データベースシステム 116－70
（1998. 7. 10）

# 部分 MAXSAT を利用した大学情報処理の自動化[1]

盛田 保文　宮崎 修一　岩間 一雄
京都大学大学院情報学研究科

複雑ではあるがそれほど大規模でないデータベース上の情報処理の高速化を目標とする．本研究では，組み合わせ問題（SAT）を利用してデータベース上の問題を解くという方針をとる．変換アルゴリズムの構築が容易であることと，SAT に対する既存の高速アルゴリズムを利用できるといった利点が考えられる．本稿では大学情報処理の一つである研究室配属問題を部分 MAXSAT に変換して解くことを試みた．ランダム例題を用いて，我々の方法と研究室配属問題を直接解くアルゴリズムとを比較した．その結果，変換を用いた方法がある程度有効であることが分かった．

# Partial MAXSAT for Combinatorial Optimization on University Databases

Yasufumi MORITA, Shuichi MIYAZAKI and Kazuo IWAMA
School of Informatics, Kyoto University

We focus on combinatorial optimization problems on databases which are complicated but not too large. In this paper, we select the Student Assignment Problem (SAP), in which we assign university students to university laboratories according to their desire and records. Our approach is not to solve the original problem directly but to translate it to Partial MAXSAT, a combinatorial optimization problem, to use fast and efficient SAT algorithms. We compare our approach with three simple algorithms and show the efficiency of our method.

## 1 はじめに

近年，情報の電子化が進むにつれて，膨大な量のデータを持つデータベースを取り扱う必要性がでてきている．したがって，大きなサイズのデータベースに対する情報の検索などの要求は当然生じ，このような単純な要求に対して高速に動作するためのデータ構造や情報処理技術の研究が重要になっている．現在のような高度情報化社会では，単純な質問処理のみならず，蓄積されたデータに対する複雑な質問処理の要求も生じるのであるが，データ構造の複雑さや，膨大なデータ量を考慮したとき，このような複雑な質問処理はしばしば現実的でないと考えられがちである．しかし，あまり大量ではないデータに対して，複雑な質問処理を行なわせたいという場面はしばしば見受けられる．例えば，大学の情報を学科単位で見ると，せいぜい教官数 30，学生数 100 といった比較的規模の小さなものである．このようなデータから時間割を作成するといった作業は人間の手で行なわれており，自動化されていないのが現状である．

本研究では，上述のような，複雑ではあるがそれほど大規模ではないデータベース上での問題をどのように取り扱うかを研究テーマとする．もちろん，個々の問題に対してそれぞれ違ったアルゴリズムを開発するといったアプローチが考えられるが，そのような効率の良いアルゴリズムを開発するのは，入力が複雑なデータからなっていることを考えると，うまく行かないであろうと思われる．本研究では，データベース上での問題を論理式の充足可能性問題（SAT）に変換するアルゴリズムを開発するという方針をとる．データベース上での問題は，複数の表から新たな表 $T$ を作る問題であると考えることができる．論理式への変換は，この $T$ の満たすべき条件を書いていくといった，比較的簡単な方法で行なうことができる．

このように，変換アルゴリズムを開発するといったアプローチには，以下のような利点が考えられる．(i) 上述のように，解法アルゴリズムの開発よりも，変換アルゴリズムの開発の方がかなり容易であり，また，変

[1]本研究は，科学研究費重点領域研究「高度データベース」の支援の元に行なわれたものである．

換アルゴリズムは高速に動作する．(ii) 変換された論理式を解くために，既存の高速なSATアルゴリズムを利用できる．(iii) 近年注目を集めている近似の概念を利用することができ，[GW94] のような，近似アルゴリズムを用いることができる．

我々は，上記のような，大学のデータベースから時間割を作成する問題をSATに変換して解くことを試み，この方法がある程度有効であることを実験により示した [MIK96]．実験には，九州大学工学部情報工学科のデータに基づいた現実的な例題を用いた．現実社会の問題は一般に，完全に満足しなければならない条件と，出来るだけ満足したい条件からなっている，このような問題を解く場合には，従来のMAXSATではなく我々が提案した部分MAXSATという最適化問題を用いるのが有効であることも示した [CIKM97]．

