最大安定集合問題に対する発見的解法のビジュアルソフトウェアの作成

飯塚　仁 (沼田 一道 助教授 , 池辺 淑子 助手)

1. はじめに

組合せ最適化問題とは、有限個の組み合わせの中から最適なものを選ぶ問題で、しらみつぶしに全部の場合を調べれば有限時間で必ず最適解が得られる。組合せ最適化問題にも効率良く解けるものもあるが、残念ながらNP-困難と呼ばれる、計算量的に非常に難しいクラスに属するものが多い。このような問題に対しては実用的な計算時間内にある程度良い解を求めるアルゴリズムが必要である。本研究では組合せ最適化問題の代表的な問題である最大安定集合問題を取り上げ、最近の多くの発見的解法の中でSTABJOIN(参考文献[1])に着目し、ビジュアルに解の改良過程を示すソフトウェアの作成を試みる。STABJOINは多くの場合においてかなり良い解を与えるが、希にうまくいかない場合がある。その原因を解明する際に、解の改良過程をビジュアルに解き示すものがあればそれは役に立つものである。

2. 最大安定集合問題

グラフ(graph)Gは、有限個の頂点(vertex)の集合Vとそれらをつなぐ枝(edge)の集合$E(E \subseteq V \times V)$からなり、$G=(V,E)$と記される。頂点$u,v$を結ぶ枝$e$を$e=(u,v)$と書き、$u$と$v$は$e$の端点といい、$e$は$u$と$v$に接続しているという。また、このとき$u$と$v$は隣接しているという。枝に方向を考えないとき無向グラフ(undirected graph)というが、以下では、$G=(V,E)$を任意の無向グラフとする。

無向グラフ$G=(V,E)$において、**安定集合**(stable set)とは頂点の部分集合SでSのどの２頂点間にも枝が存在しないものである。例えば、図１のグラフにおいて$\{1,3,5\}$や$\{5,6\}$や$\{1\}$などは安定集合である。**最大安定集合問題**とは、要素数が最大の安定集合を求める問題である。

各頂点iに対応した0-1変数x_iを導入して、安定集合Sに対し、$x_i = 0 \ / \ 1 \ (i \notin S \ / \ i \in S)$とすれば$(x_i \mid i \in V)$は$x_i + x_j \leq 1 \ , \ \forall (i,j) \in E$を満たす。よって最大安定集合問題は0-1計画問題として以下のように定式化できる。

$$
\begin{array}{lll}
\max & \sum_{i \in V} x_i & \\
\text{s.t.} & x_i + x_j \leq 1 \ , \ \forall (i,j) \in E, & \cdots\cdots (1) \\
 & x_i \in \{0,1\} \ , \ i \in V &
\end{array}
$$

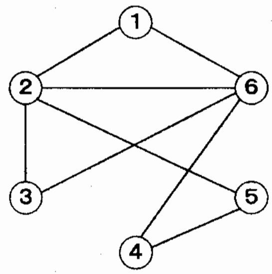

図1: グラフ G

3. 2部グラフにおける最大安定集合の求め方

グラフGが２部グラフであるとは、Gの頂点集合Vが二つの集合V_sとV_tに分割でき、Gのどの枝も一方の端点はV_sに属し、他方の端点はV_tに属するときをいう。

一般のグラフにおいて最大安定集合問題は難しい問題であるが、２部グラフに限定すれば能率良く解ける。この性質を利用してSTABJOINは一般のグラフから２部グラフを見つけ、この２部グラフにおいて最大安定集合を求めるという戦略を採っている。

２部グラフ上での最大安定集合の求め方について記す。まず、最小頂点被覆問題と最大マッチング問題について記述する。

頂点被覆(vertex cover)とは、頂点部分集合Cで、Gのどの枝も少なくとも一方の端点がCに含まれるものである。例えば、図1のグラフにおいて$\{2,4,6\}$や$\{1,2,3,4,6\}$などは頂点被覆である。**最小頂点被覆問題**とは、要素数最小の頂点被覆を求める問題である。定義より、安定集合Sに対して補集合$V \setminus S$はGの頂点被覆である。

