

グラフのフィードバック頂点集合問題に対する局所探索法

北野 将梧^{1,a)}

概要: フィードバック頂点集合問題は、単純無向グラフを対象とした組合せ最適化問題である。様々な解法が知られており、実用的にはメタヒューリスティック解法が用いられる。その一方で、近年、FPT 厳密アルゴリズムの目覚ましい計算量の改善もなされてきている。本研究では、既存のメタヒューリスティック解法において、各頂点のフィードバック頂点集合に含まれやすさを考慮することによって、改善する手法を提案する。提案手法を、既存のメタヒューリスティック解法及び FPT 解法と比較する実験を行った。実験の結果、提案する改善されたメタヒューリスティック解法は、既存のメタヒューリスティック手法によって得られたものと比較して、実行時間と解の良さの両面で優れていることが分かった。また、FPT 解法と比較すると、小さい入力に対しては、FPT アルゴリズムが実行時間で大幅に優れていたが、そのときメタヒューリスティック解法も高い確率で大域的最適解を発見できていた。大きい入力に対しては、メタヒューリスティック解法によってのみ、解を得ることができた。

キーワード: フィードバック頂点集合問題, メタヒューリスティクス, 焼き鈍し法

1. はじめに

1.1 準備

頂点集合 V , 辺集合 E からなる単純無向グラフ G を $G = (V, E)$ と表す。 $v \in V$ の次数を $\deg_G(v)$ と表す。 $V' \subseteq V$ による G の誘導部分グラフを $G[V']$ と表す。 G の頂点部分集合 F であって、 $G[V \setminus F]$ が閉路を含まない (森になる) ようなものをフィードバック頂点集合 (Feedback Vertex Set; FVS) という。 さらに、フィードバック頂点集合問題 (Feedback Vertex Set problem; FVS 問題) を次のように定義する。

フィードバック頂点集合問題

入力: 単純無向グラフ $G = (V, E)$.

出力: V の部分集合 F .

制約: F は G の FVS であり、要素数が最小である。

図 1 に FVS 問題の入出力例を示す。 図中の破線で描かれた頂点と辺は削除されていることを表す。 図 1a のグラフから頂点 b, d, h を削除したグラフには閉路 gef があるため、 $\{b, d, h\}$ は FVS ではない。 一方、頂点 d, f を削除したグラフ (図 1b) には閉路がないため、 $\{d, f\}$ は FVS であり、要素数 1 の FVS は存在しないため最小 FVS でもある。

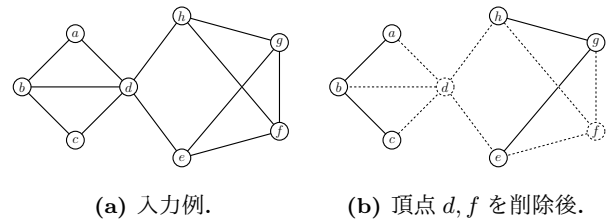


図 1: FVS 問題の入力と、頂点集合を削除後のグラフの例。

1.2 目的

FVS 問題には以下の既存手法が有効であることが知られている。

- 次章で紹介する Qin らの焼き鈍し法 [1].
- 岩田・今西による FPT^{*1} 厳密解法 [2,3].

Qin らの手法は次章で紹介する。岩田・今西らの手法は、小さなグラフ ($|V| \leq 50$ 程度) に対しては非常に高速に動作する。しかし、 $|V| \geq 100$ 程度になると現実的な時間内に終了しなくなるため、そのような大きさのグラフに対しては Qin らによる焼き鈍し法を用いることになる。本研究では、Qin らの手法に工夫を加えることで、厳密解法によって解を求めることができないようなグラフに対して、既存手法より高速に、より良い解を得ることを目指す。

¹ 電気通信大学
The University of Electro-Communications, Chofu, Tokyo
182-8585, Japan

a) sgktn28@uec.ac.jp

*1 Fixed Parameter Tractable; 固定パラメータ容易。

アルゴリズム 1 基本的な還元.

