# タブーサーチを用いたモジュール性による無向グラフのクラスタリングアルゴリズム

田村　慶一† 　高木　允†† 　森　康真† 　黒木　進† 　北上　始†

† 広島市立大学情報科学部
†† 広島市立大学大学院情報科学研究科/日本学術振興会特別研究員 DC
〒 731–3194 広島県広島市安佐南大塚東 3–4–1
E-mail: †{ktamura,mori,kuroki,kitakami}@its.hiroshima-cu.ac.jp, ††makoto@db.its.hiroshima-cu.ac.jp

**あらまし**　無向グラフの頂点集合をクラスタと呼ばれる密な構造に分割する手法として，Newman がモジュール性を利用したクラスタリングアルゴリズム（以下，Newman のアルゴリズムと呼ぶ）を提案している．Newman のアルゴリズムは貪欲アルゴリズムであり，高速にクラスタリングができる．しかしながら，貪欲アルゴリズムは局所最適解に陥る可能性があり，良いクラスタリングが得られない場合がある．そこで，本論文では，モジュール性の概念を利用し，タブーサーチを用いた無向グラフのクラスタリングアルゴリズムを提案する．タブーサーチを用いることにより，Newman のアルゴリズムよりも精度の高いクラスタリングが得られると期待される．提案アルゴリズムを評価するために，ネットワークデータと，ブログデータのトラックバックデータを無向グラフとしてみなしたグラフデータとを利用しクラスタリングの評価実験をおこなった．評価実験の結果，提案アルゴリズムは Newman のアルゴリズムと比較して精度の高いクラスタリングを求めることができることを確認した．本論文では，提案アルゴリズムの説明をおこなうとともに，評価実験の結果を報告する．

**キーワード**　クラスタリング，グラフデータ，最適化手法

# A Clustering Algorithm of a Undirected Graph by the Modularity using Tabu Search

Keiichi TAMURA†, Makoto TAKAKI††, Yasuma MORI†, Susumu KUROKI†, and Hajime KITAKAMI†

† Faculty of Infomation Sciences, Hiroshima City University
†† Graduate School of Information Sciences, Hiroshima City University/JSPS Research Fellow
Ozuka-Higashi 1–2–3, Asa-Minami-ku, Hiroshima, 731–3194 Japan
E-mail: †{ktamura,mori,kuroki,kitakami}@its.hiroshima-cu.ac.jp, ††makoto@db.its.hiroshima-cu.ac.jp

**Abstract**　As a technique of dividing vertex set of a undirected graph into a dense structure called a cluster, Newman has proposed a clustering algorithm using the modularity(the Newman's algorithm). The Newman's algorithm is a greedy algorithm and can obtain a clustering at high speed. However, a greedy algorithm may fall into a partial optimal solution, and a good clustering may not be obtained. This paper propsed a clustering algotihm of a undirected graph by the modularity using the tabu search. By using tabu search, it is expected that the proposed algorithm can obtain a high-precision clustering rather than the Newman's algorithm. In order to evaluate the proposed algorithm, the experiments used a network data and a graph data of the trackback data. As the experimental results, the proposed algorithm obtained the clustering in which accuracy is high compared with the Newman's algorithm. This paper explains the proposed algorithm and reports the experimental results.

**Key words**　Clustering, Graph Data, Optimization

## 1. はじめに

グラフの頂点集合を複数の部分集合に分類することをグラフのクラスタリングという．グラフのクラスタリングは，クラスタ分析として様々なアプリケーションで利用されており，グラフの可視化，VLSI のデザイン，タンパク質のドメインネットワーク分析，ネットワーク設計やソーシャルネットワークにおけるコミュニティ抽出にとって重要な課題となっている．

本論文は辺の構造（頂点同士のつながり方）によるグラフのクラスタリング（文献 [1]〜[3]）を扱う．辺の構造によるグラ

フのクラスタリングは，1 つのグラフを辺が密に張られたクラスタと呼ばれる「かたまり」に分割することを目的とする．ただし，本論文ではグラフとして無向グラフのみを考え，辺の重み，セルフループと多重辺とは考慮しないものとする（以下，「グラフ」はこのような単純な無向グラフを指すものとする）．図 1 に示すグラフを例として考える．図 1 に示すように 3 つのクラスタを求める問題がグラフのクラスタリングとなる．

グラフのクラスタリングアルゴリズムとして，Newman がモジュール性（Modularity）を用いたクラスタリングアルゴリズム（以下，Newman のアルゴリズムと呼ぶ）を提案している（文献 [4], [5]）．モジュール性とはクラスタリングの良さを示す尺度である．Newman のアルゴリズムは，モジュール性を表す度合いである $Q$ の値を評価値として，$Q$ の値の増加値 $\Delta Q$ が最大であるクラスタ同士を結合していく貪欲アルゴリズム（greedy algorithm）[6] となっている．

Newman のアルゴリズムはクラスタリングが高速におこなえるが，次のような課題が存在する．グラフのクラスタリングは，組合せ最適化問題のひとつで，厳密解を求めようとすると指数オーダーの計算量が必要である．そこで，Newman のアルゴリズムは近似解法である貪欲アルゴリズムを用いて準最適解を求めている．しかしながら，貪欲アルゴリズムを用いることで必ずしも良い準最適解を求めることができるとは限らない．貪欲アルゴリズムは局所最適解に陥る恐れがあり，評価値である $Q$ の値が極大になるにも関わらず最適なクラスタリングを求めることができないことがある．

本論文では，上述の課題を解決するために，メタヒューリスティック解法の 1 つであるタブーサーチ（Tabu Search）[7] を用いたモジュール性によるグラフのクラスタリングアルゴリズムを提案する．タブーサーチは局所最適解に至ってもそこでトラップされずにさらに動き回る探索手法である．タブーサーチを用いることにより，局所最適解に陥る可能性が小さくなり，より良い解に遷移する可能性が高くなる．このタブーサーチを適用することにより，Newman のアルゴリズムと比較して精度の高いクラスタリングを求めることができると期待される．

