分散グラフアルゴリズムとグラフパラメータ

若　　者

北　村　直　暉*

Distributed Algorithm and Graph Parameter

Key Words：Distributed System, Graph Theory, Graph Parameter, Algorithm

はじめに

インターネットの普及や規模の拡大により，複数の計算機で構成される分散システムが発展しており，来るべきIoT（Internet of Things）の時代に向けてその傾向が大きくなっている．それに伴い，複数の計算機が協調的な動作を行うことによって課題を解くアルゴリズム（分散アルゴリズム）の設計は重要な問題として認識されている．本稿では分散システムの主要なモデルの一つである分散グラフシステムについて説明をし，分散グラフシステムにおいて重要な問題の一つである最小全域木問題について説明をする．

従来の分散グラフシステムの研究では，ネットワークの構造に依存しないアルゴリズムを設計することが主要であった．そのような方法では，計算に最も時間がかかるグラフ（最悪時インスタンス）に関しても高速に動くアルゴリズムを設計する必要がある．一方で，そのような最悪計算時間になるのは一部の特別なグラフだけであり，その他の多くのインスタンスにおいては，一般グラフに対する最悪計算時間を打ち破るようなアルゴリズムが存在する場合も考えられる．本稿では分散グラフシステムにおけるグラフの構造と計算時間の関係について説明をする．

* Naoki KITAMURA
1994年5月生まれ
名古屋工業大学大学院　工学研究科
情報工学専攻　博士（2022年）
現在、大阪大学　大学院情報科学研究科
コンピュータサイエンス専攻　助教
工学（博士）
専門／分散グラフシステム，アルゴリズム
TEL：06-6879-4117
FAX：06-6879-4117
E-mail：n-kitamura@osaka-u.ac.jp

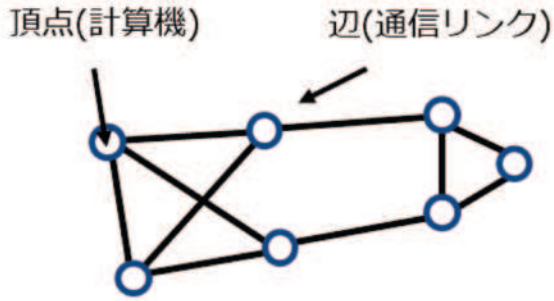

図1：ネットワークグラフの例

分散グラフシステム

分散グラフシステムとは複数の計算機と計算機間の通信リンクによって構成されるシステムである．計算機を頂点，通信リンクを辺とみなすと通信ネットワークはグラフで表現することができる（図1）．この時，ネットワークに関する何らかの問題，例えば最短経路問題や最大マッチング問題をネットワーク自身が自律的に計算するアルゴリズムを分散グラフアルゴリズムと言う．このような基礎問題をネットワーク上で解くことには多くの応用が知られている．例えば最短経路問題はネットワークにおける効率的なメッセージラウティング手法，最大マッチング問題は対戦ゲーム等の対戦相手を割当て等と関連している．

分散グラフアルゴリズムでは各頂点がメッセージ交換に要する（同期した）ラウンド数で性能が評価される．各ラウンドでは各頂点は隣接頂点にメッセージを送信・受信を行い，その結果を用いて内部計算を行う．1ラウンドで隣接頂点に通信できるメッセージ数には上限があり，一般には1ラウンドあたり送ることが可能なビット数の上限Bが分散システムのモデルによって与えられている．ネットワークの規模に比べて通信幅が小さいという仮定から，通常は$B = \log n$として設計されることが多い．また，

各頂点が内部計算を行うのにかかる時間は頂点間の通信時間に比べて極めて短いことを考慮し，内部計算の時間を無視することが多い．

分散グラフシステムの難しさ

分散グラフアルゴリズムの最も簡単な設計方法は1つの頂点にグラフの情報（トポロジー）を収集し，その頂点が逐次型のアルゴリズムを用いることで解を計算することである．このような手法を用いると理屈の上では全ての分散システム上のグラフのトポロジーに関する問題を解くことが可能であるが，グラフの情報を収集するにはグラフの辺数を m としたときに $O(m)$ ラウンド必要となり，時間がかかってしまう．そのため，分散グラフアルゴリズムの設計ではグラフ全体のトポロジーを知ることなく協調して問題を解く必要があるという点に難しさがある．現在の研究では自明な上界である $O(m)$ ラウンドよりも速いラウンド数で問題を解くアルゴリズムの設計が考えられている．