Input: $G = (V, E)$: 単無向グラフ
Output: $\langle G', F \rangle$: 還元されたグラフと, 削除された頂点からなる FVS の部分集合の組

```

1: function BASICREDUCTION( $G$ )
2:    $F \leftarrow \{\}$ 
3:    $G' \leftarrow G$ 
4:   repeat
5:     if  $G'$  に葉または孤立点  $v$  が存在する then
6:        $G'$  から  $v$  を削除する
7:     end if
8:     if  $G'$  に次数 2 の頂点  $v$  が存在する then
9:        $G'$  から  $v$  を削除する
10:       $v$  と隣接していた 2 頂点間に辺を追加する
11:    end if
12:    if  $G'$  に自己ループ辺  $vv$  が存在する then
13:       $G'$  から  $v$  を削除する
14:       $F \leftarrow F \cup \{v\}$ 
15:    end if
16:  until 上のいずれかの if 文の条件が真となった
17:  return  $\langle G', F \rangle$ 
18: end function

```

2. 既存手法

2.1 基本的な還元手法

最初に, FVS 問題に対する基本的な還元手法をアルゴリズム 1 に示す. このアルゴリズムは, 最小 FVS に明らかに含む必要のない頂点と, 含むべき頂点を取り除く操作を繰り返すものである. 以降で説明するどの手法においても前処理として行い, その後に適用される主たるアルゴリズムで得られた解と F を合併したものを最終的な解として出力する.

2.2 焼き鈍し法

焼き鈍し法 (*simulated annealing method*) [4] とは, メタヒューリスティック・アルゴリズムの 1 つである. FVS 問題に対しては, Qin らによる焼き鈍し法が現実的な時間内で最も良い解を出力することが知られている. F を V の部分集合, $S = V \setminus F$ とおく. このとき, 次の定理が成り立つ.

定理 1. 次の 2 つは同値である.

- (1) F は G の FVS である.
- (2) S には次の条件を満たす全順序 $<$ が存在する. すなわち, 任意の $v \in S$ に対し $w \in S \cap N_G(v)$ かつ $w < v$ であるものは高々 1 つである.

証明. (1) \Rightarrow (2) 条件を満たす全順序の構築法を示す. S は G の部分森を誘導する. 列 L を空に初期化する. $G[S]$ の各連結成分に対し順番に, 任意の頂点を開始点として幅優先探索を行う. その際, 頂点を訪問した順に L の末尾に追加していく. 幅優先探索木は根付き木なので, 各頂点 v の親はただ一つに定まるか, 存在しないかのいずれかであ

る. v の親が存在するとき, $S \cap N_G(v)$ の中でそのみが L において v より手前に位置する. L 上の順序によって全順序を定めれば (2) を満たす.

(2) \Rightarrow (1) 頂点 $v \in S$ に対し, $w \in S \cap N_G(v)$ を満たす w は高々 1 つである. よって w が存在する場合は w を v の親とし, 存在しない場合は v を根とする根付き木 (連結成分が複数あれば森) を構築することができる. したがって $F = V \setminus S$ は FVS である. \square

Qin らの焼き鈍し法では, S の頂点を 1 つずつ含み, 順序が上記の $<$ に従う列 L を状態とする. 遷移では, まず u を F から一様ランダムに選択する. 次に, u と隣接する頂点が L にいくつ含まれるか調べる. (a) もし 0 個ならば, u を L の先頭に挿入する. (b) もし 1 個ならば, u をその頂点の直後に挿入する. (c) もし 2 個以上ならば, (b) と同様に挿入した後, u と隣接する頂点のうち L で最も手前のもの以外を削除する. このような操作によって得た新たな列を L の近傍とし, L' とおく. 温度関数に依存した確率で, L から L' への置き換えを繰り返し行い, $|L|$ を大きく ($|F|$ を小さく) していく. L^* を開始以降に得た最長の L , $N_{\max, \text{fail}}$ を大きな自然数の定数とおく. L への頂点の追加を繰り返す中で, L^* の更新を連続して $N_{\max, \text{fail}}$ 回失敗したとき, $V \setminus L^*$ を出力して終了する.