本稿では，大学のデータベースを入力とする別の問題として，研究室配属問題をとりあげる．この問題は大学院に合格した学生を，合格順位や志望に基づき，研究室に配属させる問題である．この問題も時間割作成問題と同様に，学生の希望の書き方や評価関数（学生の満足度）の定め方により様々な定式化が考えられるが，本稿では比較的自然で，簡単だと思えるものを取り扱う．時間割作成問題 [MIK96] と同様に，部分MAXSATに変換し，従来のMAXSATアルゴリズムである局所探索法を使うことによって，学生の満足度ができるだけ大きな配属を求めることが目標である．

学生数36，研究室数9の，ある程度現実に則したランダム例題を用いて実験を行なった．研究室配属問題を直接解く3つのアルゴリズムと比較したところ，我々の手法が一番良い解を出力するような例題が存在することが分かった．本問題においても，変換を用いた方法の有効性を示すことができた．

## 2 基本的事項

### 2.1 充足可能性問題

充足可能性問題（SAT）は最も基本的な組合せ問題であり，人工知能 [KS96]，VLSI設計 [Gu96]，誤り検出 [Gu96] といった様々な応用に利用されている．また，長年に渡って解法アルゴリズムの研究もなされており，バックトラック法 [DABC93, BP93]，計数法 [Iwa89]，局所探索法 [Gu92, SLM92] といった，様々なアルゴリズムが開発されている．中でも局所探索法はかなり高速に動作することが報告されている [Mor93, CI95, CI96]．

SATおよびMAXSATは，以下のような問題である．$x_1, x_2, \cdots, x_n$ を変数といい，これらは真（1）または偽（0）の値を取る．変数 $x_i$ およびその否定 $\overline{x_i}$ を**リテラル**という．リテラルの論理和を項といい，項の論理積を**CNF論理式**という．**SAT**はCNF論理式が与えられたとき，全ての項を1にするような，変数への0，1の割り当てがあるかどうかを問う判定問題である．また，**MAXSAT**はできるだけ多くの項を1にする割り当てを見つける最適化問題である．**部分MAXSAT**[CIKM97] はMAXSATを一般化した問題である．部分MAXSATは，入力として2つの論理式 $f_A$，$f_B$ の積 $f = f_A \cdot f_B$ が与えられ，$f_A$ の全ての項を1にし，$f_B$ の項を出来るだけ多く1にする割り当てを見つける問題である．

本研究では，SATの解法アルゴリズムとして高速である局所探索法 [CI96] を用いる．局所探索法は，基本的に以下のように動作するアルゴリズムである．

(1): ランダムに，ある割り当てを選ぶ．
(2): 隣接する割り当てを見て，その中で充足される項数が一番多い割り当てに移動する．
(3): (2) を繰り返す．

ここで，隣接する割り当てとは，現在の割り当てのうち，どれか一つの変数の割り当てだけを反転させたものを指す．すなわち，$n$ 変数の論理式の場合，隣接する割り当ては $n$ 個あることになる．このアルゴリズムは，充足する項を多くするような割り当てへと進んでいく．したがって，このアルゴリズムは判定問題（SAT）のみならず，最適化問題（MAXSAT）に対しても有効である．局所探索法を部分MAXSATに適用させる場合には，入力の $f_A$ の項に適当な重みをつけることにより，ある程度良い解が得られることが実験的に分かっている [CIKM97]．

### 2.2 研究室配属問題

研究室配属問題は，大学院合格者の志望研究室リスト，成績等に基づき，合格者を研究室に配属する問題である．直観的には，出来るだけ多くの学生が，希望の研究室に行けることが望ましいわけであるが，評価関数等の定め方により様々な定式化が考えられる．ここでは比較的簡単な定式化を考えてみる．

研究室配属問題の入力は以下のものからなる．

(i): 学生の集合 $S = \{s_1, s_2, \cdots, s_N\}$.
(ii): 研究室の集合 $L = \{l_1, l_2, \cdots, l_M\}$. （ただし，$N = cM$. $c$ は定数. ）
(iii): 学生の成績順位. すなわち，$S$ から $\{1, 2, \cdots, N\}$ への全単射 $rank$. （同順の学生はいない. ）
(iv): 各学生 $s_i$ に対して，研究室の志望順位. すなわち，$L$ から $\{1, 2, \cdots, M\}$ への全単射 $choice_i$.