最小全域木問題と分散グラフアルゴリズム

分散アルゴリズムの一つの例として最小全域木問題について説明をする．最小全域木問題では入力として各辺に重みがつけられているグラフが与えられる．この問題の目的は全ての頂点を連結にするような辺集合でコストの和が最小となるものを見つけることである．最小全域木問題を逐次システムで効率的に解く手法の一つとしてブルーフカ法というものがある．ブルーフカ法のアルゴリズムは以下のとおりである．

1. アルゴリズムでは最小全域木の断片の集合（最小全域森）を管理して，最終的に一つの結合された最小全域木を求める．初期状態では全ての頂点は自身のみからなる最小全域木の断片である．
2. 各最小全域木の断片は，自身と他の木の断片をつなぐ辺（外向辺）の内，最小重みのものを選び，それを最小全域木の辺に加える[1]．辺の両端点の断片は一つの断片に結合される．

[1] 今回のアルゴリズムでは各辺の重さが異なる場合について説明をしている．同じ重さの辺がある場合はこの処理によって閉路ができる場合はあるが，その場合は任意の一つを加えるのをやめる．

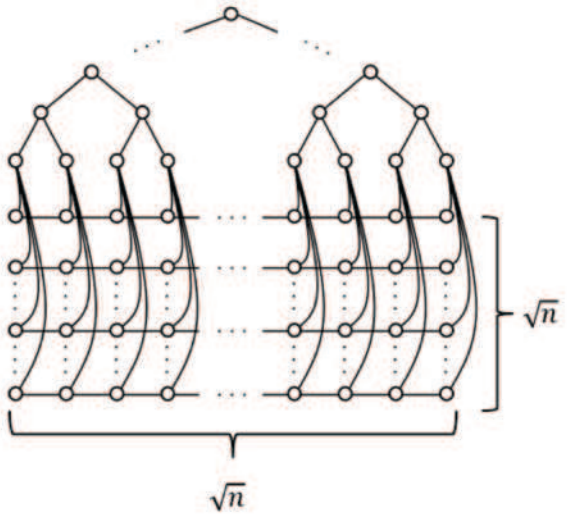

図2：分散グラフシステム上で最小全域木問題を解くのに時間がかかるグラフ

ブルーフカのアルゴリズムでは各木の断片が最小重みの外向辺を計算するのは独立して実行することが可能である．そのため，ブルーフカのアルゴリズムを分散グラフシステム上で実装するのは相性が良い．実際には各木の断片が最小重みの外向辺を見つけて結合をすると，全体の断片の数は少なくとも半分は少なくなる．従って $\log n$ 回の繰り返しを行うことによって最小全域木を構築することが可能である．このことから最小全域木問題を分散グラフシステム上で効率的に解く一つのアプローチは各木の断片が最小重みの外向辺を効率的に発見することである．Ghaffari 等は最小重みの外向辺を $O(\sqrt{n}+D)$ ラウンドで求める手法が証明されている [1]．また，最小全域木問題については分散グラフシステム上で $\Omega(\frac{\sqrt{n}+D}{\log n})$ ラウンドよりも高速に問題を解くことができないことが示されている [2]．これは図2のようなグラフで解くことができないことが示されている．図2のグラフは以下のように構成される（N は $N=o(n)$ となる変数とする）．

1. 長さが $\sqrt{N}$ のパスを $\sqrt{N}$ 個とサイズ $\log N$ の完全二分木を一つ用意する．
2. 完全二分木の各葉と $\sqrt{N}$ 個のパスの頂点を結ぶ（どの頂点を結ぶかは図を参照）．

このグラフの特徴として，全てのパスに対して，右端の頂点と左端頂点が同時にメッセージを交換するのには時間がかかることがあげられる．直観的には各パス上の辺のみを用いてメッセージを交換する

にはパスの長さが$\sqrt{N}$であるので時間がかかってしまう．そのため，効率的にメッセージを送るには完全二分木の辺を利用する必要がある．一方，分散グラフシステムでは1ラウンドに$\log n$ビットのメッセージしか送ることができないので，複数のメッセージを送るには時間がかかってしまう．

