

分散ネットワーク上のキャッシュ断片のための施設配置問題の再形式化

浮 穴 学 慈*

Re-formalization of facility location problem for cache fragments on distributed network

Satoshige Ukena

要約

近年、ブロードバンド通信環境の普及とともに、P2P型のネットワークサービスの重要性が増している。これらのネットワークサービスでは、トラヒックや耐故障性、セキュリティの観点から、ネットワーク上の適切な場所に、キャッシュなどのデータの断片を配置することが重要となっている。

本稿では、分散ネットワーク上にデータ断片を適切に配置するため、 k -メディアン問題の拡張を提案し、 Γ -色収集問題と名付ける。ここで、 Γ は集合族の添数集合であり、この集合族により各色の個数と初期配置が与えられる。この拡張は、2種類以上のデータ断片を配置する場合に該当している。グラフ彩色問題とは異なる点に注意されたい。本稿では、分散アルゴリズムで取り扱うための形式化を行った。

キーワード：分散ネットワーク、分散アルゴリズム、施設配置問題、 k -メディアン、グラフ彩色問題

(Abstract)

In recent years, P2P network services is considered more worthwhile on broadband telecommunication environment which is spread widely. For these network services, proper location of data fragments (e.g. cache data) is important for data traffic, failure resilience and security.

In this paper, I proposed an extension of k -Median problem. I named the problem as Γ -color collection problem. Here, Γ denotes an index set of a family of sets: each set gives the initial location of each color. I described some conditions for the problem to handle in the scheme of distributed algorithms.

* 提出年月日2009年11月30日、高松大学経営学部

Key words : Distributed Network, Distributed Algorithm, Facility Location Problem, κ -Median, Graph Coloring Problem

1. はじめに

近年、ブロードバンド通信環境の普及とともに、BitTorrent や Winny、Shareなどのピアツーピア型（以下P2P）ファイル共有ソフトを利用するユーザが増えている。このような、ファイル共有ソフトは、各計算機が断片化されたファイルをキャッシュデータ（あるいは、メタデータ）として蓄え、これらのデータを相互に送受信することで、1つの計算機に負荷が集中しないようにシステムが構成されている。P2Pファイル共有システムにおいては、ネットワーク上に効率良くキャッシュを配置することが重要である。

また近年、分散ネットワークの通信容量を最大限活用するための手法として、ネットワーク・コーディング（Ahlsvedeら、2000）についての研究が、盛んになっている。従来の通信では、情報源から受信者へと伝送される通信パケット1個に対し、1個の伝送経路が対応していた。それに対し、ネットワーク・コーディングでは、複数の情報源からの通信パケットを混合した上で断片化し、各断片を相異なる複数のネットワーク経路を通じて情報を伝達し、受信したノードでは複数の断片から通信パケットを復元する手法である。この手法により、経路上を流れる情報の冗長性を小さくし、通信路容量を最大限に活

