Vol.2011-MPS-82 No.1
2011/3/7

Minimum Consistent-DFA 生成問題の厳密解法に対するハイブリッドアプローチ

乾伸雄[†] 相澤彰子[††]

本稿では，ラベル付き記号列集合から最小状態数の決定性有限状態オートマトン（DFA）を効率よく生成する手法について述べる．この問題は，NP 困難な問題と知られているが，近年の SAT ソルバーの効率化によって，専用のアルゴリズムに匹敵する計算速度が得られたことが報告された．本稿では，この手法をさらに発展させるため，対称性除去のための最大クリークを MILP ソルバーで発見する方法を導入する．また，ヒューリスティックに発見した DFA を上界，最大クリークを下界としたアルゴリズムを提案する．数値実験の結果，ベンチマーク問題に対して従来手法の 50%の計算時間で結果が得られることがわかった．

A Hybrid Approach for Exact Solving of Minimum Consistent-DFA Generation Problem

Nobuo Inui[†] and Akiko Aizawa[††]

This paper proposes a new hybrid method for generating a minimum consistent-DFA (deterministic finite state automaton) of a set of labeled strings. Though this problem is well-known as an NP-hard problem, the recent paper reported that a SAT (satisfiability) problem formulation could solve it efficiently and showed compatible with specialized algorithms. This paper introduces a mixed integer linear programming model to solve the maximum clique problem for symmetry breaking and use an upper-bound found by a heuristic method to use a window search. Numerical experiment showed that our method achieved 50% reduction in running time compared with the previous SAT-based method.

1. はじめに

決定性有限状態オートマトン（以下，DFA）を使い，ラベル付き記号列の事例集合（以下，事例集合）の一般化学習を行う．文法推論の研究分野において，この問題は非常に多く研究されている．例えば，真の DFA を生成するための事例集合に関する研究[1]や任意の事例集合に関して真の DFA に近い DFA を生成する研究[2]が行われている．これらに対して，Minimum Consistent-DFA（最小無矛盾 DFA）生成問題は，与えられた事例集合を正しく判定する最小状態数の DFA を生成する問題である．この DFA はもっとも単純な形で事例集合を表現しており，様々な学習アルゴリズムの評価基準を与えると考えられる．

Minimum Consistent-DFA 生成問題は NP 困難な問題の一つ[3]であるため，効率的な解法に関する研究が行われている．一つは，DFA の一般化において二つの状態を同一なものと見なし，繰り返しマージを行う状態マージを使って可能な DFA を列挙する手法である[4]．この手法では，探索の初期に状態数が最小となる DFA を求めることと，それによって探索空間を狭くすることが効率化の鍵となる．もう一つは，ある数以下の Consistent DFA が存在しないことを証明することにより、DFA を求める手法である[5]．この手法においては，事例集合に矛盾する DFA の生成を抑制することが効率化の鍵となる．従来は，専用のアルゴリズムの開発が行われたが，様々な研究成果が反映されているソルバーを使って研究が行われている[8,9,10]．しかし，大規模な問題に対して厳密解を得ることは難しい[7]と考えられているため，一層の効率化が望まれる．本稿は後者の手法に関連して，ヒューリスティックな解法，混合整数線形計画法（MILP），充足可能性判定問題（SAT）をハイブリッドに使った効率的な解法について述べる．

本稿の構成は次のとおりである．２章では Minimum Consistent-DFA 問題に関して本稿に関連した事柄を定義する．３章では，ハイブリッド手法のアルゴリズムについて述べる．４章では数値実験を示し，５章でまとめる．

2. 記法，定義および解法

2.1 Minimum Consistent-DFA 生成問題

本稿で扱う問題を説明する．そのため，Oliveira[5]の記法に基づいて問題を説明する．DFA を生成する目的は，任意の文字列を正例・負例あるいは判別不可能と判定することである．入力記号集合をΣとし，すべての文字列集合をΣ^*とする．DFA を生成するために正例の文字列集合$I^+ \subset \Sigma^*$および負例の文字列集合$I^- \subset \Sigma^*$が与えられる．ただし，

[†] 総合研究大学院大学，国立東京工業高等専門学校
The Graduate University for Advanced Studies, Tokyo National College of Technology
[††] 東京大学，国立情報学研究所
Tokyo University, Natinal Institute of Informatics

