

d -claw free グラフの重み付き最大独立集合問題に対する タブーサーチ法の提案

青木 一正 大館 陽太 山崎 浩一

群馬大学 工学部 情報工学科
〒376-8515 群馬県 桐生市 天神町 1-5-1

概要 d -claw とは、完全 2 部グラフ $K_{1,d}$ と同型な誘導部分グラフである。 d -claw free グラフとは、 d -claw を持たないグラフである。 d -claw free グラフに対する重み付き最大独立集合問題に対しては、いくつかの近似アルゴリズムが提案されている。良い近似率を保証するアルゴリズムは、 d が小さく、かつ頂点数が少ないグラフに対しては、良い解を得られる [14]。本論文では、Berman と Krysta [3] の Optimizing Misdirection を基としたタブーサーチ法を提案し、より良い解が得られることを実験的に示す。比較のために、Pullan 等 [17, 15, 16] が提案した（一般のグラフにおける）重み付き最大クリーク問題に対するメタヒューリスティックアルゴリズムを実装し、実験を行った。

A tabu search approach for the maximum weighted independent set problem on d -claw free graphs

Kazumasa Aoki Yota Otachi Koichi Yamazaki

Department of Computer Science Gunma University
1-5-1 Tenjin-cho, Kiryu zip:376-8515, Gunma, Japan

Abstract A d -claw is an induced subgraph isomorphic to $K_{1,d}$. A graph is d -claw free if it has no d -claw. Several approximation algorithms for the maximum weighted independent set problem on d -claw free graphs have been proposed. In this paper, we propose a new tabu search algorithm based on Berman and Krysta's "Optimizing Misdirection"[3]. And we demonstrate experimentally that our tabu search works well for sparse d -claw free graphs generated in natural ways. To evaluate the performance, we compare our tabu search with a heuristic algorithm introduced by Pullan and Hoos [17, 15, 16] which works for any weighted graph.

1 はじめに

グラフにおける独立集合とは、すべての 2 頂点間に隣接関係がない頂点集合である。最大独立集合問題とは、グラフ G 上の最大独立集合を求める問題である。重み付き最大独立集合問題とは重み付き最大独立集合を求める問題である。本論文では、重み付き最大独立集合問題を MWIS と呼ぶ。

完全 2 部グラフ $K_{1,d}$ と同型な誘導部分グラフを d -claw と呼び、 d -claw を持たないグラフを d -claw free グラフと呼ぶ。いくつかの自然な交差グラフは d -claw free であることが知られており、それらはネットワークや VLSI 設計などの広い分野に応用を持つ。一般に 3-claw free グラフは単に claw free グラフと呼ばれ、よく研究されている [5]。この claw free グラフに対しては、MWIS が多項式時間で求められることが知られている [13]。

2 定義

$G = (V, E, w)$ を、 V は頂点集合、 E は辺集合、 w は非負実数を取る頂点重み関数を示す単純無向グラフとする。重み関数を拡張し、 $U \subseteq V$ に対して $w(U) = \sum_{u \in U} w(u)$ 、 $w^a(U) = \sum_{u \in U} (w(u))^a$ とする。

MWIS のインスタンス $G = (V, E, w)$ のある独立集合を $X(\subseteq V)$ とする。 X について、 X の近傍 $N(X)(\subseteq 2^V)$ を定義する。 X を独立集合 $X' \in N(X)$ に変化させることを状態遷移と呼ぶ。特に、重み関数を増加させる X' が存在するとき、 X から X' への状態遷移を局所改善と呼ぶ。

独立集合 X を、頂点集合 Y を含む独立集合に状態遷移させる演算 \oplus を、 $X \oplus Y = (X \setminus \{u \mid \{u, v\} \in E, v \in Y\}) \cup Y$ と定義する。一般に $X \oplus Y = Y \oplus X$ は成り立たない。

\mathcal{I} を、ある最大化問題 \mathcal{M} の全てのインスタンスの集合とする。 \mathcal{M} に対する近似アルゴリズム A が $I \in \mathcal{I}$ の入

力に対して出力する値を $A(I)$ とし、最適解を $Opt(I)$ とする。本論文では、 A の近似率を $\inf_{I \in \mathcal{I}} A(I)/Opt(I)$ で定義する。したがって、近似率は常に 1 以下である。