マッチング(matching)とは、枝の部分集合Mで、Mのどの枝も端点を共有しないものである。図1のグラフにおいて$\{(1,2),(3,6),(4,5)\}$や$\{(1,6)\}$などはマッチングである。点vがマッチングMに属する枝の端点のとき、Mはvを飽和しているといい、vをM-飽和点、そうでない点vをM-不飽和点という。枝数最大のマッチングを求める問題を**最大マッチング問題**という。

定理 1 (König[3]) 2部グラフにおいて、最大マッチングに含まれる枝の数と最小頂点被覆に含まれる点の個数は等しい。

図2.1はこの関係を示す例である。図2.2は最小頂点被覆の補集合が最大安定集合になっている例である。これらの関係を利用すると2部グラフ上で最大安定集合問題を能率良く解くことができる。つまり、2部グラフにおいて最大マッチング問題を解き、求めた最大マッチングから最小頂点被覆が求まる。この最小頂点被覆の補集合が要素数最大の安定集合である。

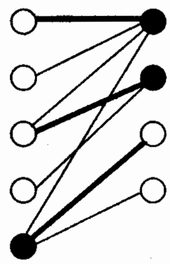

図2.1：最大マッチング(太)と最小頂点被覆(黒丸)

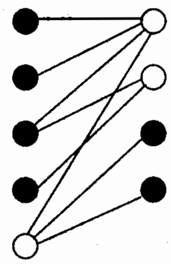

図2.2：最小頂点被覆(白)と最大安定集合(黒)

4. 最大安定集合問題に適用するSTABJOIN

最大安定集合問題に適用する発見的解法のSTABJOINを以下に示す。STABJOINにより解を求める過程を図3に示す。

STABJOINは与えられたグラフにおいて安定集合を二つ求め、その二つの安定集合を頂点とする2部グラフの最大安定集合を求める操作を繰り返して安定集合を大きくしていくアルゴリズムである。第1段階として求める二つの安定集合をS^1, S^2とし、S^1, S^2の和集合から求める最大安定集合をS^3とする。

S^1の初期解を求めるには貪欲法を用いる。貪欲法とは、頂点番号順にその頂点を入れると安定集合になるかどうかを調べて求める方法である。また、S^2を求めるには二つの方法を用いる。第一の方法は貪欲法である。第二の方法は理論的な根拠のある方法で、2部グラフ上で求まる安定集合の要素数が多くなる傾向になるようにS^2を選ぶ方法である。当然、貪欲法より良い解が得られることが多い。具体的に記すと、$S^2 \cup \{v\}$が安定集合、かつS^1に属する頂点と新たに隣接する個数が最小となるような$v \in V \setminus (S^1 \cup S^2)$を選び、$S^2$に加えていく方法である。この操作を$v$が選べるかぎり続ける。

```
アルゴリズム STABJOIN(G).
  G = (V, E) を入力；
begin
  shuffle(V);
  S¹:=greedy(V);
  while 停止条件≠yes do begin
    S²:=choose(V − S¹);
    S³:=join(S¹, S²);
    S¹:= S³;
  end;
  S¹を返す；
end;
```

STABJOIN中のそれぞれの関数の意味を示す。shuffle(·)は頂点番号のランダムな順序づけを表す。greedy(·)は貪欲法を表し、choose(·)はS^2を求める上記の二つの方法のいずれかを表す。join(S^1, S^2)は与えられた安定集合S^1, S^2に対して$S^1 \cup S^2$に含まれる最大の安定集合S^3を求める手続きとする。