このとき，以下のような制約条件を満たす解，すなわち，$S$ から $L$ への写像 $assign$ を求める.

(a): 各学生は必ずどこか1つの研究室に配属される. すなわち，各 $i$ に対して $|assign(s_i)| = 1$.
(b): 各研究室にはちょうど $c$ 人ずつの学生が配属される. すなわち，各 $j$ に対して $|\{i|assign(s_i) = l_j\}| = c$.
(c): 次のような学生 $s_{i_1}$，$s_{i_2}$ が存在しない. 学生 $s_{i_1}$ の方が $s_{i_2}$ よりも成績が良い. すなわち，$rank(s_{i_1}) < rank(s_{i_2})$. 学生 $s_{i_2}$ の配属された研究室 $assign(s_{i_2})$ の志望順位 $choice_{i_2}(assign(s_{i_2}))$ は，学生 $s_{i_1}$ のその研究室の志望順位 $choice_{i_1}(assign(s_{i_2}))$ よりも低い. すなわち，$choice_{i_2}(assign(s_{i_2})) > choice_{i_1}(assign(s_{i_2}))$. 学生 $s_{i_1}$ に関して，自分の配属された研究室 $assign(s_{i_1})$ の志望順位 $choice_{i_1}(assign(s_{i_1}))$ は $choice_{i_1}(assign(s_{i_2}))$ より低い. すなわち，$choice_{i_1}(assign(s_{i_1})) > choice_{i_1}(assign(s_{i_2}))$.

次に，解のコストを定義する. これは，各学生が志望順位第 $i$ 位の研究室に配属されたときの不満度を $i$ と定義し，その総和を解のコストとする. すなわち，$\Sigma_i(choice_i(assign(s_i)))$ を最小にする解を求める問題である.

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $s_3$ | $l_1$ | $l_2$ | $l_3$ | $l_4$ | $l_5$ |
| $s_{10}$ | $l_2$ | $l_1$ | $l_4$ | $l_3$ | $l_5$ |

表 1: 学生の配属希望例

ここで，条件 (c) について簡単に触れておく. 条件 (c) は成績を考慮に入れたものである. 簡単のため，以下の例を考えてみる. 2人の学生 $s_3$ と $s_{10}$ は，$s_3$ の方が成績が良かったとする. 5つの研究室に対する志望順位が表1の通りであったとする. このとき，$s_3$ が研究室 $l_5$ に，$s_{10}$ が研究室 $l_3$ に配属された場合 $s_3$ は，自分よりも成績が低く，研究室 $l_3$ への志望順位を低くつけている $s_{10}$ に研究室 $l_3$ をとられてしまったことになる. しかも，$s_3$ はその研究室 $l_3$ よりも低い志望の研究室に配属されたわけである. (c) はこのようなことが起こらないことを表す条件である.

## 3 変換アルゴリズム

本節では，研究室配属問題を部分 MAXSAT に変換する. 変換は時間割作成問題からの変換 [MIK96] と同様に，部分 MAXSAT を利用して研究室配属問題を解くことが出来るようにする. すなわち，部分 MAXSAT の解（変数への 0，1 割り当て）から各学生の研究室への配属を求めることができ，部分 MAXSAT で $f_B$ の項をより多く充足しているほど，それから求まった配属のコストが小さいものになっているという変換を行なう.

### 3.1 変換アルゴリズム 1

まず，以下のような変換方法を考えてみる. 合格者数を $N$，研究室数を $M$ （$N = cM$）とし，変数は，$x_{i,j}$ （$1 \leq i \leq N$，$1 \leq j \leq M$）を用いる. $x_{i,j} = 1$ を合格者 $s_i$ が研究室 $l_j$ に配属された状態と考える.

**変換アルゴリズム 1.**

ステップ 1： 各 $i$ （$1 \leq i \leq N$）に対して，$(x_{i,1} + x_{i,2} + \cdots + x_{i,M})$ を作る. 合格者 $s_i$ がどの研究室にも配属されないときこの項は 0 になる.
ステップ 2： 各 $i$，$j_1$，$j_2$ （$1 \leq i \leq N$，$1 \leq j_1 < j_2 \leq M$）に対して項 $(\overline{x_{i,j_1}} + \overline{x_{i,j_2}})$ を作る. 合格者 $s_i$ が 2つ以上の研究室に配属になったとき，これらのうちのいずれかの項が 0 になる.