3 既存の近似アルゴリズムの概要

この節では、過去に研究された d -claw free グラフに対する近似アルゴリズムの、近似率と時間計算量について述べる。この節で示すすべての近似アルゴリズムは、局所改善の繰り返しによるものである。まず、 $O(|V|^2)$ 時間で $1/d$ 近似を保証するアルゴリズムが Hochbaum [10] によって示された。その後、 $|V|^{O(1/\epsilon)}$ 時間で $1/(d-2+\epsilon)$ 近似を実現するアルゴリズムが Bafna [1] により示されている。また、 $1.5/d$ 近似アルゴリズムが Chandra と Halldórsson [4] によって示されたが、時間計算量が $\Omega(|V|^d)$ であった。Berman [2] は Chandra 等 [4] のアルゴリズムを改良し、 $2/d$ 近似アルゴリズムを実現したが、時間計算量はやはり $\Omega(|V|^d)$ であった。表 1 に、近似率と時間計算量をまとめる。

表 1 既存の近似アルゴリズム

提案者	近似率	時間計算量
Hochbaum [10]	$1/d$	$O(V ^2)$
Bafna [1]	$1/(d-2+\epsilon)$	$ V ^{O(1/\epsilon)}$
Chandra 等 [4]	$1.5/d$	$\Omega(V ^d)$
Berman [2]	$2/d$	$\Omega(V ^d)$

4 Berman と Krysta の近似アルゴリズム Optimizing Misdirection

ここでは、今回基とした Berman と Krysta [3] のアルゴリズムについて述べる。

Berman [2] のアルゴリズムは時間計算量が $\Omega(n^d)$ であったが、Berman と Krysta [3] は、時間計算量が d に関する指数関数にならない近似アルゴリズム Optimizing Misdirection [3] (以下、OM) を提案した。アルゴリズム OM を Algorithm 1 に示す。

Algorithm 1 Optimizing Misdirection

Input: $G = (V, E, w), d$.

Output: X

- 1: $X := \emptyset$
- 2: **while** there exists $\{u, v\}$ that $w^{\alpha(d)}(X) < w^{\alpha(d)}(X \oplus \{u, v\})$
do
- 3: $X := X \oplus \{u, v\}$
- 4: **end while**

OM は、(重みなし) 独立集合問題で $2/(d-1)$ 近似を実現する Bafna のアルゴリズム [1] を改良したもので、近似率は $1/(\beta(d) \cdot (d-1))$ を保証している。 $\beta(d)$ の具体的な

数値を、以下の式に示す。

$$\beta(d) = \begin{cases} 2^{-1/\log_2 3} & (6 \leq d) \\ (\sqrt{13}-1)/4 & (d=5) \\ 2/3 & (d=4) \end{cases}$$

さらに具体的な実数で表すと、近似率は表 2 のようになる。この近似率は、Berman [2] の近似率 $2/d$ より小さいが、Chandra 等 [4] の近似率 $1.5/d$ より大きい。

表 2 OM の近似率

d	近似率
6 以上	$1.548/(d-1)$
5	$1.535/(d-1)$
4	$1.5/(d-1)$

OM は局所改善の繰り返しによるアルゴリズムである。局所改善をおこなう際の具体的な近傍 $N(X)$ は、式 (1) で表せる。

$$N(X) = \{X \oplus \{u, v\} \mid \forall \{u, v\} \subseteq V, u \text{ or } v \in V \setminus X, u \neq v, \{u, v\} \notin E\}. \quad (1)$$

$N(X)$ 中に、条件式 (2) を満たす $\{u, v\}$ が存在すれば、OM は X から $X \oplus \{u, v\}$ への状態遷移を行う。

$$w^{\alpha(d)}(X) < w^{\alpha(d)}(X \oplus \{u, v\}). \quad (2)$$

式 (2) を満たす $\{u, v\}$ が存在しなくなった時、アルゴリズムは停止する。ここで $\alpha(d)$ は、式 (3) により決定される。

$$\alpha(d) = \begin{cases} \log_2 3 & (6 \leq d) \\ \log_{(\sqrt{13}-1)/4} \frac{1}{2} & (d=5) \\ \log_{2/3} \frac{1}{2} & (d=4) \end{cases} \quad (3)$$

