リンクの追加・削除が連続的に生じる動的ネットワークにおけるGrundyノード彩色アルゴリズム

Grundy node coloring on the dynamic network where link addition/deletion occurs continuously

井本　宗一郎 *　　角川　裕次 *　　増澤　利光 *
Soichiro Imoto　　Hirotsugu Kakugawa　　Toshimitu Masuzawa

Abstract

モバイル・アドホック・ネットワークなどの動的ネットワークが普及しつつある．それにともない動的ネットワーク上で動作する分散システムの研究が注目を集めている. 本報告ではリンクの追加・削除が連続的に生じることによりトポロジーが変化する動的ネットワークにおいて,Grundy ノード彩色を実現する分散アルゴリズムを提案するこの分散アルゴリズムは, 初期状況から高々 n ステップ (n はノード数) で隣接ノードの色が異なるように, すべてのノードを彩色する．さらに，トポロジ変化が生じても，各ラウンドの終了時には，彩色された隣接ノードが必ず異なる色を持つことを保証する．さらに，トポロジ変化が $4\Delta - 1$ ラウンドの間生じなければ, Grundy 彩色を実現する．ここで Δ はノードの最大次数を表す.Grundy 彩色は色数の局所最小性を満たすノード彩色であり，少ない色数でノード彩色を実現できる．

1　はじめに

1.1　本研究の背景と成果

動的ネットワークとは, ノードの参加・離脱, リンクの追加･削除などによってネットワークの構造が変化するネットワークのことである. 無線接続されたスマートフォンやセンサなどのモバイル端末を含むモバイル・アドホック・ネットワーク (MANET) などの普及が進み，現在のネットワークはその多くを動的ネットワークとみなすことができる．

分散システムとは，ネットワークを通じて多数の計算機（以下，ノード）が互いに通信することにより協調動作するシステムのことである．またその分散システム上で動作するアルゴリズムを**分散アルゴリズム**という．これまでに, 静的ネットワークを対象とした分散アルゴリズムの研究は，さまざまな問題に対して数多く行われている（文献 [1],[2]）．それらのほとんどは，分散アルゴリズム実行中にトポロジ変化が生じうる動的ネットワーク上では正しく動作しない．そのため, 動的ネットワーク上で動作する分散アルゴリズムに対する必要性が近年, 高まっている．動的ネットワークにおいては，分散アルゴリズム実行中に生じるトポロジ変化に対応する（どのようなトポロジ変化が生じても問題の解を得る）必要があるため，分散アルゴリズムには新たな手法が要求される．また，分散アルゴリズムが対象とする問題についても，ネットワークのトポロジ変化を考慮した，新たな問題が提示されている．

*大阪大学大学院情報科学研究科,Graduate School of Information Science and Technology, Osaka University

これまでに，動的ネットワークのモデルもいくつか提案されている（文献 [3],[4],[5]）．最も代表的で一般的なモデルは TVG (Time-Varying Graph) と呼ばれるものである．このモデルでは，各時刻におけるトポロジを指定することで動的ネットワークを定義している．また，離散的な時刻を導入し，TGV をグラフの系列として表すことも多い．本稿では，この離散的な時刻を導入した，同期式動的ネットワークを対象とする．

ノード彩色とは, 隣接するノードが同じ色にならないように全ノードを彩色することである. ノード彩色はスケジューリングや局所排他的制御, 無線ネットワークにおけるノードへの周波数割り当てなどで応用される. 最小の色数でノード彩色を行う問題は NP-困難であることが知られており，なるべく少ない色でノード彩色を行うために，局所探索法などの手法が利用されている．

Grundy ノード彩色とは，なるべく少ない色でノード彩色を実現するための一つの方法である．Grundy ノード彩色では，色に全順序関係があることを想定し，各ノードにおいて, そのノードよりも小さいすべての色がそのノードの隣接ノードに割り当てられていることを満たすノード彩色である．つまり，Grundy ノード彩色は，色数の局所最小性を満たすノード彩色であり，少ない色数のノード彩色を期待できる．

