした最適設備問題

全対距離和を指標とした最適設備における性質について議論する．

5.2.3.1　最適な連続木形状設備

距離和最小の連続木形状設備は，距離和最小の連続木形状設備を求めることにより得られることを示す.

距離和最小の連続木形状設備は性質 22を満たす.

性質 22 木構造ネットワークを T，Tの重心を q，距離和最小の連続木形状設備を Fとする．すると，Fは qを含むか，さもなくば qの隣接頂点 v_jと qの間の辺 e_jに含まれる．ただし，後者の場合，v_jも重心である.

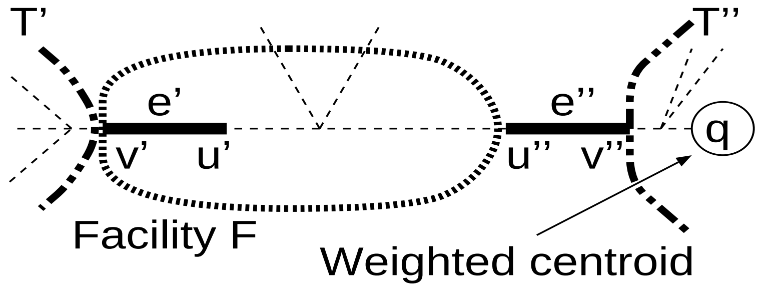


図 5.6: 木 Tの重心 qを含まない設備 Fと長さϵの部分辺 e'，e''

証明　Fは qを含まないとする．長さϵ（> 0）の部分辺を $e' = (u', v')$，$e'' = (u'', v'')$ とする．ただし， 最も qに近い Fの端点を u''として $F \cap e'' = u''$，$e'' \subseteq P(u'', q)$，また u''以外の Fの端点を v'として $v' \subseteq e'$，$e' \subseteq F$とする．また，$T' \cap e' = v'$なる最大の部分木を T'，$T'''' \cap e'' = v''$ なる最大の部分木を T''とする（図 5.6）．すると，F/e'が頂点を含むか，F/e'が頂点を含まず $w^{T''} > w^T/2$ の場合，性質 16より $D(F \cup e''/e') < D(F)$ なので，Fは距離和最小でない．F/e'が頂点を含まない場合，設備 F，部分辺 e'，e''は，一つの辺 $e_j = (v_j, v_k) \in E$に含まれる．そして，$w^{T'} = w^{T''} = w^T/2$ なので，頂点 v_j，v_kはともに重心となり，そのうちの一つは qである．したがって，Fは qを含むか，さもなくば qの隣接頂点 v_jと qの間の辺 e_jに含まれる．∎

同様に，全対距離和最小の連続木形状設備は性質 23を満たす.

性質 23 木構造ネットワークを T，Tの重心を q，全対距離和最小の連続木形状設備を Fとする．すると，Fは qを含むか，さもなくば qの隣接頂点 v_iと qの間の辺 e_iに含まれる．ただし，v_iは，qを含まない Tの重み最大の部分木に含まれる頂点である．

証明　Fは qを含まないとする．そして，性質 22と同様に e'，e''，T'，T''を表す（図 5.6）．すると，F/e'が頂点を含まない場合，性質 20より $P(F \cup e''/e') = P(F)$ である．よって，Fと全対距離和の等しい設備 F'で，F'/e'が頂点を含むものがある．一方，F/e'が頂点を含む場合，性質 20より $P(F \cup e''/e') < P(F)$ なので，Fは全対距離和最小でない．したがって，Fは qを含むか，さもなくば qの隣接頂点 v_jと qの間の辺 e_jに含まれる．さて，以下では qを含まず qの隣接頂点 v_jを含む Tの最大の部分木を T_j，T_jの重みを w^{T_j}と表す．Fが qの隣接頂点 v_iと qの間の辺 e_iに含まれるとすると，性質 18より，同じ全対距離和で qを含む設備 F'

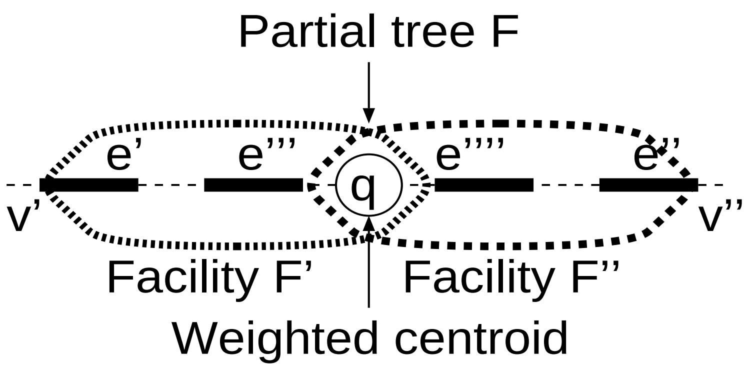


図 5.7: 木 Tの重心 qを含む，最適な連続木形状設備 F'と距離和最小の連続木形状設備 F''

がある．もし，$w^{T_k} > w^{T_i}$なる子頂点 v_kがあるならば，qを含む e_k上の長さϵの部分辺を e''，F'の q以外の端点を含む F'上の長さϵの部分辺を e'とすると，$P(F \cup e''/e') < P(F)$ なので，Fは全対距離和最小でない．したがって，qのすべての隣接頂点 v_jに対して $w^{T_i} \geq w^{T_j}$． ∎

よって，距離和最小の連続木形状設備と最適な連続木形状設備は，次の性質 24，25，26 を満たす.

性質 24 距離和最小の連続木形状設備は，最適な連続木形状設備である．

証明　木 Tのサイズは設備サイズ上限 lより大きいとする．このとき，最適な連続木形状設備も距離和最小の連続木形状設備もサイズは lとなる．距離和最小の連続木形状設備を Fとする．Fが Tの重心 qを含む場合，性質 21より Fは最適な連続木形状設備である．Fが qを含まない場合，性質 22より Fは Tの 2 つの重心間の辺 e_jに含まれる．よって，性質 14より Tの重心を含み辺 e_jに含まれる距離和最小の設備 F'がある．性質 21より，F'は最適な連続木形状設備である．したがって，性質 18より Fは最適な連続木形状設備である． ∎

性質 25 2 つ以上の頂点を含む設備に対し，最適な連続木形状設備と距離和最小の連続木形状設備は等しい．

証明　木 Tのサイズは設備サイズ上限 lより大きいとする．このとき，最適な連続木形状設備も距離和最小の連続木形状設備もサイズは lとなる．Tの重心を q，最適な連続木形状設備を F'，距離和最小の連続木形状設備を F''とする．ここで，性質 22，23より，F'と F''は qを含む．すると，F'と F''の積 $F = F' \cap F''$は qを含む部分木である．Fのサイズ l'は

設備 F' と F'' のサイズ l より小さいとする．長さ ϵ（>0）の部分辺 e' は，$e' \subseteq F'$ で $e' \cap F''$ が空か点である部分辺で，その一つの端点 v' は F' の端点とする．同じく長さ ϵ の部分辺 e'' は，$e'' \subseteq F''$ で $e'' \cap F'$ が空か点である部分辺で，その一つの端点 v'' は F'' の端点とする．パス $P(q, v')$ 上の長さ ϵ の部分辺 e''' は，$e''' \subseteq F'$ で $e''' \cap F''$ が点となる部分辺とする．パス $P(q, v'')$ 上の長さ ϵ の部分辺 e'''' は，$e'''' \subseteq F''$ で $e'''' \cap F'$ が点となる部分辺とする（図 5.7）．すると，性質 13より，$DD(e''') \leq DD(e'')$．性質 17，21より，$DD(e'''') \leq DD(e')$．性質 15より，$DD(e') \leq DD(e''')$，$DD(e'') \leq DD(e'''')$．したがって，$DD(e''') = DD(e'')$ となる．よって，部分木 $F''' = F'' \cup e'''/e''$ は距離和最小の連続木形状設備であり，積 $F' \cap F'''$ のサイズは $l' + \epsilon$ となる．以上により，最適な連続木形状設備 F' に対し，積 F のサイズが l となる距離和最小の連続木形状設備が存在する．すなわち，F' は距離和最小の連続木形状設備である．逆は，性質 24による．∎

性質 26 木構造ネットワーク T の各辺の長さが 1 の場合，最適な離散木形状設備と距離和最小の離散木形状設備は等しい．

証明　離散木形状設備は 2 つ以上の頂点を含む．よって，性質 25の証明において，$\epsilon = 1$ とすればよい．∎

距離和最小の連続木形状設備や，木構造ネットワーク T の各辺の長さが 1 の場合の距離和最小の離散木形状設備の算出は $O(n)$ 時間で算出できるので [40]，この場合の最適な木形状設備は $O(n)$ 時間で求まる．

5.2.3.2　最適な離散木形状設備

図 5.8に，頂点数 $n = 6$ の木 T に対するサイズ 3 の離散木形状設備を示す．図の F' はサイズ $l = 3$ の距離和最小の木形状離散設備，F'' は最適な離散木形状設備であり，$D(F') = 14 < 15 = D(F'')$，$P(F') = 64 > 63 = P(F'')$ である．したがって，最適な離散木形状設備と距離和最小の離散木形状設備が一致しないことが分る．

5.2.3.3　最適な連続パス形状設備

先ず，最適な連続パス形状設備が距離和最小の離散パス形状設備と異なることを示す．図 5.9の F' はサイズ $l = 20$ の最適な連続パス形状設備，F'' は距離和最小の連続パス形状設備であり，$D(F') = 23 > 22 = D(F'')$，$P(F') = 129 < 134 = P(F'')$ である．