しかし、このままではアルゴリズムの多項式時間停止性を保証できない。保証するためには、定数 k を用いた重み関数 w のリスケージング等を行うようにアルゴリズムを変更する必要がある [3]。その結果近似率は $(k-1)/(k \cdot \beta(d) \cdot (d-1))$ となり、時間計算量は $O(k^{\alpha(d)}(dn)^{2+\alpha(d)})$ となる。本論文では、インスタンスの重み関数 w が取る値を整数に制限した。このような場合には、リスケージング等を行わなくても、多項式時間停止性が保証される [3]。

5 タブーサーチの概要

ここではタブーサーチ法の概要を説明する。Glover [8, 9] によって提案されたタブーサーチは、局所改善を繰り返すメタヒューリスティックの一種である。最大独立集合問題に対しても、いくつかの適用例があり、その有効性が確認されている (例えば Fujisawa 等の Life Span Method [7])。ある解 X の近傍 $N(X)$ に、評価関数を増加させる解 X' が存在しないとき、 X を局所最適解と呼ぶ。

ある実行可能解から局所改善を繰り返し続けると、局所最適解に陥る場合がある。これを回避するために、**禁止リスト**と呼ばれるキュー構造を用いることが、タブーサーチの特徴である。

ある状態遷移が行われたとき、その遷移情報を禁止リストに追加する。アルゴリズムは、禁止リストに含まれる遷移を実行しないように状態遷移を続ける。つまり、タブーサーチにおける状態遷移には、禁止リストによる制限がかかることになる。禁止リストの大きさは、整数値 $TabuLength$ で制限されており、古い遷移情報から捨てられていく。

6 Tabu Search with Optimizing Misdirection

この節では、本研究で提案する Tabu Search with Optimizing Misdirection (以下、TSOM) について述べる。TSOM は、OM とタブーサーチを組み合わせたアルゴリズムである。アルゴリズムの概観を Algorithm 2 に示す。本節では今後、 X は現在の独立集合の状態、 $Best$ は TSOM が発見した最大値を表す変数と定義する。

Algorithm 2 Tabu Search with Optimizing Misdirection

Input: $G = (V, E, w), d, MaxCount, TabuLength$

Output: $Best$

```

1:  $Best := 0, count := 0$ 
2:  $X := OM(G, d)$  { 6.4 節 }
3: while  $count < MaxCount$  do
4:   choose  $\{u, v\}$  with maximum  $w^{\alpha(d)}(X \oplus \{u, v\})$  except
     tabu { 6.3 節 }
5:    $X := X \oplus \{u, v\}$ 
6:   update tabu list  $TL$  { 6.2 節 }
7:    $count := count + 1$ 
8:   if  $Best < w(X)$  then
9:      $Best := w(X)$ 
10:     $count := 0$ 
11:  end if
12: end while

```

6.1 近傍の定義

タブーサーチのような局所改善を基礎とするアルゴリズムを構築するうえで最も重要なのは、近傍構造の設計である。その近傍に、Berman [3] が示した OM で用いられた $N(X)$ を使う。理由は、近傍のサイズが $|V|^2$ と小さいため実行時間を少なくでき、また、OM は良い近似率を保証しているため、良い解が期待できるからである。

X の近傍 $N(X)$ として、OM と同様の式 (1) を用いる。確認のため、ここに再掲する。

$$N(X) = \{X \oplus \{u, v\} \mid \forall \{u, v\} \subseteq V, u \text{ or } v \in V \setminus X, u \neq v, \{u, v\} \notin E\}.$$

6.2 禁止リスト管理

一般的には、禁止リストは $TabuLength$ サイズのキュー構造で表現される。TSOM では、禁止リストの管理に、Fujisawa 等が提案した Life Span Method [7] で用いられる配列構造 TL を使用する。ある頂点 v が禁止リストに含まれるかどうかは、 $TL(v) > 0$ かどうかで判定できる。 $TL(v) > 0$ であるときに、頂点 v が禁止リストに含まれるという。 X から $X \oplus \{u, v\}$ への遷移は、頂点 u, v が両方も禁止リストに含まれる場合に禁止される。禁止リストをふまえた具体的な近傍と状態遷移は、6.3 節で示す。