既存研究として文献 [6] で Hedetniemi らは最大次数 Δ の静的なネットワークに対して Grundy 彩色を高々 n ステップで色数 $\Delta + 1$ で彩色を行う自己安定アルゴリズムを提案している. 自己安定アルゴリズムとは，どのような初期状況から実行を開始しても，いずれ問題の解を得て安定するという特長を持つ分散アルゴリズムである．自己安定アルゴリズムは，動的ネットワークにおいてどのようなトポロジ変化が生じても，十分に長い間, 新たなトポロジ変化が生じなければ，解を得て安定する．この特長から，自己安定アルゴリズムは動的ネットワークに適する分散アルゴリズムとして期待されている．しかし，本稿が対象とするような，トポロジ変化が連続して発生する動的ネットワークでは，自己安定アルゴリズムがいずれ解を求めて安定することは保証できない．また，十分に長い間, トポロジ変化が生じない場合は，自己安定アルゴリズムがいずれ解を求めて安定することは保証できるが，安定するまでの振る舞いについては何も保証できない．

本稿では, リンクの追加・削除が連続的に生じる動的ネットワークにおける Grundy ノード彩色に対して，以下の特長をもつ分散アルゴリズムを提案する．ただし，同期式ネットワークを仮定し，トポロジ変化としては高々 1 つのリンクの追加または削除だけが同時に生じるものと仮定している．

(1) 初期状況から高々n ステップですべてのノードが彩色される．
(2) トポロジ変化にかかわらず各ラウンド終了時には，彩色されたノードが同じ色のノードと隣接することはない．
(3) 十分に長い間，トポロジ変化が生じなければ，Grundy ノード彩色が達成される．
特長 (2) は，トポロジ変化が生じても，隣接ノードで色の衝突がないという安全な状況にすばやく復帰できることを意味している．また，特長 (3) は，安全な状況を保ちながら，徐々に色数の削減を行うことを意味している．

2 諸定義

2.1 動的ネットワーク

本稿では，通信リンク（以下，単にリンクとよぶ）で相互接続されたノード（計算機）からなるネットワークを考える．リンクは相異なるノードを接続し，ノード u, v がリンクで接続されているとき，ノード u, v は双方向全二重通信が可能である．V をノード集合，E をリンク集合とし，ネットワークを $G=(V,E)$，ノード数を $n=|V|$，リンク数を $m=|E|$ とする．ノード u, v を接続するリンクを (u, v)（または (v,u)）と表し，リンク (u, v) が存在するとき，ノード u と v は隣接しているという．またノード v の隣接ノードの集合を $N(v)$ とする．ネットワーク $G=(V,E)$ はノードを頂点，リンクを辺とする単純無向グラフ（自己ループも多重辺もないグラフ）と見なせるので，グラフに関する用語をネットワークに対しても用いる．本稿では連結ネットワークのみを扱う．

本稿では，ラウンドとよぶ時間間隔で同期している同期式ネットワークを考える．また，リンクの追加・削除が生じる動的ネットワークを扱う．リンクの追加・削除はラウンド間で発生し，ラウンド内では発生しないものとする．従って，ラウンド $t(t\geq 0)$ のネットワークを $G_t=(V_t,E_t)$ と表し，動的ネットワークをネットワーク系列 $G_0,G_1,G_2,\ldots$ として表す．ただし，2 つのラウンド間に発生するのは，高々 1 つのリンクの追加または削除のみと仮定する．つまり，各 $t(t\geq 0)$ について

$$|(E_{t+1}-E_t)\cup(E_t-E_{t+1})|\leq 1 \qquad (1)$$

が成り立つ．またノードの参加と離脱は考えないため，各 $t(t\geq 0)$ について

$$V_t=V \qquad (2)$$

とする．

2.2 計算モデル

ノードはメッセージを送受信できる状態機械としてモデル化する．各ノード u は，全順序集合から選ばれた大域的に一意な識別子 ID_u を持つ．各ノード u は各ラウンド $t(t\geq 0)$ で以下の動作を順に行う．
(1) ネットワーク G_t での隣接ノードへのメッセージ送信．
複数の隣接ノードにメッセージを送信できる．
(2) ネットワーク G_t での隣接ノードからのメッセージ受信．
ラウンド t に隣接ノードから u に送信されたメッセージをすべて受信する．
(3) 受信したメッセージに基づき，現状態を次状態に更新．