また，次の性質 27が示される．

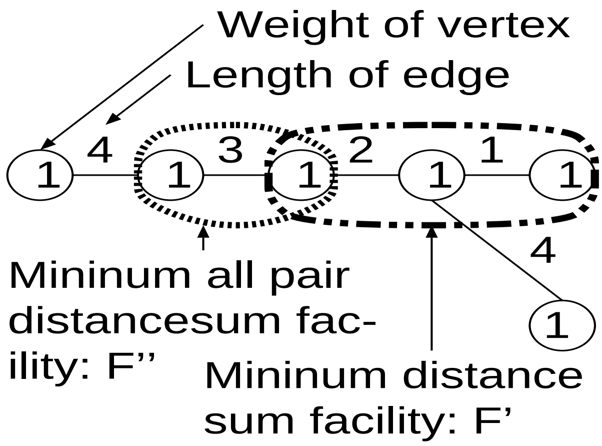


図 5.8: 木 Tのサイズ $l = 3$ の最適な離散木形状設備 F'と距離和最小の離散木形状設備 F''

性質 27 最適な連続パス形状設備は，$O(n^3)$ 時間で求まる．

証明　文献 [35] の距離和最小の連続パス形状設備配置アルゴリズムをそのまま用いる．すなわち，先ず，Tの二つの葉 u，vを選択する．そして，$P(u, v)$ 上にある頂点 u を含むサイズ lのパス Fを求め，Fの全対距離和 $P(F)$ を求める．これらは，$O(n)$ 時間で求まる．次に，パス Fを頂点 v方向に，Fの端点の一方が頂点に達するまで移動させ，パス F'とする．パス F'は $O(1)$ 時間で求まり，性質 17より全対距離和の差 $P(F') - P(F)$ も $O(1)$ 時間で求まる．そして，パスの移動回数は $2|V(P(u, v))|$ 以下，すなわち $O(n)$ である．葉 u，v の組合せ数は $O(n^2)$ なので，最適な連続パス形状設備の算出は $O(n^3)$ 時間で求まる．∎

5.2.3.4　最適な離散パス形状設備

最適な離散パス形状設備と距離和最小の離散パス形状設備が異なることは，図 5.9の設備 F'，F''が離散設備であることで示される．

また，次の性質 28が示される．

性質 28 最適な離散パス形状設備は，$O(n^3)$ 時間で求まる．

証明　文献 [37] の距離和最小の離散パス形状設備配置アルゴリズムをそのまま用いる．すなわち，Tの二頂点の組み合わせ数が $O(n^2)$ であり，設備 Fによる全対距離和の削減量 $DP(F)$ は性質 17により $O(n)$ で求まることから，最適な連続パス形状設備は $O(n^3)$ 時間で求まる．∎

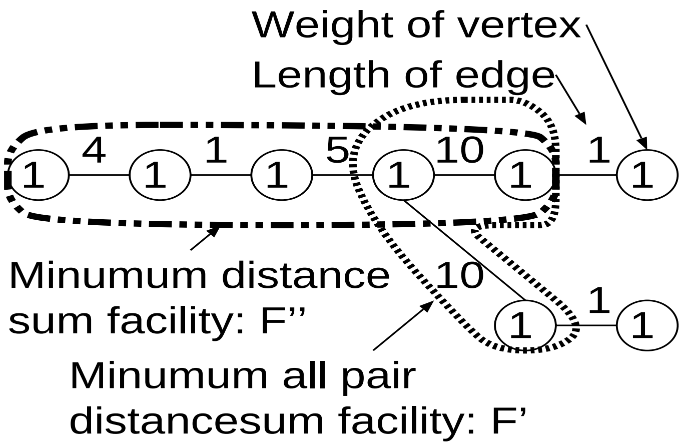


図 5.9: 木 T のサイズ $l = 20$ の最適な連続パス形状設備 F' と距離和最小の連続パス形状設備 F''

5.2.4　最適な離散木形状設備の配置

ここでは，最適な離散木形状設備の配置が NP 困難であることを示す．また，最適な離散木形状設備配置を求める疑似多項式時間アルゴリズムや，完全多項式近似な離散木形状設備配置を求めるアルゴリズムを示す.

5.2.4.1　最適な離散木形状設備算出の実行時間

文献 [37] のキャタピラ木の手法をそのまま使用することにより，次の性質 29が成り立つ.

性質 29 最適な離散木形状設備の算出は NP 困難である.

証明　二分割問題は，n 個の整数の集まり $A = \{a_1, a_2, ..., a_n\}$，その総和 $2k = \sum_1^n a_i$ に対し，A の要素の集まりで和が k となるものがあるかどうかを求めるものであり，NP 完全なことが知られている [42]．二分割問題に対し，図 5.10に示す頂点数 $2n + 2$ のキャタピラ木 T を作成する．図の $a_1, ..., a_n$，$2k$ は辺の長さであり，各頂点の重みは 1 である．木 T にサイズ $l = (2n + 1)k$ 以下で全対距離和が $(2n + 1)k$ 以下の離散木形状設備があることと，A が二分割できることは等しい．したがって，最適な離散木形状設備の算出は NP 困難である．∎

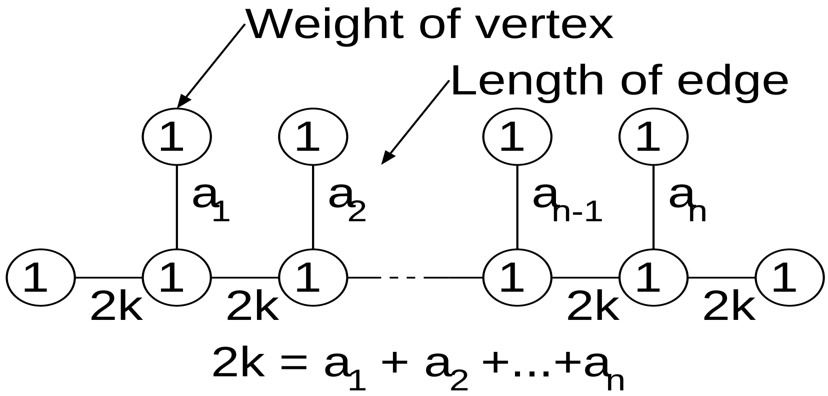


図 5.10: 分割問題に対応する木 T

5.2.4.2　最適な離散木形状設備算出のアルゴリズム

文献 [40] の疑似多項式時間アルゴリズムを用いることにより，最適な離散木形状設備を求めることができる．また，文献 [40] の完全多項式近似アルゴリズムを用いることにより，完全多項式近似な離散木形状設備を求めることができる．

重み関数 Wは，木 Tの各頂点 v_iとその重み w_iの組みの集合とする．重み関数として Wを用いた場合の Tの重心を q，全対距離和を $P^W(F)$ と表す．以下では，全対距離和 $P^W(F)$ が最小となる設備を，重み Wで最適な離散木形状設備とよぶ．

（１）重心 qを含む，重み Wで最適な離散木形状設備の算出

文献 [40] のアルゴリズムで使用されている，木 Tの各頂点 v_jに対する変数 W_jと D_jの定義を，次式に変更する．

$$\begin{aligned} W_j &= w^{T'} w^{T''}, \\ D_j &= \textstyle\sum_{v_i \in T', v_i \neq v_j} a_i W_i. \end{aligned} \tag{5.10}$$

ただし，v_jの親を v_kと表したとき，v_kを含み v_iを含まない最大の離散部分木を T'，v_kを含み v_jを含まない最大の離散部分木を T''とする．

すると，文献 [40] のアルゴリズムにより，木 Tの重心 qを含む，重み Wで最適な離散木形状設備を求めることができる．すなわち，木 Tの根 v_1を木 Tの重心 qとする．このとき式 5.10で定義した W_jは，各頂点の重み w_iの値に関わらず，頂点 v_jとその親頂点 v_iに対して $W_j < W_i$である．したがって，文献 [40] で示された，Greedy 法による 2 近似アルゴリズム，Left-Right 法による疑似多項式時間アルゴリズム，$(1+\epsilon)$ 法による完全多項式近似

アルゴリズムをそのまま使用できる．その結果，$O(n \log n)$ 時間で 2 倍近似な離散木形状設備を，$O(n \ \min(l, \mathrm{WD}(l)))$ 時間で最適な離散木形状設備を，$O(n^2)$ 時間で $1+\epsilon$倍近似な離散木形状設備を求めることができる．ただし，$\mathrm{WD}(l)$ は，サイズ l以下の最適な離散木形状設備の全対距離和の値である．

（２）重み Wで最適な離散木形状設備の算出

木 Tの重心 qを含む，重み Wで最適な離散木形状設備の算出は，上記（１）項のアルゴリズムにより求まる．

一方，Tの重心 qを含まない重み Wで最適な離散木形状設備の算出は，次のように行う．木 Tの根 v_1の子頂点を v_jとする．頂点 v_jを含み v_1を含まない，Tの最大の離散部分木を T'とする．また，重み関数 W'を，$w'_i = w_i$（$v_i \in V(T'), i \neq j$），$w'_j = w^T - w^{T'} + w_j$とする．明らかに，木 T'の重み W'で最適な離散木形状設備は，T'に包含される重み Wで最適な離散木形状設備となる．よって，木 Tの重み Wで最適な離散木形状設備は再帰的に求まる．再帰において Tの各頂点 v_jは一度しか根として選択されないので，上記（１）項のアルゴリズムの実行回数は n 回である．

したがって，2 倍近似な離散木形状設備は $O(n^2 \log n)$ 時間，最適な離散木形状設備は $O(n^2 \ \min(l, \mathrm{WD}(l)))$ 時間，$1+\epsilon$倍近似な離散木形状設備は $O(n^3)$ 時間で求めることができる．

5.2.5　まとめ

本節では，実際の計算機ネットワークの構築を考慮した実用的な設備の評価指標である全対距離和が提案され，各辺の長さが任意である木構造の通信ネットワークに対する，全対距離和が最小となる木形状やパス形状設備の性質や配置方法が示された．