3. 提案手法

提案する焼き鈍し法をアルゴリズム 2 に示す. BIASED-SELECT (アルゴリズム 3) については次の段落で説明する. ADD は, 列 L に対して頂点 u を追加し, 不要な頂点を削除する補助関数である. EVALADD は ADD を実行した際の $|L|$ の変化量を評価する補助関数である.

Qin らの手法では, 近傍を生成する際に挿入する頂点を一様ランダムに選択していた. 提案手法では頂点の重み関数 $w : V \rightarrow \mathbb{R}$ を導入し, それによる評価値が大きい頂点ほど選ばれやすいように, 確率分布に偏りを与える. w は $w(v) = c_1 w_1(v) + c_2 w_2(v)$ と定義する. c_1, c_2 は 2 つの重み関数の影響の強さを調整する正の実数である. 近傍を選択する際の際に, w が大きい頂点ほど高い確率で選択されるようにするのが BIASEDSELECT の役割である.

3.1 頂点の次数を反映する重み付け

w_1 は頂点の次数を反映する重み付けであり, $w_1(v) = \deg_G(v)/|V|$ と定義する. $e \in E$ を一様ランダムに選択した後, e の端点を一様ランダムに (各 1/2 の確率で) 選択する.

この重み付けの根拠を示す. アルゴリズム 1 による還元を施した後の $G = (V, E)$ に対し次の補題が成り立つ.

補題 1. F を G の任意の FVS とし, $S = V \setminus F$ とおく. また, $E_F = \{uv \mid u, v \in F\}$, $E_S = \{uv \mid u, v \in S\}$,

$E_{F,S} = \{uv \mid u \in F, v \in S\}$ とおく. このとき, $|E_S| < |E_{F,S}|$ が成り立つ.

証明. $G[S]$ は森なので $|E_S| < |S|$ が成り立つ. また, アルゴリズム 1 を施した後のグラフでは任意の $v \in V$ に対し $\deg_G(v) \geq 3$ なので,

$$\begin{aligned} 3|S| &\leq \sum_{v \in S} \deg_G(v) \\ &= |E_{F,S}| + 2|E_S| \\ &< |E_{F,S}| + 2|S|. \end{aligned}$$

したがって $|S| < |E_{F,S}|$ が成り立つ. \square

この補題から次の定理が導かれる.

定理 2. 任意の FVS F を考える. E から一様ランダムに $e \in E$ を選んだときに, e のいずれかの端点が F に含まれる確率は $1/2$ 以上である.

証明. $e \in E$ のいずれの端点も F に含まれない確率は,

$$\frac{|E_S|}{|E_F| + |E_{F,S}| + |E_S|} < \frac{|E_S|}{|E_F| + 2|E_S|} \leq \frac{1}{2}.$$

したがって, e のいずれかの端点が F に含まれる確率は $1/2$ 以上である. \square

定理 2 より, 次の系が直ちに導かれる.

系 1. 任意の FVS F を考える. E から一様ランダムに $e \in E$ を選んだ後, e のいずれかの端点を一様ランダムに (それぞれ $1/2$ の確率で) 選ぶと, それが F に含まれる確率は $1/4$ 以上である.

系 1 は, 定理 2 の方法によって任意の $v \in F$ が選ばれる確率は $\sum_{w \in N_G(v)} 1/4m$ 以上であると主張する. これは $\deg_G(v)/4m$ に等しい. つまり, 定理 2 のような頂点の選び方による確率分布と, 選ばれる確率が各頂点の度数に比例するような確率分布は等価である. これは「度数が高いほど FVS に含まれやすい」という直感の裏付けを与える.