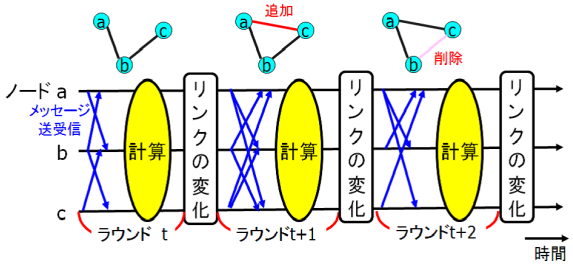

Figure 1: ラウンド動作

Figure1 では，3 つのノード a,b,c で構成されるネットワークにおけるラウンド $t,t+1,t+2$ の動作例を示す．ネットワークトポロジの変化はこの図のように，ラウンドの間で生じる．この図では例として，ラウンド t と $t+1$ の間でリンク (a,c) が追加され，ラウンド $t+1$ と $t+2$ の間でリンク (b,c) が削除されている．もちろんリンクの変化ではリンクの追加も削除も生じないこともありえる．

2.3 ノード彩色

以下にノード彩色問題を定義する．各ノード v は，$color(v)$ という色を格納する出力変数を所持している．$color(v)$ は $\{\perp,1,2,\ldots,2\Delta\}$ から選んだ一つの値をノード v の色として格納する．ここで $\perp$ は未彩色であることを示すために導入する特別な記号である．ラウンド t においてノード v に与えられた色を $color_t(v)\in\{\perp,1,2,...,2\Delta\}$ と表現する．ここで，n はそのネットワークのノード数とする．ノード彩色とは隣接ノードと異なるように各ノードに $\perp$ 以外の色を割り当てることである．つまりノード彩色されたネットワークは以下の条件を満たす．

$$\forall v, color(v)\in 1,2,\ldots,2\Delta\wedge(\nexists u\in N(v), color(u)=color(v)) \quad (3)$$

またノードの初期状態として任意のノード v に対して $color(v)=\perp$ が与えられているとする．

未彩色なノードを含むネットワークでの彩色が以下の条件を満たすとき，彩色が無矛盾であるという．

$$\forall v,\forall t, color_t(v)\neq\perp\Rightarrow\nexists u\in N(v)(color_t(u)=color_t(v)) \quad (4)$$

2.4 Grundy ノード彩色

ノード彩色されたネットワーク（未彩色のノードを含まない）において，以下が成り立つとき Grundy 彩色されているという．

$$\forall v, color(v)=k(\neq\perp)\Rightarrow(\forall k'<k,\exists u\in N(v), color(u)=k') \quad (5)$$

つまり，Grundy 彩色されているならば，ノード v は隣接ノードと同じとならないような最小の色で彩色されている．また，ある色 $h(\neq\perp)$ について，h 以下の色で彩色されたすべてのノードで上の条件が成立するとき，色 h 以下について Grundy 彩色されているという．

3 提案手法

Grundy ノード彩色問題に対して，提案アルゴリズムを以下に示す．

3.1 アルゴリズムの概要

形状変化が生じない静的ネットワークであれば，未彩色ノードを順に，隣接ノードに彩色されていない最小の色で彩色すればGrundyノード彩色を実現できる．しかし，動的ネットワークでは,(Grundy）彩色された状況でも，リンクの追加・削除によって以下の問題が生じる．
(a) 同じ色で彩色されたノード間にリンクが追加されると彩色に矛盾が生じる．
(b) リンクが削除されると，彩色の矛盾は生じないがGrundy彩色ではなくなる．

そこで本稿では，次の3つの条件を満たす分散アルゴリズムを提案する．
(1) ラウンド終了時には，彩色は常に無矛盾である．つまり，ラウンド終了時には，$\perp$ 以外で彩色されたノードは同じ色の隣接ノードを持たない．
(2) すべてのノードはいずれ彩色される．つまり，各ノード v の出力変数は，ある時点以降 $\perp$ 以外の色を格納する．ただし，ネットワークの形状変化が継続的に発生するとノード彩色が変化し続けることもある．
(3) リンクの追加・削除が十分に長い間生じなければ，Grundy彩色を実現し彩色が変化しない．

提案アルゴリズムでは，各ラウンドにおいて，すべての隣接ノード間で (ID, 現在の色，次の候補色) の3項組を交換する．以降では，現在の色を現色，次の候補色を次色とよぶ．ここで，次色は現色より小さい色で，隣接ノードに使われていない最小の色とする．ただし，そのような色が存在しない場合は，次色を $\perp$ とする．そして，すべての隣接ノードから受信したこの3項組に基づいて，そのラウンドで色を変更するかどうか，また，変更する場合にはどの色に変更するかを決定する．その詳細は次節以降で説明する．

3.2 各ノードの定数，変数

各ノード v は，以下の定数，変数を持つ．

- ID_v: 自身の ID を格納する定数.
- $color_v$: 自身の現色を格納する変数．初期値は $\perp$.
- $nextc_v$: 自身の次色を格納する変数．初期値は $\perp$.
- MSG_v: そのラウンドで受信したメッセージ（上記の3項組）の集合.