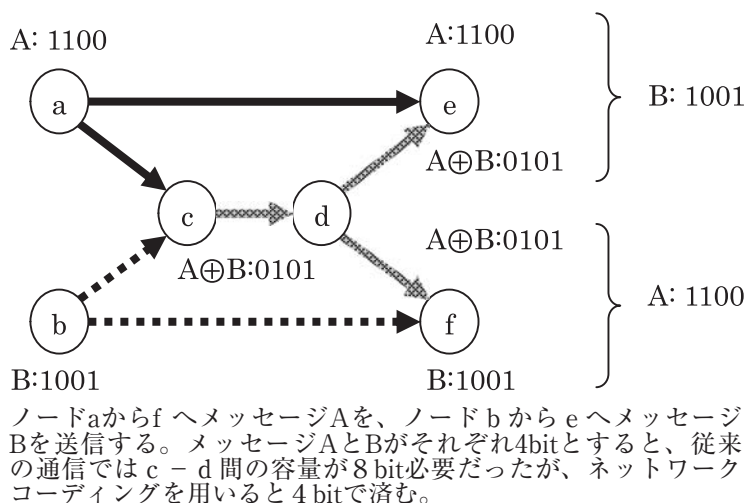


図1 ネットワーク・コーディング

用することができる（図1）。このようなネットワーク・コーディング環境においても、どのノードにキャッシュを置けばよいか重要となる。

このように、ネットワーク上の計算機にキャッシュを適切に配置することは、ネットワークを効率的に利用する上で重要である。ここで、どの計算機上のキャッシュも断片化しておらず、同じデータであるならば、ネットワーク上の適切なノードにキャッシュを配置する問題は、離散施設配置問題として研究されてきた。特に通信リンクに容量制限を置かないモデルである、容量制約なし施設配置問題は1960年代から盛んに研究されてきた。また、この問題は、NP困難なクラスに属することが Guha らによって示されている（Korteら、2009）。1997年に Shmoys らが近似保障付の近似アルゴリズムを提案して以降（Shmoysら、1997）、様々な手法に基づく近似保障付近似アルゴリズムが提案されるようになった。Korupoluらは局所探索により施設配置問題の近似保障ができることを示している（Korupolu ら、2000）。この局所探索により良い近似性能比が達成されていることは、通常の近似アルゴリズムだけでなく、後述する分散アルゴリズムの枠組みにおいてこの問題を解くための重要な要素となっている。

さて、このように分散ネットワークを効率的に利用するための仕組みが求められている。分散ネットワーク環境下で、解決のために計算機同士の協調動作を必要とする問題を分散問題と呼び、分散問題を解決するアルゴリズムを分散アルゴリズムという。分散アルゴリズムのなかでも、システムの初期化が不要であり、従って、どのような一時故障に対しても自律的に復旧する自己安定アルゴリズムが、Dijkstraをはじめとして盛んに研究されてきた（Dijkstra ら、1974）。また、分散ネットワークの特性上、非同期ネットワークにおいては、最も基礎的な分散問題である合意問題が、計算途中にただ1個の計算機が故障停止することにより解けなくなることが、Lynchらにより示されるなど（Lynchら、1983）、分散ネットワーク上の故障が問題解決に致命的な影響を及ぼすことが明らかになる中で、さまざまな故障モデルとそれらの存在下での問題の可解性について研究が行われてきた。このように、分散アルゴリズムは分散システムの故障に対応し、分散システムの可用性を保証するための基礎的なアルゴリズムとして、重要な役割を持っている。

本稿では、分散ネットワーク上にキャッシュ断片を配置する問題に対応するため、離散施設配置問題の1つの形式化であるネットワーク上のメトリック k -メディアン問題の拡張問題を提案し、これを Γ -色収集問題と名付けた。この拡張はキャッシュ断片が2種類以上のクラスに分類される場合に該当し、拡張された問題は k -メディアン問題を含んでいる。

さらに、分散アルゴリズムで取り扱うための問題の形式化を行う。逐次アルゴリズムでは、問題の形式化は入出力を定義することで行われるが、分散アルゴリズムで取り扱うために、分散ネットワークが満たすべき制約条件を定義することで問題を形式化する。

本稿は、以後、次のように構成される。2において、本稿で使用する分散ネットワークについて、その計算モデルを説明する。次に、3において、分散アルゴリズムが扱う分散問題を説明し、さらにアルゴリズムが問題を解くとはどういうことであるのかを説明する。そして、4において、複数種類の施設に対応するように k -メディアン問題を拡張し、 Γ -色収集問題として形式的に定義する。さらに、5において、分散アルゴリズムの計算モデルに合わせて、この Γ -色収集問題を形式的に定義する。最後に、まとめと今後の課題を述べる。

2. 分散ネットワーク

ここで、本稿で用いる分散ネットワークについて、その構造と構成要素、さらに、分散ネットワークの動作実行を定義する。

2.1 分散ネットワークの構造

本稿では、分散ネットワークのモデルとして、メッセージ交換型の非同期式分散システムのモデルを使用する。モデルは亀田らのモデルを用いる(亀田ら, 1994)。分散ネットワークは、計算単位として複数の自律動作する計算ノード(以後、プロセス)と、これらを相互接続する通信リンク(以後、リンク)により構成される。本稿においては、これらを単純無向グラフ $G = (V, E)$ を用いて表す。頂点集合 V はプロセスの集合であり、辺集合 E はリンクの集合である。また、辺 $(p_i, p_j) \in E$ は、プロセス p_i と p_j がリンクにより直接接続されて双方向の通信が可能であることを表し、このとき p_i と p_j は隣接するという。