先ず，全対距離和を用いた設備配置を提案し，その性質を示した．

次に，全対距離和最小の設備の性質について議論した．先ず，全対距離和最小の連続木形状設備は距離和最小の連続木形状設備に等しいことを示した．この結果，全対距離和最小の連続木形状設備は $O(n)$ 時間で求まることが明らかになった．次に，全対距離和最小の離散木形状設備と距離和最小の離散木形状設備が異なることを示した．また，全対距離和最小の連続パス形状設備と距離和最小の連続パス形状設備が異なること，全対距離和最小の連続パス形状設備を求める $O(n^3)$ 時間のアルゴリズムが存在することを示した．更に，全対距離和最小の離散パス形状設備と距離和最小の離散パス形状設備が異なること，全対距

離和最小の連続パス形状設備を求める $O(n^3)$ 時間のアルゴリズムが存在することを示した．

更に，全対距離和最小の離散木形状設備の配置について議論し，離散木形状設備の配置は NP 困難であることを明らかにした．また，最適な離散木形状設備を求める疑似多項式時間アルゴリズムと完全多項式近似アルゴリズムについて議論した．

本節の議論により，超並列計算システムの通信ネットワークに二点間の通信の平均通信コストを重視した設備を配置することができ，実用的な通信ネットワークシステムを構築することができる．

5.3　高速化率一定の設備の配置

超並列・分散計算システムの通信ネットワークシステムに配置された高速通信路は，その通信路を通過する通信の通信コストを削減することができる．これは，該当する辺の長さを縮小することに対応する．

本節では，各辺の長さが任意である木構造の通信ネットワークに対する，二点間の平均距離が最小となる木形状やパス形状設備について議論する．その結果，高速通信設備の高速化率が一定の場合に，二点間の平均距離が最小となる木形状やパス形状設備は 5.2節の手法により求まることが示される．実際の通信ネットワークシステムの構築において，高速通信設備の高速化率一定という条件を満たす場合は多い．たとえば，10Mbps の通信路よりなる通信ネットワークシステムに 100Mbps の高速通信設備を配置する場合，通信路の長さ d に対する通信コスト削減関数 $f(d) = d/10$ となり，高速化率は 10 である．これにより，超並列計算機システムの通信ネットワークに設備内の通信コストを考慮した設備を配置することができ，実用的な通信ネットワークシステムを構築することができる．

本節の内容を以下に述べる．先ず，設備内の通信コストを 0 とした場合の二点間の距離の総和を用いた設備配置を提案し，その最適な設備が全対距離和最小の設備に等しいことを示す．次に，高速通信設備の配置により当該辺の長さ d を $f(d)$ に縮小する場合を議論し，高速化率が一定の場合は設備内の通信コストを 0 とした場合の設備配置に等しいことを示す．この結果，高速通信設備の高速化率が一定の場合の最適な設備は，5.2節で示した方法を用いることにより配置できることを明らかにする．

5.3.1　設備配置の評価関数

先ず，設備内の通信コストを 0 にするモデルを提案する．

無向の木構造ネットワークを $T = (V, E)$ とする．ただし，頂点集合 $V = \{v_1, v_2, ..., v_n\}$，辺集合 Eとする．各辺は単間隔で表され，内点を使用可能とする．また，$A(T)$ を Tの辺上にある点の集合とする．$A(T)$ は，$n-1$ 個の単間隔の接続で表される，接続され閉じた集合である．$P(v_i, v_j)$ を，$A(T)$ に含まれる，点 v_iと v_jを接続するパスとする．v_1を Tの根とする．$i = 2, ..., n$ に対し，$g(v_i)$ を v_iの親，すなわち $P(v_1, v_i)$ 上にあり，v_i以外で最も近くにある $v \in V$の点である．集合 $E = \{e_2, e_3, ..., e_n\}$ は，v_jと $g(v_j)$ を接続する辺 e_jの集合である．点 (x, j)（$0 \le x \le 1$）は，e_j上で，$g(v_j)$ からの距離が x である点である．

そして，$T' \subseteq A(T)$ で，接続され閉じているものを部分木とする．部分木 T'は，どの二辺の交点も空であるか Vに含まれる点である部分辺の集合で表される．$E(T')$ をこの部分辺の集合とし，$V(T')$ を Vに含まれる $A(T')$ の点の集合とする．部分木 T'が離散であるとは，$E(T')$ の要素すべての端点が Vに含まれる点であることである．

各辺 e_jの長さを $l(e_j)$ で表し，点 (x, j) と (y, j) を接続する部分辺の長さを $l(e_j)|x-y|$ とする．そして，$A(T)$ の距離関数は $a_2, ..., a_n$で与えられ，点 (x, i) と (y, j) の距離 $d((x, i), (y, j))$ は，そのパスに含まれる部分辺の長さの和となる．また，T'のサイズ $L(T')$ は $E(T')$ の各部分辺の長さの和とする．部分木 $T' \in A(T)$ と点 $v_j \in V$との距離 $d(v_j, T')$ は，v_jから最も近い T'上の点までの距離とする．

各頂点 v_iの重みを w_iで表す．また，部分木 T'の重み $w^{T'}$は，T'の各頂点の重みの和とする．すなわち，次式で表される．

$$w^{T'} = \textstyle\sum_{v_i \in V(T')} w_i. \tag{5.11}$$

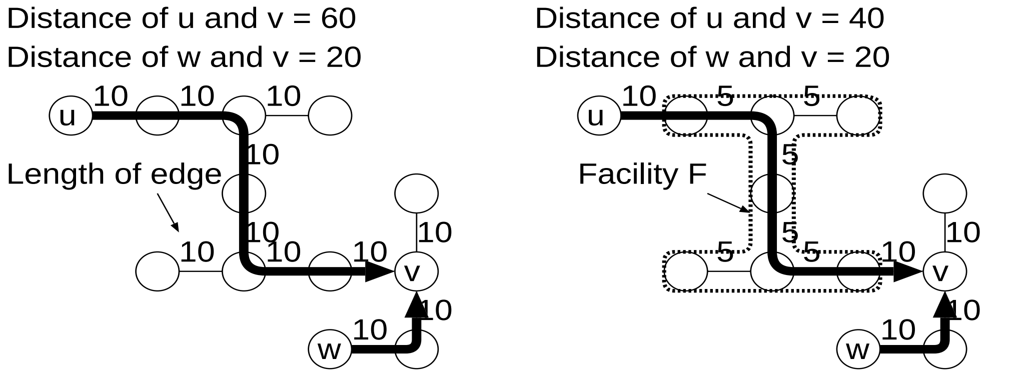
部分木 T'，T''の結合 $T' \cup T''$は，$v_i \in T'$または $v_i \in T''$なる点 v_iの集合とする．

設備 Fは部分木である．特に離散部分木である設備を離散設備とよぶ．離散設備と区別するとき，部分木である設備を連続設備とよぶ.

設備 Fの（重み付き）全対コスト $C(F)$ は，設備 Fを配置後の二点間の距離に二点の重みの積を乗じたものの総和であり，次式で表される．

$$C(F) = \textstyle\sum_{v_i, v_j \in V} \{w_i w_j d(T_F; v_i, v_j)\}. \tag{5.12}$$

ただし，$d(T_F; v_i, v_j)$ は木構造ネットワーク T_Fの二点 v_i，v_j間の距離であり，T_Fは Tと同



(a) Network T.　　(b) Network T_F generated from facility F.

図 5.11: 設備配置と全対コスト

一形状で各辺 e_j の長さ $l(e_j)$ を次式で表す $l'(e_j)$ に置き換えたものである．

$$l'(e_j) = \begin{cases} 0 & (e_j \in E(F)) \\ l(e_j) & (otherwise). \end{cases} \tag{5.13}$$

図 5.11に設備と全対コストの例を示す．各点の重み w_j は 1 とし，設備により通信路の通信コストが半減するとする．図 5.11(a) のネットワーク T において，頂点 u，v 間の距離は 60，頂点 w，v 間の距離は 20 である．設備 F に対するネットワーク T_F は図 5.11(b) のようになる．T_F における頂点 u，v 間の距離は 40，頂点 w，v 間の距離は 20 となる．そして，全対コスト $C(F) = 1570$ となる．

本節では，全対コスト $C(F)$ が最小となる設備を最適な設備とする．

木構造ネットワーク T に木形状やパス形状の設備を配置する場合，明らかに，全対コスト $C(F)$ と全対距離和 $P(F)$ は等しい．したがって，全対コストモデルにおける最適な木形状やパス形状設備の設備配置は，明らかに 5.2節で議論した全対距離和を用いた設備配置に等しい．

なお，設備が複数の部分設備からなる場合は，全対コストと全対距離和の両評価関数は異なる．たとえば，各辺を共有することなく，かつ各頂点を共有可能として，複数個の木形状部分設備からなる設備を配置する場合を考える．図 5.12に，木 T にサイズ 2 の木形状の部分設備 2 つからなる設備を配置する場合を示す．図の F' は全対コスト最小の設備，F'' は全対距離和最小の設備であり，$C(F') = 88 < 100 = C(F'')$，$P(F') = 88 > 40 = P(F'')$ である．特に，各辺の長さが 1 である木構造ネットワークに複数個の木形状部分設備から

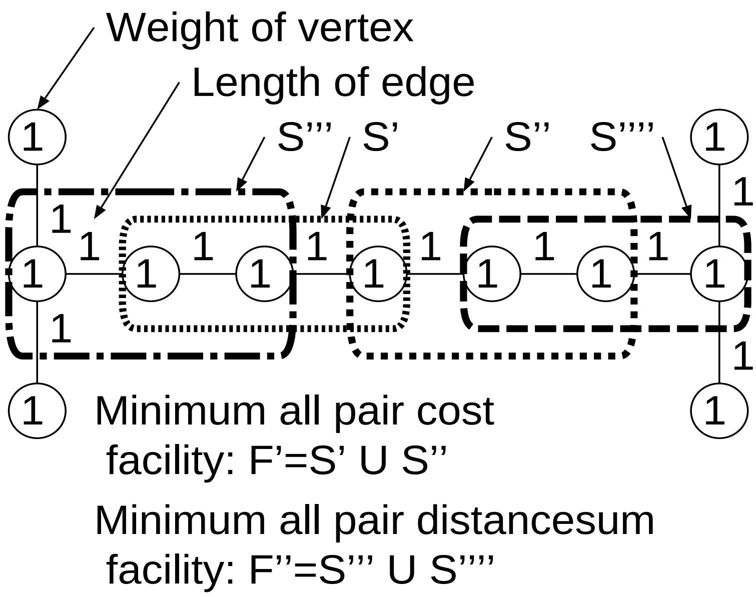


図 5.12: サイズ 2 の木形状の部分設備 2 つからなる，全対コスト最小の設備 F'と全対距離和最小の設備 F''