$I^+ \cap I^- = \emptyset$とする．

Augmented Prefix Tree Acceptor（APTA）はI^+とI^-から作成される木構造をした DFA である．図１に例を示す．DFA は○で示された状態と，状態間の矢印で示された入力記号に対する状態遷移で表現される．APTA は与えられた事例の正負を正しく判定し，それ以外の文字列は判別不可能と判定する．

I^+とI^-を正しく判定する DFA は Consistent-DFA と呼ばれる．与えられた事例集合だけに関して，APTA は Consistent-DFA の中で最大の状態数を持っている．より少ない状態数を持つ Consistent-DFA は APTA を起点として状態マージによって作成されるラティス中で表現されている．図２に図１の APTA についてのラティスの一部を示す．状態iと状態jがマージされるとき T, マージされないとき F とする論理値をもつ論理変数P_{ij}を使うことによって，二つの状態がマージ可能であるかどうかは次の論理式によって表現できる．ここで，Q は APTA における状態の集合を表す．δ_{ia}は遷移関数とよばれ，状態 i において記号 a が入力したときに遷移する状態を示す．APTA 中にδ_{ia}に対応する状態が存在する場合は$\delta_{ia} \in Q$，存在しない場合，すなわち，判定不可能である場合は$\delta_{ia} \notin Q$となる．また，記号列の遷移が状態 i で終了した場合，λ(i)は出力関数と呼ばれ，記号列に対する判定を表し，正例の場合は{+}，負例の場合は{-}，判定不可能の場合は{+,-}の値を持つ．

$$\wedge_{i,j,\delta_{ia},\delta_{ja}\in Q,a\in\Sigma}\left(P_{ij}\rightarrow P_{\delta_{ia}\delta_{ja}}\right),\wedge_{i,j,k\in Q}\left(P_{ij}\wedge P_{jk}\rightarrow P_{ik}\right),\wedge_{(i,j)\in Q,\lambda(i)\cap\lambda(j)=\emptyset}\overline{P_{ij}} \qquad (1)$$

上記の式において, 第１項は遷移関数に対して, 二つの状態 i,j がマージする場合は, 任意の記号aによって遷移するδ_{ia}, δ_{ja}もマージすることを表す．第２項は三つの状態のマージに関するもので，遷移の決定性を保証する．第３項は異なる判定を行う状態のマージを禁止する．

$\Sigma = \{a, b\}$
$\Lambda = \{+, -\}$
$I^+ = \{aab, bb\}$
$I^- = \{b, ba\}$

$Q = \{0,1,2,3,4,5,6\}$
$\delta_{0a} = 1, \delta_{0b} = 2, \delta_{1a} = 3,$
$\delta_{2a} = 4, \delta_{2b} = 5, \delta_{3a} = 6$
$\lambda(0) = \{+,-\}, \lambda(1) = \{+,-\},$
$\lambda(2) = \{+\}, \lambda(3) = \{+,-\},$
$\lambda(4) = \{-\}, \lambda(5) = \{+\},$
$\lambda(6) = \{+\}$

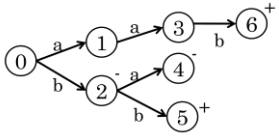

図１　事例と APTA

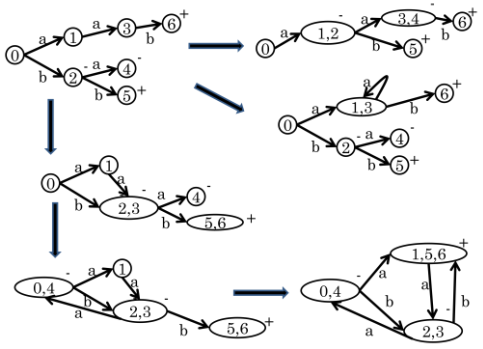

図２　APTA から作られるラティス(一部)

図２は(1)を使って得られるラティスの一部である．Minimum Consistent-DFA はこのラティス中で状態数が最小となる DFA であり，この DFA を求める問題を Minimum Consistent-DFA 生成問題と呼ぶ．