3.2 簡易的な焼き鈍し法の結果を反映する重み付け

w_2 は事前に行った簡易的な焼き鈍し法の結果を反映する重み付けである. Qin らの焼き鈍し法は, 定数パラメータを調整することで, 高い温度から生成された乱雑な解から高速に局所解へ収束し, 局所解からの抜け出しを積極的に試みないような振る舞いをさせることができる. これを異なる乱数のシードで 50 回実行する. このようにして得られた 50 個の FVS のうち, 各 $v \in V$ を含むものの個数を i_v とし, $w_2(v) = i_v/50$ と定義する. 簡易的な焼き鈍しを行うことで, 各頂点がどの程度 FVS に含まれやすいかという傾向を掴む.

4. 数値実験

Qin らの焼き鈍し法, 提案する焼き鈍し法, 岩田・今西の FPT 厳密アルゴリズムを比較する実験を行う.

アルゴリズム 2 提案する焼き鈍し法.

Input: $G = (V, E)$: グラフ, $T_0, \alpha, N_{\max_fail}$: 定数

Output: FVS 問題の局所解

```

1: function SIMULATEDANNEALING( $G$ )
2:    $L_1 \leftarrow [v_1]$ 
3:    $L^* \leftarrow L_1$ 
4:   for  $k = 1, 2, \dots$  do
5:      $u \leftarrow \text{BIASEDSELECT}(V, L_k, w, k)$ 
6:      $r \leftarrow [0, 1]$  の一様ランダムな実数
7:      $\Delta = \text{EVALADD}(G, L_k, u)$ 
8:      $T_k \leftarrow \alpha T_{k-1}$ 
9:     if  $\Delta \geq 0$  または  $r < \exp(\Delta/T_k)$  then
10:       $L_{k+1} \leftarrow \text{ADD}(G, L_k, u)$ 
11:     else
12:       $L_{k+1} \leftarrow L_k$ 
13:     end if
14:     if  $|L^*| < |L_{k+1}|$  then
15:       $L^* \leftarrow L_{k+1}$ 
16:       $n_{\text{fail}} \leftarrow 0$ 
17:     else
18:       $n_{\text{fail}} \leftarrow n_{\text{fail}} + 1$ 
19:     end if
20:     if  $n_{\text{fail}} = N_{\max\_fail}$  then
21:      return  $V \setminus L^*$ 
22:     end if
23:   end for
24: end function

```

アルゴリズム 3 追加する頂点の選択アルゴリズム.

Input: V : 頂点集合, L : V の部分集合, w : 頂点の重み関数, k : ループカウンタ

Output: $u \in V \setminus L$ から偏った乱数によって選択した 1 つの頂点

```

1: function BIASEDSELECT( $V, L, w, k$ )
2:   if  $|V \setminus L| \geq 3$  かつ  $k \not\equiv 0 \pmod{5}$  then
3:      $u_1, u_2, u_3 \leftarrow V \setminus L$  から一様ランダムに 3 つ選択
4:      $u \leftarrow u_1, u_2, u_3$  のうち  $w(u_i)$  が最大のもの
5:   else
6:      $u \leftarrow V \setminus L$  から一様ランダムに頂点を 1 つ選択
7:   end if
8:   return  $u$ 
9: end function

```

4.1 入力生成方法

入力は辺確率一定のランダムグラフとランダム正則グラフを用いる. 辺確率一定ランダムグラフは, 頂点数 n は 50, 100, 150, 200 から, 辺確率 p は 0.1, 0.2, 0.3, 0.4, 0.5 から選び, 全ての n, p の組み合わせで, 任意の 2 頂点間に辺が存在する確率が p であるようなランダムグラフを生成する. ランダム正則グラフは, 頂点数 n は 50, 100, 150, 200 から, 辺確率 d は $0.1n, 0.2n, 0.3n, 0.4n, 0.5n$ から選び, 全ての n, d の組み合わせで, 各頂点の度数が d であるようなランダムグラフを生成する.