提案アルゴリズムを実際に実装し，提案アルゴリズムと Newman のアルゴリズムの比較実験をおこなった．実験では，まず，人工的なグラフデータとネットワーク網とを使用して評価実験をおこなった．この評価実験では，提案アルゴリズムを用いると，Newman のアルゴリズムと比較して $Q$ の値が大きいクラスタリングを求めることができることを確認した．次に，ブログのトラックバックデータを用い，コミュニティ抽出という観点でクラスタリングの評価をおこなった．この評価実験では，Newman のアルゴリズムでは 1 つのクラスタとして抽出される複数のクラスタを，提案アルゴリズムを用いると，それぞれ個別のクラスタとして抽出できることを確認した．

本論文の構成は以下の通りである．第 2 章では，Newman のアルゴリズムとその課題について説明する．第 3 章では，タブーサーチを用いたモジュール性によるグラフのクラスタリングアルゴリズムを提案し，第 4 章では関連研究を述べる．第 5 章で評価実験を示し，第 6 章で本論文をまとめる．

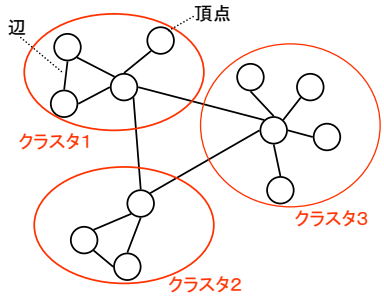



図 1　グラフのクラスタリング

## 2. Newman のアルゴリズムとその課題

本章では Newman のアルゴリズムとその課題を説明する．

### 2.1 問題定義

無向グラフ $G = (V, E)$（$V$:頂点集合，$E$:辺集合）を考える．クラスタ $i$ に所属する頂点集合を $C_i \in V (1 \leqq i \leqq n)$ と表すと，クラスタリング $C$ は以下のようにすべてのクラスタからなる集合として表現される．

$$C = \{C_1, C_2, \ldots, C_n\} \ (C_i \subset V) \tag{1}$$

このとき，以下の関係が成り立っている．

$$V = \bigcup_{i=1}^{n} C_i \qquad C_i \cap C_j = \phi \ (i \neq j)$$

ここで，$\mathcal{F}$ をクラスタリングの良さを示す評価関数，$\mathcal{F}(G, C)$ が返す値を評価値とする．グラフのクラスタリング問題は，以下のように $\mathcal{F}(G, C)$ の値を最大にするクラスタリング $C$ を求める組合せ最適化問題となる．

$$\max_{C \in \mathbf{S}} \mathcal{F}(G, C) \tag{2}$$

式（2）において，$\mathbf{S}$ はグラフ $G$ のクラスタリング $C$ として考えれられる実行可能解の集合である．クラスタ数の最大数を $n$，頂点数を $|V|$ とすると，考えられるクラスタリングの組合せの数は $n^{|V|}$ になる．

### 2.2 モジュール性

モジュール性とはクラスタリングの良さを示す尺度である．「クラスタ内に含まれている辺の数の割合が，クラスタ外に出ている辺の割合よりも大きいクラスタ」は独立性の高いクラスタである．そして，そのようなクラスタを多く含むクラスタリングがモジュール性の高いクラスタリングとなる．

例えば，図 1 の各クラスタは，内部に含まれている辺の数に比べて外部に向かっている辺の数が少ないので，このクラスタリングはモジュール性が高いといえる．図 1 において，クラスタ 3 は必ずしもクリークといった密な構造とはいえないが，独立性があるという点では 1 つのクラスタとして抽出されるべきである．このようなクラスタも求めることができるのが Newman のアルゴリズムの特徴でもある．

モジュール性の度合い示す評価値は $Q$ で表される．この評価値は，クラスタとクラスタとを結び付けている橋の部分を識別することを目的に設計されている．具体的には，$Q$ の値は，「クラスタに含まれている辺の割合がクラスタから出ている辺

の割合よりもどれだけ大きいか」ということと「グラフが適度に分割されているか」ということのトレードオフをとる値となる（詳しくは文献 [8] を参照のこと）．

以下に $Q$ の定義式を示す．

$$Q = \sum_i (e_{ii} - a_i^2) \tag{3}$$

$$e_{ii} = \frac{\text{クラスタ } i \text{ 内部の辺数}}{m}$$

$$a_i = \sum_j e_{ij}$$

$$e_{ij} = \frac{\text{クラスタ } i \text{ とクラスタ } j \text{ 間の辺数}}{2m}$$

$m$ はグラフの辺数である．また，$Q$ の値を構成する各クラスタに対応する $(e_{ii} - a_i^2)$ の値を各クラスタ毎の $Q$ の値と呼び，ここでは $Q_i$ と表記する．図 1 のクラスタリング結果を用いて $Q$ の値を示す．クラスタ 1 に関しては，内部には 4 本の辺が存在し，外部に 2 本の辺が出ているので，$Q_1 = 4/14 - 81/28^2$ となる．同様に，クラスタ 2 は，$Q_2 = 3/14 - 64/28^2$，クラスタ 3 は，$Q_3 = 4/14 - 100/28^2$ となる．よって，$Q = 11/14 - 245/28^2$ となる．

### 2.3 Newman のアルゴリズム

Newman のアルゴリズムは，近似解法である貪欲アルゴリズムであり，モジュール性の度合いである $Q$ の値を評価値として使用する．貪欲アルゴリズムは，組合せの要素をそれぞれ独立に評価し，評価値の高い順に要素を組合せに取り込んでいくことで解を得る手法である．