2.2 Minimum Consistent-DFA 生成問題とグラフ彩色問題

グラフ彩色問題（頂点彩色問題）とは頂点集合および隣接辺集合が与えられたときに，隣接辺で隣り合う頂点を同じ色に塗らず，最小彩色数ですべての頂点を塗り分ける問題である．Minimum Consistent-DFA 生成問題は，状態を頂点と考えることで頂点彩色問題の拡張であると考えることができる[6]. 頂点彩色問題として式(1)を考えると, 第２項は３個の頂点の色の一致を, 第３項は二つの状態間の隣接辺を表す. Coste ら[6]は第１項を dynamic constraint と呼び，頂点彩色問題の外側に存在する制約条件として扱った．

Oliveira ら[5]は，ラティスを使わずに，少ない彩色数から順番に調べ，初めて見つかった Consistent-DFA を Minimum Consistent-DFA とする手法を提案した．Inui ら[8]は MILP ソルバーを使って問題を定式化し，MILP による直接的なアプローチの困難さを示した．Heule ら[9]は，SAT ソルバーを使ってこの問題を定式化し，専用に作成されたプログラムと同等の計算時間で解が得られることを示した．ソルバーを用いる手法としては現在 SAT によるアプローチが最も効率的であると考えられる．

これらの方法では，APTA 中の遷移関数を色の間の遷移関数として扱う．色の集合をCとする．論理変数X_{ic}は状態$i \in Q$の色が$c \in C$であるとき T，そうでないとき F となり，論理変数$Y_{c_1ac_2}$は色$c_1 \in C$において入力記号が$a \in \Sigma$であるとき色$c_2 \in C$に遷移するとき T，そうでないとき F となる．このとき，Consistent-DFA は次の論理式を満足する．

$$\wedge_{i\in Q}\left(\vee_{c\in C} X_{ic}\right),\wedge_{i,\delta_{ia}\in Q,c_1,c_2\in C,a\in\Sigma}\left(X_{ic_1}\wedge X_{\delta_{ia}c_2}\rightarrow Y_{c_1ac_2}\right),\wedge_{c,c_1,c_2\in C,a\in\Sigma,c_1\neq c_2}\left(\overline{Y_{cac_1}}\vee\overline{Y_{cac_2}}\right),$$
$$\wedge_{i,j\in Q,\lambda(i)\cap\lambda(j)=\emptyset,c\in C}\left(\overline{X_{ic}}\vee\overline{X_{jc}}\right) \qquad (2)$$

第１項はすべての状態は少なくとも一つの色に彩色される条件，第２項は遷移関数の存在に関する規則，第３項は遷移関数の一意性に関する規則，第４項は異なる判定を行う状態は異なる色を持つための規則である．

2.3 Minimum Consistent-DFA 生成問題の下界値問題

前節で述べた手法では少ない彩色数から順次 Consistent-DFA の存在を調べるため，事前にある彩色数未満に可能な解がないことを知っていれば解を求める効率化につながる．頂点彩色問題については，お互いに辺で結ばれた完全グラフ（クリーク）が彩色数の下限値を与えることがわかっている．そこで，式(2)の第４項を拡張した頂点彩色問題を考える．

式(1)の第３項, 式(2)の第４項では異なる出力関数の値を持つ状態がマージできないことを表現した．これは，式(1)を使うことにより，任意の二つの状態がマージできるかどうかを判定することに一般化できる．この頂点彩色問題は式(3)で表現できる．

$\overline{Exp(1)(P_{ij})}$は$P_{ij}=T$としたとき式(1)の評価が F であることを示す．

$$\wedge_{i\in Q}\left(\vee_{c\in C} X_{ic}\right),\wedge_{i,j\in Q,\overline{Exp(1)(P_{ij})},c\in C}\left(\overline{X_{ic}}\vee\overline{X_{jc}}\right) \quad (3)$$