なる設備を配置する場合，全対コスト最小の設備は明らかに部分木であるが，全対距離和最小の設備は部分木とは限らない．

5.3.2 設備の通信コストを考慮したモデル

設備の通信コストを考慮したモデルについて検討する．ここでは，通信コスト削減関数 fを与えた場合の最適な設備配置について議論する．

通信コスト削減関数 fは，設備内の長さ d の辺を長さ $f(d)$ に削減する関数である．ただし，$0 \leq f(d) < d$ とする．

通信コスト削減関数 fに対する設備 Fの（重み付き）全対コスト $C_f(F)$ は，設備 Fを配置後の二点間の距離の総和であり，次式で表される．

$$C_f(F) = \sum_{v_i, v_j \in V} \{w_i w_j d(T_F; v_i, v_j)\}. \tag{5.14}$$

ただし，$d(T_F; v_i, v_j)$ は木構造ネットワーク T_Fの二点 v_i，v_j間の距離であり，T_Fは Tと同

一形状で各辺 e_jの長さ $l(e_j)$ を次式で表す $l'(e_j)$ に置き換えたものである．

$$l'(e_j) = \begin{cases} f(l(e_j)) & (e_j \in E(F)) \\ l(e_j) & (otherwise). \end{cases} \tag{5.15}$$

そして，全対コスト $C_f(F)$ が最小となる設備を最適な設備とする．

なお，設備を配置しない場合の木 Tの全対コストは $C(\phi)$ で表す．

5.3.3　高速化率一定の場合

高速化率一定の場合，すなわち通信コスト削減関数 $f(d) = \alpha \cdot d$ （αは $0 \leq \alpha < 1$ の定数）の場合の最適な設備が，設備内の通信コストが 0 の場合，すなわち $f(d) = 0$ の場合の最適な設備に等しいことを示す．

木構造ネットワークを T，通信コスト削減関数を fとする．また，部分辺 $e = (u, v)$ に対し，点 u を含み u 以外の e の点を含まない Tの最大の部分木を T_1^e，点 vを含み v以外の e の点を含まない Tの最大の部分木を T_2^eと表す．すると，明らかに次の性質 30が成り立つ．

性質 30 木構造ネットワークを T，通信コスト削減関数 f，設備を Fとする．また，Fの部分辺の集合を $S = \{e^1, e^2, ..., e^m\}$ と表す．ただし，$F = e^1 \cup e^2 \cup ... \cup e^m$であり，各 $e^i \cap e^j$は頂点か空である．このとき，設備 Fの配置による全対コストの削減量 $DC_f(F) = C(\phi) - C_f(F)$ は，次式で表される．

$$\begin{aligned} DC_f(F) &= \textstyle\sum_{v_i, v_j \in V(T)} \sum_{e^k \in S, e^k \subseteq P(v_i, v_j)} (l(e^k) - f(l(e^k))) \\ &= \textstyle\sum_{e^k \in S} \{(l(e^k) - f(l(e^k))) w^{T_1^{e^k}} w^{T_2^{e^k}}\}. \end{aligned} \tag{5.16}$$

性質 30より，次の性質 31が成り立つ．

性質 31 高速化率が一定，すなわち通信コスト削減関数 $f(d) = \alpha \cdot d$（$0 \leq \alpha < 1$）の場合の最適な設備は等しい．

証明　木 Tのサイズは設備サイズ lより大きいとする．通信コスト削減関数 $f'(d) = \alpha \cdot d$，$f''(d) = \beta \cdot d$ とする．ただし，α，βは定数で，$0 \leq \alpha, \beta < 1$ とする．通信コスト削減関数が f'の場合の最適な設備を F'，f''の場合の最適な設備を F''とする．F'の部分辺の集合を $S' = \{e'^1, e'2, ..., e'^m\}$，$F''$の部分辺の集合を $S'' = \{e''^1, e''^2, ..., e''^m\}$ と表す．ただし，$F' = e'^1 \cup e'^2 \cup ... \cup e'^m$，$F'' = e''^1 \cup e''^2 \cup ... \cup e''^m$であり，各 $e'^i \cap e'^j$，$e''^i \cap e''^j$は頂点か空

である．定義より，$DC_{f'}(F') \geq DC_{f'}(F'')$ である．また，$DC_{f''}(F') \leq DC_{f''}(F'')$ なので，性質 30 より，

$$\textstyle\sum_{e^k \in S'}\{(1-\beta)l(e^k)w^{T_1^{e^k}}w^{T_2^{e^k}}\} \leq \sum_{e^k \in S''}\{(1-\beta)l(e^k)w^{T_1^{e^k}}w^{T_2^{e^k}}\}. \tag{5.17}$$

両辺を $(1-\alpha)/(1-\beta)$ 倍して，

$$\textstyle\sum_{e^k \in S'}\{(1-\alpha)l(e^k)w^{T_1^{e^k}}w^{T_2^{e^k}}\} \leq \sum_{e^k \in S''}\{(1-\alpha)l(e^k)w^{T_1^{e^k}}w^{T_2^{e^k}}\}. \tag{5.18}$$

すなわち，$DC_{f'}(F') \leq DC_{f'}(F'')$．よって，$DC_{f'}(F') = DC_{f'}(F'')$ となる．同様に，$DC_{f''}(F') = DC_{f''}(F'')$．したがって，全対コスト $C_{f'}(F') = C_{f'}(F'')$，$C_{f''}(F') = C_{f''}(F'')$．すなわち，通信コスト削減関数 f' の場合の最適な設備は通信コスト削減関数 f'' の場合の最適な設備であり，逆も成り立つ．∎

特に，設備内の通信コスト 0，すなわち $f(d) = 0 \cdot d$ の場合の最適な設備と，高速化率が一定の場合の最適な設備は等しい．なお，性質 30，31は，設備の形状に関わらない．

5.3.4 まとめ

本節では，実際の計算機ネットワークの構築を考慮した，設備内の通信コストを考慮した実用的な評価指標である全対コストが提案され，各辺の長さが任意である木構造の通信ネットワークに対する全対コストが最小となる木形状やパス形状設備について議論した．

先ず，設備内の通信コストを 0 とした場合の二点間の距離の総和を用いた設備配置を提案し，木形上やパス形状の設備配置において，全対コスト最小の設備と，5.2節で議論した全対距離和最小の設備とが等しいことを明らかにした．

次に，高速通信設備の配置により当該辺の長さ d を $f(d)$ に縮小する場合を議論した．そして，高速化率一定の場合，すなわち通信コスト削減関数 $f(d) = \alpha \cdot d$（αは $0 \leq \alpha < 1$ の定数）の場合の最適な設備が，設備内の通信コストを 0 とした場合の最適な設備に等しいことを示した．すなわち，高速化率一定の場合は，5.2節で示したアルゴリズムを用いることにより，最適な木形状やパス形状設備を求めることができる．

本節の議論により，設備内の通信コストを考慮した設備配置による，超並列計算機システムの実用的な通信ネットワークシステムの構築指針を与えることができる．

5.4 実用的な設備の配置

従来の設備配置は，各辺の長さ a_iを設備サイズの制約と評価関数の両方に用いていた．しかし，各辺に対する設備サイズの制約と評価関数を分離して取り扱うことにより，幅広い計算機ネットワークシステムに適合した効率的な設備を配置することが可能となる．そこで本節では，各辺に対し，設備の構築コストに対応する辺の長さ a_i，設備の評価に対応するコスト c_iを用いる，より実用的な設備配置を提案し，サイズ指定の木形状やパス形状設備の配置について議論する．従来の距離和や全対距離和が最小となる離散木形状設備を算出する近似アルゴリズムでは，実用設備配置における最適な離散木形状設備を算出できない．そこで，文献 [40] の疑似多項式時間アルゴリズムを用いることにより実用設備配置における最適な離散木形状設備が求まることを示す．通信ネットワークの各辺の長さが 1 の場合には，このアルゴリズムの実行時間は $O(n^3)$ 時間である．

更に，従来の距離和を用いる設備配置，全対距離和を用いる設備配置，全対コストを用いる設備配置が，距離和を用いる実用設備配置により求まることを示す．これにより，距離和を用いる実用設備配置により，距離和を用いる従来の設備配置よりも幅広い通信ネットワークシステム構築を行うことができることを示す．特に，設備の高速化率が一定でない場合の全対コスト最小の設備も，距離和を用いる実用設備配置により求めることができる．

先ず，設備構築コスト a_iと設備評価コスト c_iを有する実用的な設備配置を定式化する．次に，サイズ指定の木形状やパス形状設備の実用設備配置について議論し，文献 [40] の疑似多項式時間アルゴリズムを用いることにより最適な離散木形状設備が求まることを示す．また，文献 [35][37] のアルゴリズムを用いることにより，最適な連続および離散パス形状設備が求まることを示す．更に，従来の距離和を用いる設備配置，全対距離和や全対コストを用いる木形状やパス形状の設備配置が，距離和を用いる実用設備配置により求まることを示す．

5.4.1 実用設備

無向の木構造ネットワークを $T = (V, E)$ とする．ただし，頂点集合 $V = \{v_1, v_2, ..., v_n\}$，辺集合 Eとする．各頂点は深さ順に並ぶものとする．また各辺は単間隔で表され，内点を使用可能とする．また，$A(T)$ を Tの辺上にある点の集合とする．$A(T)$ は，$n-1$ 個の単間隔の接続で表される，接続され閉じた集合である．$P(v_i, v_j)$ を，$A(T)$ に含まれる，点 v_iと