Newman のアルゴリズムを以下に示す．

**Algorithm 1** Newman 入力:G(V,E) 出力:C

```
1:  C := φ;
2:  for all v ∈ V do
3:      C := C ∩ {v};  /* 各頂点を 1 つのクラスタとする */
4:  end for
5:  while (1) do
6:      ΔQ := ∅;  /* ΔQ は |C| 行 |C| 列の行列 */
7:      ΔQ := CALC_DQ(G, C);
        /* 関数 CALC_DQ(G, C) はクラスタリング C についてクラスタ
        i とクラスタ j とを結合したときに増加する Q の値 ΔQ_ij を算
        出する関数 */
8:      {max_dq, i, j} := GET_MAX_DELTA_Q(ΔQ);
        /* 関数 GET_MAX_DELTA_Q は ΔQ_ij の最大値 max_dq とその
        添字 {i, j} を返す関数 */
9:      if max_dq > 0 then
10:         C := RECLUSTERING(C, i, j);
            /* 関数 RECLUSTERING(C, i, j) は，C において C_i と C_j を
            結合して 1 つのクラスタとする関数 */
11:     else
12:         return C; /* 得られたクラスタリング結果を返す */
13:     end if
14: end while
```

Newman のアルゴリズムの流れを簡単に説明する．最初に，グラフの各頂点を 1 つのクラスタとする．次に，関数 **CALC_DQ** を用いて 2 つのクラスタ $i$ とクラスタ $j$ を結合したときのモジュール性の増加値 $\Delta Q_{ij}$ を計算する．そして，関数 **GET_MAX_DELTA_Q** を用いて増加値 $\Delta Q_{ij}$ の最大値を求める．最後に，増加値 $\Delta Q_{ij}$ が最大値であるクラスタ同士を結合して 1 つのクラスタとする．以下，5 行目に戻り，同じ処理を繰り返しクラスタを結合していく．増加値 $\Delta Q_{ij}$ がすべて負の値となれば，これ以上結合を続けたとしても $Q$ の値が減少するため，処理を終了する．

$\Delta Q_{ij}$ は以下の式により求めることができる．

$$\Delta Q_{ij} = 2(e_{ij} - a_i a_j) \tag{4}$$

例えば，図 1 のクラスタ 1 とクラスタ 3 とを結合することを考える．$\Delta Q_{13} = 2(1/14 - 9/28 \times 9/28) < 0$ となり，クラスタ 1 とクラスタ 3 とを結合すると $Q$ の値が減少してしまう．同様に，クラスタ 1 とクラスタ 2 とをクラスタ 2 とクラスタ 3 とを結合した場合，$\Delta Q_{13}$ と $\Delta Q_{23}$ とは 0 より小さくなる．よって，図 1 のクラスタリングは $Q$ の値が極大なクラスタリングといえる．

### 2.4 Newman のアルゴリズムのその課題

Newman のアルゴリズムは，評価値が負になる方向への探索は許さないため，局所最適解に陥ることが考えられる．また，Newman のアルゴリズムは，クラスタの結合のみしかおこなえないため，探索の終盤になるほど解の探索範囲が狭まってしまうという問題点がある．

例えば，図 2-左のグラフを Newman のアルゴリズムによりクラスタリングすると，図 2-中のクラスタリングが得られる．しかしながら，$Q$ の値が最大となるクラスタリングは図 2-右である．

この例は，探索の終盤に解の探索範囲が狭まったため発生したクラスタリングの失敗例といえる．図 2-中において，頂点 17，頂点 18，頂点 19 と頂点 20 を 1 つのクラスタとして抽出できない理由は次の通りである（以下，各頂点 $i$ を $v_i$ と表記する）．

最初に，$v_1$，$v_2$，$v_3$ と $v_4$ とがクラスタ 1 に，$v_5$，$v_6$，$v_7$ と $v_8$ とがクラスタ 2 に，$v_9$，$v_{10}$，$v_{11}$ と $v_{12}$ とがクラスタ 3 に，$v_{13}$，$v_{14}$，$v_{15}$ と $v_{16}$ とがクラスタ 4 として結合する．次に，$v_{17}$ と $v_{18}$，$v_{19}$ と $v_{20}$ がそれぞれペアになるよりも，各 $v_{17}$，$v_{18}$，$v_{19}$ と $v_{20}$ とが隣接するクラスタと結合する方が $\Delta Q$ の値が大きいため，$v_{17}$，$v_{18}$，$v_{19}$ と $v_{20}$ とがそれぞれ隣接するクラスタと結合して 4 つのクラスタが形成される．この時点で，$Q$ の値が極大となり，Newman のアルゴリズムでは図 2-中のようなクラスタリングとなる．

Newman のアルゴリズムは準最適解を求めることが目的ではあるが，このような小規模なグラフの例でさえ，良いクラスタリングを求めることができていない．よって，より良い準最適解を求めるには局所最適解に陥らない手法を考える必要がある．

## 3. タブーサーチによるクラスタリングアルゴリズム

本章では，タブーサーチによるクラスタリングアルゴリズムを説明する．

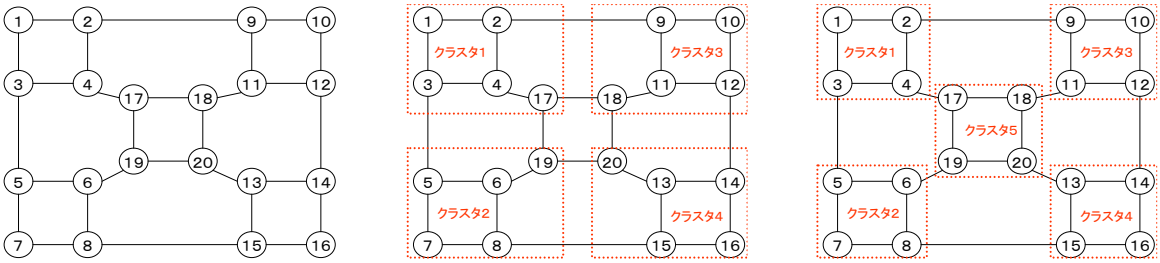



図 2　Newman のアルゴリズムでのクラスタリング例

### 3.1　タブーサーチ

タブーサーチの探索方針は次の通りである．現在得られている解 $C$ の近傍集合 $\mathcal{N}(C)$ の中で最良の近傍解 $\tilde{C}(\in \mathcal{N}(C))$ を求める．$\mathcal{N}(C)$ は大きくなるため，通常，$\tilde{C}$ からランダムに複数の近傍解を生成し，生成した近傍解の集合内で最良の近傍解 $\tilde{C}$ を選ぶことが多い．このとき，この近傍解がたとえ改悪であったとしても $\tilde{C}$ を次の解として選ぶ．このルールにより，局所最適解に至ったとしても改悪をゆるす方向に探索が進むため，局所最適解に陥る可能性が低くなる．