式(3)は式(2)の緩和問題となっているため，式(3)を使って求められた最小彩色数は式(2)を満足する彩色数の下限値を与える．さらに頂点彩色問題の下限値は最大クリーク（頂点数が最大となる完全部分グラフ）で与えられることが知られている．論理変数Z_iがTのとき状態iはクリークを構成する状態，Fのときそうでないとする．すると，クリークは次のように表現することができる．$Exp(1)(P_{ij})$は$P_{ij}=T$としたとき式(1)の評価が T，つまり二つの状態は同じ色に彩色できる状態であることを示す．

$$\wedge_{i,j\in Q,Exp(1)(P_{ij})}\left(\overline{Z_i}\vee\overline{Z_j}\right) \quad (4)$$

最大クリークは彩色数の下限値を与え，最大クリークに属する状態には別々に固定した色を割り付けることができる．このような色の固定は，SAT ソルバーなどで解を得るとき，対称性の除去に効果がある．

図３に図１の APTA に対する隣接グラフ表現を示す．Minimum Consistent-DFA 生成問題は遷移に関する条件も満たす必要があるため，一般に N 個の状態について，すべての N-1 個の状態が同じ色に彩色可能であっても，N 個は彩色できない場合がある．そのため，厳密に隣接グラフで APTA を表現する場合は，２個以上の頂点の集まりを一つの辺と考え，少なくとも一つの辺の一対の頂点を異なる色に塗らなければならないハイパーグラフ彩色問題[13]に帰着する必要がある．しかしながら，APTA からハイパーグラフを作成することは，全てのマージ可能な頂点集合を列挙する問題と等しくなり，実現困難である．そのため，３状態以上の辺を持つハイパーグラフ彩色問題ではなく，２状態に関する頂点彩色問題として，Minimum Consistent-DFA 生成問題の下界値問題を扱う．

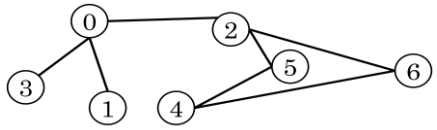

図３　APTA の隣接グラフ

頂点彩色問題および最大クリーク問題はともに NP 困難な問題である．頂点彩色問題に関しては効率化の研究が行われている[12]が，実用的に解を求めることは難しい．これに対して，最大クリーク問題は，辺の数が多い順番にソートし，先頭の頂点から順番に完全グラフを構成するヒューリスティックな手法が存在する．Heule ら[9]は，これで求まった最大クリークを Minimum Consistent-DFA 生成問題の下界値および対称性除去に用いることで効率的に解が求められることを報告している．

3. 提案手法

前章では，Minimum Consistent-DFA 生成問題を解くための基本的な考え方について述べた．特に，式(2)および最大クリークを使った解法は成功を収めており，SAT ソルバーの開発とともに扱える問題の規模も大きくなっていくと思われる．本稿では，さらに SAT ソルバーを使った解法の効率化を図る手法を提案する．本稿で提案する手法は次のとおりである．

① MILP ソルバーを使って，最大クリーク問題による下界値を厳密に求める．
② ヒューリスティックに求めた Minimum Consistent-DFA により上界値を求める．
③ 両者を組み合わせたアルゴリズムを設計する．

最大クリーク問題は NP 困難な問題であるため，ヒューリスティックな解法よりも高速に解を求めることはできないが，最近の MILP ソルバーの改良により，隣接グラフによっては解が高速に求まる可能性が高い．もし，ヒューリスティックに求められたクリークよりも大きいクリークが短い時間で求められれば，SAT ソルバーを使った解法の効率化を行うことができる．

Heule ら[9]は，少ない状態数から順次 SAT を使って consistent DFA を探していた．SAT の適用回数の削減を考えれば，ある状態数以下の DFA を探すアルゴリズムが効果的であると考えられる．具体的には，最大クリークが最小状態数に近くない場合などは効率化を図ることができる．

本章では，提案アルゴリズムとその個別手法について説明する．

3.1 提案アルゴリズム

次に，提案するアルゴリズムを示す．

入力：APTA，出力：Minimum Consistent-DFA
APTA の各頂点間について，状態マージが可能か求める
ヒューリスティックに Minimum Consistent-DFA(D1)を求める
ヒューリスティックに最大クリーク(C1)を求める
if D1 の状態数=C1 のサイズ: return D1
C1 を初期解として，MILP ソルバーにより最大クリーク（C2）を求める
if D1 の状態数=C2 のサイズ: return D1
upperSize=D1 の状態数-1
lowerSize=C2 のサイズ
lowerSize 以上，upperSize 以下で Minimum Consistent DFA(D2)を求める
if D2 が存在：return D2
return D1

