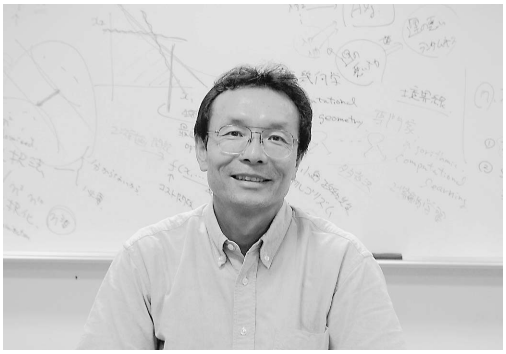
In Laboratory Now

研究室訪問 1

NP困難問題とアルゴリズム

渡辺 治 研究室〜数理・計算科学専攻


渡辺 治 教授

現在、コンピュータは我々の産業の中心に位置している。その性能は現在もなお発達し続けており、より難しい処理も高速に行えるようになっている。しかし、その功績はコンピュータ自身の発展だけによるものではない。コンピュータにどのように処理をさせるかを考えている人たちの努力の賜物でもある。

今回取材に伺った渡辺研究室もその一端を担っている。この記事では、コンピュータに効率よく処理を行わせるために渡辺研究室がどのような工夫を行っているのかを紹介しよう。

効率化とアルゴリズム

皆さんはコンピュータサイエンスという学問をご存知だろうか。コンピュータサイエンスとは、コンピュータ上の処理に関して研究を行う学問である。コンピュータ上の処理には、高度なシミュレーションからワープロやWeb上での処理まで様々なものがある。それらの処理をコンピュータにどのように行わせるかについて論理的に記した手順はアルゴリズムと呼ばれ、渡辺研究室では効率よく問題を解くアルゴリズムについての研究を行っている。

皆さんの中にはコンピュータ自体の性能が上がれば、効率よく問題を解くアルゴリズムについて研究する必要がなくなると思う人がいるかもしれない。確かに、近年コンピュータの性能は格段に向上しており、かつてのコンピュータが時間をかけないと解けなかった問題でも今のコンピュータであれば高速で処理してくれるだろう。しかし、以前に比べてコンピュータの性能も向上したが、扱うべき問題の方もその問題量が格段に増大したことにより難しくなっているのである。

Webの検索がその代表例である。インターネットが普及し始めた頃はまだWeb上の情報量も少なかった。だが、現在ではあらゆる企業や個人が情報発信をしており、その情報量は爆発的に増えている。この原因の一つはコンピュータの性能が上昇したことであり、コンピュータの性能が上昇すればする程その増加に拍車がかかっていくのは明らかである。

また、今までには存在しなかった問題も現れてきている。その代表例としては、ICカードのサービスが挙げられる。ICカードによるサービスは性質上、データを安全かつ高速に処理することが必要であり、これには高度な暗号技術が関わっている。このようなサービスが盛んになった背景には集積回路の技術的な進歩がある。同様な技術の進歩によって今後も新たな問題が生まれてくると予想される。

このように、現在コンピュータが発達しているといっても、コンピュータの性能の向上のみでは現状の問題の難易度の上昇や新たな問題に対応するのは難しいと考えられる。そこで重要となってくるのが、どのようにコンピュータに処理をさせるかであり、渡辺研究室で行われている効率よく問題を解くアルゴリズムの研究である。

では、どのように効率よく問題を解くのかを具体的に見ていくため、最小包含円問題（**図１**）について考えてみる。

この問題を解くアルゴリズムの一つとして次のようなものが考えられる。

アルゴリズムA1

STEP1 . n個の点から3点選び、
3点を囲む最小円を求める
STEP2 . 円が点全体を囲むかどうか調べる
囲んでいればSTEP3へ、
囲んでいなければSTEP1へ
STEP3 . その円が解

このアルゴリズムは簡素であり、プログラミングの手間も少なくて済むという点では悪いアルゴリズムではない。だが、効率のよいアルゴリズムではない。なぜなら、このアルゴリズムを用いた場合、点の数をnとすると問題を解くのにはn^3に比例した時間がかかり、点の数nが10倍になると処理にかかる時間は約1000倍になってしまう。つまり、点の数nの増え方に比べ処理にかかる時間の増え方が大きいので、点の数nが増えた時にすぐに対処できない可能性がある。よって、アルゴリズムA1は効率がよいとはいえない。