4.2 結果の比較方法

2 つの生成方法によって 8 種類の異なる乱数シードから入力を生成し, 3 つの手法を計算機上で実行する. そして,

得られた FVS の要素数と実行時間の平均と標準偏差を計算する。

提案手法には $T_0, \alpha, N_{\max.\text{fail}}$, そして c_1, c_2 という 5 つのパラメータがあった。それぞれ、焼き鈍し法の初期温度、冷却速度、最良解の更新の連続した失敗を許容する回数、重み付けにおける影響の強さを決める定数である。そのうち $N_{\max.\text{fail}}$ と c_1 は $N_{\max.\text{fail}} = 10^9$, $c_1 = 1$ に固定する。このような $N_{\max.\text{fail}}$ の値は、入力されるグラフの大きさに対して十分大きな値である。また、 c_1 と c_2 はそれらの比のみが重要であるから、 c_1 は固定してよい。残る T_0, α, c_2 はグリッドサーチによって最適な（最も要素数の小さい解を出力する）ものを求め、それによる結果を採用する。つまり、 T_0 は 0 から 1000 までの 100 刻みの値、 α は $1 - 10^{-x}$ と表現したときの x を 7 から 10 まで 0.3 刻みの値、 c_2 は 0 から 3 まで 0.3 刻みの値の全てに対して焼き鈍し法のアルゴリズムを実行し、最適な組み合わせを結果として用いる。

4.3 実験環境

各入力と手法の組に対して、実行時間は最大 120 分とする。Qin らの手法と提案手法は、C++14 (GCC 4.9.3) によって本稿の著者が実装した。岩田・今西の手法は Java8 (JDK 8u74) によって本人らが実装したもの [5] を用いた。どちらも各言語の標準的な作法による実装である。つまり、アーキテクチャに依存した命令の使用や、プログラムの可読性を犠牲にするような、極度の高速化は行わない。使用した計算機の OS は Linux (Red Hat Enterprise Linux Server release 6.9), プロセッサは Intel Xeon CPU E5-4640 である。

5. 実験の結果と考察

5.1 結果

解の大きさと実行時間に関する実験の結果を図 2 に示す。また、最適なパラメータに関する実験の結果を表 1 に示す。

時間は全て基本的な還元を行うのにかかった時間を含む。図中の赤色・● は岩田・今西らの手法による結果、緑色・× は提案手法、青色・▲ は Qin らの手法による結果を表す。プロットは異なる乱数シードで生成した入力に対する結果の平均であり、エラーバーは標準偏差を平均に加減算した値である。岩田・今西の手法の結果における表示の欠落は、120 分以内に結果が得ることができず、探索が打ち切られたことを表す。各 n, p または n, d の組み合わせで異なるシードを用いて作成した 8 種類の入力を用いたが、全て時間内に探索を終了したか、全て打ち切られたかのいずれかであった。提案手法の実行時間には w_2 を求めるのにかかった時間も含まれている。ただし、実験に用いた範囲の大きさのグラフにおいては高々数秒という小さいオーダーであった。

5.2 考察

5.2.1 厳密解法との比較

$n = 50, 100$ では厳密解法で制限時間内に解が得られたが、要素数が提案手法によるものと一致している。ゆえに、提案手法も大域的最適解を発見できたと言える。それにかかった実行時間は、 $n = 50$ では厳密解法が大幅に優れているが、 $n = 100$ では、提案手法の方が高速であった。また、厳密解法が時間内に解を発見できなかった入力に対しても、提案手法は n_{fail} が $N_{\max.\text{fail}}$ に達した時点の暫定的な最良解を時間内に出力することができた。しかし、その解が大域的最適解と比較してどれほど大きいものであるかは不明である。さらに、 $n = 100, p = 0.1, 0.2$ では、岩田・今西の手法は最適解を発見できたが、要した時間の標準偏差が非常に大きい。すなわち、同じパラメータで生成したランダムな入力でも、シードによって実行時間に大きな差が生まれている。一方、Qin らの手法と提案手法では、厳密解法と比較して小さい偏差に収まっており、実行時間が非常に長くなることも無かった。