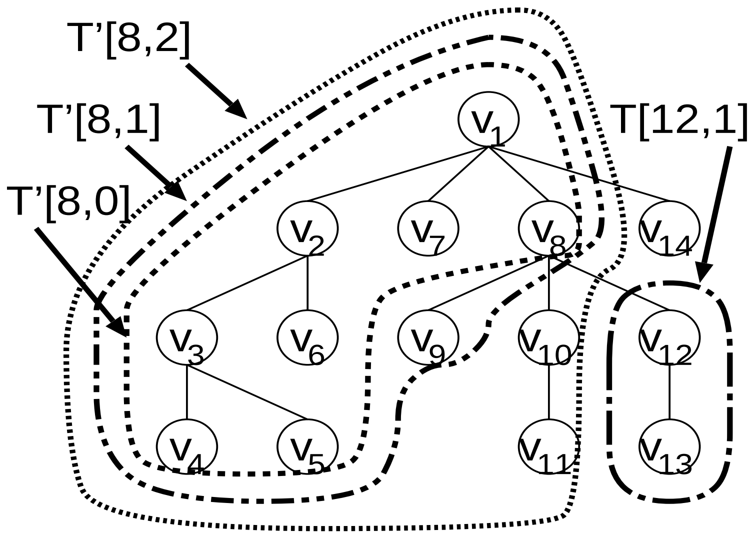


図 5.13: 部分木 $T[j,t]$ と $T'[j,t]$

v_jを接続するパスとする．v_1を Tの根とする．$j = 2, ..., n$ に対し，$g(v_j)$ を v_jの親，すなわち $P(v_1, v_j)$ 上にあり，v_j以外で最も近くにある $v \in V$の点である．また $v_j = g(v_i)$ となる頂点 v_iを，v_jの小頂点とよぶ．v_jの小頂点の集合を S_j，小頂点数を s_jと表す．小頂点は順序付けし，$t = 1, ..., s_j$に対し $v_{j(t)}$と表す．頂点 v_iに対し，パス $P(v_1, v_i)$ 上に v_jがあるならば，v_iを v_jの子孫とよぶ．集合 $E = \{e_2, e_3, ..., e_n\}$ は，v_jと $g(v_j)$ を接続する辺 e_jの集合である．点 (x, j)（$0 \leq x \leq 1$）は，e_j上で，$g(v_j)$ からの距離が x である点である．

そして，$T' \subseteq A(T)$ で，接続され閉じているものを部分木とする．部分木 T'は，どの二辺の交点も空であるか Vに含まれる点である部分辺の集合で表される．$E(T')$ をこの部分辺の集合とし，$V(T')$ を Vに含まれる $A(T')$ の点の集合とする．部分木 T'が離散であるとは，$E(T')$ の要素すべての端点が Vに含まれる点であることである．図 5.13は，部分木 $T[j,t]$ と $T'[j,t]$ の例を示す．部分木 $T[j,t]$ は，頂点 v_jおよび，v_jの最初の t 個の小頂点 $v_{j(i)}$（$i = 1, ..., t$）とその子孫を含み，他の頂点を含まない離散部分木とする．また部分木 $T'[j,t]$ は，頂点 $v_1, ..., v_{j-1}$および $V(T[j,t])$ を含み，他の頂点を含まない離散部分木とする．

各辺 e_jの長さを $l(e_j)$ で表し，点 (x, j) と (y, j) を接続する部分辺の長さを $l(e_j)|x-y|$ とする．そして，$A(T)$ の距離関数は $a_2, ..., a_n$で与えられ，点 (x, i) と (y, j) の距離 $d((x, i), (y, j))$ は，そのパスに含まれる部分辺の長さの和となる．また，T'のサイズ $L(T')$ は $E(T')$ の各

部分辺の長さの和とする．部分木 $T' \in A(T)$ と点 $v_j \in V$との距離 $d(v_j, T')$ は，v_jから最も近い T'上の点までの距離とする．

また，各辺 e_jにおける単位長さ当りのコストを $c(e_j)$ で表し，点 (x, j) と (y, j) を接続する部分辺のコストを $c(e_j)|x-y|$ とする．そして，$A(T)$ のコスト関数は $c_2, ..., c_n$で与えられ，点 (x, i) と (y, j) 間のコスト $c((x, i), (y, j))$ は，そのパスに含まれる部分辺のコストの和となる．部分木 $T' \in A(T)$ と点 $v_j \in V$の間のコスト $c(v_j, T')$ は，v_jと v_jから最も近い T'上の点までの間のコストとする．

各頂点 v_iの重みを w_iで表す．また，部分木 T'の重み $w^{T'}$は，T'の各頂点の重みの和とする．すなわち，次式で表される．

$$w^{T'} = \textstyle\sum_{v_i \in V(T')} w_i. \tag{5.19}$$

設備 Fは，サイズ $L(F) \leq l$である部分木である．特に離散部分木である設備を離散設備とよぶ．離散設備と区別するとき，部分木である設備を連続設備とよぶ．

設備 Fの（重み付き）距離和 $D(F)$ は，次式で表される．

$$D(F) = \textstyle\sum_{v_i \in V} \{w_i \cdot c(v_i, F)\}. \tag{5.20}$$

設備 Fの（重み付き）偏差 $ECC(F)$ は，次式で表される．

$$ECC(F) = \max_{v_i \in V} \{w_i + c(v_i, F)\}. \tag{5.21}$$

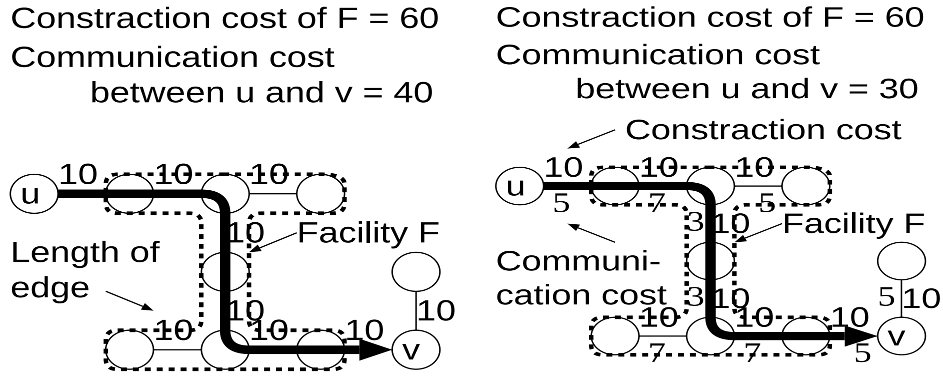
設備 Fの（重み付き）全対距離和 $P(F)$ は，次式で表される．

$$P(F) = \textstyle\sum_{v_i, v_j \in V} \{w_i w_j \min(d(v_i, F) + d(v_j, F), d(v_i, v_j))\}. \tag{5.22}$$

実用設備配置は，この距離和 $D(F)$，偏差 $ECC(F)$，全対距離和 $P(F)$ の評価関数のいずれかが最小または最大となる，サイズ l以下の設備を求めるものである．

図 5.14に実用設備配置の例を示す．図 5.11(a) は従来の設備配置であり，二点間の距離やコストは辺の長さで定まる．図 5.11(a) において，頂点 u，v間の距離は 40，設備 Fのコストは 60 である．実用設備配置では，図 5.11(b) のように距離とコストを分離する．図 5.11(b) において，頂点 u，v間の距離は 30，設備 Fのコストは 60 となる．実用設備配置では，評価関数の値はこの距離のみで定まり，コストに対して独立である．

本節では，評価関数の値が最小となる設備を最適な設備とよぶ．



(a) Usual facility location.　(b) Practical facility location.

図 5.14: 従来の設備配置と実用設備配置

5.4.2　実用設備配置

実用設備設備について議論し，最適な離散木形状設備，連続および離散パス形状設備を求める．

先ず，最適な離散木形状設備を求める．

前処理として，Tの各辺 e_jの重み W_j，コスト D_jを算出する．重み W_j，コスト D_jは，評価関数により次のように定義する．すなわち，距離和 $D(F)$ に対しては，

$$\begin{aligned} W_j &= w^{T'} \\ D_j &= \textstyle\sum_{v_k \in V(T'), k \neq j} (c_k W_k). \end{aligned} \tag{5.23}$$

偏差 $ECC(F)$ に対しては，

$$\begin{aligned} W_j &= 1 \\ D_j &= \max_{v_k \in V(T')} \{w_k + c(v_i, v_k)\}. \end{aligned} \tag{5.24}$$

全対距離和 $P(F)$ に対しては，

$$\begin{aligned} W_j &= w^{T'} w^{T''} \\ D_j &= \textstyle\sum_{v_k \in V(T'), k \neq j} (c_k W_k). \end{aligned} \tag{5.25}$$

ただし，頂点 v_iは v_jの親，部分木 T'は v_jを含み v_iを含まない最大の離散部分木，部分木 T''は v_iを含み v_jを含まない最大の離散部分木である．Tのすべての辺 e_jの重み W_j，コスト D_jは，明らかに $O(n)$ 時間で求めることができる．

また，変数 D_j^+ は次式で表す．

$$D_j^+ = D_j + W_j c_j. \tag{5.26}$$

部分木 $T'[j,t]$ に対する設備 F の評価関数 $H(j,t,F)$，関数 $E(g,l)$ は，次のように表す．すなわち，距離和 $D(F)$ に対しては，

$$\begin{aligned} &E(g,l) = g + l \\ &H(j,t,F) = \textstyle\sum_{v_i \in V(T'[j,t])} w_i c(v_i, F). \end{aligned} \tag{5.27}$$

偏差 $ECC(F)$ に対しては，

$$\begin{aligned} &E(g,l) = \max(g,l) \\ &H(j,t,F) = \max_{v_i \in V(T'[j,t])}\{w_i + c(v_i, F)\}. \end{aligned} \tag{5.28}$$