ただし，現在の解 $C$ が局所最適解である場合，$\mathcal{N}(C)$ の最良解 $\tilde{C}$ に移ったとしても再び $C$ に戻る可能性が高い．そこで，タブーサーチでは，この堂々巡りを避けるために，タブーリストを用意し，$C$ に戻るような操作を禁止するようになっている．

### 3.2　近　傍　解

タブーサーチでは近傍解を定義する必要があるが，ここでは近傍解の定義を示す．

あるクラスタに所属している頂点を別のクラスタへ所属を変更する操作を「異動」と呼ぶこととする．この「異動」の操作を 1 回実行して得られるクラスタリングを近傍解とする．つまり，現在得られているクラスタリング $C$ において，クラスタ $C_i$ に所属するある頂点 $v$ をクラスタ $C_j$ に所属を変更したクラスタリングが近傍解となる．

ただし，他のどのクラスタとも隣接していない頂点を異動したとしても良い探索をおこなっているとはいえないため，「異動」の対象となる頂点はクラスタ内で他のクラスタの頂点と隣接している頂点のみとする（図 3(a)）．

図 3(b) と図 3(c) とに近傍解の例を示す．図 3(b) を現在のクラスタリングと考える．クラスタ 1 の頂点 $v$ をクラスタ 2 に「移動」させたクラスタリングが図 3(c) となる．このクラスタリングは図 3(b) のクラスタリングの近傍解である．

また，提案アルゴリズムでは，近傍解は各クラスタ毎に 1 つ生成し，全体で $|C|$ 個の近傍解を生成することにする．各クラスタ毎に近傍解を生成することで，各クラスタ毎に探索が均等に進み，解の収束速度を早めることができる．

クラスタ $C_i$ に所属する頂点 $v$ をクラスタ $C_j$ に異動したときの $\Delta Q_{ij}(v)$ は次の式で求めることができる．

$$\Delta Q_{ij}(v) = e_{ij}(v) + \frac{a_j - a_i}{m} - 2\left(\frac{k_v}{2m}\right)^2 \qquad (5)$$

$$e_{ij}(v) = \frac{C_i \text{と} C_j \text{間の辺で端点に} v \text{を含む辺の数}}{2m}$$

$k_v$ : 頂点 $v$ の次数

### 3.3　処 理 手 順

タブーサーチを用いたモジュール性によるクラスタリングアルゴリズムの処理手順を次に示す．タブーリストの設計と初期解については次節以降で詳しく述べる．

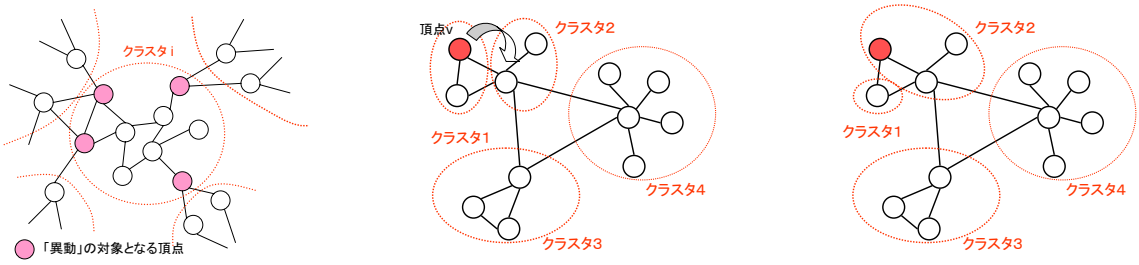
**Algorithm 2** TABU_CLUSTERING 入力:$G(V, E)$ 出力:$C$

```
1:  C := INIT(G);
    /* INIT(G) は初期解を求め，初期解を返す関数 */
2:  T := φ;
3:  Q := Q(G, C);  /* Q(G, C) は Q の値を返す関数 */
4:  C_best := C;  Q_best := Q; /* 最良解の保存 */
5:  while (終了条件) do
6:      {C̃, dq, v, i, j} := GET_BEST_NEIGHBOR(G, C, T);
        /* 関数 GET_BEST_NEIGHBOR(G, C, T) を実行し，近傍解の中で最も Q の増加値が大きい近傍解 C̃，その増加値 dq,「異動」の対象となった頂点 v,「移動」元のクラスタ番号 i と「異動」先のクラスタ番号 j を返す．詳細は Algorithm3 に記載する． */
7:      C := C̃;  Q := Q + dq;  /* 次の解へ移動 */
8:      if dq ≦ 0 then
9:          if Q ≧ Q_best then
10:             C_best := C;  Q_best := Q; /* 最良解の保存 */
11:         end if
12:     else
13:         UPDATE_TABULIST(T, v, i, j);
            /* タブーリストの更新 */
14:     end if
15: end while
16: return C_best;
```

**Algorithm 3** GET_BEST_NEIGHBOR 入力:$G(V, E)$, $C$, $T$ 出力:$\tilde{C}, max\_dq, v, i, j$

```
1: for all c ∈ C do
2:     クラスタ c 内から「異動」の対象となる頂点 v と「異動」先とをランダムに選び，近傍解を生成する．また，Q の増加値 dq を計算する．ただし，タブーリスト T に登録してある頂点は「異動」の対象とはならない．
3: end for
4: 生成した近傍解の中から，Q の増加値 dq が最大の近傍解 C̃ を取り出す．
5: C̃，max_dq,「異動」の対象となった頂点 v,「移動」元のクラスタ番号 i と「異動」先のクラスタ番号 j とを返す．
```

提案アルゴリズムの処理の流れを簡単に説明する．最初に，関数 **INIT** で初期解を作成する．次に，関数 **GET_BEST_NEIGHBOR** を実行する．この関数では，現在のクラスタリング $C$ から，各クラスタごとに「異動」の対象を




