

拡張極大 P -スター分割に対する自己安定アルゴリズム

茶円 春希[†] 江口 僚太[†] 大下 福仁[‡] 井上 美智子[†]

奈良先端科学技術大学院大学[†]

福井工業大学[‡]

1 はじめに

分散システムは複数の計算主体 (プロセス) が協調動作するシステムである。大規模分散システムにおいて故障の発生は避けられず、故障に対する耐性が重要である。高い故障耐性を実現する方法の一つに自己安定アルゴリズムがある。自己安定アルゴリズムは、一時故障によりシステムがどのような状況に陥ったとしても、最終的に目的の状況に達する性質を持つ。

文献 [1] では極大 p -スター分解に対する自己安定アルゴリズムが提案されている。 p -スター分解 (p は 1 以上の整数) は、グラフを互いに素な p -スター (完全 2 部グラフ $K_{1,p}$) に分解するものであり、並列計算・負荷分散の分野で応用が可能である。極大 p -スター分解は、 p -スターに属さないプロセスで p -スターを作成できない分解であるが、 p -スターに属さないプロセスが存在する場合があります、並列計算・負荷分散などの効率が低下する場合がある。

本研究では、全てのプロセスがスターに参加するように、**拡張極大 P -スター分割**を新たに提案し、それに対する自己安定アルゴリズムを提案する。拡張極大 P -スター分割では P -スターが作成できない部分において、可能な限り大きな p' -スター ($0 \leq p' < P$) を作成することを目的としている。提案アルゴリズムは公平で

ない D デーモンにおいて動作し、 $O(n)$ ラウンドで拡張極大 P -スター分割を達成する。ここで n はプロセス数とする。

2 諸定義

2.1 分散システム

ネットワークを無向グラフ $G = (V, E)$ でモデル化する。 V はプロセス集合、 $(v, w) \in E$ はプロセス v と w が通信可能であることを示す。各プロセスの状態は、変数とその値で定義される。通信モデルは、各プロセスが隣接するプロセスの状態を直接参照することができる状態通信モデルとする。

分散アルゴリズムは条件付きアクションの集合である。プロセスは、条件を満たす場合にアクションを実行でき、アクションの実行により状態を更新する。スケジューラーとして公平でない D デーモンを仮定する。公平ではない D デーモンは各ステップでアクションを実行可能なプロセスを 1 つ以上選択し、選択されたプロセスはアクションを実行する。

提案アルゴリズムの性能評価は、非同期ラウンドを用いる。最初の状況においてアクションを実行可能な全てのプロセスが、1 回以上アクションを実行する、もしくは実行可能でなくなった最初の状況までの系列を最初のラウンドとする。以降のラウンドも同様に定義される。

2.2 拡張極大 P -スター分割

定義 1 に P -スター分割と拡張極大 P -スター分割の定義を示す。

Extended Maximal P -star Partition and Its Self-Stabilizing Solution

[†] Haruki Chaen, Ryota Eguchi, Michiko Inoue, Nara Institute of Science and Technology

[‡] Fukuhito Ooshita, Fukui University of Technology

定義 1 グラフ $G = (V, E)$ の互いに素な部分グラフの集合 SP は、次の条件を満たす場合に P -スター分割である。任意の $G' \in SP$ はある $p (\leq P)$ について p -スターであり、任意の $v \in V$ がある G' に属する。ここで $G_{SP}[h]$ を SP において大きさが h 以上のスターに属する頂点を除いたグラフとする。 G の P -スター分割 SP は、任意の $h (0 \leq h \leq P)$ について $G_{SP}[h]$ が h -スターを含まない場合、拡張極大である。

3 提案アルゴリズム

提案アルゴリズムは 0 以上 P 以下の全ての任意の整数 h について、自身がスターの親となった場合の最も大きなスターの作成を目的とする動作を行う。この動作を繰り返すことで拡張極大 P -スター分割を達成する。

図 1 を用いて、スターの作成手順について述べる。プロセス a がプロセス b を招待することを a から b への赤矢印で、プロセス a がプロセス b の招待に応じることを b から a への青矢印で示す。スターの親プロセスは青矢印を自身に向けることで、隣接プロセスに親プロセスであることを示す。自身の変数の数字はプロセスが属するスターの大きさを示す。

図 1 においてプロセス 1 は 2-スターを、プロセス 3 は 3-スターを作ることができる (図 1a を参照)。スターに属さないプロセスはできるだけ多くの隣接プロセスを招待する (図 1a は招待が完了した後の状況を示す)。招待されたプロセスは自身を招待しているプロセスのうち、最も大きなスターを作るプロセスの招待に応じる (図 1b は招待に応じた後の状況を示す)。プロセス 1 は 8 の作る 2-スターより 3 の作る 3-スターが大きいため、3 の招待に応じる。複数のプロセスから同じ大きさのスターに招待された場合、識別子が最も小さなプロセスの招待に応じる。招待した全てのプロセスが応じた場合、そのプロセスは親プロセスとなる (図 1c はプロ

セス 3 が親プロセスになった状況を示す)。スターの親または子になったプロセスは、そのスターの大きさを自身の変数に保存する。招待に応じたプロセスが親プロセスとなった場合、そのプロセスは子プロセスとなる (図 1d はプロセス 1, 8, 6 が子プロセスになった状況を示す)。

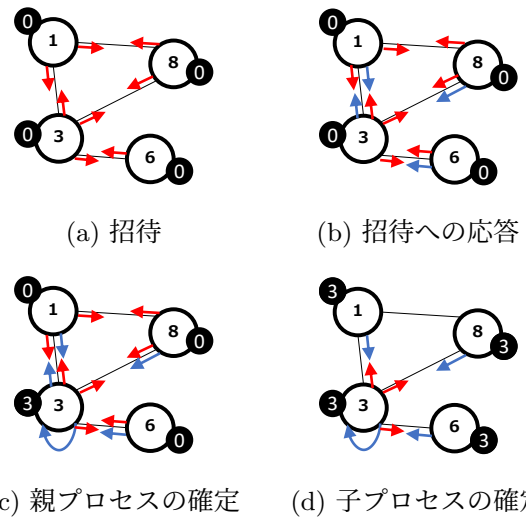


図 1: スターの作成手順

定理 1 提案アルゴリズムは拡張極大 P -スター分割に対する自己安定アルゴリズムである。公平でない D デーモンにおいて、このアルゴリズムは $O(n)$ ラウンドで拡張極大 P -スター分割を達成する。アルゴリズムの空間計算量は $O(P \log n)$ である。

4 まとめ

本稿では拡張極大 P -スター分割とそれに対する自己安定アルゴリズムを提案した。提案アルゴリズムは公平ではない D デーモンにおいて、 $O(n)$ ラウンドで拡張極大 P -スター分割を達成する。今後はラウンド数の改善やステップ数での時間複雑度の導出などを予定している。

参考文献

[1] Haddad, Mohammed et al., Polynomial Silent Self-Stabilizing p -Star Decomposition. The Computer Journal, 63, 253-266