分散グラフシステムとグラフパラメータアルゴリズム

前節では最小全域木問題に対する分散グラフシステムでのアルゴリズムと下界について説明した．$\log n$のファクターを除くとブルーフカのアルゴリズムを元にした分散グラフアルゴリズムは最適な計算時間を達成していることになる．しかし，実際には最悪な計算時間になる場合はごく一部のインスタンスだけであり，その他の多くのインスタンスでは高速に解くことが可能であることが考えられる．入力に対する制約を問題のサイズ以外で定量化し，そのサイズが小さいときに動作するアルゴリズムをパラメータ化アルゴリズムと言う．逐次計算の分野では，木や平面等のグラフが持つ性質に対して特徴付けをし，概念を拡張するために様々なパラメータが提案をされており研究がされている．本稿ではグラフパラメータの一つである直径について最小全域木問題のパラメータ化アルゴリズムを説明する[3]．ブルーフカのアルゴリズムを分散システムで実装するときに時間がかかる部分は各木の断片が最小重みの外向辺を見つける部分であった．分散グラフシステムでは，一つの辺を複数のメッセージを送るために利用すると，通信幅の制約によって時間がかかってしまう．しかしながら，直観的には，グラフの直径が小さいとき，各頂点間には複数の長さが短いパスが存在するように見える．この直観を用いると以下のアルゴリズムが考えられる．

1. 初めに各木の断片内の直径について考える．直径が$n^{1/3}$より小さいグラフでは各木の断片内の辺を用いて，最小重みの外向辺を見つける．
2. 直径が$n^{1/3}$以上であるような木の断片に含まれる各頂点は，自分の隣接頂点に自身に接続する最小重みの辺の情報を伝える．
3. 最小重みの辺の情報が送られた頂点は，各隣接辺に対して，確率$\frac{1}{n^{1/3}}$の確率で自分の隣接頂点に最小重みの辺の情報を送る．ただし同じ木の断片から送られてきた最小重みの辺の情報は最も辺重みが小さいものだけを送る[2]．
4. 直径が$n^{1/3}$以上であるような木の断片に含まれる各頂点はこれまでに送られてきた情報の中で最小の辺重みを持つ辺の情報を自身とその接続辺を用いて送る．ただし，情報を送るのは自身から距離$n^{1/3}+D$未満の頂点のみである．

このアルゴリズムでは直径が小さい木の断片に関しては事前に最小重みの外向辺の情報を送っている．この処理は分散グラフシステムでは$O(n^{1/3})$ラウンドで行うことができる．一方で直径が大きい木の断片の数は高々$n^{2/3}$個しかない．そのため確率$1/n^{1/3}$で辺の情報を伝えると1つの辺に対する伝えるメッセージの数は高確率[3]で$O(n^{1/3})$となる．このようにすることによって$O\left(\left(n^{\frac{1}{3}}+D\right)\log n\right)$ラウンドのアルゴリズムを作成できる．

おわりに

本稿では分散グラフシステムの基礎的な問題である最小全域木問題と最小全域木問題に対するパラメータ化アルゴリズムについて説明をした．本稿では主に直径が3のグラフに対するアルゴリズムについて説明したが，直径のサイズが定数である場合に直径のサイズと計算時間のトレードオフについても知られている[4]．またGhaffari等によって示された最小全域木問題を解くアルゴリズムのフレームワークは他の様々な問題に対しても利用できることが知られている．具体的には分散グラフシステム上で最小全域木問題を解くのに必要な計算時間と近似最短経路問題や近似最小カット問題を解くのに必要な

[2] 2つの頂点が同じ木の断片であるかということは事前計算によって求める必要がある．詳細は省くが，分散グラフシステム上のブルーフカのアルゴリズムでは木の断片を結合するときに，同時に同じ木の断片に含まれる頂点に対して共通のラベルを割り当てている．

[3] ここで言う高確率とは確率$1-1/n^{o(1)}$以上であることを表す．分散グラフシステム上では大規模なネットワークを想定しているので良く使われる指標である．

計算時間はほとんど同じ時間必要であることが知られている．今後の課題としてはどのような問題に対して，分散グラフシステム上でパラメータ化アルゴリズムを作ることが可能であるかを明らかにすることである．

参考文献

[1] Mohsen Ghaffari and Bernhard Haeupler. Distributed algorithms for planar networks II: low-congestion shortcuts, mst, and min-cut. In Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms (SODA), pages 202–219, 2016.

[2] David Peleg and Vitaly Rubinovich. A near-tight lower bound on the time complexity of distributed minimum-weight spanning tree construction. SIAM Journal on Computing, pages 1427–1442, 2000.

[3] Naoki Kitamura, Hirotaka Kitagawa, Yota Otachi and Taisuke Izumi, Low-Congestion Shortcut and Graph Parameters, Distributed Computing, pages 349-365, 2021.

[4] Shimon Kogan and Merav Parter, In Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing (PODC), pages 203-211, 2021.