5.2.2 Qin らの手法との比較

提案手法と Qin らの手法を、得られた解の要素数によって比較すると、平均で見れば、提案手法が Qin らの手法よりも大きい解を出力することは無かった。実行時間に関しては、提案手法の方が短い入力が多かったが、Qin らの手法の方が短いことも無視できない個数の入力で確認された。これは、暫定的な解を更新する回数が多くなり、さらに、重み関数 w_1 の効果により次数が高い頂点を選ばれやすくなっているため、 L への頂点の追加の実行に必要な時間が増加したからであると考えられる。実際に、そのような入力を与えた際に L に頂点を追加する処理に要する時間を、プロファイリングツール GPROF を用いて測定した。その結果、プログラム全体の実行時間のうち Qin らの手法では 30 % 程度、提案手法では 45 % 程度であり、確かに提案手法の方が大きかった。

5.2.3 提案手法の最適なパラメータ

表 1 に示すように、入力を様々に変化させた際も、 T_0 は 400 から 550 の範囲、 x は 7.0 から 8.0 の範囲が最適であり、特に何らかの値との強い相関は見られなかった。一方で、 c_2 は辺数が多いほど大きい方が良いという傾向が見られた。 c_2 が大きいということは、近傍の選択で簡易的な焼き鈍し法の結果をより強く、頂点の次数をより弱く反映させるということであるが、なぜそれが良い影響を及ぼすかについては不明である。

6. おわりに

既存の FVS 問題に対する焼き鈍し法において、状態の近傍を得るために頂点を選択する際の確率分布に偏りを与えることによって、出力を改善するアルゴリズムを提案した。具体的には、頂点の次数を反映させる重み付けと、事

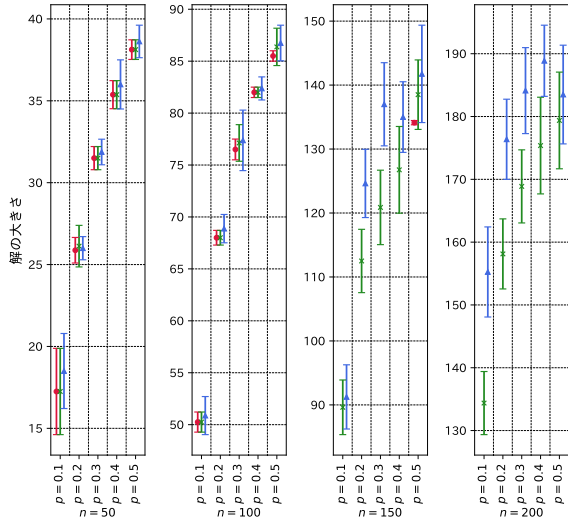
前の簡易的な焼き鈍し法の結果を反映させる重み付けによって、確率分布に偏りを与えた。それによって、FPT 厳密解法では解が得られないような大きさの入力に対して、既存のメタヒューリスティクス手法より良い解を、より短い時間で得ることを目指した。

また、FPT 解法、既存の焼き鈍し法、そして提案する焼き鈍し法を比較する実験を行った。その結果、頂点数が小さい入力に対しては FPT 解法と提案手法はともに最適解を得られたが、実行時間は FPT 解法の方が短かった。一方、頂点数が大きい入力に対しては、FPT 解法では現実的な時間内に解を得ることができず、提案手法では解を求めることができた。提案手法と既存の焼き鈍し法を解の要素数で比較すると、全ての入力の作成方法において平均的に提案手法の方が優れていた。実行時間でも、提案手法の方が短い時間で終了することの方が、全ての入力においてではないが多かった。既存手法の方が短い実行時間で終了するような入力では、提案手法のボトルネックである状態遷移にかかる時間が増加していた。提案手法には、焼き鈍し法の初期温度と冷却速度を決める定数がパラメータとして存在する。それらの適した値は入力の大きさに依存していなかった。辺が密に存在するようなグラフでは頂点の次数よりも事前の簡易的な焼き鈍し法の結果を強く反映させる方がより良い解を出力した。