2.2 プロセス(計算ノード)

本稿では、分散ネットワークに対し、次の仮定を置く。各プロセスは、同一の決定性アルゴリズムに基づいて非同期的に自律動作し、故障しない。つまり、各プロセスは同一のアルゴリズムに基づいたプログラムを持ち、計算を実行するが、プロセスによって動作の

速度は異なる。また、同一のプロセスであっても一定の動作速度で計算を行っているとは限らない。各プロセスは、個々に変数の値を管理し、個別にプログラムを実行している。これらを合わせてプロセスの状態と呼ぶ。個々のプロセスの状態は個々のプロセスが管理している。ここで、本稿では、プロセスは故障により停止することがなく、プロセス内に格納された変数の値、および、プログラム自体が改変されることもないと仮定する。

また、各プロセス p_i は、相異なる識別子 $\text{id}(p_i)$ を持つ。TCP/IPネットワークでは、IPアドレスが該当する。識別子がどのような値であるかということについては、特に定める必要はないが、本稿では簡単のため、 $O(\log n)$ bit の自然数であるとする。さらに、プロセス p_i とその識別子 $\text{id}(p_i)$ を同一視する。この識別子により、プロセス p_j が送信したメッセージをプロセス p_i が受信したとき、プロセス p_i は、そのメッセージの送信者が p_j であることを識別できる。識別子を除いて、どのプロセスも等価である。識別子の存在を仮定するのは、次の理由による。匿名ネットワーク（識別子が存在しないネットワーク）では、対称なネットワーク構造（リング）において、決定性アルゴリズムを用いてリーダー選挙をはじめとする様々な問題が非可解であることが、Rosenstiehl らにより示されている。（Rosenstiehlら、1972）また、匿名ネットワークへの識別子割当の問題（名前付け問題）とリーダー選挙問題は等価であり、本質的に、ネットワークの対称性の破れを利用している。識別子が存在しない完全に対称なネットワークでは、決定性アルゴリズムでは対称性を破ることができない。

2.3 通信リンク（プロセス間通信）

プロセス間通信は、2のプロセスが隣接する（プロセス間に通信リンクが存在する）ときのみ、直接のメッセージ伝達が可能である。各通信リンクは全二重双方向FIFOリンクであり、故障しない。つまり、メッセージは、改変、複製、損失などの影響を受けることなく、送信された順序どおり確実に伝達される。ただし、メッセージ伝達のタイミングは決まっていない。プロセス p_i, p_j 間の双方向リンクは、 p_i から p_j へのメッセージ伝達のみ出来る単方向リンクと、その逆方向である p_j から p_i への単方向リンクの対によって構成される。各単方向リンク L は、プロセス p_i の出力ポート OUT_L 、および、プロセス p_j の入力ポート IN_L と接続され、入力ポート IN_L は入力待行列を持つ（図2）。この入力待行列に長さの上限は設けなため、送信されたメッセージは、受信側プロセスが無限にしばしば（infinitely often）動作をすることで、いずれ受信されることになる。各双方向リ