禁止リストの更新は以下に示す**追加**、**減少**、**全削除**の3ステップで構成され、順に実行される。

■**追加** 直前に発生した遷移 $X \oplus \{u, v\}$ に対応し、頂点 u, v を禁止リストへ追加する。具体的には、 $TL(u) = TabuLength$ 、 $TL(v) = TabuLength$ を適用する。このとき、 $TL(u) > 0$ 、 $TL(v) > 0$ である場合でも、同様の操作を行い禁止期間を延長する。

■**減少** $TL(v) > 0$ であるすべての頂点 v に対して、 $TL(v) := TL(v) - 1$ を適用する。もしも、すべての頂点 v が禁止リストに含まれるならば、減少ステップを繰り返す。

■**全削除** X に含まれないすべての頂点が、禁止リストに含まれるとする。このような場合は、禁止リストを全削除する。すなわち、すべての頂点 v に対して、 $TL(v) := 0$ を実行する。

6.3 禁止リストで制限された近傍と状態遷移

禁止リスト $TL(v)$ により制限された近傍 $N'(X)$ を式 (4) で定義する。

$$N'(X) = \{X_i \in N(X) \mid TL(u) = 0 \text{ or } TL(v) = 0\}. \quad (4)$$

X の次の状態 X' は、 $N'(X)$ の中から以下の条件を満たすものを選択する。

$$w^{\alpha(d)}(X') = \max_{X_i \in N'(X)} \{w^{\alpha(d)}(X_i)\}.$$

$\alpha(d)$ もまた OM と同様に式 (3) により決まる。

6.4 初期化と終了判定

近傍から遷移状態を決定するためのパラメータ $\alpha(d)$ は OM に準じ、式 (3) により決定する。また、TSOM における独立集合の初期状態は、OM によって生成する。

終了判定は、パラメータ $MaxCount$ を用いる。発見した最大値 $Best$ が更新されないまま、連続で $MaxCount$ 回の遷移が行われた時、アルゴリズムを終了する。

7 比較対象 Phased Local Search

実験の比較対象として、Pullan 等の提案した Phased Local Search [17, 15, 16] (以下、PLS と表記) を採用した。PLS は最大重み付きクリーク問題に対するアルゴリズムであるが、一般にクリーク問題は独立集合問題と同等である。比較対象として採用した理由は二つ挙げられる。

- 局所改善の繰り返しによるアルゴリズムである,
- MWIS に対する最新の結果である.

PLS はランダムアルゴリズムであり, 状態遷移などの候補が複数ある場合は, 一様乱数によって動作が決定される.

PLS は, 入力としてグラフ $G = (V, E, w)$, *TargetWeight*, *StopCount* を受け取る. *StopCount* 回以内の局所改善で, 重みが *TargetWeight* 以上のクリークを見つけばそれを返し, 発見できなければ **false** を返す. 本論文では, TSOM に要した計算時間とほぼ同じ時間だけ動作させるため, 停止条件を回数から時間に変更し, *StopTime* 以内とした.

8 実験データ

いくつかのグラフクラスは *d-claw free* であることが知られているが, 本節では実験に使用した *unit disk* グラフと *d-SP* グラフを紹介する.

8.1 Unit disk グラフ

単位円の交差グラフは *unit disk* グラフと呼ばれる. Marathe 等 [12] は *unit disk* グラフが *6-claw free* であることを示した. この *unit disk* グラフは, ネットワークなどに広い応用を持ち, 盛んに研究されている (例えば, Diaz 等による最近の研究 [6]). 図 1 に, *unit disk* の *5-claw* の例を示す.

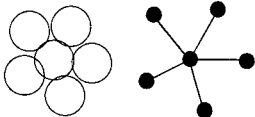


図 1 Unit disk グラフの 5-claw の例

8.2 *d-SP* グラフ

重み付き集合充填問題とは, 重みの付いた部分集合の族が与えられたとき, 互いに素な部分集合からなる族のうち重みが最大のものを見つける問題である. 特に, 部分集合の最大サイズが k である問題を *k-set packing problem* (*k-SP*) という. 集合充填問題は独立集合問題に帰着することが可能であり, 特に *k-SP* のインスタンス I をグラフ独立集合問題のインスタンス G_I に帰着すると, G_I は $(k+1)$ -*claw free* であることが知られている [11]. 本論文では $(d-1)$ -*SP* のインスタンスから生成するグラフを, に *d-SP* グラフと呼ぶことにする. *d-SP* グラフは *d-claw free* である. 図 2 に, *4-SP* の *3-claw* の例を示す.