3.3 各ノードの動作

ここでは, 各ラウンドにおける各ノードの動作を示す. すべてのノードは同じアルゴリズムを実行する.

1. 3項組（$ID_v, color_v, nextc_v$）をすべての隣接ノードに送信する.
2. すべての隣接ノード u から3項組 $(ID_u, color_u, nextc_u)$ を受信し，変数 MSG_v に格納する.
3. $color_v = \perp$ ならば，初期彩色法（後述）に従って $color_v$ を更新する．6. へ.
4. 変数 MSG_v に保持されたメッセージ集合に $color_v = color_u$ となる隣接ノード u が存在する場合，色衝突解消法（後述）に従って $color_v$ を変更する．6. へ.
5. $col = min\{c \mid c \neq color_u \ for\ any\ u \in N(v)\} < color_v$ なる色 col が存在する場合，色詰め法（後述）に従って，$color_v$ を変更する.
6. $col = min\{c \mid c \neq color_u \ for\ any\ u \in N(v)\} < color_v$ （上記 3 ,4 または 5 で $color_v$ が変更された場合，変更後の $color_v$) なる色 col が存在する場合，$nextc_v = col$ とする．このような色 col が存在しない場合 $nextc_v = \perp$ とする.

上記のステップ 3 は，未彩色のノードを初めて彩色するための動作である．隣接するノードが同じ色で彩色されないように，後述の初期彩色法に従って彩色する．ただし，隣接ノードと同時に同じ色に彩色されることを避ける必要があり，ノード v が現ラウンドで彩色されるとは限らない.

ステップ 4 は，ノード v とその隣接ノード u が同じ現色を持つ（色衝突とよぶ）場合の動作である．このような色衝突は現ラウンド開始時にリンク (v, u) が追加されたときのみに生じる．提案アルゴリズムでは，各ラウンド終了時にノード彩色が実現している（色衝突がない）ことを保証するために，後述の色解消法によってただちに色衝突を解消する．ただし，この色衝突解消法の結果はGrundyノード彩色になっているとは限らない.

ステップ 5 はノード v がGrundyノード彩色の条件を満たさない際の動作である．このような状況は現ラウンド開始時にノード v に接続するリンクが削除された場合，あるいは，前ラウンドで v もしくは v のある隣接ノードが現色を変更した場合にのみ生じる．提案アルゴリズムでは，Grundyノード彩色を実現するため，後述の色詰め法により現色を変更する．ただし，隣接ノードと同時に現色を同じ色に変更することを避ける必要があり，ノード v が現ラウンドでGrundyノード彩色の条件を満たすように現色を変更できるとは限らない.
隣接ノードが同時に現色を変更した結果, これらのノード間で色衝突が発生することを避けるために, 各ステップにおいて以下のルールを適用する.

初期彩色法

- 隣接ノードに自分よりもIDが小さく色 $\perp$ をもつノードがある（$\exists u \in N(v) \wedge color_u = \perp, ID_u < ID_v$）場合,$color_v$ は変更しない.

以下, $\nexists u \in N(v), color_u = \perp \wedge ID_u < ID_v$ を満たす場合の彩色法を記述する.

- 次色がどの隣接ノードの現色または次色とも異なる場合は $color_v = nextc_v$ とする.
- 次色が隣接ノード u の現色と同じ（$nextc_v = color_u \neq \perp$）場合
  - リンクの追加によって色衝突が発生したなら $color_v = nextc_v$
  - リンクの追加以外によって色衝突が発生したなら $color_v = max\{c \mid c \notin \{color_w, nextc_w\}\ for\ any\ w \in N(v)\}$ とする.

この初期彩色法によって, 未彩色ノードのうち,ID 最小のノードは必ず彩色される. したがって,n ラウンドですべてのノードが彩色される.