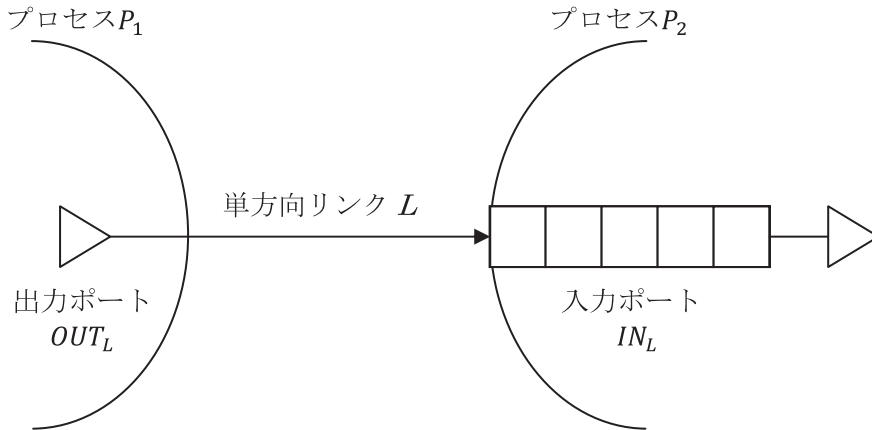


図2 単方向通信リンク

リンクが含む伝達中のメッセージとその順序、そして入力待行列の状態を合わせてリンクの状態と呼ぶ。リンクの状態は、個々のリンクが管理している。

各プロセス P_j は、メッセージ送信命令 *Send*、および、受信命令 *Receive* を持つ。プロセス P_i から P_j への単方向リンク L があるとき、メッセージ m の送受信は次のように行われる。まず、プロセス P_i は、局所変数 M_i 内に格納されたメッセージ m を、 OUT_L を通じて P_j に対して送信するため、送信命令 $Send(M_i, OUT_L)$ を実行する。すると、通信遅延時間の後に、単方向リンク L は、メッセージ m をプロセス P_j の入力ポート IN_L の入力待行列の末尾に追加する（図3 - a）。次に、 IN_L の入力待行列の先頭にメッセージ m が存在するとき、プロセス P_j が受信命令 $Receive(M_j, IN_L)$ を実行すると、命令実行遅延時間の後にメッセージ m はプロセス P_j の局所変数 M_j に読み込まれ、入力待行列からは取り除かれる（図3 - b）。

本稿では送受信に関して次の仮定をおく。送信命令 *Send* はアトミックであり、ノンブロッキング送信とする。つまり、受信側プロセスがメッセージを受信するまで次の処理に移ることなく待ち続けるブロッキング送信とは異なり、受信側プロセスによってメッセージが受信されたことを、送信側プロセスは知ることができない。また、通信遅延時間 δ は有限であるが上界を仮定しないため、いつまでにメッセージが受信されるのかを知ることができない。また、受信命令 *Receive* は、ブロッキング受信であり、入力待行列が空でなければ、1個のメッセージの受信動作が完了した後に、次の処理に移る。入力待行列が空のときは述語 *empty* を返す。この述語 *empty* によって、ノンブロッキング受信と能力的に等価であることが保証される。ここで、受信命令 *Receive* を実行してから命令実行遅