全対距離和 $P(F)$ に対しては，

$$\begin{aligned} &E(g,l) = g + l \\ &H(j,t,F) = \textstyle\sum_{v_i, v_j \in V(T'[j,t])}\{w_i w_j \min(c(v_i,F) + c(v_j,F), c(v_i,v_j))\}. \end{aligned} \tag{5.29}$$

すると，明らかに以下の性質 32が成り立つ．

性質 32 部分木 $T'[j,t]$，離散設備を F とする．F が頂点 v_j を含み頂点 $v_{j(t)}$ を含まないならば，次式が成り立つ．

$$H(j,t,F) = E(H(j,t-1,F), D_{j(t)}^+). \tag{5.30}$$

性質 32より，以下の性質 33が成り立つ．

性質 33 実用設備設備において，サイズが l 以下で距離和最小，偏差最小，全対距離和最小の離散木形状設備配置は，いずれも $O(n^2 l)$ 時間で求まる．

証明　Tamir の Left-Right アルゴリズム [40] をそのまま用いる．すなわち，$g[j,t,l']$ は，式 5.31の問題の最適解とする．

$$\begin{aligned} &\text{Minimize } H(j,t,F) \\ &\text{ s.t. } L(F) \le l' \\ &\qquad F \text{ is discrete tree of } T'[j,t] \text{ containing } v_1 \text{ and } v_j. \end{aligned} \tag{5.31}$$

―――

1. $j=1, t=0$ の場合
   $G[j,t]=\{(0,0)\}$
2. $j>1, t=0$ の場合（v_jは v_iの k番目の子頂点とする）
   $G[j,0]$ は，組 $(g[i,k-1,l'], l'+a_j)$ のリストとする．
   ただし，組 $(g[i,k-1,l'], l'+a_j)$ は，$G[i,k-1]$ の各
   要素 $(g[i,k-1,l'], l')$ のうち $l'+a_j \leq l$となる組である．
3. $t>1$ の場合
   次の手順で $G[j,t]$ を求める
   1) $G'[j,t-1]$ は，組 $(E(g[j,t-1,l'], D^+_{j(t)}), l')$ のリストとする．ただし，組 $(E(g[j,t-1,l'], D^+_{j(t)}), l')$ は，$G[j,t-1]$ の各要素 $(g[j,k-1,l'], l')$ のうち $E(g[j,t-1,l'], D^+_{j(t)}) \leq M$となる組である．
   2) G は，$G'[j,t-1]$ と $G[j(t), s_{j(t)}]$ を lの大きさによりマージしたリストとする．
   3) $G[j,t]$ は，G から優性要素を除いたリストとする．すなわち，G の二要素 (g^1, l^1)，(g^2, l^2) が $g^1 \leq g^2$ かつ $l^1 \leq l^2$ならば，(g^2, l^2) を除く．

―――

図 5.15: リスト $G[j,t]$ を算出するアルゴリズム

$G[j,t]$ は，最適解と設備サイズの組 $(g[j,t,l'], l')$ の順序付リストとする．ただし，$l' \leq l$，$g[j,t,l'] \leq M$で，Mは評価関数の上限値として予め与えられる値とする．各 j,t に対して図5.15のアルゴリズムを実行することで $G[j,t]$ が求まる．各 $G[j,t]$ のサイズは $O(\min(l,M))$ となるので，各 $G[j,t]$ は $O(\min(l,M))$ 時間で求まり，よって全ての $G[j,t]$ は $O(n\min(l,M))$ 時間で求まる．そして，$G[1,s_1]$ の要素のうち $g[j,t,l']$ が最小となるものが，v_1 を含む最適な離散木形状設備となる．したがって，v_1を含む Tの最適な離散木形状設備は $O(n\min(l,M))$ 時間で求まる．これを，Tの各頂点を根 v_1として繰り返すことで，最適な離散木形状設備は $O(n^2\min(l,M))$ 時間で求まる．したがって，$O(n^2 l)$ 時間で求めることができる．∎

なお，サイズ lの設備に対する評価関数の最小値 $\mathrm{WD}(l)$ が予め分っていれば，$M=\mathrm{WD}(l)$ とすることで，最適な離散木形状設備は $O(n^2\min(l,\mathrm{WD}(l)))$ 時間で求まる．また，木 Tの

各辺の長さが 1 の場合，$l \leq n-1$ なので，最適な離散木形状設備は $O(n^3)$ 時間で求まる．
次に，最適な連続および離散パス形状設備については，以下の性質 34，35が示される．

性質 34 実用設備設備において，サイズが l以下で距離和最小，偏差最小，全対距離和最小の連続パス形状設備配置は，いずれも $O(n^3)$ 時間で求まる．

証明　性質 27に同様．∎

性質 35 実用設備設備において，サイズが l以下で距離和最小，偏差最小，全対距離和最小の離散パス形状設備配置は，いずれも $O(n^3)$ 時間で求まる．

証明　性質 28に同様．∎

5.4.3　実用設備の一般性

従来の距離和を用いる設備配置，全対距離和を用いる設備配置，全対コストを用いる設備配置が，距離和を用いる実用設備配置により求まることを示す．
すなわち，以下の性質 36，37，38 が成り立つ．

性質 36 従来の距離和を用いる設備配置は，距離和を用いた実用設備配置により求めることができる．

証明　$c_i = a_i$とすればよい．∎

性質 37 全対距離和を用いる木形状やパス形状設備配置は，距離和を用いた実用設備配置により求めることができる．

証明　全対距離和を用いた根 v_1を含む設備の配置は，性質 13，17より，$c_i = a_i w^{T'}$として距離和を用いた実用設備配置を行うことで求まる．ただし，部分木 T'は，頂点 v_iを含まず v_iの親を含む最大の離散部分木である．これを，Tの各頂点を根 v_1として繰り返すことで，全対距離和を用いた最適な設備が求まる．∎

性質 38 全対コストを用いる木形状やパス形状設備配置は，距離和を用いた実用設備配置により求めることができる．

証明　全対コストを用いた根 v_1 を含む設備の配置は，性質 13，30 より，$c_i = (a_i - f(a_i))w^{T'}$ として距離和を用いた実用設備配置を行うことで求まる．ただし，部分木 T' は，頂点 v_i を含まず v_i の親を含む最大の離散部分木であり，$f(a_i)$ は長さ a_i に対する通信コスト削減関数の値である．これを，T の各頂点を根 v_1 として繰り返すことで，全対距離和を用いた最適な設備が求まる．∎

5.4.4　まとめ

本節では，各辺に対し設備の構築コスト a_i，設備の評価コスト c_i を用いた実用設備配置問題を定式化し，サイズ指定の木形状やパス形状設備の配置について議論した．その結果，文献 [40] の疑似多項式時間アルゴリズムを用いることにより，実用設備配置における最適な離散木形状設備が求まることを示した．特に，木 T の各辺の長さが 1 の場合は，このアルゴリズムの実行時間は $O(n^3)$ 時間であることを示した．また，最適な連続および離散パス形状設備が $O(n^3)$ 時間で求まることを示した．

また，従来の距離和を用いる設備配置，全対距離和や全対コストを用いる木形状やパス形状設備配置が，距離和を用いる実用設備配置により求まることを示した．特に，設備の高速化率が一定でない場合の全対コスト最小の設備も，距離和を用いた実用設備配置により求まることを明らかにした．

5.5　むすび

本章では，実際の木構造通信ネットワークシステム構築に実用的な，通信コストを考慮した設備配置について議論した．

先ず，全対距離和を提案し，全対距離和が最小となる木形状設備の配置について議論した．この結果，特に各辺の長さが 1 である木構造ネットワークに対する全対距離和最小の木形状設備が $O(n)$ 時間で配置できることを示した．

次に，より実際の通信ネットワークに近いモデルについて議論した．先ず，木構造ネットワークの各辺の通信コストを d としたとき，通信コスト低減関数 $f(d)$ を用いて設備内の通信コストを $f(d)$ に削減するモデルを提案した．そして，高速化率一定の場合，すなわち $f(d) = \alpha \cdot d$（α は定数）の場合の最適な設備が，全対距離和が最小となる設備に等しいことを示した．この結果，特に各辺の長さが 1 である木構造ネットワークに対する最適な

設備が $O(n)$ 時間で配置できることが示された．これは，超並列計算機の通信ネットワークなど各辺の長さを 1 と考えることのできる木構造通信ネットワークシステムにおいて，たとえば 10Mbps のネットワークに 100Mbps の高速通信設備を配置する場合（このとき $f(d) = d/10$）に，効率的に高速通信設備を配置できることを示している．

更に，設備サイズの制約に対しては辺の長さ a_iを，設備の評価に対しては辺のコスト c_i を用いる実用的な設備配置問題について議論した．この結果，各辺における設備の構築コストと通信コストの異なる通信ネットワークシステムに対する，幅広い計算機ネットワークシステムに適合した効率的な設備の配置することが可能となる．そして，この実用設備配置に対する，サイズ指定の木形状やパス形状設備の配置について議論した．実用設備配置における最適な離散木形上設備算出において，従来の距離和や全対距離和が最小となる離散木形上設備を算出する近似アルゴリズムを用いることはできない．ここでは，疑似多項式時間アルゴリズムにより実用設備配置における最適な離散木形状設備が求まることを示した．特に，通信ネットワークの各辺の長さが 1 の場合には，実用設備配置における最適な離散木形状設備が $O(n^3)$ 時間で求まることを示した．また，最適な連続および離散パス形状設備が $O(n^3)$ 時間で求まることを示した．更に，従来の距離和を用いた設備や，全対距離和や全対コストを用いた木形状やパス形状の設備配置を，距離和を用いた実用設備配置問題により求めることができることを示した．