提案するアルゴリズムはヒューリスティックな手法と MILP による手法と SAT によ

る手法を用いたハイブリッドな手法である．個々の手法は，全体の計算時間を短くするために適所において使われる．まず，ヒューリスティックに Consistent-DFA を求める．これは，状態数の上界を与える．次に，APTA の状態間の隣接辺を求め，最大クリークを求める．クリークは異なる色に割り当てなければならない状態の集合を表現している．そのため，Consistent-DFA の下界を与える．上界＝下界であれば，解が見つかったことになる．最大クリークは，ヒューリスティックに求めることができるが，MILP ソルバーを使って厳密に求めることもできる．このとき，ヒューリスティックに求めた解を初期解として与えることで，効率化を図ることができる．最大クリークの頂点数よりも上界が高ければ，その間に Minimum Consistent-DFA が存在することになる．従来研究では上界を用いていないが，本研究では上界も用いて，Minimum Consistent-DFA を探す．この方法には，下界から順次求める，上界から順次求める，二分探索を用いる手法を実現した．下界からの場合は，特定の彩色数で解があるかを調べ，上界からの場合は，ある彩色数以下に解があるかを調べ，２分探索の場合はある彩色数の範囲に解が存在するかを調べる．これには，SAT を使うことができる．

次節より，各処理で使われている手法について説明する．

3.2 Minimum Consistent-DFA 問題のヒューリスティック解の導出

まず，提案アルゴリズムではヒューリスティックな手法を使って Minimum Consistent-DFA を求める．これは，Blue-fringe[4]アルゴリズムを使って行うことができる．APTA の状態数を|Q|としたとき，APTA は[0,…,|Q|-1]というリストで表現できる．APTA の各状態は開始状態から各入力記号に対して幅優先で番号づけられているものとする．このリストでは，状態iがi未満のどの状態とマージしたかを表現する．番号i未満の状態とマージしていなければ，i とする．この表現を使って，次のようにアルゴリズムを書くことができる．関数 stateMerge は map で表現された DFA において状態iとjをマージし，作成された DFA を返す．

```
入力：APTA，出力：DFA
map=[0,1,…,|Q|-1]
for i in [1,…,|Q|-1]:
   if map[i]==i:   #i 未満の状態とマージしていないことを表現
      for j in [0,…,i-1]:
        if 状態 i が状態 j にマージ可能:
           map=stateMerge(map,i,j)
           break
return map
```

3.3 最大クリーク問題のヒューリスティック解の導出

ヒューリスティックにクリークを求めるためには，APTA の各状態を隣接辺の数を使って降順にソートし，先頭からクリークを作成する．

```
入力: 隣接辺数で降順にソートされた APTA の状態のリスト，出力：クリーク
l=[q0, q1, …, q|Q|−1]
C=[q0]
for i in [1,…, |Q|-1]:
   if 状態 l[i]は C のどの状態ともマージできない:  C.append(i)
return C
```

3.4 最大クリーク問題の厳密解の導出

本問題には，3.3 で求めたクリークを初期解として与えた，次の MILP の問題を解く．初期値を与えることによって効率的に解を求めることができる．次の定式化で，1 の値を持つ変数c_iが最大クリークのメンバーとなる．

最大化　$c_0 + c_1 + \cdots + c_{|Q|-1}$

$c_i + c_j \leq 1, \forall i, j \in Q$, 状態 i と状態 j はマージ可能

$c_i \in \{0,1\}, \forall i \in Q$

3.5 Minimum Consistent-DFA の厳密解の導出

3.1 節で述べたように，上界である Minimum Consistent-DFA のヒューリスティック解の状態数が下界を与える最大クリーク問題の厳密解での状態数と一致しない場合，下界以上，上界未満の状態数を持つ解が存在する可能性がある．2.3 節で述べたように，Minimum Consistent-DFA 生成問題において，頂点彩色問題は最大クリーク問題よりもより良い解を与える可能性があるが，計算時間がかかる割にはあまり下界をあげることはない．そのため，最大クリークで得られる状態数を彩色数の下界として与える．