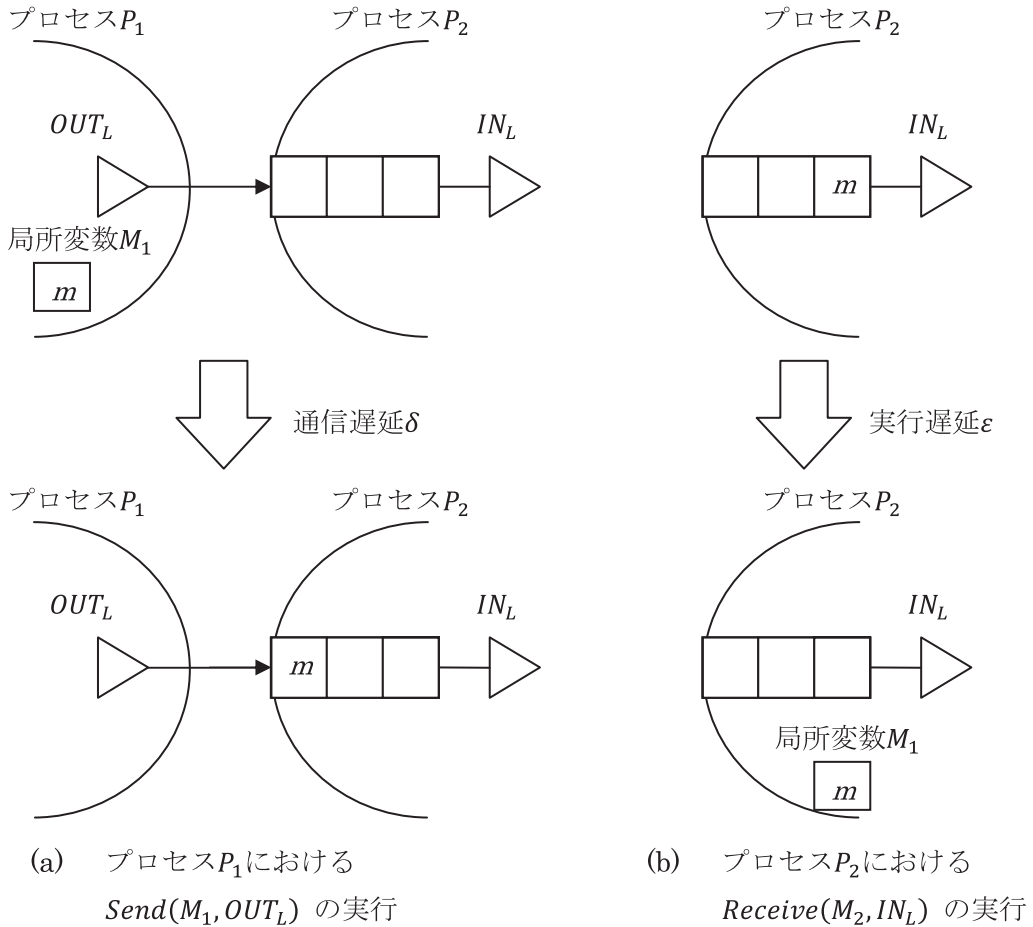


図3 メッセージの送信と受信

延時間 ε の後に受信動作が完了することと、受信命令がアトミックに動作し、次の処理を実行するまでに待ち時間 ε が経過することが等価であるため、本稿では受信命令 $Receive$ はアトミックであるとする。

2.4 分散システムの実行

さて、ここで改めて本稿の分散ネットワークが完全非同期システムであることを述べておく。つまり、本稿の分散ネットワークは、次の3条件を満たしている。

1. プロセスの実行速度に対して仮定を設けない。ただし、プロセスの実行はフェアネスを満たす。つまり、プロセスが命令を実行しようとするれば、有

- 限時間内に命令は実行されるが、その時間の上限は設けない。
2. 通信遅延に対して仮定を設けない。ただし、通信遅延時間は有限であるが、上限は設けない。
 3. プロセス間の局所時計の差に対して仮定を設けない。

留意されたいのは、プロセスは同一の決定性アルゴリズムに基づいて動作し、メッセージもリンクを通じて FIFO 順序で確実に伝達されるが、プロセスの動作タイミングや、メッセージ伝達のタイミングが一定とは限らないため、システム全体は非決定性的のように振る舞う場合があることである。システムに含まれる全プロセスの状態と全リンクの状態を合わせたものを**システムの状況**（以下、**状況**）と呼ぶが、ある決められた1つのシステムの状況から実行が開始されても、システム全体としては同じ実行動作になるとは限らない。システム全体の実行動作を略して**実行**と呼ぶ。実行は、システムの状況と送受信イベントの交互列として定義される。システムの動作として、状況 c_i において送受信イベント e_{i+1} が発生し、状況 c_{i+1} に到達することを、 $c_i e_{i+1} c_{i+1}$ と表す。システムの実行 σ は、 $\sigma = c_0 e_1 c_1 e_2 c_2 \dots$ のように定義され、状況 c_0 をシステムの**初期状況**と呼ぶ。ここで、システムの実行に関して、フェアネスの仮定を置く。実行が無限系列であるならば、各プロセスに属する送受信イベントが無限にしばしば含まれているというものである。この仮定によって、各プロセスは、いずれ計算を進行させることになる。

3. 分散アルゴリズム

本節では、分散問題が一般的にどのように定義されるのかを述べ、分散アルゴリズムが問題を解くとはどういうことかを説明する。そして、分散アルゴリズムによってシステムの実行がどのような条件を満たさなければならないのかを説明する。