第 6 章

結論

6.1　研究のまとめ

最先端科学技術分野では，コンピュータによる大規模計算が行われるようになり，膨大なコンピューティングパワーと巨大なメモリを有する超並列計算機が必要とされている．実際に，多数の高速プロセッサを相互接続網で結合したさまざまな超並列計算機が開発され，商用並列計算機として利用されている．また，ネットワーク上の多数のコンピュータを用いた超分散コンピュータ環境も，ネットワークコンピューティングとして多いに期待されている．しかし，現在の超並列計算機では各機種独自のプログラム開発が必要なことや，高性能でコストパフォーマンスの高い超分散ネットワーク構築法が明らかにされてないなどの問題がある．そこで本論文では，超並列・分散コンピュータネットワークにおける実用的並列計算モデルの構築および，コストや実用性を重視した木構造ネットワークにおける最適設備配置法について議論した．

第 2 章では，超並列計算機における並列プログラム開発に必要な各種のパラメータや物理的制約を考慮した実用的な並列計算モデルについて議論した．先ず，Culler らの提案した並列計算モデル LogP モデルに対し，実用的側面から LogP モデルの詳細な検討を行い，並列アルゴリズムを記述するには不十分な点のあることを示した．そして，実際の通信ネットワークのバッファ動作や通信集中に着目し，LogP モデルにおける通信路をキュー（Queue）の結合で表す新しい実用並列計算モデル LogPQ を提案した．LogPQ モデルは，通信路にキューを付加することで，実際の通信ネットワークのバッファ動作や通信集中の影響を反映することができる．また，通信パラメータを 1 ワード メッセージに対して構築すること

で，通信メッセージサイズの通信遅延に対する影響を考慮できることを明らかにした．更に，キューパラメータで与えられる制限による，通信路のバッファ動作の明示的なオーバーフロー制御により，実用的な通信処理の解析が可能であることを示した．次に，計算機アーキテクチャの観点から LogPQ モデルの構造に関する検討を行い，LogPQ モデルにより並列計算機上での並列アルゴリズムの詳細な検討が可能であることを明らかにした．また，LogPQ モデルと計算機ネットワークの通信路の関係，LogPQ モデルと LogP モデルの関係などを明らかにした．

第 3 章では，さまざまな並列アルゴリズムを実際の並列計算機やエミュレータで実行することにより，2 章で提案した実用並列計算モデル LogPQ の有用性や実用性に対する実験的評価を行った．この結果，変形 CG アルゴリズムの LPRAM モデル上での実験的評価により，同期処理や通信遅延が並列アルゴリズムの実行時間に大きな影響を与えることを明らかにした．また，並列多倍長 GCD アルゴリズムの LogPQ モデル上での実験的評価により，並列計算機の通信性能の各成分がどのように並列アルゴリズムの性能に影響を与えるかを示し，LogPQ モデルが効率的な並列アルゴリズム構築に対し有用であることを明らかにした．更に，並列行列乗算アルゴリズムの並列計算機 CM5 上での実験的評価により，通信にバッファを用いることにより非同期なプロセッサ間の通信遅延を隠蔽でき，並列アルゴリズムの効率を LogPQ モデルを用いて詳細に検討できることを示した．

第 4 章では，超並列・分散コンピュータネットワークの設備配置問題を検討し，木構造の通信ネットワークに対して構築コストを重視した木形状の高速通信路設備を配置する問題について議論した．先ず，平衡二分木における設備配置問題を取り上げ，木形状設備を配置することにより，低速通信路と高速通信路の混在する低コストで高性能な木構造通信ネットワークシステムを構築できることを明らかにした．そして，実用性の観点から，実際に通信ネットワークシステムを構築する問題点について検討し，同一サイズの部分設備で構成される設備の配置問題や，同一サイズの複数個の設備の配置問題が重要であることを示した．次に，低速通信路や高速通信路が混在する非均質な木構造通信ネットワークを，同一サイズの部品からなる高速通信路を用いて構築する問題について議論した．その結果，木構造ネットワークに対する距離和が最少となる等分割可能な木形状設備問題の定式化を行い，最適設備配置アルゴリズムを提案した．そして，この設備配置アルゴリズムは各頂点の次数が$\alpha \log n / \log\log n$ 以下の木構造ネットワークにおいて $O(n^{\alpha(1+\log\alpha)+3})$ 時間で実行できることを示した．更に，低速通信路や高速通信路が混在する非均質な木構造通信ネットワー

クを，同一サイズの高速通信路を複数個用いて構築する問題について議論した．その結果，木構造ネットワークに対する距離和の総和が最小となる複数個の同一サイズの木形状設備配置問題の定式化を行い，最適設備配置アルゴリズムを提案した．この配置アルゴリズムは，各頂点の次数が$\alpha \log n / \log\log n$以下の木構造ネットワークに対し，$O(n^{\alpha(1+\log \alpha)+3})$ 時間で実行できることを示した．これらの設備配置法により，コストパフォーマンスの高い通信ネットワークシステムや，通信量の多い場合に高性能な通信ネットワークシステムを構築できることを明らかにした．

第 5 章では，多種類の高速通信設備を用いた複雑な通信ネットワークシステム構築に必要な高速通信設備内の通信コストに着目し，木構造ネットワークに対する二点間の平均距離が最小となるサイズ lの木形状設備やパス形状設備の配置問題について議論した．先ず，全対距離和を提案し，全対距離和が最小となる木形状設備の配置問題を定式化し，特に各辺の長さが 1 である木構造ネットワークに対する全対距離和最小の木形状設備が $O(n)$ 時間で配置できることを示した．次に，実際の通信ネットワークを考え，木構造ネットワークの各辺の通信コストを d としたとき，通信コスト低減関数 $f(d)$ を用いて設備内の通信コストを $f(d)$ に削減する設備配置を提案した．そして，高速化率一定の場合，すなわち $f(d) = \alpha \cdot d$（αは定数）の場合の最適な設備が，全対距離和が最小となる設備に等しいことを示した．この結果，特に各辺の長さが 1 である木構造ネットワークに対する最適な設備が $O(n)$ 時間で配置できることを示した．更に，従来のサイズ指定設備配置問題の拡張として，設備サイズの制約には辺の長さ a_iを，設備の評価には辺のコスト c_iを用いる実用設備配置問題を定式化し議論した．この実用設備配置問題により，各辺における設備の構築コストと通信コストの異なる幅広い計算機ネットワークシステムに対し，設備を最適配置することができる．そして，疑似多項式時間アルゴリズムにより実用設備配置問題における最適な離散木形状設備配置が求まることを示した．特に各辺の長さが 1 である木構造ネットワークに対する最適な離散木形状設備配置が $O(n^3)$ 時間で求まることを示した．また，距離和を用いた実用設備配置問題により，従来の距離和を用いる設備や全対距離和や全対コストを用いる木形状やパス形状の設備配置を求めることができることを示した．

6.2　今後の課題

本論文で提案した実用的な並列計算モデル LogPQ は従来の並列計算モデルより詳細に並列アルゴリズムの解析や並列処理時間の評価を行うことができることを示した．また，木

構造ネットワークにおける設備配置法により効率的な超並列・分散コンピュータネットワークシステムを構築できる．しかしながら，現実の通信ネットワークシステムの複雑性に対応するためには多くの課題が残っている．ここでは，そのうちの 4 点について問題点と今後の課題を述べる．

1. 多種類の高速通信設備への対応
   本研究では高速通信設備の性能を一種類としていた．これは，たとえば 10Mbps の低速通信路に 100Mbps の高速通信路を追加配置する場合に有用である．しかしながら，実際には 100Mbps と 1Gbps の高速通信路を併用するなどの，さまざまな通信ネットワークシステムが存在する．これに対し，高速通信設備内の通信コストを考慮し，各種速度の高速通信設備を配置することにより，実際の通信ネットワークに即した実用性の高い通信ネットワークシステムの構築を行うことができる．

2. ネットワークの次数制限の緩和
   本研究では主にネットワークの次数が $O(\log n/\log\log n)$ 以下となるような大規模ネットワークを扱ってきた．次数制限を緩和することにより，より次数の大きい中小規模ネットワークの構築を行うことができる．

3. 大規模分散システムへの対応
   本研究では主にネットワークの各辺の長さを 1 に固定した．この長さを任意に拡張することにより，よりグローバルな分散通信ネットワークシステムを構築することができる．

4. 一般的な構造のネットワークへの対応
   本研究では木構造ネットワークのみを扱ってきた．一般のネットワーク構造に拡張することにより，現存する幅広い構造の通信ネットワークシステムの構築や改良を行うことができる．

上記の改良を行うことにより，現在の超並列・分散コンピュータネットワークに対してコストパフォーマンスの高い効率的な通信ネットワークシステムの構築法を確立することができる．

6.3　おわりに

本研究は，実用的な超並列計算機モデルの提案を行い，超分散コンピュータネットワーク上の設備配置問題を定式化し最適設備配置法を提案した．これらの結果は，今後益々大規模化が進む超並列計算機や超並列コンピュータネットワークなどの分野において，ネットワークを考慮に入れた並列プログラムの開発あるいは低コストで効率のよい高速通信ネットワークを構築する際に一助になると考えられる．

謝辞

本研究は，北陸先端科学技術大学院大学情報科学研究科情報システム学専攻在学中に，堀口進教授のもとで行われました．本研究を進めるにあたり，堀口進教授には終始懇切なる御指導・御鞭撻を賜わりました．心から感謝するとともに，ここに御礼申し上げます.

北陸先端科学技術大学院大学情報科学研究科 木村正行教授には，特に副テーマに関して有意義な御指導を賜わりましたことを，深く感謝致します.

北陸先端科学技術大学院大学情報科学研究科 阿部亨助教授には，終始懇切なる御指導を賜わりましたことを，心から感謝致します.

北陸先端科学技術大学院大学情報科学研究科 日比野靖教授と弘前大学理工学部 吉岡良雄教授には，審査にあたり御教示・御検討を賜わりましたことを，深謝致します.

また，日頃から有益な御助言をいただき，多面に渡って励ましていただいた，北陸先端科学技術大学院大学 情報科学研究科 山森一人博士と情報科学センター 井口寧博士に感謝致します.

最後に，本論文をまとめるに当たって大変お世話になりました，堀口・阿部研究室の諸兄に厚く御礼申し上げます.

参考文献

[1] 宮野悟, 並列アルゴリズム, 近代科学社 (1993).