SAT による定式化は式(2)に与えられている．同様の定式化は MILP によっても可能であるが，従来研究[8]で示されているように，実質的に解くことは困難である．そのため，本稿では，SAT による定式化で解を求めるようにした．

SAT による定式化では，色の上限および下限を与える定式化が可能である．これを使って，Minimum Consistent-DFA 生成問題の厳密解を求める．手法としては，前述のように，下界から探す，上界から探す，２分探索を使って探す手法を実現した．

4. 数値実験

本章では，提案手法に関する数値実験について述べる．数値実験は，二つのベンチマークを用いた．一つは，Abbadingo[2]においてベンチマーク問題として公開されているものである．もう一つは，ランダムに生成した DFA から生成される事例を使う．

実験では，使用言語は Python2.5.5 を用い，MILP ソルバーとして CPLEX12.1，SAT ソルバーとして CLASP13.1[11]を利用した．

4.1 Abbadingo コンペティションでのベンチマークによる評価

このベンチマークは web より入手可能である．このベンチマーク問題は４～２１状態の DFA からランダムに生成された長さ０～２９の文字列が提示されている．ラベル数が２以上（実際は２）の 770 個のデータセットからなっている．入力記号数は２，出力記号数は２である．記号列数は 516～549 であった．作成された APTA に関して，状態数は 497～530 であった．密度（実際の遷移数／可能な遷移数）の平均はすべての事例についてほぼ 0.96 であり，ほとんどの状態で２種類の入力記号に対して遷移が定義されていることになる．また，全ての状態について，一つの出力記号を持っている．隣接グラフ（二つの状態がマージできない場合は辺を持ち，マージできる場合は辺を持たない）の密度（実際の辺の数／可能な辺の数）は，最低 0.02，最大 0.64，平均 0.51 であった．770 個のデータセット中，ヒューリスティックに求めた Minimum Consistent-DFA の彩色数と最大クリークの状態数が一致した事例が 286 個あり，これらは評価より除外する．

まず，ヒューリスティックに求めた最大クリークのサイズと MILP ソルバーを使って厳密に求めた最大クリークのサイズを比較する．図４は全体的な傾向をグラフ化したものであるが，(厳密な最大クリーク-近似による最大クリーク)は最大４，最小０，平均 0.92，分散 0.99 であり，ヒューリスティックな手法により求められた最大クリークはかなり良い近似を与えている．図５は(最小の彩色数-最大クリークの状態数)をグラフ化した．図５を見ると，最大クリークの最適解を求めることで，最適解との差が縮まっていることがわかる．

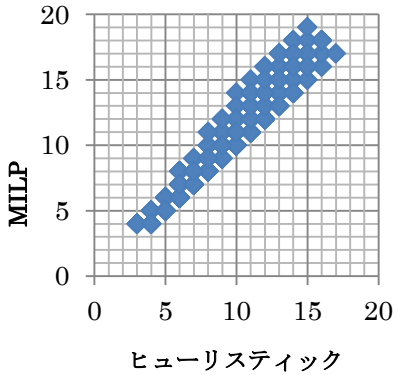

図４　最大クリークのサイズの比較

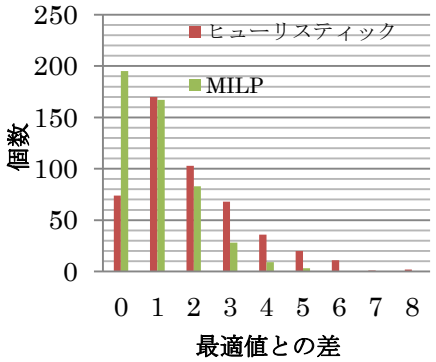

図５　最適値と最大クリークの比較

この実験データにおいては，最大クリークと最適解の差が小さいことから，SAT を使った厳密解の算出では，最大クリークの数から一つづつ彩色数を増やし，最初に見つけた Consistent-DFA を Minimum Consistent-DFA 生成問題の解とする．このとき，ヒューリスティックに求めた最大クリークを使って SAT で解いた場合に必要な時間と MILP で最大クリークを厳密に求め SAT で解いた場合の合計時間を比較する．図６に示したように，一次式で近似した場合の傾きから，平均的には MILP の厳密解を使うことで 50%の時間で解が求められていることがわかる．図７には MILP による最大クリークがヒューリスティックによる Minimum Consistent-DFA の彩色数と一致しなかった 484 データセットについて，SAT 一回当たりの平均時間をグラフ化した．このグラフより，MILP による最大クリークを用いることによって平均 70%の時間で解が求まることがわかる．これは SAT において固定される色が増えたことによると考えられる．