(a) 近傍解の候補　(b) 例（現在の解）　(c) 例（近傍解）

図 3　近傍解について

選択し，近傍解を $n$ 個生成する（$n$ はクラスタの数）．近傍解生成は，タブーリストを用いて堂々巡りになる可能性がある近傍解は生成されないように制御される．そして，各近傍解について $Q$ の増加値を計算し，増加値 $dq$ が最大の近傍解 $\tilde{C}$ を返す．最後に，$\tilde{C}$ を $C$ と置き換え，次の解へ遷移する．終了条件としては試行回数を使用する．

### 3.4　タブーリスト

タブーリストとして「異動」した頂点を登録する．タブーリストに登録されている頂点は，しばらくの間，近傍解生成の対象とならない．タブーリストへの登録は，改悪の方向に探索が進んだ場合のみ登録がおこなわれ，局所最適解に陥ったときに探索が堂々巡りすることを回避することができる．

通常，タブーリストは 1 つ作成するが，グラフのクラスタリングにおいて解の堂々巡りを回避するためにはクラスタの数以上の長さのタブーリストを作成する必要がある．この場合，タブーリストの長さが大きくなるため効率が悪い．そこで，1 つのタブーリストを持つのではなく，各クラス毎にタブーリストを作成する．

### 3.5　初　期　解

初期解としては，ランダムに頂点を分割することが考えられる．しかしならが，何個に分割するのかということと頂点同士のつながりを考える必要があり，初期解の生成に時間がかかる．また，各頂点を 1 つのクラスタとすることも考えられるが，大規模なグラフの場合，収束に時間がかかってしまう．そこで，提案アルゴリズムでは，次数の大きい頂点とその隣接頂点を 1 つのクラスタとしていく（但し，すでに他のクラスタに分類されているものは含めない）方法で，初期解を設定する．

## 4.　関 連 研 究

近年，ソーシャルネットワーク分野において頻繁に Newman のアルゴリズムが利用され，その有用性が確認されている．文献 [9] は mixi の人と人とのつながりを示すグラフ構造に対して，文献 [10] はたんぱく質の相互作用ネットワークに対して文献 [11] ではブログデータに対して Newman のアルゴリズムを適用している．その他，Newman らは様々なグラフに対してモジュール性を用いたグラフのクラスタリング手法を適用して有用性を検証している．

コミュニティ抽出という点では，カットを利用したコミュニティ抽出手法が数多く提案（文献 [12]）されている．カットを利用したコミュニティ抽出手法では個々のコミュニティを取り出すことを目的としている．一方，Newman のアルゴリズムはグラフの頂点を分類することを目的としている．よって，Newman のアルゴリズムはグラフ全体の構造を把握しやすいという長所がある．ただし，Newman のアルゴリズムは，無向グラフにしか適用できないという制約も存在する．

クラスタリングという点では，同様にカットを利用したクラスタリングアルゴリズムが提案（文献 [1]）されている．これらのアルゴリズムでは，クリークや疑クリークといった密接なクラスタを抽出することを目的としているため，ハブや疎な構造ではあるが 1 つのかたまりとして判断できるものが抽出できない．Newman のアルゴリズムでは，図 1 に示すように，クリークのような構造ではないクラスタ 3 のような構造も取り出すことができる．

文献 [11] では，ブログのトラックバックデータに Newman のアルゴリズムを使用すると，少数の大きなクラスタが抽出され，それらのクラスタでは複数の話題が含まれていたという報告がされている．この事例では，局所最適解に陥っていると考えられる．文献 [11] では，この問題をトラックバックの内容を考慮し，辺に対して重み付けをすることで緩和できることを示している．本研究は，グラフの構造のみを考慮し，局所最適解におちいることを回避する手法の開発を目指している．

文献 [13] では，Newman のアルゴリズムにおける結合過程を工夫することでトータルの処理ステップを減少させる手法を提案している．提案されている手法では $\Delta Q_{ij}$ が最大ではないクラスタ同士が結合していくが，最終的に得られる $Q$ の値がオリジナルのアルゴリズムよりも大きくなることが報告されている．この結果は，貪欲アルゴリズムで得られる準最適解よりもより評価の高い準最適解が存在することを示唆している．

一方，タブーサーチはグラフの分割問題によく使用されている．グラフの分割問題では，はじめに，適当にグラフの頂点を 2 つの集合に分割する．次に，2 つの集合の頂点同士を交換することで近傍解を生成することでタブーサーチ進めていく．このアルゴリズムでは，分割数や分割された 2 つの頂点集合の集合数が均一であるため，本研究で扱う問題よりも非常に簡単である．

また，文献 [14] では，タブーサーチと同様にメタヒューリスティック解法の 1 つである確率的進化手法を使用したクラスタリングアルゴリズムが提案されている．提案されているアルゴリズムは，完全グラフ内の頂点を辺の重みによりクラスタリングする目的で開発されており，疎なグラフのクラスタリングを目的としている Newman のアルゴリズムには適用できない．

表 1　テストデータの詳細

| | 頂点数 | 辺数 | 内容 |
|---|---|---|---|
| テストデータ 1 | 16 | 17 | サンプルデータ |
| テストデータ 2 | 16 | 30 | サンプルデータ |
| テストデータ 3 | 20 | 28 | サンプルデータ |
| テストデータ 4 | 28 | 30 | サンプルデータ |
| テストデータ 5 | 332 | 2125 | アメリカ合衆国航空網（1997 年） |
| テストデータ 6 | 4432 | 28733 | ブログデータ（2006 年 6 月） |
| テストデータ 7 | 3147 | 18986 | ブログデータ（2006 年 7 月） |
| テストデータ 8 | 3951 | 23966 | ブログデータ（2006 年 8 月） |
| テストデータ 9 | 2284 | 10760 | ブログデータ（2006 年 9 月） |