まず、逐次アルゴリズムで取り扱う通常の問題は、入力と出力を用いて記述される。まず、入力が満たしている制約条件が記述され、その制約条件を満たす任意の入力に対する解（計算結果）として出力が満たすべき条件を記述する。逐次アルゴリズムにおいては、解が出力された後、アルゴリズムは停止する。これに対し、分散問題はしばしば次のように定義される。まず、入力に代わって、システムの初期状況が満たす制約条件を定義する。そして、制約条件を満たす任意の初期状況に対するシステムの計算実行の結果、シス

テムの状況が満たすべき条件を定義することで出力の定義に代える。出力として定義されたシステムの状況を**解状況**と呼ぶ。

ここで、分散問題の解に着目すると、分散問題を静的・動的の2つのクラスに分類することができる。まず、**静的分散問題**としては、全プロセスが合意形成を行い、各プロセスの持つ値を一致させる問題である合意問題、ネットワーク上に論理的な木構造を構築する全域木問題などが例として挙げられる。これらの問題は、解を得て以降はシステム状況が変化しないため、静的分散問題と呼ばれる。解を得た後はユーザ側に出力を行った後に停止してもよい。それに対し、**動的分散問題**の例として、排他的に利用される共有資源に対するアクセス制御を行う分散相互排除問題を挙げる。分散相互排除問題では、共有資源を利用したいプロセスが次々と現れ、プロセス間の情報交換の結果、排他性を満たすように共有資源に対するアクセス制御が行われることになる。このように、動的分散問題では、計算結果は刻々と変化する。動的分散問題の解はシステムの実行であり、出力の代わりにシステムの実行に対して満たすべき条件を定義することになる。

さて、本稿で取り扱う問題である施設配置問題は、分散ネットワークにおいては静的分散問題として分類される。静的分散問題を解く分散アルゴリズムは、与えられた制約条件下の任意の初期状況に対し、かつ、任意のシステムの動作に対して、解状況に到達することが保証されなければならない。言い換えると、静的分散問題を解く分散アルゴリズムの動作によるシステムの任意の実行は、そのなかに含まれるすべての状況が解状況であるような接尾部 (suffix) を必ず持つことになる。

4. Γ -色収集問題 (複数種類の施設に対応する k -メディアン問題の拡張)

本稿では、分散ネットワーク上にどのようにキャッシュを配置すればよいのかを考え、分散アルゴリズムのために施設配置問題を拡張する。施設配置問題の文脈では、施設利用コストと施設開設コストを考慮するが、本稿では、キャッシュを利用したい計算ノードが、どれだけ迅速にキャッシュ断片を集めてデータ全体を再構成できるのかが重要であると考え。従って、施設開設コストと比較して施設利用コストが重要であると考え。そこで、簡単のため、施設開設コストを0であるとみなし、施設利用コストとしてネットワーク上の距離を考える。このとき、施設配置問題はメトリック k -メディアン問題と呼ばれる。まず、 Γ -色収集問題への拡張のための準備として、 k -メディアン問題を記述する。

施設利用コスト

2つのプロセス p_i と p_j の間のネットワーク上の距離を $\text{dist}(p_i, p_j)$ と表す。キャッシュが配置されるプロセスの集合を $V_\gamma (V_\gamma \subseteq V)$ と表すと、プロセス p_i において最も近い距離に置かれたキャッシュを利用するために必要なコストは次のように定義される。

$$\text{cost}(p_i, V_\gamma) := \min_{p_j \in V_\gamma} \text{dist}(p_i, p_j)$$

これを用いて、通常の k -メディアン問題は次のように形式化される。

k -メディアン問題

次式で定義される施設利用コストの総和 $\text{median}_G(\gamma)$ が最小となるような、 $V_\gamma \subseteq V$ (ただし $|V_\gamma| = k$ である) を求める。

$$\text{median}_G(\gamma) := \sum_{p_i \in V} \text{cost}(p_i, V_\gamma)$$

本稿では、キャッシュ断片が複数種類のクラスに分類される場合に、ネットワーク上にもどのように配置すればよいのかを考える。各プロセス p_i において、全種類 (色) のデータを集めるというように考えると、これは、 k -メディアン問題を拡張することで定義できる。本稿では、これを Γ -色収集問題と名付けた。