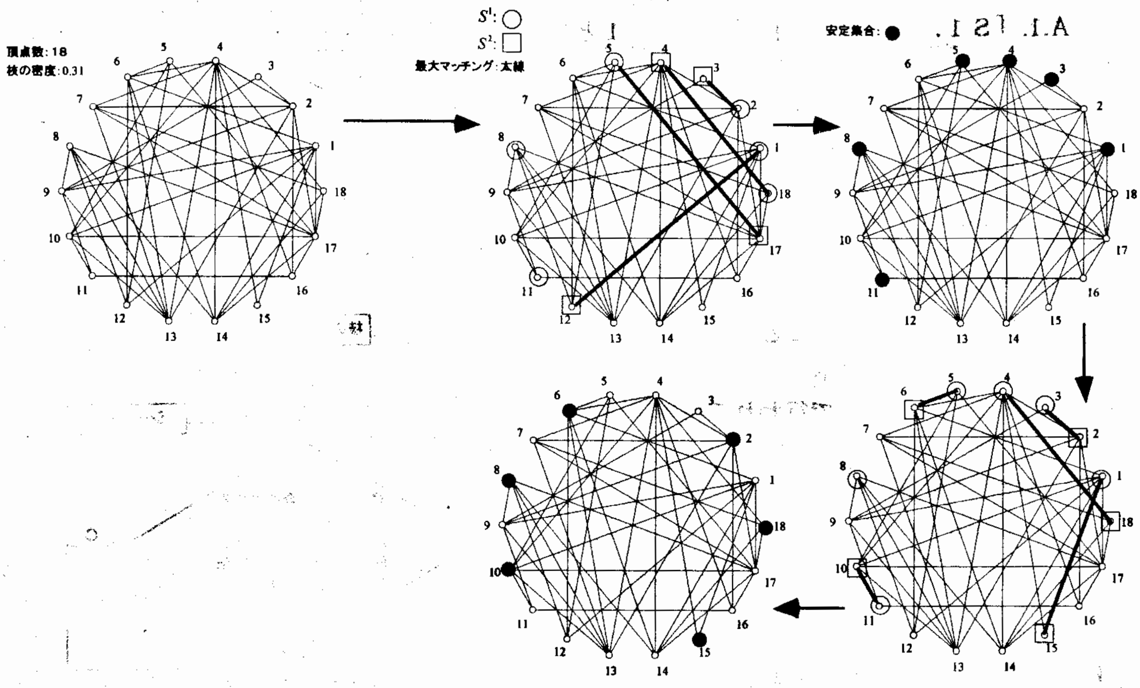

図３：STABJOINにより解を求める過程(S^2を求める方法は貪欲法)

5. 作成したソフトウェアの概要

作成したソフトウェアの仕様を記述し、機能の流れを図５に示す。なお、使用言語ソフトはDelphi Ver2.0J(Borland社)である。

5.1 グラフの入力

「グラフの設定」ボタンを押して、グラフの枝の接続関係をファイルから読み込むか、ランダム発生にするかを選択する。

ａ：ランダム発生

ランダム発生を選択した場合には、まず頂点数を入力する。次に、枝の密度（任意の２頂点間に枝を張る確率）pを入力する$(0 \leq p \leq 1)$。

ｂ：ファイルからグラフを読み込む

この項目が選択された場合、入力ファイルを選択するダイアログボックスが表示される。

<グラフの入力形式>

頂点iと$j(i \neq j)$との関係によって次のように第(i,j)成分γ_{ij}を定める隣接行列(γ_{ij})を使用する。

$$\gamma_{ij} = \begin{cases} 1 & \text{頂点}i\text{と}j\text{の間に枝が存在} \\ 0 & \text{その他} \end{cases}$$