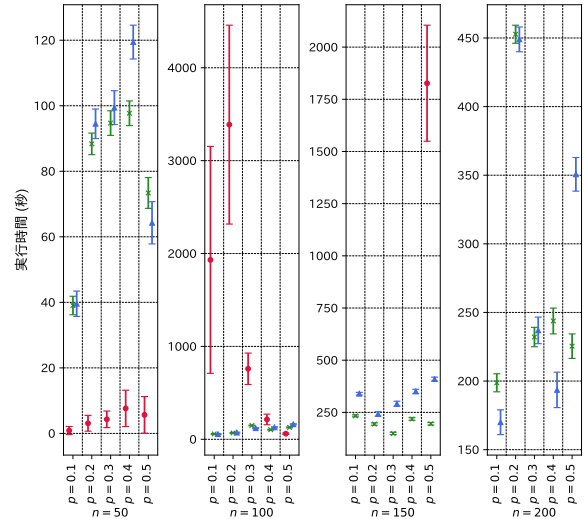
謝辞 本研究を進めるにあたり、ご指導を頂いた主任指導教員の岡本吉央先生、そして日常の議論を通じて多くの示唆を頂いた研究室の皆様へ感謝致します。また、数値実験を行うに際し大きな助けとなった電気通信大学の教育系サーバー sol の管理を担っている皆様にも感謝致します。

参考文献

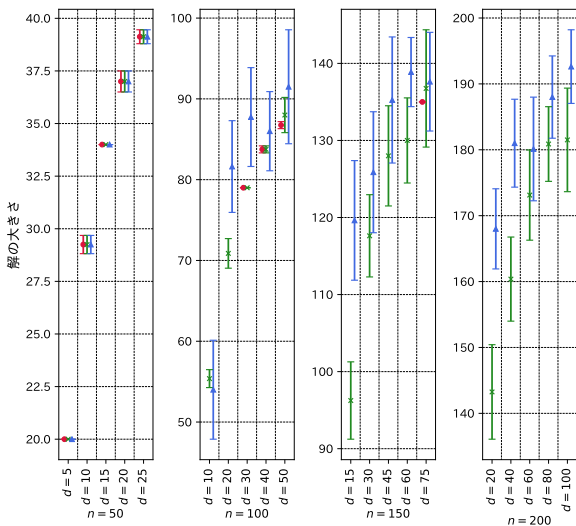
- [1] Qin, S.-M. and Zhou, H.-J.: Solving the undirected feedback vertex set problem by local search, *The European Physical Journal B - Condensed Matter and Complex Systems*, Vol. 87, No. 11 (2014-11-01).
- [2] Iwata, Y.: Linear-time Kernelization for Feedback Vertex Set, *CoRR*, Vol. abs/1608.01463 (online), available from (<http://arxiv.org/abs/1608.01463>) (2016).
- [3] Iwata, Y., Wahlström, M. and Yoshida, Y.: Half-integrality, LP-branching, and FPT Algorithms, *SIAM J. Comput.*, Vol. 45, No. 4, pp. 1377–1411 (online), DOI: 10.1137/140962838 (2016).
- [4] 久保幹雄, Pedroso, J. P.: メタヒューリスティクスの数理論, 共立出版 (2009).
- [5] Iwata, Y.: wata-orz/fvs: feedback vertex set solver, <https://github.com/wata-orz/fvs/>. version fda5514, 2016-08-10T02:25:21+09:00.



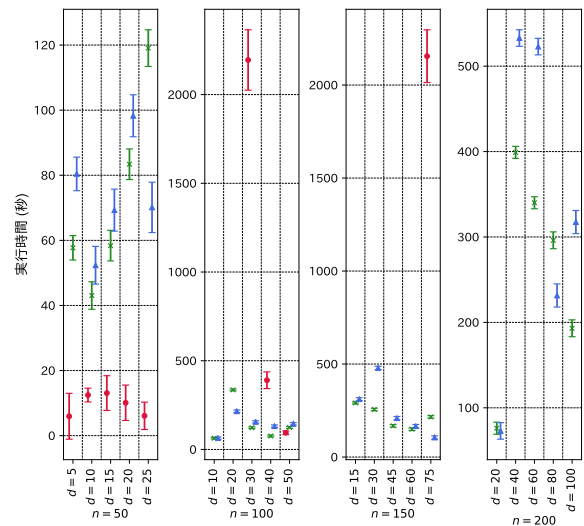
(a) 辺確率一定ランダムグラフで得られた解の大きさ.