このデータセットの場合，ヒューリスティックに最大クリークを求める平均時間は 0.31 秒であるのに対し，MILP により最大クリークを求める平均時間は 6.46 秒と非常に高速であり，SAT で解く時間を考えた時に MILP による最大クリークを用いる利点は大きい．図８に最小彩色数に対する計算時間を示した．SAT で解くべき変数が増えることから最小彩色数が増えると計算時間も増える傾向にあるが，MILP によって最大クリークの最適解を得ることで，計算時間の短縮に効果があることが分かった．

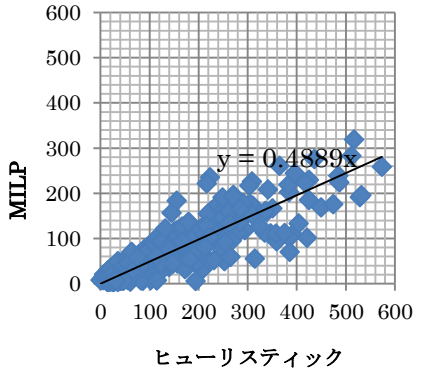

図６　解が求まるまでの時間比較（秒）

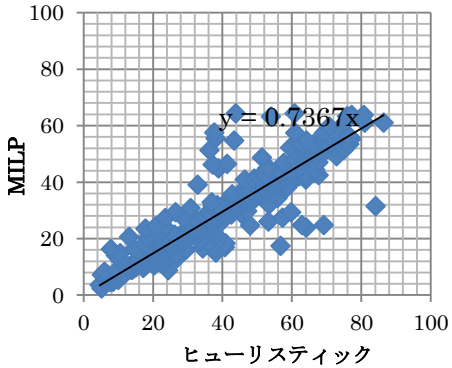

図７　SAT 一回当たりの平均時間（秒）

4.1 ランダムに生成したデータセットによる数値実験

次に，ランダムに生成した DFA を使った実験結果について述べる．この場合，事例を生成する元となる DFA の状態数は 15 とした．入力記号・出力記号ともに２種類である．最大記号列長は 15 とし，最低は１とした．生成した事例数は，300，400，800

とし，各 10 回ずつ同じ DFA より事例を生成した．この実験では，比較的スパースな APTA を生成している．APTA の平均密度は 0.8，出力記号が一つである状態数の割合は 0.35，隣接グラフの密度は 0.08 である．APTA の状態数は，各事例数について，平均 965，1227，2101 であった．

各事例数について，（最小彩色数-最大クリークサイズ）の平均は，ヒューリスティックによる最大クリークでは，4.4，2.7，2.1 であり，MILP による最大クリークでは，2.9，2.3，1.8 となった．事例数が増えるとヒューリスティックによる最大クリークの値がよくなっている．

SAT による探索の手法として，最大クリークから一つずつ彩色数を増やす場合と，ヒューリスティックに求めた Minimum Consistent-DFA の彩色数から一つずつ減らす場合と，２分法で探索場合の実行時間を比較する．図９はヒューリスティックによる最大クリークを用いた場合のグラフである．各データセットについて，下界値より探索した場合を１とした時間をプロットしている．これらをみると，上から探索したものの計算時間が最も短いことがわかる．これは，多くのデータセットでヒューリスティックに生成した Minimum Consistent-DFA が厳密解であるか，彩色数が近いことが原因である．

MILP によって最大クリーク問題の厳密解を求める手法についても実験を行った．この場合，SAT についての計算時間自体は，最高で 45%，最悪で 1.1 倍，平均 90%の計算時間で済む．ただし，隣接グラフがスパースであるため，MILP で最大クリーク問題を解くのに多くの時間を必要とした．この場合，ヒューリスティックな最大クリークを使って SAT で解く時間に比べ，最良で 63%，最悪で 41.1 倍，平均 12.1 倍の計算時間を必要とした．このような場合は，最大クリーク問題を MILP で解くのは好ましくないことを示している．