## 5. 評価実験

提案アルゴリズムと Newman のアルゴリズムを比較するために評価実験をおこなった．

### 5.1 データセット

表 1 に示す 9 つのデータセットに関して，Newman のアルゴリズムと提案アルゴリズムとで得られたクラスタリングの比較をおこなう．

テストデータ 1 とテストデータ 2 とは最適解がすでに分かっているデータであり，クラスタリングが正確におこなえるかを検証するために用いる．テストデータ 3 とテストデータ 4 とは Newman のアルゴリズムを用いると局所最適解に陥る例である．テストデータ 5 はアメリカ合衆国の航空網（1997 年）を無向グラフで示したデータである．頂点が空港で辺が航路にあたる．

テストデータ 6，7，8 と 9 はブログのトラックバックデータを無向グラフとみなしたデータである．これらのデータは，Newman のアルゴリズムが頻繁に利用されているコミュニティ抽出において，提案アルゴリズムがどのような効果をあげるかを検証するために用いた．このデータは頂点がブログのサイトを示し，辺はそのブログ間にトラックバックが 1 本以上張られていることを示す．

### 5.2 実験環境

実験に使用した計算機はデスクトップパソコン（CPU:PentiumD 2.8GHz, Memory:2Gbyte, Disk:250GB）である．各クラスタが持つタブーリストの長さは 1 とする．また，提案アルゴリズムの試行回数は 5000 回に設定する．

### 5.3 実験結果 1

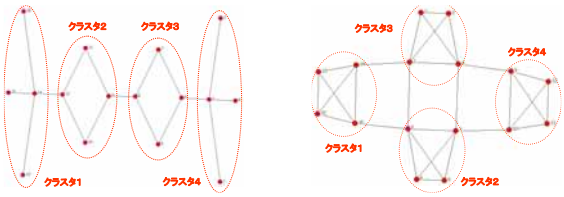
本節では，テストデータ 1 からテストデータ 5 までの実験結果を示す．

図 4 にテストデータ 1 とテストデータ 2 とのクラスタリング結果を図示する．Newman のアルゴリズムと提案アルゴリズムともに同様のクラスタリングが得られた．その時の $Q$ の値は，それぞれ，0.565744 と 0.548889 とで最適値が得られている．

図 5 と図 6 とにテストデータ 3 とテストデータ 4 とのクラスタリング結果を図示する．いずれの結果も，Newman のアルゴリズムを用いた場合，最適解が得られなかったが，提案アルゴリズムでは最適解が得られている．

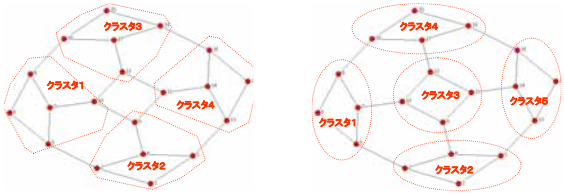
### 5.4 実験結果 2

本節ではテストデータ 4 の実験結果を示す．まず，表 2 に得



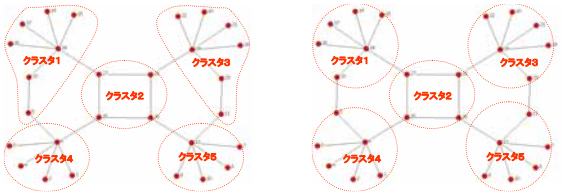

(a) テストデータ 1　　(b) テストデータ 2

図 4　クラスタリング結果（テストデータ 1,2）




(a) Newman のアルゴリズム　　(b) 提案アルゴリズム

図 5　クラスタリング結果（テストデータ 3）




(a) Newman のアルゴリズム　　(b) 提案アルゴリズム

図 6　クラスタリング結果（テストデータ 4）

られたクラスタ数と $Q$ の値とを示す．表 2 から分かるように，提案アルゴリズムの方が良い $Q$ の値のクラスタリングが得られていることが分かる．

図 7 にクラスタの頂点数と各クラスタの $Q$ 値の関係を散布図にしたグラフを示す．各クラスタの $Q$ の値とは $(e_{ii} - a_i^2)$ の値を示す．グラフは横軸がクラスタの頂点数で縦軸が $Q$ の値である．提案アルゴリズムでは，モジュール性の高いクラスタ（頂点数 103 個）が 1 つ得られており，その結果，Newman のアルゴリズムよりも良い $Q$ の値が得られたといえる．

図 8 に図 7 中の丸の破線に対応する 2 つのクラスタを図示する．Newman のアルゴリズムではクラスタ 1 とクラスタ 2 とは 1 つのクラスタとして抽出された．図 8 では，媒介中心性が高い頂点ほど頂点のサイズを大きくして頂点を描画している．媒介中心性とはクラスタとクラスタとを結ぶ橋の結合部分である度合いを示すものであり，その部分（頂点）を中心にグラフを分割できることを示唆している．

1 つのクラスタとして抽出したときのクラスタの $Q$ の値は 0.143487 であり，2 つのクラスタとして抽出したときの 2 つのクラスタの $Q$ の値の合計は 0.178874 である．2 つのクラスタに分離した方が評価が高いといえる．これを裏付けるように，媒介中心性が高い頂点がクラスタ 1 とクラスタ 2 の間に存在している．つまり，クラスタ 1 とクラスタ 2 との間に隔離性があり，2 つのクラスタに分離した方が良いことを示している．

以上の結果より，提案アルゴリズムの方が Newman のアルゴリズムと比較して良いクラスタリング結果を求めることができているといえる．

### 5.5 実験結果 3

本節では，テストデータ 6 からテストデータ 9 までの実験結果を示す．

表 2　テストデータ 5 の実験結果

| | クラスタ数 | Q |
|---|---|---|
| Newman のアルゴリズム | 7 | 0.320392 |
| 提案アルゴリズム | 10 | 0.350094 |