なお、このように処理にかかる時間がn^3に比例している場合は、この問題を解くのには$O(n^3)$かかると一般的に表現される。

では、改良した方法ではどれほど効率よく解けるか。そのアルゴリズムについて見ていこう。

アルゴリズムA2

STEP1 . n個の点からランダムにk個の点を選ぶ
STEP2 . k個の点を囲む最小円を求める
（kは小さい値なのでA1を使用）
STEP3 . 円がn個の点全体を囲むか調べる
囲んでいればSTEP6へ
囲んでいなければSTEP4へ
STEP4 . 円の外側の点を選ぶ確率を高くする
STEP5 . STEP1へ
STEP6 . その円が解

このアルゴリズムを用いた場合にかかる時間は$O(n\log n)$と分かっている。つまり、点の数nが10個から100個へと10倍になっても処理にかかる時間は約20倍であり、先程の約1000倍と比べればこちらのアルゴリズムの方がはるかに効率的であることが分かる。

このように、アルゴリズムを工夫することで解くのにかかる時間が大幅に削減される問題は多い。その工夫方法のひとつとしては、今回のようにランダムを用いる手法があり、渡辺研ではランダムを利用したアルゴリズムに関しても数多く研究している。

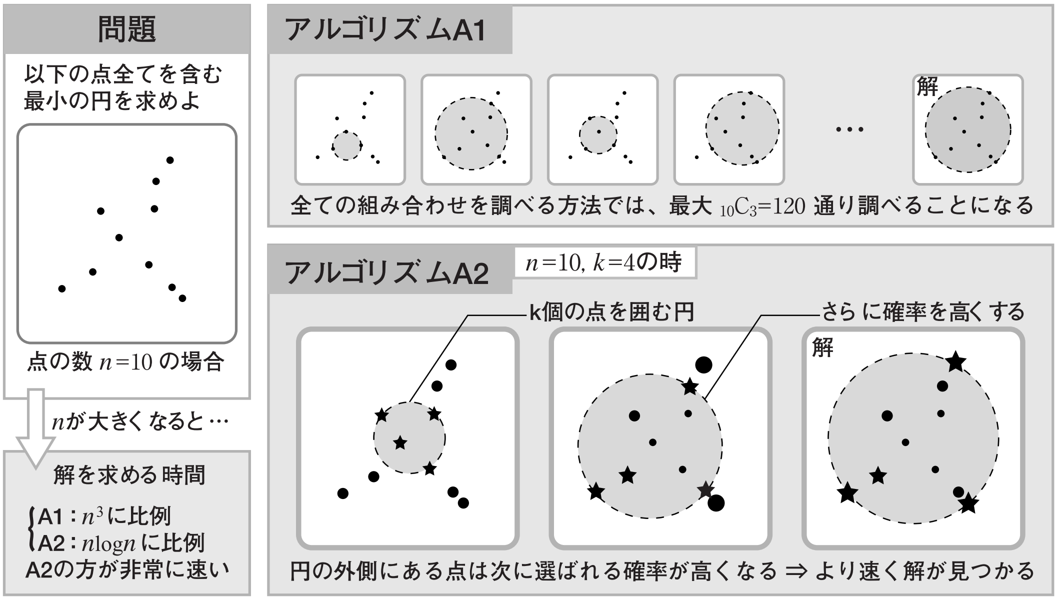


図１　最小包含円問題

NP困難問題とは

だが、世の中にはより難しい問題も多く、効率よく解くアルゴリズムが見つかっていない問題もある。この章では難しい問題の例として三彩色問題（**図2**）について説明していく。

この問題は解候補が正しいのかを確かめるのはたやすいが、解を見つけ出すまでに大変手間がかかる。おおよそどれくらいの手間がかかるのか単純なアルゴリズムで見てみよう。

アルゴリズムA3

STEP1．まだ調べていない塗り方で点を塗る
STEP2．隣り合う点と色を比較する
重なっていれば解でないのでSTEP3へ
重なっていなければSTEP4へ
STEP3．調べてない塗り方がなければSTEP5へ
調べてない塗り方があればSTEP1へ
STEP4．この塗り方が解 → 終了
STEP5．解はなし → 終了