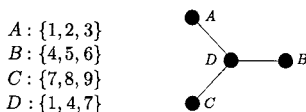


図 2 *4-SP* グラフの *3-claw* の例

9 計算機実験

この節では, 実験に使用したグラフについて説明し, 実験結果を述べる. 実験環境は, CPU 2.6GHz (AMD Athlon64X2 5000+), RAM 2GB, OS Windows Server 2003 64bit 版の PC である. 実装は C++ で行い, Visual Studio 2005 Professional Edition の amd64 でコンパイルした.

9.1 使用したグラフデータ

実験対象のグラフデータは, *unit disk* グラフ (*6-claw free*), *6-SP* グラフ (*6-claw free*), *10-SP* グラフ (*10-claw free*) の 3 種類とする. 頂点数は 200, 500, 1000 の 3 種類とし, それぞれに対して 10 個のグラフを生成し, 使用した. 頂点重みは, 一様乱数で $[1, 200]$ の範囲で割り当てた. 使用するグラフの辺密度を決定するために, 1000 頂点の *unit disk* グラフに対して予備実験を行った. その結果, 辺密度が高くなると実験時間が膨大になることが分かったため, 辺密度は約 1% となるように調整した (詳細は 9.3 節で述べる).

9.1.1 Unit disk グラフの生成方法

$k \times k$ サイズの正方形に対し, 一様乱数で点 (x, y) を打つ (x, y は実数とする). 打たれた点を頂点とし, 距離が 1 未満である点同士に辺を与えることで, *unit disk* グラフを表現する. このような方法で生成されたグラフは, *random geometric graph* とも呼ばれる [6].

グラフの辺密度を約 1% に限定するために, $k = 17$ としてグラフの生成を行った.

9.1.2 *d-SP* グラフの生成方法

各頂点 v に対し, 台集合 S から要素 $s \in S$ を, 一様にランダムに $d-1$ 回だけ復元抽出した集合 S_v を割り当てる. このとき, $S_u \cap S_v \neq \emptyset$ ならば対応する 2 頂点 u, v 間に辺を加える.

辺密度を約 1% に制限するため, *6-SP* グラフでは $|S| = 2500$, *10-SP* グラフでは $|S| = 8000$ としている.

9.2 実験手順と記録方法

インスタンス $G = (V, E, w)$ に対し, 以下の順で実験を行う.

まず, TSOM を実行する. $TabuLength = |V|/4$, $MaxCount = 5 \cdot |V|$ に固定し, インスタンスに応じたパラメータのチューニングは行わない. TSOM の実行結果として, $Best$, T_s , および T_e を記録する. それぞれ,

- $Best$: TSOM が発見した解の重みの最大値,
- T_s : 重みが $Best$ である解を初めて発見した時間 [秒],
- T_e : アルゴリズム TSOM の停止時間 [秒].

を表す.

次に PLS を実行する. パラメータは, $TargetWeight = Best$, $StopTime = T_s$ とする. ただし, $T_s < 1.00$ である場合は, $StopTime = 1.00$ とした. PLS はランダムアルゴリズムであるため, 5 回試行し, 条件を満

たす解を発見した回数を記録する。

9.3 実験結果

表 3~表 11 に、実験値を示す。また、予備実験として行った辺密度についての比較を、表 12 に示す。予備実験の比較対象は unit disk グラフであり、頂点数を 1000 に固定し辺密度を変化させている。辺密度が 10% を超えると、TSOM の停止まで 30,000 秒以上かかるインスタンスが存在したため、5.0, 3.0, 1.5, 1.0 % についてのみ結果を示している。

表の読み方は以下の通りである。

- 表題：実験したグラフの頂点数と種類、
- 辺密度：実験したグラフの辺密度、
- No.：グラフごとに付けられた識別番号、
- OM：アルゴリズム OM による実験値、
- TSOM：アルゴリズム TSOM による実験値
 - Best：TSOM の出力 Best、
 - IMP：OM と比較した改善値、
 - T_s, T_e ：アルゴリズム停止時間 [秒] (9.2 節)、0.00 は、0.01 未満を表す、
- PLS：アルゴリズム PLS の成功回数。