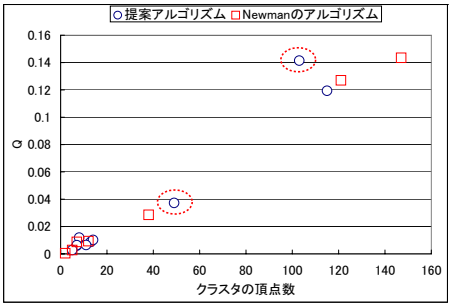



図 7　クラスタの頂点数と Q 値の比較（テストデータ 5）

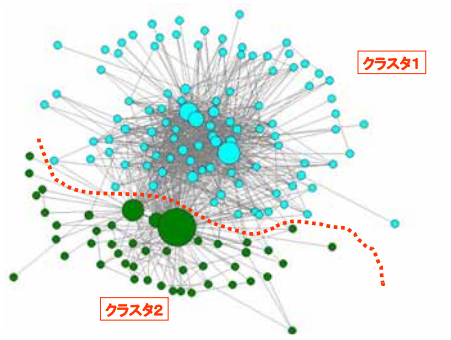



図 8　クラスタの図示（テストデータ 5）

各データにおいて得られてたクラスタリングの結果を表 3, 表 4，表 5 と表 6 とに示す．いずれの結果も，提案アルゴリズムの方が Newman のアルゴリズムと比較して，$Q$ の値が良いクラスタリングが得られている．

図 9，図 10，図 11 と図 12 とにクラスタの頂点数と各クラスタの $Q$ 値の関係を散布図にしたグラフを示す．グラフから分かるように，頂点数が増加すると急激に $Q$ の値が増加している．これは，クラスタの頂点数が増えるほどクラスタ内に張られる辺の密度が大きくなっているためである．

提案アルゴリズムと Newman のアルゴリズムを比較すると，クラスタの頂点数に対する $Q$ の値は提案アルゴリズムの方が大きくなっている．頂点数が同じで，$Q$ の値が大きいということはより密なクラスタを抽出できているといえる．$Q$ の値にそれほど差がなかったテストデータ 9 では，提案アルゴリズムと Newman のアルゴリズムはともに同じような傾向で $Q$ の値が増えている．

いくつかのクラスタにおいて Newman のアルゴリズムと比較して良い $Q$ の値が得られていないように見えるクラスタがある．これは以下の理由で特に問題ないと考える．図 9 で頂点数が 200 前後の 3 つのクラスタは，Newman のアルゴリズムでもそれほど良い $Q$ の値が得られていないクラスタが分離して得られたものであった．また，図 10 で頂点数が最大のクラスタは，Newman のアルゴリズムで得られた頂点数が最大のクラスタから頂点数を増やしたものではなく，Newman のアルゴリズムで得られた頂点数 600 近くで $Q$ の値が小さいクラスタにさらに頂点を加えたクラスタであった．

表 3　テストデータ 6 の実験結果

| | クラスタ数 | Q |
|---|---|---|
| Newman のアルゴリズム | 44 | 0.579466 |
| 提案アルゴリズム | 166 | 0.602565 |

表 4　テストデータ 7 の実験結果

| | クラスタ数 | Q |
|---|---|---|
| Newman のアルゴリズム | 44 | 0.515584 |
| 提案アルゴリズム | 151 | 0.562549 |

表 5　テストデータ 8 の実験結果

| | クラスタ数 | Q |
|---|---|---|
| Newman のアルゴリズム | 49 | 0.58815 |
| 提案アルゴリズム | 201 | 0.614505 |

表 6　テストデータ 9 の実験結果

| | クラスタ数 | Q |
|---|---|---|
| Newman のアルゴリズム | 52 | 0.636598 |
| 提案アルゴリズム | 124 | 0.638274 |

次に，テストデータ 6 から得られたクラスタリング結果についてさらに詳しく見ていく．Newman のアルゴリズムで抽出された 4 つクラスタ（クラスタ 8，クラスタ 9，クラスタ 12，クラスタ 19）は，提案アルゴリズムで抽出された複数のクラスタから構成されていることが分かった．例えば，Newman のアルゴリズムで抽出されたクラスタ 8 は，提案アルゴリズムのクラスタ 9，12，13，15 と 23 とを結合して得られるクラスタになっていた．

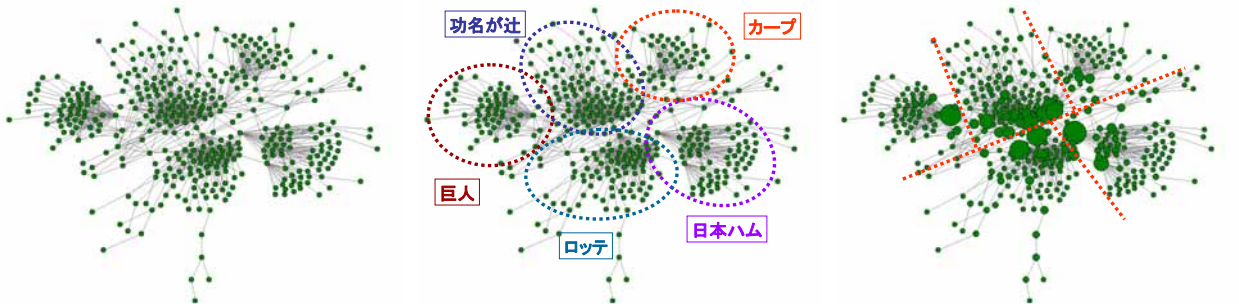
クラスタ 8 について各ブログサイトの記事に対して tf-idf 解析をおこなった．その結果,「日本ハム」,「カープ」,「ロッテ」,「巨人」と「功名が辻」という複数のトピックを含むことが分かった．同様に，提案アルゴリズムで抽出された 5 個のクラスタに対して同様に tf-idf 解析をおこなったところ，クラスタ 12 が「日本ハム」，クラスタ 15 が「カープ」，クラスタ 13 が「ロッテ」，クラスタ 23 が「巨人」，クラスタ 9 が「功名が辻」をトピックとしていることが分かった．

以上のことから，Newman のアルゴリズムで得られたクラスタ 8 は 5 つのクラスタが結合いることが分かった．図 13(a) にクラスタ 8 を描画した図を示すが，図からもいくつかのまとまりが結合していることが分かる．図 13(b) にクラスタ 8 内のトピックの配置を示す．

最後に，図 13(c) に媒介中心性を解析した結果を図示する．この結果からも，クラスタ間に媒介中心性の高い頂点が存在している．よって，複数のクラスタに分離した方が評価が高いといえる．他，クラスタ 9，クラスタ 12 とクラスタ 19 についても同様の結果が得られた．