ステップ3： 各 $j$ $(1 \leq j \leq M)$ および，$i_1, i_2, \cdots, i_{c+1}$ $(1 \leq i_1 < i_2 < \cdots < i_{N-c+1} \leq N)$ に対して，項 $(x_{i_1,j} + x_{i_2,j} + \cdots + x_{i_{N-c+1},j})$ を作る．研究室 $l_j$ への配属人数が $c-1$ 人以下のとき，これらのうちのいずれかの項が 0 になる．

ステップ4： 各 $j$ $(1 \leq j \leq M)$ および，$i_1, i_2, \cdots, i_{c+1}$ $(1 \leq i_1 < i_2 < \cdots < i_{c+1} \leq N)$ に対して，項 $(\overline{x_{i_1,j}} + \overline{x_{i_2,j}} + \cdots + \overline{x_{i_{c+1},j}})$ を作る．研究室 $l_j$ に $c+1$ 人以上の学生が配属されたとき，これらのうちのいずれかの項が 0 になる．

ステップ5： 以下のような条件を満たす学生 $s_{i_1}$，$s_{i_2}$ および研究室 $l_j$ を考える．(i) $rank(s_{i_1}) < rank(s_{i_2})$，(ii) $choice_{i_2}(l_j) > choice_{i_1}(l_j)$．ここで，$choice_{i_2}(l_j) = t$ とし，$choice_{i_1}(l_{p_q}) = q$ $(1 \leq q \leq t)$ とする．（つまり，学生 $s_{i_1}$ の第 $q$ 志望研究室が $l_{p_q}$ である．） このとき，項 $(\overline{x_{i_2,j}} + x_{i_1,p_1} + x_{i_1,p_2} + \cdots + x_{i_1,p_t})$ を作る．合格者 $s_{i_1}$，$s_{i_2}$，研究室 $l_j$ に関して条件 (c) のようなことが起こった場合，この項が 0 になる．このような $s_{i_1}$，$s_{i_2}$，$l_j$ の組み合わせ全てに対して項を作る．

ステップ6： 各 $i$，$j$ $(1 \leq i \leq N,\ 1 \leq j \leq M)$ に対して，項 $(\overline{x_{i,j}})$ を $choice_i(l_j)$ 個作る．

部分 MAXSAT として，$f_A$ にステップ1からステップ5の項を，$f_B$ にステップ6の項を割り当てる．ステップ1，2の項を全て充足させると，各合格者はちょうど1つの研究室に割り当てられ，条件 (a) が満たされる．また，ステップ3，4の項が全て充足されると，条件 (b) が満たされる．変換中にも書いたように，ステップ5の項を全て充足させることは，条件 (c) を満たすことに対応している．ステップ6の項は，各 $i$ に対して，$1+2+\cdots+(M-1) = \frac{M(M-1)}{2}$ 個存在する．$s_i$ が志望順位 $t$ 番目の研究室に配属されたとき，これらのうち $t$ 項が不充足になる．すなわち，志望順位の高い研究室に配属されるとより多くの項が満たされることになる．

この変換に必要な項数は，ステップ1が $O(N)$，ステップ2が $O(NM^2)$，ステップ5が $O(N^2M)$，ステップ6が $O(NM^2)$ である．ところが，ステップ3には $O(N^{c-1}M)$，ステップ4には $O(N^{c+1}M)$ の項が必要である．1つの研究室に（例えば7人といった）多くの学生が配属されるような問題の場合にはこの変換では，項の数が多くなってしまい，変換のオーバーヘッドが MAXSAT アルゴリズムの効果を打ち消してしまう．そこで，次の変換を考えてみる．