10 考察

多くのインスタンスに対する実験結果で、PLS の成功回数が 0 であることから、疎な d -claw free グラフに対する、TSOM の PLS に対する優位性を確認することができる。また、頂点数 200 を除き、ほぼすべてのグラフに対して、OM の解を改善することができている。特に SP グラフについては、大幅な改善がみられると共に、PLS との比較でも良い結果が得られている。

実験結果のうち、特に 200 頂点の unit disk グラフ (表 3) に注目する。10 個の実験データのうち、8 個は IMP が 0 である。つまり、TSOM による改善がされていない。それにも関わらず、PLS は 8 個中 7 個成功していない。この結果から、頂点の少ない unit disk グラフに対して、アルゴリズム OM は効果的であることが伺える。

11 まとめと今後の課題

本研究では、Berman 等 [3] の OM を取り入れたタブーサーチ法 TSOM を提案し、Pullan 等の PLS [16] との比較実験を行った。

TSOM は、MWIS に対してシンプルにタブーサーチ法を適用したアルゴリズムである。今後の課題として、タブーサーチ法を発展させた、より新しいメタヒューリスティックの近似アルゴリズムへの適用がある。例えば、タブーサーチ法を発展させた Life Span Method では、TabuLength のランダム性や、我々も取り入れた禁止リストの管理方法など、様々な工夫がされている。本研究では、実験時間の制約のため TabuLength のランダム性を取

り入れなかったが、ランダム要素を取り入れることによって解の改善が期待される。

参考文献

- [1] V. Bafna, B. Narayanan, and R. Ravi. Nonoverlapping local alignments (weighted independent sets of axis-parallel rectangles). *Discrete Applied Mathematics*, Vol. 71, pp. 41–53, 1996.
- [2] P. Berman. A $d/2$ approximation for maximum weight independent set in d -claw free graphs. *Nordic Journal of Computing*, Vol. 7, No. 3, pp. 178–184, 2000.
- [3] P. Berman and P. Krysta. Optimizing misdirection. *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 192–201, 2003.
- [4] B. Chandra and M. M. Halldórsson. Greedy local improvement and weighted set packing approximation. *Journal of Algorithms*, Vol. 39, pp. 223–240, 2001.
- [5] M. Chundnovsky and P. Seymour. The structure of claw-free graphs. *Surveys in combinatorics, London Mathematical Society Lecture Note Series*, Vol. 372, pp. 153–171, 2005.
- [6] J. Díaz, D. Mitsche, and X. Pérez. Sharp threshold for hamiltonicity of random geometric graphs. *SIAM Journal on Discrete Mathematics*, Vol. 21, pp. 57–65, 2007.
- [7] K. Fujisawa and M. Kubo. Experimental analyses of the life span method for the maximum stable set problem. *The Institute of Statistical Mathematics Cooperative Research Report*, Vol. 75, pp. 135–165, 1995.
- [8] F. Glover. Tabu search – Part I. *ORSA Journal on Computing*, Vol. 1, No. 3, pp. 190–206, 1989.
- [9] F. Glover. Tabu search – Part II. *ORSA Journal on Computing*, Vol. 2, No. 1, pp. 4–32, 1990.
- [10] D.S. Hochbaum. Efficient bounds for the stable set, vertex cover, and set packing problems. *Discrete Applied Mathematics*, Vol. 6, pp. 243–254, 1983.
- [11] C.A.J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio heuristics for packing problems. *SIAM Journal on Discrete Mathematics*, Vol. 2, No. 1, pp. 68–72, feb 1989.
- [12] M.V. Marathe, H. Brey, H.B. Hunt III, S.S.Ravi, and D.J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, Vol. 25, pp. 59–68, 1995.
- [13] D. Nakamura and A. Tamura. A revision of minty’s algorithm for finding a maximum weight stable set of a claw-free graph. *Journal of the Operations Research Society of Japan*, Vol. 44, No. 2, pp. 194–204, 2001.
- [14] Y. Otachi and K. Yamazaki. Experimental analyses of

approximation algorithms for the maximum weighted independent set problem on d -claw free graphs. *IEICE technical report. Theoretical foundations of Computing*, Vol. 105, pp. 25–30, 2006.