Γ -色収集問題

次式で定義される施設利用コストの総和 $\text{newmedian}_G(\Gamma)$ が最小となるように、すべての $\gamma \in \Gamma$ に対して V_γ を求める。ただし、 $\gamma \neq \gamma'$ のとき $V_\gamma \cap V_{\gamma'} = \emptyset$ 、かつ、 $\cup_{\gamma \in \Gamma} V_\gamma \subseteq V$ でなければならない。

$$\text{newmedian}_G(\Gamma) := \sum_{\gamma \in \Gamma} \sum_{p_i \in V} \text{cost}(p_i, V_\gamma)$$

ここで、 $|\Gamma| = 1$ のとき、 Γ -色収集問題は k -メディアン問題と一致することから、 Γ -色収集問題は、 k -メディアン問題の自然な拡張となっている。また、色の配置を考える点でグラ

フ彩色問題との類似性があるように錯覚するかもしれないが、隣接頂点に同じ色が彩色されても構わない点が、グラフ彩色問題とは異なっている。

5. 分散アルゴリズムのための Γ -色収集問題の形式化

前節において定義した Γ -色収集問題を分散アルゴリズムの枠組みで解くためには、問題の入力、および、出力を分散アルゴリズムの枠組みで形式化する必要がある。分散アルゴリズムにおいては、問題の入力は初期状況として定義し、問題の出力は解状況として定義する。これを分散 Γ -色収集問題と呼ぶことにする。

インスタンス

各プロセス p_i は、次の内部変数を持つ。他の変数は計算に必要な変数でありアルゴリズムによって与えられる。

$$\text{color}_i := \gamma \quad (\text{ただし、} \gamma \in \Gamma)$$

初期状況 (入力)

問題の入力は、各色の初期状況における配置により与えられる。すなわち、初期状況 c_0 における color_i の値を color_i^0 とすると、各色 γ の初期配置は、次のように記述できる。

$V_\gamma^0 := \{p_i \mid \text{color}_i^0 = \gamma\}$ (ただし、 $\gamma \neq \gamma'$ のとき $V_\gamma^0 \cap V_{\gamma'}^0 = \emptyset$ 、かつ、 $\Pi_{\gamma \in \Gamma} V_\gamma^0 \subseteq V$)
この V_γ^0 により、各色 γ の初期配置とその個数が与えられる、これを問題の入力とする。

解状況 (出力)

実行 σ に次を満たす接尾部(suffix) σ_s が存在し、 σ_s に含まれるすべての状況 c_i において、評価関数 $\text{newmedian}_G(\Gamma)$ の値が最小値 m を持つ。この状況を解状況と呼ぶ。ここで、各色の個数は初期状況における色の個数と一致し、色の配置は実行の接尾部 σ_s において変化しない。つまり、状況 c_i における各色の配置を V_γ^i とすると、定義されるすべての i 、および、各 $\gamma \in \Gamma$ について $|V_\gamma^i| = |V_\gamma^0|$ であり、かつ、 $V_\gamma^i = V_\gamma^{i+1}$ 、 $\gamma \neq \gamma'$ のとき $V_\gamma^i \cap V_{\gamma'}^i = \emptyset$ でなければならない。

以上のように、分散アルゴリズムで取り扱うための分散問題として分散 Γ -色収集問題を形式化した。分散アルゴリズムでは、直接にメッセージを交換し得られる情報は、隣接プロセスのものである。前述のように Korupolu らによって、局所探索で施設配置問題の近似保証ができることが示されていることから (Korupolu ら, 2000)、直接得られる情報が局所的なものである分散アルゴリズムの枠組みであっても、近似保証ができる可能性があることが予想できる。また、コンバージキャスト・アルゴリズムなどを用いてネットワーク全体の情報を収集し、1つのプロセスのみが計算を行うことで、逐次アルゴリズムへの変換が容易に可能であることから、可解であること、さらに、仮に最適解を得ることが困難であっても、近似解が得られる可能性が高いことが予想される。