## 3.2 変換アルゴリズム 2

変換アルゴリズム1は，研究室 $l_j$ に対して $N$ 変数 $x_{i,j}$ $(1 \leq i \leq N)$ を使用し，このうちちょうど $c$ 個の変数に1が入るという条件を項で表した．直観的には，1つの箱を用意し，その中にちょうど $c$ 人の学生を入れたことになる．そのため，項の数が $O(N^{c+1})$ 必要になった．本節で取り扱うアルゴリズムには $N^2$ 変数を使用する．各研究室は $c$ 個の箱を用意し，1つの箱にちょうど1人の学生を入れるような変数の使い方をする．変数は $x_{i,j,k}$ $(1 \leq i \leq N, 1 \leq j \leq M, 1 \leq k \leq c)$ を使用し，$x_{i,j,k} = 1$ という割り当ては，学生 $s_i$ が研究室 $l_j$ の第 $k$ 枠に配属されることを意味する．従って，各学生に対する変数はちょうど1つが1で，各枠に対する変数もちょうど1つが1という項を作るだけで良く，変換アルゴリズム1よりも項の数を減らすことができる．

**変換アルゴリズム 2．**

ステップ1： 各 $i$ $(1 \leq i \leq N)$ に対して，$(x_{i,1,1} + x_{i,1,2} + \cdots + x_{i,1,c} + x_{i,2,1} + x_{i,2,2} + \cdots + x_{i,2,c} + \cdots + x_{i,M,1} + x_{i,M,2} + \cdots + x_{i,M,c})$ を作る．合格者 $s_i$ がどの研究室にも配属されないときこの項は 0 になる．

ステップ2： 各 $i$，$j_1$，$j_2$，$k_1$，$k_2$ $(1 \leq i \leq N,\ 1 \leq j_1 \leq M,\ 1 \leq j_2 \leq M,\ 1 \leq k_1 \leq c,\ 1 \leq k_2 \leq c,\ j_1 \neq j_2$ または $k_1 \neq k_2)$ に対して項 $(\overline{x_{i,j_1,k_1}} + \overline{x_{i,j_2,k_2}})$ を作る．合格者 $s_i$ が2つ以上の研究室に（または，同じ研究室に重複して）配属になったとき，これらのうちのいずれかの項が 0 になる．

ステップ3： 各 $j$，$k$，$(1 \leq j \leq M,\ 1 \leq k \leq c)$ に対して，項 $(x_{1,j,k} + x_{2,j,k} + \cdots + x_{N,j,k})$ を作る．研究室 $l_j$ の $k$ 枠に誰も配属されないとき，この項が 0 になる．

ステップ4： 各 $j$，$k$，$(1 \leq j \leq M,\ 1 \leq k \leq c)$ および，$i_1, i_2$ $(1 \leq i_1 < i_2 \leq N)$ に対して，項 $(\overline{x_{i_1,j,k}} + \overline{x_{i_2,j,k}})$ を作る．研究室 $l_j$ の $k$ 枠に2人以上配属されたとき，これらのうちいずれかの項が 0 になる．

ステップ5： 以下のような条件を満たす学生 $s_{i_1}$，$s_{i_2}$ および研究室 $l_j$ を考える．(i) $rank(s_{i_1}) < rank(s_{i_2})$，(ii) $choice_{i_2}(l_j) > choice_{i_1}(l_j)$．ここで，$choice_{i_2}(l_j) = t$ とし，$choice_{i_1}(l_{p_q}) = q$ $(1 \leq q \leq t)$ とする．（つまり，学生 $s_{i_1}$ の第 $q$ 志望研究室が $l_{p_q}$ である．） このとき，$1 \leq k \leq c$ に対して，項 $(\overline{x_{i_2,j,k}} + x_{i_1,p_1,1} + x_{i_1,p_1,2} + \cdots + x_{i_1,p_1,c} + x_{i_1,p_2,1} + x_{i_1,p_2,2} + \cdots + x_{i_1,p_2,c} + \cdots + x_{i_1,p_t,1} + x_{i_1,p_t,2} +$

$\cdots + x_{i_1,p_t,c})$ を作る．合格者 $s_{i_1}$，$s_{i_2}$，研究室 $l_j$ に関して条件 (c) のようなことが起こった場合，この項が 0 になる．このような $s_{i_1}$，$s_{i_2}$，$l_j$ の組合せ全てに対して項を作る．

ステップ 6： 各 $i$，$j$ $(1 \le i \le N,\ 1 \le j \le M)$ に対して，項 $(\overline{x_{i,j,1}})(\overline{x_{i,j,2}})\cdots(\overline{x_{i,j,c}})$ を $choice_i(l_j)$ 個ずつ作る．