このアルゴリズムを用いた場合、最大で判断すべき点の塗り方は3^n通りあり、問題を解くのにかかる時間は$O(3^n)$である。つまり、問題の点の数が10倍に増えるとかかる時間は3^{10}(=59049)倍にもなるのである。これを見ると、前章の最小包含円問題に比べ桁違いに手間がかかることがよく分かるだろう。

このように解くのにかかる時間、すなわち難易度は問題によって大きく異なり、この違いを利用することにより問題を分類することができる。例えば前の章で紹介した最小包含円問題は、アルゴリズムA1を用いた場合には解を見つけるのに$O(n^k)$かかる（この場合k=3)。このような問題はP問題と呼ばれている。また、NP問題という問題がある。これは解候補が正しいことを確かめるのに$O(n^k)$かかる問題であり、三彩色問題がそれに該当する。なお、P問題と分類される問題はNP問題でもある。なぜなら、P問題は検証を含めた解く時間が$O(n^k)$で済むことが分かっており、検証にそれより多くの時間がかかってしまうことがないからである。ゆえに、最小包含円問題はNP問題でもある。

そして、NP問題よりも難しい、または同等の難しさである問題にNP困難問題と呼ばれるものがあり、三彩色問題もこれに含まれる。渡辺研究室はNP困難問題について数多く研究している。このような問題を実際に解かなければならない場合も多い。その例としてスケジューリング問題と呼ばれるものがあり、スポーツのリーグ戦のスケジュール作りなどがこれに該当する。次章では、具体的に渡辺研究室がどのようなアルゴリズムを用いて難しい問題に取り組んでいるのかについて見ていこう。

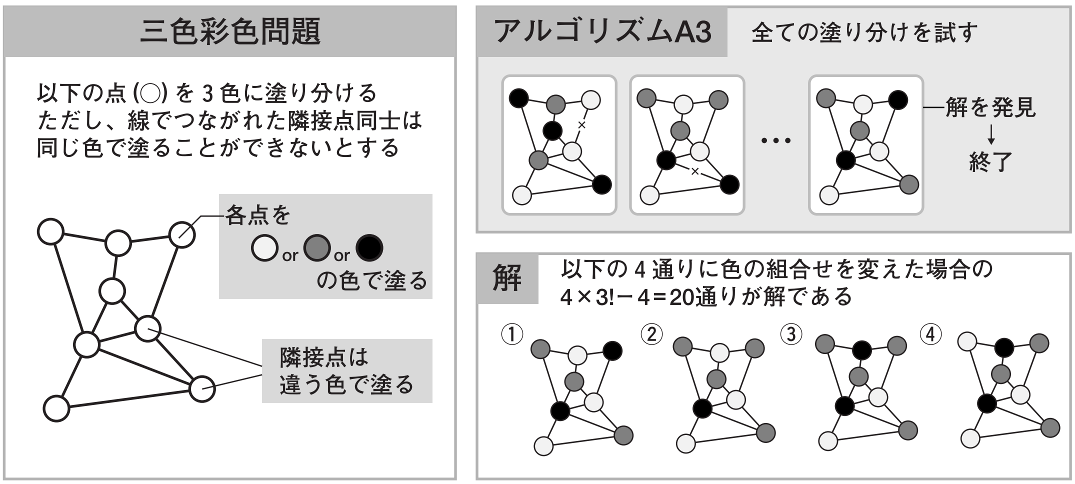


図2　三彩色問題

グラフ分割問題への信念伝播法の応用

前の章で紹介した通り、三彩色問題を代表とするNP困難問題は大変手間のかかる問題であり、全ての場合に対し$O(n^k)$の時間で解くアルゴリズムは見つかってない。だが、全ての場合に対し効率よく解くのは難しくても、ほとんどの場合に対して効率よく解く方法が存在するNP困難問題もある。渡辺研究室ではNP困難問題の一種であるグラフ分割問題を効率よく解く研究を行っている。この章ではその手法について紹介しよう。

まず、**図3**を見てもらいたい。**図3**のような、いくつかの点とそれらを結ぶ辺の集合はグラフと呼ばれており、これを分割する問題がグラフ分割問題である。渡辺研究室ではグラフ分割問題の一種であるMostLikelyPartition(MLP)problemについて研究している。