隣接行列は対称行列なので、入力ファイルは頂点数nと隣接行列の下三角成分を使って図４のような形式にする。

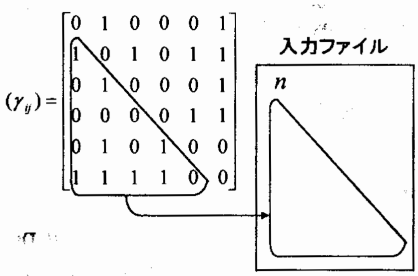

図4: 図1のグラフの隣接行列と入力ファイル

5.2 最大安定集合の表示

最大安定集合を求める過程の表示方法を３通り用意した。それぞれの方法で表示させる際に、最初にS^2を求める方法を選択する。

Ａ：ボタンを押して表示

A.1.「Ｓ１，Ｓ２の生成」ボタン：Ｓ１とＳ２がそれぞれ黄色と青色で表示される。

A.2.「最大マッチング」ボタン：S^1，S^2を頂点とする２部グラフの最大マッチングを赤色で表示。

A.3.「安定集合」ボタン：２部グラフにおける最大安定集合を緑色で表示する。

再び、「Ｓ１，Ｓ２の生成」ボタンを押すと、求めた安定集合をS^1とし、S^2を求めて同様に表示する。この３つのボタンで上記のことを繰り返す。

Ｂ：一定時間ごとに表示

上記の３つのボタンで行われる処理が一定時間ごとに表示される。表示時間はメインメニューの“設定”で自由に変えられる。

Ｃ：直ぐに解を表示

「解」ボタンを押すとすぐに解が表示される。この解は、反復において安定集合の大きさがｋ回変わらなければ停止したときの安定集合である。このｋも“設定”で自由に変えることができる。

5.3 その他の機能

- 描画：設定したグラフの描画。
- シャッフル：頂点番号をランダムにする。
- ファイルに保存：グラフをファイルに保存する。(入力ファイルと同様の形式)

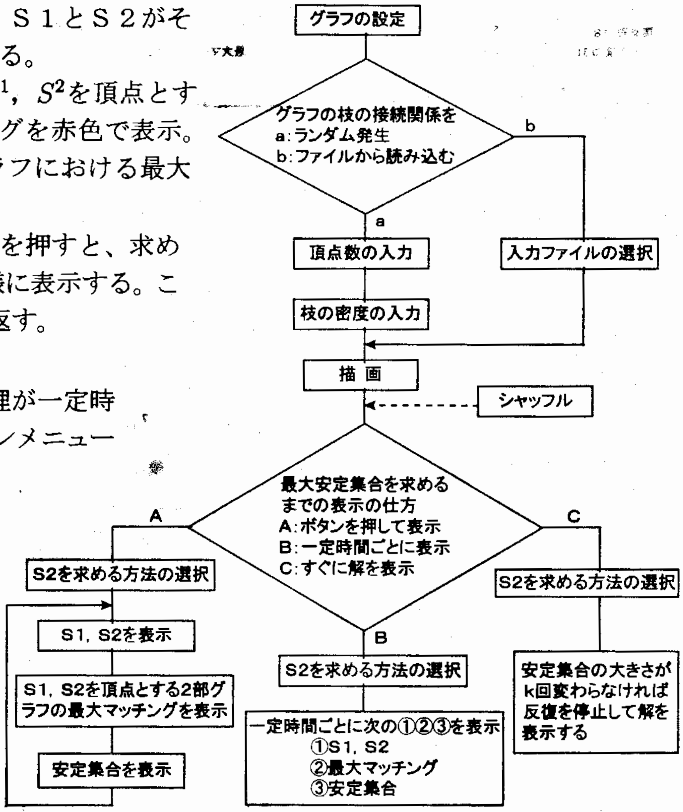

図5:ソフトウェアの機能 の流れ

6. おわりに

S^2を求める方法として、頂点番号の昇順、降順を交互に繰り返して求める方法と、ランダムに選ぶ方法を試みたが、両者とも第一の方法より良い解が得られるか同程度の傾向にあり、理論的な根拠のある第二の方法を用いた解には及ばなかった。

最大安定集合をより求めやすくするためには、見つけた２部グラフから求める安定集合に多様性を持たせる、つまり、反復において再び同じ２部グラフが求められたとしても、そこで前回とは異なった安定集合を求めるようにすれば、解が変化しやすくなり最大安定集合に辿り着きやすくなると考えられる。

描画における問題点としては、枝の数が非常に多くなると殆ど見分けがつかないことである。これを解決するためには、枝の交差が少ない奇麗なグラフを描画する描画アルゴリズムを採用する必要がある。

参考文献

[1] Y.T.Ikebe , A.Tamura , “Ideal polytopes and face structures of some Combinatorial optimization problems,” Mathematical Programming 71, pp.11-13 (1994)

[2] 田村 明久, “多面体的組合せ論—最大クリーク問題を例として”, ＲＡＭＰシンポジウム論文集 (1994)

[3] Ulrich Derigs, “Programming in Networks and Graphs,” ,Lecture Notes in Economics and Mathematical Systems 300 , Springer-Verlag, pp.118-133 (1988)