アルゴリズム 2 の各ステップはアルゴリズム 1 のステップに対応している．このアルゴリズムの，各ステップによって作られる項の数は，それぞれ，$O(N)$，$O(N^3)$，$O(N)$，$O(N^3)$，$O(N^3)$，$O(N^2M)$ である．変数の数はアルゴリズム 2 の方が多くなってしまうが，$c$ の値が大きい例題に対しては，項の数を格段に減らすことができる．

また，この変換は，$N$（学生数）が $M$（研究室数）の整数倍でない場合をも取り扱うことができる．例えば各研究室に 5 人または 6 人を配属することになったとする．このとき，各研究室では 6 枠分の変数を用意し，1 枠から 5 枠までは必ず 1 人が配属され（ステップ 3 とステップ 4），6 枠目には 1 人または 0 人が配属される（ステップ 4 のみ）とすれば良い．

## 4 実験

実験には，学生数 36，研究室数 9 の例題を使用した．ある程度のランダム性を持たせ，また，研究室の人気の違いにある程度偏りを持たせるため，以下のような方法で例題を生成した．まず，各学生の第 $i$ 希望研究室を $l_i (1 \le i \le 9)$ とする．次に，各学生に対し，ランダムに研究室を 2 つ選びその 2 つの研究室の希望順位を入れ換える．この入れ換え操作を 1 人の学生につき 3 回行なう．こうすることにより，$i$ の値が小さいほど研究室 $l_i$ の人気が高いことになる．上記のように作られた例題 10 個に対して，以下の 4 つのアルゴリズムを用いて解を求め，解の質を比較した．これらのアルゴリズムにより得られる解は，明らかに条件 (a)，(b)，(c) を満たしていることに注意されたい．

**成績優先アルゴリズム:** このアルゴリズムは，成績上位の学生から，まだ空いている研究室の中で一番志望順位の高いものに配属していくアルゴリズムである．このアルゴリズムにより得られる解は，明らかに条件 (a)，(b)，(c) を満たしている．

**希望優先アルゴリズム:** 各研究室 4 人配属の例で説明する．全ての学生の第一希望のみを見る．このとき，もし希望が 4 人以下の研究室があれば，そこを希望している学生は優先的に配属が決まる．5 人以上が志望している研究室は，成績上位 4 人をその研究室に配属させる．次に，残った学生，残った研究室枠に対して，第二希望で同様のことを行なう．第三希望，第四希望，… と，同様のことを行なっていく．

**第一希望優先アルゴリズム:** 第一希望に対して希望優先アルゴリズムを使い，残った学生および研究室枠に成績優先アルゴリズムを適用させる．

**部分 MAXSAT を利用する方法:** 与えられた例題を変換 2 を使い部分 MAXSAT に変換する．局所探索法を使い解を求め，その解から配属を求める．

結果を表 2 に示す．表中の数字はコストを表す．例題 1 および例題 7 は部分 MAXSAT による方法が最良の解を出力した．このうち，例題 7 の結果を表 3 〜 6 に示す．例題 7 を部分 MAXSAT に変換した例題の変数，項，$f_A$ 項の数はそれぞれ 1296, 65624, 59144 であった．

| 例題番号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 成績優先 | 68 | 73 | 74 | 74 | 78 | 76 | 75 | 92 | 74 | 77 |
| 希望優先 | 73 | 65 | 69 | 66 | 73 | 70 | 69 | 90 | 70 | 81 |
| 第一希望優先 | 80 | 70 | 74 | 65 | 76 | 70 | 73 | 89 | 70 | 81 |
| 部分 MAXSAT | 67 | 65 | 69 | 67 | 73 | 70 | 68 | 97 | 70 | 82 |

表 2: 実験結果

## 5 おわりに

本研究では研究室配属問題を部分 MAXSAT を利用して解くことを試み，その有効性を示した．本問題においては，コストの定め方により様々な定式化が考えられるので，他のコスト関数においても部分 MAXSAT が効力を発揮できるのかどうかを検証する．

## 参考文献


[BP93] K. M. Bugrara and P. W. Purdom, jr, "Average time analysis of clause order backtracking," *SIAM J. Comput.*, Vol.22, pp.303-317, 1993.

[CI95] B. Cha and K. Iwama, "Performance test of local search algorithms using new types of random CNF formulas," *Proc. IJCAI-95*, pp.304-310, 1995.