ここで、分散システムの可用性の観点から、計算の途中であってもサービス提供が継続され、キャッシュ断片が取得可能であるが望ましい。そのために、計算の途中であっても、色の個数が初期状況と一致することが望ましい。しかしながら、メッセージ交換によって色データを配置しなおす過程において、各プロセスが所有する色がどの色なのかを一般的な形で一意に定義することは難しい。ただし、問題を解くアルゴリズムが、コミットメントアルゴリズムなど、同期を取る仕組みを備えているのであれば、アルゴリズムに応じて無矛盾な実行断面 (仮想的なスナップショット) を定義することが可能であろうと予想できる。

本節では、このように分散 Γ -色収集問題を形式化した。これを解く分散アルゴリズムを設計し、分散ネットワーク上で実装することで、高可用性を有する効率的なキャッシュデータ断片の配置が可能になる。

6. まとめ

本稿では、分散ネットワーク上にキャッシュ断片を配置し、効率良く元データを再構成することを目的とし、施設配置問題の一つの形式的記述である k -メディアン問題を拡張し、これを Γ -色収集問題と名付けた。さらに、分散アルゴリズムで取り扱うために、この問題の初期状況、解状況を定義することで、分散 Γ -色収集問題として形式化した。さらに、解状況に至る過程においてもサービスが提供可能であるために、計算中に分散システムが満たすべき条件について示唆を与えた。分散 Γ -色収集問題を解く分散アルゴリズムを設計し、分散ネットワーク上で実装することで、効率的なキャッシュ配置のシステム

を高可用性を持って運用することができる。分散 Γ -色収集問題を解くアルゴリズムを提案することが必要であるが、問題の困難性を考慮すると近似解を得るにとどまる可能性が高い。また、問題を解くための時間計算量であるメッセージ交換回数、および、空間計算量であるメッセージサイズを見積もることが必要であり、今後の課題として挙げられる。

参考文献

- 亀田恒彦、山下雅史 (1994) 分散アルゴリズム、近代科学社、5-13
- Ahlsweede, R., Cai, N., Li, S.-Y.R. and Yeung, R.W. (2000) Network information flow. IEEE Trans. on Information Theory, vol. 46, 1204-1216
- Diestel, R., 根上生也ほか訳 (2000) グラフ理論 (訳書) (Graph Theory), シュプリンガー・ジャパン
- Dijkstra, E. (1974) Self-stabilizing systems in spite of distributed control. CACM, vol.17, no.11, 643-644
- Fischer, M.J., Lynch, N.A. and Paterson, M.S. (1983) Impossibility of distributed consensus with on faulty process. Proc.2nd. ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, 1-7
- Korte, B., Vygen., J., 浅野孝夫ほか訳 (2009) 組合せ最適化 第2版 理論とアルゴリズム (訳書) (Combinatorial Optimization 2nd.ed. Theory and Algorithms), シュプリンガー・ジャパン
- Korupolu, M., Plaxton, C. and Rajaraman, R. (2000) Analysis of a local search heuristic for facility location problems. Journal of Algorithms 37, 146-188
- Lynch, N.A. (1997) Distributed Algorithms. Morgan Kaufmann
- Rosenstiehl, P., Fiksel, J.R. and Holliger A. (1972) Intelligent graphs in Graph Theory and Computing, Academic Press, 219-265
- Shmoys, D.B., Tardos, E. and Aardal, K. (1997) Approximation algorithms for facility location problems. Procs. of 29th Annual ACM Symposium on Theory of Computing, 265-274
- Zhang, P. (2007) A new approximation algorithm for the k-facility location problem. Theoretical Computer Science 384, 126-135

研 究 紀 要
第52・53合併号

平成22年 2月25日 印刷

平成22年 2月28日 発行

編集発行 高 松 大 学
高 松 短 期 大 学
〒761-0194 高松市春日町960番地
TEL (087) 841-3255
FAX (087) 841-3064

印 刷 株式会社 美巧社
高松市多賀町1-8-10
TEL (087) 833-5811