- [15] W.J. Pullan. Phased local search for the maximum clique problem. *Journal of Combinatorial Optimization*, Vol. 12, No. 3, pp. 303–323, 2006.
- [16] W.J. Pullan. Approximating the maximum vertex/edge weighted clique using local search. *Journal of Heuristics*, to appear.
- [17] W.J. Pullan and H.H. Hoos. Dynamic local search for the maximum clique problem. *Journal of Artificial Intelligence Research*, Vol. 25, pp. 159–185, 2006.

表 3 200 頂点の unit disk グラフ

No.	OM	TSOM				PLS
		Best	IMP	T_s	T_e	
0	11498	11498	0	0.01	2.94	0
1	10498	10498	0	0.01	2.00	0
2	10682	10682	0	0.00	2.00	0
3	10325	10516	191	1.81	3.72	0
4	11479	11479	0	0.00	1.97	0
5	10563	10833	270	0.13	2.09	0
6	11499	11499	0	0.00	1.95	0
7	10087	10087	0	0.01	2.03	0
8	10888	10888	0	0.01	1.98	5
9	10568	10568	0	0.01	1.95	0

表 4 500 頂点の unit disk グラフ

No.	OM	TSOM				PLS
		Best	IMP	T_s	T_e	
0	16365	16746	381	10.36	72.03	0
1	17242	17431	189	18.28	77.44	0
2	15929	15997	68	19.14	81.56	4
3	16715	16784	69	16.16	75.13	0
4	15955	16064	109	4.38	65.38	5
5	15865	16549	684	20.59	84.41	0
6	16451	16627	176	6.78	67.27	0
7	16022	16205	183	2.99	65.55	5
8	15640	15923	283	54.22	121.00	5
9	16433	16631	198	4.58	64.33	0

表 5 1000 頂点の unit disk グラフ

No.	OM	TSOM				PLS
		Best	IMP	T_s	T_e	
0	20548	20898	350	454.94	1655.81	0
1	20275	20394	119	335.59	1534.02	5
2	20163	21032	869	581.05	1987.63	0
3	19938	20489	551	1297.56	2795.06	5
4	20101	20849	748	830.06	2210.20	0
5	20555	20802	247	1518.42	2928.72	0
6	20571	21356	785	1943.64	3234.22	0
7	20634	21178	544	1644.47	2748.49	0
8	20531	20831	300	2079.20	3392.27	0
9	19951	20453	502	1186.38	2499.55	2

表 6 200 頂点の 6-SP グラフ

No.	OM	TSOM				PLS
		Best	IMP	T_s	T_e	
0	13257	13280	23	0.02	2.03	0
1	13622	13685	63	0.03	1.86	0
2	12792	12862	70	0.69	2.61	5
3	13353	13455	102	0.05	1.95	0
4	13522	13522	0	0.00	1.94	0
5	13059	13078	19	1.58	3.41	0
6	13405	13443	38	0.03	1.92	0
7	12809	12809	0	0.00	2.05	1
8	12613	12626	13	0.02	1.95	5
9	13374	13460	86	0.06	1.97	0

表 9 200 頂点の 10-SP グラフ

No.	OM	TSOM				PLS
		Best	IMP	T_s	T_e	
0	13488	13519	31	0.03	1.61	5
1	12782	12784	2	0.05	2.02	0
2	13354	13362	8	0.05	1.74	0
3	13815	13841	26	0.05	1.84	0
4	11983	12087	104	0.11	2.28	0
5	13042	13044	2	0.05	2.11	0
6	13319	13368	49	0.77	2.66	0
7	12587	12652	65	0.05	2.03	5
8	12553	12751	198	0.17	2.02	0
9	12620	12857	237	0.09	1.92	0

表 7 500 頂点の 6-SP グラフ

No.	OM	TSOM				PLS
		Best	IMP	T_s	T_e	
0	22039	22361	322	10.14	70.08	0
1	22638	22936	298	1.55	64.00	0
2	23182	23578	396	1.08	61.70	0
3	24139	24169	30	1.53	62.06	0
4	23873	24297	424	67.75	127.49	3
5	23295	23429	134	0.67	63.59	0
6	23449	23829	380	4.80	63.75	0
7	22306	22654	348	68.81	131.80	0
8	24645	24645	0	0.09	62.44	0
9	23252	23710	458	2.47	64.28	0