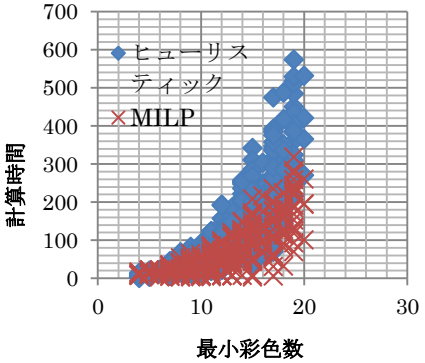

図８　最小彩色数に対する計算時間（秒）

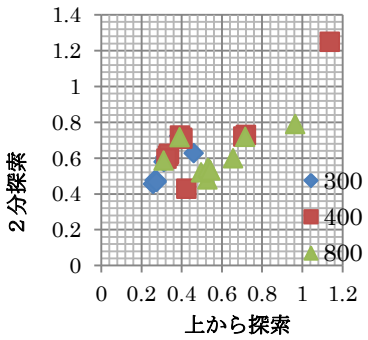

図９　下から探索を１としたときの時間比

5. おわりに

本稿では，ヒューリスティックな手法，MILP による手法，SAT による手法を組み合わせた Minimum Consistent-DFA 生成問題のハイブリッド解法を提案し，有効性を数値実験で検証した. Abbadingo ベンチマーク問題のような密な隣接グラフを持つ APTA の場合，MILP によって高速に最大クリーク問題を解くことができ，解を高速に求めることが示せた. 隣接グラフが疎な場合は, 最大クリーク問題を解く時間が長いため，MILP を用いるのは不利である．また，ヒューリスティックに見つけた上界とクリークによる下界を使った手法を提案し，効果があることを確かめた．

参考文献

1) Oncina, J., Garca, P. : Inferring regular languages in polynomial update time, vol. 1 of Machine Perception and Artificial Intelligence, pp.49-61 (1992).
2) Lang, K. J., Pearlmutter, B. A. and Price, R. A.: Results of the Abbadingo One DFA Learning Competition and a New Evidence-Driven State Merging Algorithm, Grammatical Inference, Lecture Notes in Computer Science, Volume 1433, pp.1-12 (1998).
3) Gold, E.M.: "Language Identification in the Limit". Inf. and Control, 10, pp.447-474 (1967).
4) Lang, K.J.: Faster algorithms for finding minimal consistent DFAs. Technical report, NEC Research Institute (1999).
5) Oliveira, A., Silva, J. M.: Efficient Algorithms for the Inference of Minimum Size DFAs, Machine Learning, 44, pp.93–119 (2001).
6) Coste, F. and Nicolas, J.: Regular Inference as a graph coloring problem, Workshop on Grammar Inference, Automata Induction, and Language Acquisition (ICML' 97) (1997).
7) Bugalho, M. and Oliveira, A. L.: Inference of regular languages using state merging algorithms with search, Pattern Recognition 38, pp.1457-1467 (2005)
8) Inui, N. and Shinano, Y.: Minimizing State Transition Model for Multiclassification by Mixed-Integer Programming, MICAI2005 ADVANCES IN ARTIFICIAL INTELLIGENCE LNCS 3789, pp.473-482 (2005).
9) Heule M. J. H., Verwer S.: Exact DFA identification using SAT solvers, ICGI'10, pp.66-79 (2010)
10) Lucas, S. M., Reynolds T. J.: Learning Deterministic Finite Automaton with a Smart State Labeling Evolutionary Algorithm, PATTERN ANALYSIS AND MACHINE LEARNING, 27, 7, pp.1063-1074 (2005).
11) Clasp: http://www.cs.uni-potsdam.de/**clasp**/
12) Méndez-Díaz, I., Zabala P.: A cutting plane algorithm for graph coloring, Discrete Applied Mathematics 156, pp.159-179 (2008).
13) Czumaj, A., Sohler C.: Testing hypergraph colorability, Theoretical Computer Science 331, pp.37-52 (2005).