色衝突解消法

- 次色をもたない（$nextc_v = \perp$）場合
  - 色衝突している隣接ノード u が次色をもつ（$nextc_u \neq \perp$）なら 3. へ.
  - ノード u が次色をもたないとき $ID_v < ID_u$ なら 4. へ.$ID_u < ID_v$ なら 3. へ.
  - リンクの追加によって色 $\perp$ をもつノード u の次色と衝突した場合（$color_v = nextc_u$）は,4. へ.
- 次色をもつ（$nextc_v \neq \perp$）場合
  - 色衝突している隣接ノード u が次色をもたない（$nextc_u = \perp$）なら 1. へ.
  - ノード u が次色をもち, かつ, 次色が同じ（$nextc_v = nextc_u$）とき $ID_v < ID_u$ なら 1. へ.$ID_u < ID_v$ なら 3. へ.
  - ノード u が次色をもち, かつ, 次色が異なる（$nextc_v \neq nextc_u$）とき $ID_v < ID_u$ なら 1. へ.$ID_u < ID_v$ なら 2. へ.

1. 色詰め法（後述）に従って $color_v = nextc_v$ が可能なら行う. 不可能なら 4. へ.
2. 色詰め法（後述）に従って $color_v = nextc_v$ が可能なら行う. 不可能なら 3. へ.
3. ノード u が $color_u$ を変更し色衝突を解消するため, なにも変更しない.
4. $color_v = min\{c \mid c \neq color_w \vee nextc_w \; for \; any \; w \in N(v)\}$ とする.

色詰め法

- 次色が同じ隣接ノードが存在しない（$\nexists u \in N(v), nextc_v = nextc_u$）なら 2．へ.
- 次色が同じ隣接ノードが存在する（$\exists u \in N(v), nextc_v = nextc_u$）なら 2．へ.

1. 次色が同じどの隣接ノードよりも現色がそれ以下（$\forall u, color_v \leq color_u (u \in N(v) \wedge nextc_v = nextc_u)$ （ただし $\perp < 1$ とする)）なら 2. へ. そうでないなら 3．へ.
2. $color_v = nextc_v$
3. 隣接ノードが $netc_v$ と同じ色になる可能性があるため変更しない.

4 正当性の証明

以下の 3 つの定理に関して証明を行う. ただし，以下では，ラウンド t 終了時のノード v の出力変数 $color_v$ の値を $color_t(v)$ と表す.

定理 1 任意のノードに対して, ラウンド終了時にはリンクでつながっている隣接ノードと色が異なるように割り当てられている

- $\forall v, \forall t, color(v) \neq \perp \Rightarrow \nexists u \in N(v)[color_t(v) = color_t(u)]$

定理 2 ラウンド開始時に未彩色のノードがあれば, そのラウンドで少なくとも一つのノードが彩色される.

- $1 \leq t \leq n, \; |\{v| \; color_t(v) \neq \perp\}| \geq min\{n|\,, |\{v| \, color_{t-1}(v) \neq \perp\}| + 1\}$

定理 3 全ノードが彩色されているなら, リンクの追加・削除が最後に生じてから $4\Delta - 1$ ラウンド経てば,Grundy ノード彩色が実現している

4.1 定理 1 の証明

任意のラウンドにおいて, ラウンド終了時には隣接ノードが異なる色で彩色されていることを証明する.

まず, ノードの色変更によって色衝突が発生することがないことを示す. ノードが色を変える方法は 2 通りある.1 つは次色以外の色（次色が $\perp$ の場合も含む）に変更するもの. もう 1 つは次色に変更するものである.

前者の変更は 2 パターンある.1 つは新しいリンクの追加により隣接ノードと色が競合した際に, どちらか 1 方がその隣接ノードの現色と次色以外の最小の色に変更する場合. もう 1 つは未彩色のノードがリンクの追加によらず隣接ノードの現色と色衝突したときに MSG に含まれない最大の色になる時である（初期彩色法）.$\perp$ が現色のときに色を変更するのは現色が $\perp$ であるすべての隣接ノードの ID が自分よりも大きいときのみなので, この変更によって隣接ノードと色が衝突することはない.

後者のように次色に変更するのは隣接ノードの現色と, 次色が自分の次色と異なる場合のみである.