表 10 500 頂点の 10-SP グラフ

No.	OM	TSOM				PLS
		Best	IMP	T_s	T_e	
0	24466	24762	296	11.49	72.42	0
1	22611	22820	209	16.75	77.38	0
2	22897	22995	98	5.03	65.47	0
3	23860	24077	217	2.11	63.88	0
4	24028	24578	550	24.49	83.19	0
5	25269	25476	207	15.75	73.52	0
6	24892	25339	447	88.24	148.21	0
7	24032	24410	378	91.95	155.46	0
8	23268	23573	305	38.81	99.53	0
9	21450	21752	302	35.38	96.98	0

表 8 1000 頂点の 6-SP グラフ

No.	OM	TSOM				PLS
		Best	IMP	T_s	T_e	
0	32362	33540	1178	1490.94	2704.13	0
1	32734	34142	1408	1496.80	2670.08	0
2	32116	32697	581	13.36	1027.38	0
3	33049	33882	833	1294.80	2533.63	0
4	32097	33581	1484	864.53	2064.77	0
5	33910	34387	477	41.66	1411.05	0
6	32735	33561	826	31.63	1241.63	0
7	33925	35141	1216	454.69	1464.92	0
8	32909	33803	894	1870.16	3075.45	0
9	33679	34436	757	15.09	1258.98	0

表 11 1000 頂点の 10-SP グラフ

No.	OM	TSOM				PLS
		Best	IMP	T_s	T_e	
0	33352	33858	506	16.28	1152.16	0
1	34938	35631	693	382.50	1602.11	0
2	33841	34931	1090	480.52	1706.36	0
3	34472	34820	348	50.33	1379.31	0
4	34511	35082	571	1464.33	2760.34	0
5	34162	35680	1518	573.56	1784.11	0
6	34117	34576	459	317.36	1603.16	0
7	35500	36099	599	1265.89	2437.48	0
8	34381	35050	669	193.25	1200.13	0
9	34982	35883	901	1170.67	2205.16	0

表 12 1000 頂点の unit disk グラフに対する辺密度による比較

辺密度 [%]	No.	OM	TSOM				PLS
			<i>Best</i>	IMP	T_s	T_e	
5.0	0	5794	5884	90	10934.60	18218.80	5
	1	5587	5798	211	327.36	7580.88	5
	2	5668	5840	172	9927.94	17055.40	0
	3	5692	5897	205	3165.38	10189.10	5
	4	5811	5941	130	11564.80	18699.70	0
	5	5617	5769	152	11226.40	18433.30	5
	6	5592	5653	61	6674.50	13901.30	5
	7	5740	5802	62	150.16	7308.91	0
	8	5451	5684	233	9207.42	16366.60	1
9	8162	8670	508	1126.52	8275.47	0	
3.0	0	9071	9334	263	2596.61	6385.53	5
	1	9017	9172	155	3965.84	7787.78	5
	2	8880	9026	146	6960.56	10946.00	5
	3	8676	9070	394	10163.50	14028.80	5
	4	8775	9006	231	3959.49	7804.02	5
	5	8886	9147	261	5318.28	9353.08	5
	6	8834	9319	485	2666.91	6353.39	4
	7	9154	9311	157	2503.39	6437.06	0
	8	8576	9183	607	3265.97	7246.97	0
9	8949	9206	257	4026.34	7968.38	4	
1.5	0	14844	15300	456	709.34	2908.11	5
	1	15342	15496	154	198.67	2291.02	2
	2	15270	15909	639	4609.53	6873.53	0
	3	14616	15168	552	1379.81	3564.14	5
	4	15352	15684	332	3811.19	6582.94	0
	5	15045	15238	193	2557.05	4659.95	5
	6	14634	15412	778	3864.64	6013.48	5
	7	14766	15300	534	1575.13	4303.47	4
	8	15087	15488	401	4716.48	7089.61	3
9	15001	15546	545	1026.91	3032.14	0	
1.0	0	20548	20898	350	454.94	1655.81	0
	1	20275	20394	119	335.59	1534.02	5
	2	20163	21032	869	581.05	1987.63	0
	3	19938	20489	551	1297.56	2795.06	5
	4	20101	20849	748	830.06	2210.20	0
	5	20555	20802	247	1518.42	2928.72	0
	6	20571	21356	785	1943.64	3234.22	0
	7	20633	21178	545	1644.47	2748.49	0
	8	20531	20831	300	2079.20	3392.27	0
9	19951	20453	502	1186.38	2499.55	2	