ーグラフ分割問題ー

- 点は人で、辺で結ばれた点同士は友達同士
- 映画の好みが二つに分かれている
  （例：アクション派とロマン派）
- 誰が何派かはわからない
- 友達同士は映画の好みが同じの場合が多い

↓

映画の好みを予想し、２つにグループ分け

だが、これを解くアルゴリズムの研究をする上でひとつ問題がある。それは、このグラフ分割問題は解の検証を行うのに手間がかかるということである。なぜなら、問題の解候補が見つかったときに、それが最もふさわしいかを検証する必要があるからである。そこで、PlantedSolutionModelという手法が役立つ。先の映画の好みを例を元に説明しよう。

まず、映画の好みを先に決め、問題（グラフ）を作成し、その問題をアルゴリズムを用いて解く。これがPlantedSolutionModelである。この際渡辺研究室ではある工夫を行っている。それは、映画の好みが同じ人が友達である確率pと、映画の好みが異なる人が友達である確率rの差を一定以上にするというものである($p-r \geqq \Omega((\log n)/n)$)。この条件下で作成された問題の解は最初に定めたひとつになることが分かっており、解候補が解であるかをすぐに確実に検証できる。

渡辺研究室では、このPlantedSolutionModelを用いて、どのようなアルゴリズムを用いればグラフ分割問題を効率よく解けるかについての研究を行っている。

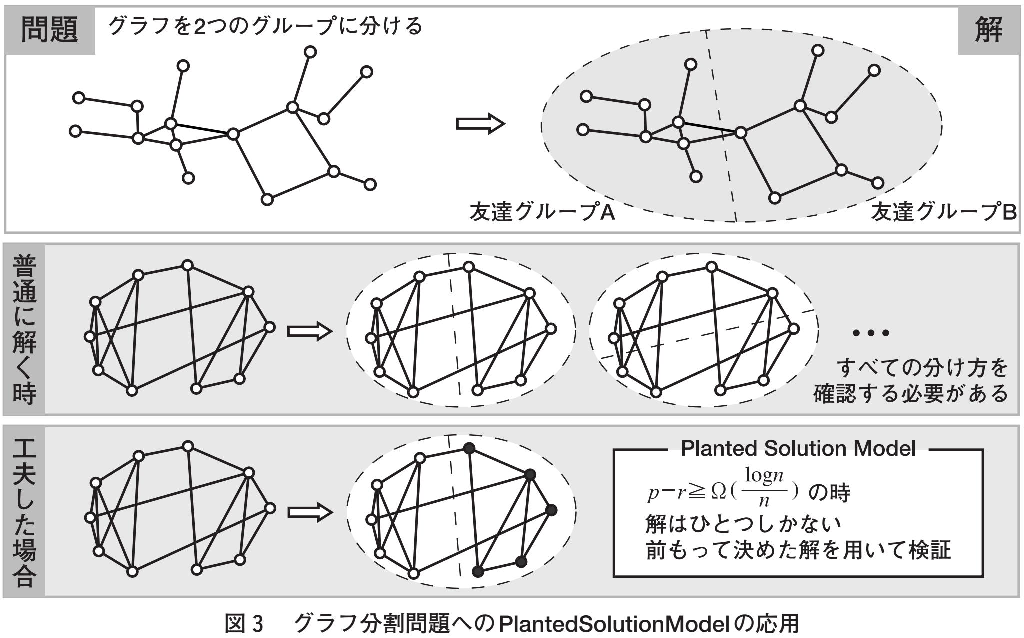


図３　グラフ分割問題へのPlantedSolutionModelの応用

では渡辺研究室はどのようなアルゴリズムを用いてグラフ分割問題を効率よく解こうとしているのか。それは、信念伝播法というアルゴリズムである。それでは、映画の好みの例に則して説明していこう。

まず便宜上、最初に誰か一人の信念の値を＋∞と固定し、他の人の信念の値を0にする。次に、各人が各人に対し同時に信念を送り合う。例えば、アクション派の人は、友達もアクション派である可能性が高いので、友達へ＋の信念を送り、逆に友達でない人へは－の信念を送るのである。これを幾度か繰り返すと映画の好み別に信念の値が＋－で分かれる。信念の値が＋の人がアクション派かロマン派かは分からないが、ここではグループ分けができればよいので問題ない。