(b) 辺確率一定ランダムグラフで解を得るのにかった時間.



(c) ランダム正則グラフで得られた解の大きさ.



(d) ランダム正則グラフで解を得るのにかった時間.

図 2: 実験の結果. 赤色・● は岩田・今西の厳密解法, 緑色・× は提案手法, 青色・▲ は Qin らの手法である. 縦軸は出力された解の大きさ及びそれまでにかかった時間である. プロットは平均を, エラーバーは標準偏差を平均に加減算した値を表す. 岩田・今西の手法における表示の欠落は 120 分以内に終了せず打ち切られたことを表す.

表 1: 最も良い解を出力した際のパラメータ.

入力			T_0		x		c_2	
作成方法	n	p/d	平均	標準偏差	平均	標準偏差	平均	標準偏差
辺確率一定の ランダムグラフ	50	0.1	412.50	33.07	7.45	0.34	0.49	0.26
	50	0.2	412.50	59.95	7.45	0.34	0.56	0.23
	50	0.3	375.00	66.14	7.70	0.39	0.56	0.23
	50	0.4	387.50	78.06	7.45	0.34	0.64	0.23
	50	0.5	487.50	59.95	7.60	0.40	0.94	0.23
	100	0.1	412.50	33.07	7.55	0.37	0.56	0.18
	100	0.2	387.50	33.07	7.35	0.35	0.60	0.21
	100	0.3	375.00	43.30	7.80	0.20	0.64	0.18
	100	0.4	375.00	66.14	7.50	0.33	0.94	0.23
	100	0.5	462.50	48.41	7.50	0.39	0.82	0.20
	150	0.1	412.50	59.95	7.65	0.31	0.45	0.21
	150	0.2	400.00	50.00	7.45	0.34	0.52	0.20
	150	0.3	375.00	66.14	7.60	0.35	0.52	0.20
	150	0.4	375.00	66.14	7.90	0.35	0.94	0.18
	150	0.5	487.50	59.95	7.75	0.34	0.82	0.13
	200	0.1	525.00	43.30	7.70	0.33	0.64	0.18
	200	0.2	400.00	50.00	7.45	0.34	0.75	0.15
	200	0.3	512.50	78.06	7.75	0.28	0.94	0.23
	200	0.4	512.50	33.07	7.65	0.37	0.90	0.26
	200	0.5	550.00	50.00	7.55	0.37	0.86	0.23
ランダム正則グラフ	50	5	387.50	78.06	7.30	0.26	0.64	0.10
	50	10	375.00	66.14	7.65	0.31	0.56	0.23
	50	15	425.00	66.14	7.85	0.28	0.64	0.23
	50	20	412.50	59.95	7.65	0.31	0.97	0.25
	50	25	475.00	43.30	7.60	0.28	0.90	0.21
	100	10	400.00	50.00	7.65	0.31	0.64	0.18
	100	20	387.50	78.06	7.60	0.40	0.56	0.10
	100	30	425.00	66.14	7.70	0.33	0.86	0.28
	100	40	412.50	33.07	8.05	0.34	0.90	0.26
	100	50	487.50	33.07	7.60	0.35	0.94	0.23
	150	15	400.00	70.71	7.65	0.37	0.60	0.21
	150	30	400.00	50.00	7.40	0.28	0.86	0.28
	150	45	412.50	59.95	7.60	0.35	0.86	0.23
	150	60	425.00	43.30	7.55	0.31	0.90	0.26
	150	75	487.50	33.07	7.55	0.31	0.86	0.28
	200	20	487.50	59.95	7.65	0.37	0.64	0.18
	200	40	412.50	33.07	7.44	0.34	0.86	0.18
	200	60	450.00	50.00	7.55	0.37	0.94	0.23
	200	80	450.00	50.00	7.75	0.28	0.98	0.13
	200	100	537.50	69.60	7.65	0.31	0.86	0.23