[2] Helmut Alt, Torben Hagerup, Kurt Mehlhorn and Franco P.Preparata, “Deterministic simulation of idealized parallel computers on more realistic ones”, SIAM Journal on Computing, 16, pp.808-835 (1987).

[3] Clyde P.Kruskal, Larry Rudolph and Marc Snir, “A Complexity Theory of Efficient Parallel Algorithms”, Theoretical Computer Science, 71, pp.95-132 (1990).

[4] Alok Aggarwal, Ashok K.Chandra and Marc Snir, “Communication Complexity of PRAMs”, Theoretical Computer Science, 71, pp.3-28 (1990).

[5] “nCUBE2: Technical Overview”, nCUBE Corporation (1992).

[6] “Connection Machine CM-5 Technical Summary”, Thinking Machines Corporation (1992).

[7] William J.Dally, Andrew Chien, Stuart Fiske, Waldemar Horwat, John Keen, Michael Larivee, Rich Lethin, Peter Nuth and Scott Wills, “The J-Machine: A Fine-Grain Concurrent Computer”, Information Processing 89, pp.1147-1153 (1989).

[8] Thorsten von Eicken, David E.Culler, Seth Copen Goldstein and Klaus Erik Schauseret, “Active Messages: a Mechanism for Integrated Communication and Computation”, Proceeding of the 19th Annual International Symposium on Computer Architecture pp.256-266 (1992).

[9] David Culler, Richard Karp, David Pattersom, Abhijit Sahay, Klaus Erik Schauser, Eunice Santos, Ramesh Subramonian and Thorsten von Eicken, “LogP: Towards a

Realistic Model of Parallel Computation", Proceeding of the 4th ACM SIGPLAN Symposium on Principles and Parallel Programming (1993).

[10] Takeshi Horie, Youichi Koyanagi, Nobutaka Imamura, Kenichi Hayashi, Toshiyuki Shimizu and Hiroaki Ishihata, "Effect of Message Communication on Distributed-Memory Parallel Computer Performance (in Japanese)", Transactions of IPSJ, Vol.35 No.4, pp.609-618 (1994).

[11] Albert Alexandrov, Mihai F.Ionescu, Klaus E. Schauser and Chris Scheiman, "LogGP: Incorporating Long Messages into the LogP Model", Proceedings of the 7th Symposium on Parallel Algorithms and Architectures, pp.95-105 (1995).

[12] R.P.Brent and H.T.Kung, "Systolic VLSI Arrays for Linear Time GCD Computation", VLSI'83, pp.145-154, North-Holland (1983).

[13] B.Chor and O.Goldreich, "An Improved Parallel Algorithm for Integer GCD", Algorithmica, 5, pp.1-10 (1990).

[14] A.Chandra and S.Fortune, "Unbounded Fan-in Circuit and Associative Functions", J. Comput. & Syst. Sci., 30, pp.222-234 (1985).

[15] V.Kumar et al., "Introduction to Parallel Computing", Benjamin/Cummings (1994).

[16] P.Hämäläinen, "PRAM EMULATOR", University of Joensuu, Department of Computer Science, Joensuu, Finland (1993).

[17] S.Juvaste, "PM2 compiler", University of Joensuu, Department of Computer Science, Joensuu, Finland (1992).

[18] E.Brickell, "A Fast Multiplication Algorithm with Application to Two Key Cryptography", Advances in Cryptology, Proceeding of Crypto'82, pp.51-60, Plenum Press (1982).

[19] 亀山充隆　川人 祥二　樋口 龍雄, "Signed-Digit 数系に基づく双方向電流モード多値基本演算回路とその評価", 信学論 D, Vol.J71-D No.7, pp.1189-1198 (1988).

[20] 當山孝義, 堀口進, “実用並列計算モデル LogPQ の動作検討”, 情処学アーキテクチャ研報, 94-ARC-109-8, pp.57-64 (1994).

[21] Vipin Kumar, Ananth Grama, Anshul Gupta and George Karypis, “Introduction to Parallel Computing”, Benjamin/Cummings Publishing (1994).

[22] Mark S.Daskin, “Network and Discrete Location”, John Wiley & Sons, Inc. (1995).

[23] P. Jaervinen, J. Rajala and H. Sinervo, “A Branch-and-Bound Algorithm for Seeking the p-median”, Operations Research, 20, pp.173-178 (1972).

[24] Giovanni Righini, “A Double Annealing Algorithm for Discrete Location/Allocation Problems”, European Journal of Operational Research, 86, pp.452-468 (1994).

[25] Michel X. Goemans and David P. Williamson, “The Primal-Dual Method for Approximation Algorithms and Its Application to Network Design Problem”, In D.S.Hochboum, editor. Approximation Algorithms for NP-Hard Problems, PWS Publishing (1995).

[26] Bezalel Gavish and Suresh Sridhar, “Computing the 2-Median on Tree Networks in $O(n\ lg\ n)$ Time”, Networks, 26, pp.305-317 (1995) (Erratum: Networks, 27, pp.169 (1996)).

[27] Arie Tamir, “An O(pn^2) Algorithm for the p-median and Related Problems on Tree Graphs”, Operations Research Letters, 19, pp.59-64 (1996).

[28] Pierre Chardaire and Alain Sutter, “Solving the Dynamic Facility Location Problem”, Networks, 28, pp.117-124 (1996).

[29] S.O. Krumke, “On a Geberalization of the p-Center Problem”, Information Processing Letters, 56, pp.67-71 (1995).

[30] Juan A.Mesa and T.Brian Boffey, “A Review of Extensive Facility Location in Networks”, European Journal of Operational Research, 95, pp.592-603 (1996).

[31] James F. Campbell, “Hub Location and the p-Hub Median Problem”, Operations Research, 44, 6, pp.923-935 (1996).

[32] Peter J.Slater, “On Locating a Facility to Services Area in a Network”, Operations Research, 29, 3, pp.523-531 (1981).

[33] Peter J.Slater, “Locating Central Paths in a Graph”, Transportation Science, 16, 1, pp.1-18 (1982).

[34] Christine A.Morgan and Peter J.Slater, “A Linear Algorithm for a Core of a Tree”, Journal of Algorithms, 1, pp.247-258 (1980).

[35] Edward Minieka, “The optimal location on a path or tree in a tree network”, Networks, 15, pp.309-321 (1985).

[36] Arie Tamir and Timothy J. Lowe, “The Generalized P-Forest Problem on a Tree Network”, Networks, 22, pp.217-230 (1992).

[37] S. L. Hakimi and Martine Labbe, “On Locating Path- or Tree-Shaped Facilities on Networks”, Networks, 23, pp.543-555 (1993).

[38] W. J. Dally, “Performance Analysis of k-ary n-cube Interconnection Networks”, IEEE Transactions on Computers, 39, 6, pp.775-785 (1990).

[39] Shietung Peng and Win-Tsung Lo, “Efficient Algorithms for Finding a Core of a Tree with a Specified Length”, Journal of Algorithms, 20, pp.445-458 (1996).

[40] Arie Tamir, “Fully Polynomial Approximation Scheme for Locating a Tree-Shaped Facility”, Technical Report of Tel Aviv Univercity, pp.1-18 (1993).

[41] Shietung Peng and Win-Tsung Lo, “The Optimal Location of a Structured Facility in a Tree Network”, Parallel Algorithm and Applications, 2, pp.43-60 (1994).

[42] Michael R. Garey and David S. Johnson, “Computers and Intractability: A Guide to the Theory of NP-Completeness”, W.H.Freeman (1979).

本研究に関する発表論文

査読付き論文

[**1**] 當山孝義, 堀口進, “Chor-Goldreich 並列 GCD アルゴリズムの動作解析”, 電子情報通信学会論文誌, Vol.J77-D-I, No.11, pp.770-773, (1994 Nov)

[**2**] 當山孝義, 堀口進, “並列多倍長 GCD アルゴリズムとその性能評価”, 電子情報通信学会論文誌, Vol.J80-D-I, No.5, pp.450-455 (1997 May)

[**3**] 當山孝義, 堀口進, “木構造ネットワークへの等分割可能な木形状設備配置”, 電子情報通信学会論文誌, Vol.J81-D-I, No.2 (1998, Feb)

[**4**] 當山孝義, 堀口進, “複数個の木形状設備配置による高速木構造ネットワークの構築”, 電子情報通信学会論文誌, D-I (投稿中)

[**5**] 當山孝義, 堀口進, “並列アルゴリズムの動作解析のための実用並列計算機モデル LogPQ ”, 情報処理学会論文誌 (投稿中)

査読付き国際会議発表論文

[**6**] Takayoshi TOUYAMA and Susumu HORIGUCHI, “Parallel Computation Model LogPQ ”, International Symposium on High Performance Computing ’97, LNCS1336, pp.327-334 (1997 Nov)

研究会・口頭発表

[**7**] 當山孝義, 阿部亨, 堀口進, “PRAM モデルによる並列 GCD アルゴリズム”, 平成 5 年度電気関係学会北陸支部連合大会, E-22, pp.273, (1993 Sep)

[**8**] 當山孝義, 堀口進, “超並列計算機の実用並列計算モデル LogPQ と数値計算の実行評価”, 情報処理学会第 49 回全国大会, 1P-6, pp.89-90 (1994 Sep)

[**9**] 當山孝義, 堀口進, “実用並列計算モデル LogPQ の動作検討”, 情報処理学会計算機アーキテクチャ研究会, 94-ARC-109-8, pp.57-64 (1994 Dec)

[**10**] Takayoshi TOUYAMA and Susumu HORIGUCHI, “Parallel Computation Model LogPQ and Its Evaluation”, JAIST Research Report, IS-RR-97-0020A, (1997 Apr)

[**11**] 當山孝義, 堀口進, “実用並列計算機モデル LogPQ による並列アルゴリズムの動作解析”, 情報処理学会ハイパフォーマンスコンピューティング研究会, 97-HPC-68-6, pp.33-38 (1997 Oct)