次に、これを計算としてどう実現しているのか見ていこう。ここで、信念伝播法の詳細を見ていくにあたり、いくつかの変数を定義する。

n ：対象者の人数

$\{v_1,\cdots,v_n\}$：対象者の集合

b_i：i番目の人の信念（値）

N_i：i番目の人の友人の集合

では、具体的なアルゴリズムをみていこう。

A4 信念伝播法

STEP1 . 各人の信念値を初期設定する

$$b_1=+\infty,\quad b_i=0\ (2\leqq i\leqq n)$$

STEP2 . 各点が各点に信念を送る

各b_iに対して、以下の式を同時に計算

$$b_i=\sum_{v_j\in N_i}h_+\ \mathrm{Th}_+(b_j)+\sum_{v_j\notin N_i}h_-\ \mathrm{Th}_-(b_j)$$

STEP3 . 好み別に分かれるまでSTEP2を実行

STEP4 . 信念値の十一でグループを分けて終了

まず、基準の人を一人決めてその人の信念の値を＋∞に設定する。次のSTEP2において、友達へ送るときの信念は$h_+\mathrm{Th}_+(b_j)$ (>0)であり、友達でない人へ送る場合は$h_-\mathrm{Th}_-(b_j)$ (<0)となっている。そして、これを繰り返し送ることにより最終的にアクション派の信念の値は$b_i>0$となり、これを基にグループ分けができるのである。

このような信念伝播法を用いると、このグラフ分割問題は劇的に速く解を求めることができる。ただ、現在のところはまだp,rなどが限られている条件での適用しか証明されていない。今後の課題としては、複数へ分割する場合への適用や、より広い条件での適用があり、現在渡辺研究室ではそれらについて日夜研究を行っている。

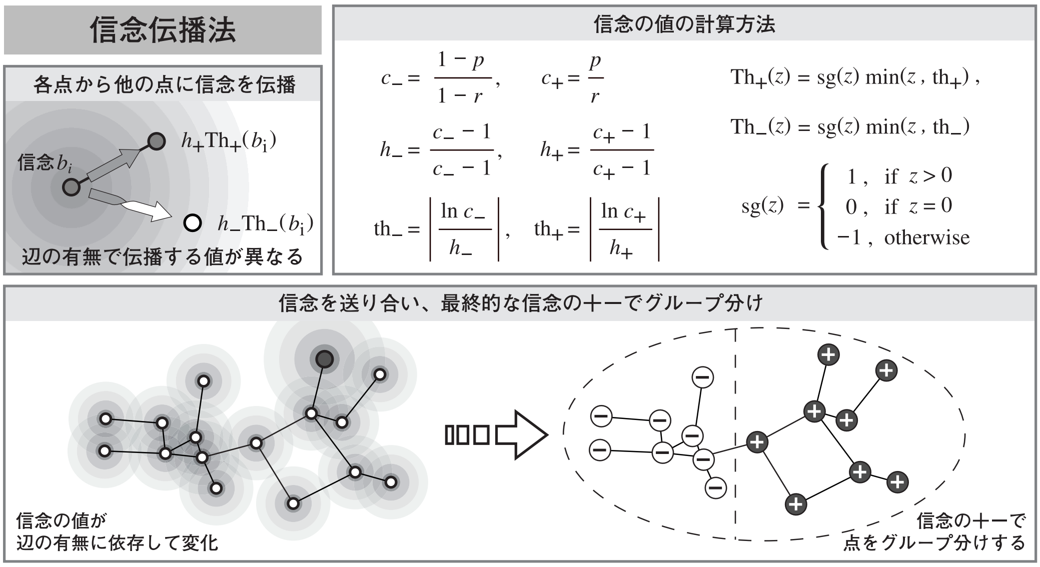


図4　信念伝播法のグラフ分割問題への応用

今回の取材は学べたことが多く、とても充実したものでした。最後になりましたが、お忙しい中度重なる取材に快く応じてくださった渡辺先生に厚くお礼申し上げます。（桂　悠哉）