## 6. ま と め

本研究では，タブーサーチを用いたモジュール性による無向グラフのクラスタリングアルゴリズムを提案した．提案アルゴリズムと Newman のアルゴリズムとを比較すると，タブーサーチによるモジュール性を用いたクラスタリング手法の方がより適切にクラスタリングがおこなえることを評価実験により示した．




(a) クラスタ 8　(b) トピック　(c) 媒介中心性

図 13　クラスタ 8 について

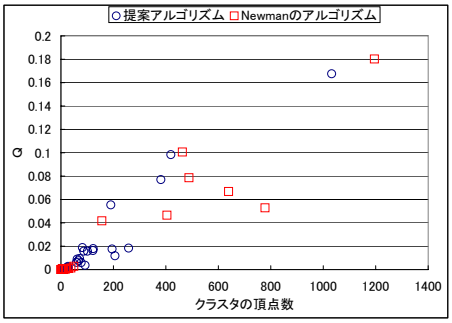



図 9　クラスタの頂点数と Q 値の比較（テストデータ 6）

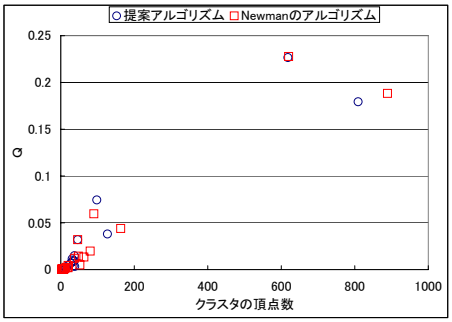



図 12　クラスタの頂点数と Q 値の比較（テストデータ 9）

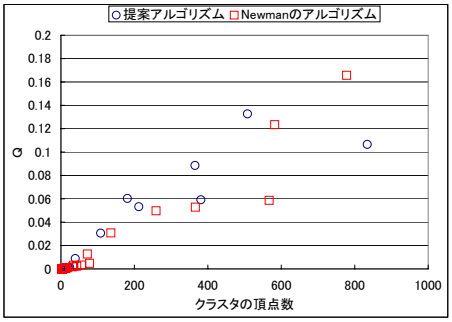



図 10　クラスタの頂点数と Q 値の比較（テストデータ 7）

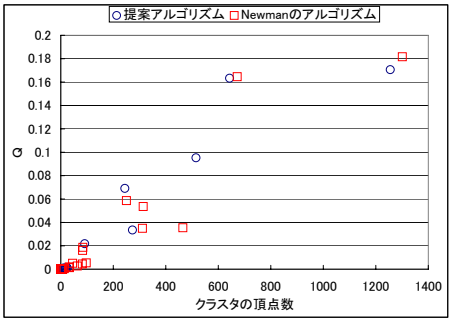



図 11　クラスタの頂点数と Q 値の比較（テストデータ 8）

これからの課題として，まず，NTCIR などのテストコレクションを使った定量的な評価があげられる．テストコレクションを使用して，提案アルゴリズムの有用性を検証していきたい．次に，SA や進化的計算など他のメタヒューリスティック解法の適用があげられる．タブーサーチはメタヒューリスティック解法として強力なアルゴリズムではあるが，進化的計算を適用することでさらに良い準最適解が得られると期待される．

## 謝　辞

本研究の一部は，日本学術振興会・科学研究費補助金（基盤研究（C），（一般），課題番号：17500097），文部科学省・科学研究費補助金（課題番号：18700094）の支援によりおこなわれた．

## 文　献

[1] E. Hartuv and R. Shamir: “A clustering algorithm based on graph connectivity”, Information Processing Letters, **76**, 4–6, pp. 175–181 (2000).

[2] M. Brinkmeier: “Communities in graphs.”, IICS, pp. 20–35 (2003).

[3] U. Brandes, M. Gaertler and D. Wagner: “Experiments on graph clustering algorithms” (2003).

[4] M. E. J. Newman: “Fast algorithm for detecting community structure in networks”, Physical Review E, **69**, p. 066133 (2004).

[5] A. Clauset, M. E. J. Newman and C. Moore: “Finding community structure in very large networks”, Physical Review E, **70**, p. 066111 (2004).

[6] T. H. Cormen, C. E. Leiserson and R. L. Rivest: “Introduction to Algorithms”, MIT Press/McGraw-Hill (1990).

[7] F. Glover and F. Laguna: “Tabu Search”, Kluwer Academic Publishers, Norwell, MA, USA (1997).

[8] M. E. J. Newman: “Modularity and community structure in networks”, PROC.NATL.ACAD.SCI.USA, **103**, p. 8577 (2006).

[9] 湯田聴夫, 小野直亮, 藤原義久：“ソーシャル・ネットワーキング・サービスにおける人的ネットワークの構造”, 情報処理学会論文誌, **47**, 3, pp. 865–874 (2006).

[10] 辻尚, Ｍｄ.Altafm-Amm, 有田正規, 西尾泰和, 真保陽子, 黒川顕, 金谷重彦：“生体ネットワーククラスタの可視化に関する研究”, 情報処理学会研究会報告, **No.2006-BIO-005**, pp. 9–15 (2006).

[11] 安藤潤, 吉井伸一郎：“Www ナビゲーション向けコミュニティ分割手法に関する一考察”, 情報処理学会研究会報告, **No.2006-ICS-142**, pp. 115–122 (2006).

[12] G. Flake, S. Lawrence and C. L. Giles: “Efficient identification of web communities”, Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, pp. 150–160 (2000).

[13] 鶴見敏行, 脇田建：“大規模社会ネットワークからのコミュニティ抽出”, 第 7 回 Web インテリジェンスとインタラクション研究会, pp. 109–114 (2006).

[14] 波平光洋, 名嘉村盛和, 岡崎威生, シバスンタランスハルナン：“複数の最小木を考慮した確率的進化計算による遺伝子データ・クラスタリング”, 情報処理学会研究会報告, **No.2006-BIO-005**, pp. 59–64 (2006).