従って, このときも色変更によって色衝突が発生することはない. したがって隣接ノードの色が等しくなるのは同じ現色を持つノード間にリンクが追加された場合のみである. このとき, 必ず一方のノードが色変更を行うが, 色変更によって色衝突は発生しないので, 任意のラウンド終了時には隣接ノードは異なる色で彩色されている． □

4.2 定理 2 の証明

ノードが彩色されると（$color_v \neq \perp$ になると）, それ以降, 未彩色（$color_v \neq \perp$） になることはない. したがって, ラウンド n 終了時までにすべてのノードが彩色されることを示すには，未彩色ノード（現色が $\perp$ のノード）がある限り，各ラウンドで少なくとも 1 つの未彩色ノードが彩色されることを示せば十分である.

未彩色ノードを彩色できないケースは, 自分よりも ID が小さい未彩色の隣接ノードを待つ場合のみである. したがって未彩色ノードの内，ID が最小のノードは必ず彩色される. □

4.3 定理 3 の証明

全ノードの彩色が終了しているなら, リンクの追加・削除が最後に生じてから $2k-1$ ラウンド経てば, 色 k 以下のすべてのノードが Grundy ノード彩色の条件を満たしていることを証明する. これにより, 最大色が 2Δ であることから定理 3 が証明される.

4.1 節よりリンクの追加・削除が行われても 1 ラウンドで必ず彩色は完了する. 従って，ノード彩色がされている（色の衝突がない）状況から $2k-2$ ラウンドの間，リンクの追加・削除が発生していないと考えてよい.

リンクの追加によってあるノード v の色が i から他の色に変更されたとする. これによって色詰めしなければならないのは i よりも大きい色をもつ v の隣接ノードであり,v が色の変更をしてから 2 ラウンド後に色を変更する. また $N(v)$ に属するあるノード u の色変更によって，さらにその隣接ノードで Grundy ノード彩色条件が破綻し，色詰めしなければならない場合がある．しかし，この場合も，色を変更するノードの現色は u の元の色よりも大きい色を持つノードのみである. リンクの削除が行われた場合は Grundy ノード彩色の条件を満たさないノードが高々 1 つ発生するだけである.

以上のことより, 彩色が完了されているなら, リンクの追加・削除が最後に生じてから $2k-1$ ラウンド経てば色 k 以下に関して色詰めされた状態である. □

5 おわりに

本稿では,Grundy ノード彩色に対する分散アルゴリズムを提案した．この分散アルゴリズムでは，彩色されていないリンクの追加･削除が連続的に生じる動的ネットワークに対して各ノードが隣接ノードの色をうまく考慮し変更することで Grundy ノード彩色を実現した．この分散アルゴリズムは，以下の長所を持つ.

1. すべてのノードは n ラウンド以内に彩色される
2. 各ラウンド終了時，どの隣接ノードも異なる色を持つ
   (リンク追加による色の衝突を 1 ラウンドで解消)
3. トポロジ変化が $4\Delta - 1$ ラウンドの間なければ,Grundy ノード彩色を実現

本稿では，同期式動的ネットワークにおいて，高々 1 つのリンクの追加または削除だけが同時に発生すると仮定した．しかし，提案アルゴリズムは，互いに独立したリンク（端点を共有しないリンク）であれば，複数リンクの追加や削除が同時に発生しても，同じ特長を持ちながら Grundy ノード彩色を実現できることを示せる．今後の課題は，互いに独立とは限らない複数リンクの追加または削除が同時に発生する場合にも対応できる分散アルゴリズムを開発することである.

謝辞

本研究は JSPS 科研費 JP26280022, JP16K00018, JP17K19977 の助成を受けたものです.

References

[1] Aspenes, J. and Shah, G. : *Skip graphs.* In 14th Symposium on Distributed Algorithms (SODA),Baltimore, MD, pp. 384393, 2003.

[2] Gallager, R. G., Humblet, P. A., Spira, P. M. : *A distributed algorithm for minimum-weight spanning trees.* ACM Transactions on Programming Languages and Systems 5 (1), 6677, 1983.

[3] Nicola,Santoro. : *Time to Change: On Distributed Computing in Dynamic Networks.* OPDIS, 2015.

[4] Fabian Kuhn. Rotem Oshman : *Dynamic Networks: Models and Algorithms.* ACM SIGACT News vol.42, no1, 2011.

[5] Giuseppe, Di, Luna1, Roberto, Baldoni. : *Non Trivial Computations in Anonymous Dynamic Networks.* OPDIS, 2015.

[6] Hedetniemi,S.T., Jacobs, D.P. and Srimani. P.K. : *Linear time Self-stabilizing colorings*. In-formation Processing Letters 87，pp.251-255，2006.