[CI96] B. Cha and K. Iwama, "Adding new clauses for faster local search," *Proc. AAAI-96*, pp.332-337, 1996.

[CIKM97] B. Cha, K. Iwama, Y. Kambayashi and S. Miyazaki, "Local search algorithms for Partial MAXSAT," *Proc. AAAI-97*, pp.263-268, 1997.

[DABC93] O. Dubois, P. Andre, Y. Boufkhad and J. Carlier, "SAT versus UNSAT," *2nd DIMACS Challenge Workshop*, 1993.

[Gu92] J. Gu, "Efficient local search for very largescale satisfiability problem," *SIGACT Bull*, Vol.3, pp.8-12, 1992.

[Gu96] J. Gu, "Satisfiability (SAT) problems in VLSI engineering," *DIMACS volume Series on Discrete Mathematics and Theoretical Computer Science*, 1996.

[GW94] M. X. Goemans and D. P. Williamson, ".878-approximation algorithms for MAX CUT and MAX 2SAT," *Proc. 26th ACM Symposium on Theory of Computing*, pp.422-431, 1994.

[Iwa89] K. Iwama, "CNF satisfiability test by counting and polynomial average time," *SIAM J. Comput.*, pp.385-391, 1989.

[KS96] H. Kautz and B. Selman, "Pushing the envelope: Planning, propositional logic, and stochastic search," *Proc. AAAI-96*, pp.1194-1201, 1996.

[MIK96] S. Miyazaki, K. Iwama and Y. Kambayashi, "Database queries as combinatorial optimization problems," *Proc. CODAS'96*, pp.448-454, 1996.

[Mor93] P. Morris, "The breakout method for escaping from local minima," *Proc. AAAI-93*, 1993.

[SLM92] B. Selman, H. J. Levesque and D. G. Mitchell, "A new method for solving hard satisfiability problems," *Proc. AAAI-92*, pp.440-446, 1992.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $s_1$ | ① | 2 | 3 | 4 | 5 | 8 | 6 | 9 | 7 |
| $s_2$ | ① | 2 | 9 | 3 | 5 | 7 | 6 | 8 | 4 |
| $s_3$ | ① | 8 | 3 | 4 | 5 | 2 | 7 | 6 | 9 |
| $s_4$ | ① | 9 | 3 | 6 | 5 | 4 | 2 | 8 | 7 |
| $s_5$ | ② | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| $s_6$ | ② | 5 | 3 | 4 | 1 | 6 | 7 | 8 | 9 |
| $s_7$ | ④ | 1 | 2 | 3 | 5 | 6 | 7 | 8 | 9 |
| $s_8$ | 1 | ② | 3 | 4 | 5 | 7 | 6 | 8 | 9 |
| $s_9$ | ⑧ | 2 | 3 | 4 | 1 | 6 | 7 | 9 | 5 |
| $s_{10}$ | 1 | ② | 6 | 4 | 5 | 8 | 7 | 3 | 9 |
| $s_{11}$ | ⑥ | 2 | 3 | 1 | 8 | 4 | 7 | 5 | 9 |
| $s_{12}$ | 1 | ⑧ | 9 | 4 | 2 | 6 | 7 | 5 | 3 |
| $s_{13}$ | 1 | 2 | ④ | 7 | 5 | 6 | 9 | 8 | 3 |
| $s_{14}$ | 1 | 2 | ⑦ | 4 | 5 | 6 | 3 | 8 | 9 |
| $s_{15}$ | ③ | 5 | 1 | 6 | 2 | 4 | 7 | 8 | 9 |
| $s_{16}$ | 1 | 2 | ③ | 4 | 6 | 5 | 7 | 8 | 9 |
| $s_{17}$ | 1 | ⑥ | 2 | 3 | 5 | 4 | 7 | 8 | 9 |
| $s_{18}$ | 1 | 2 | ⑤ | 3 | 4 | 6 | 8 | 7 | 9 |
| $s_{19}$ | 1 | 2 | ③ | 9 | 5 | 4 | 7 | 6 | 8 |
| $s_{20}$ | ⑨ | 2 | 6 | 8 | 5 | 3 | 7 | 4 | 1 |
| $s_{21}$ | 1 | ⑥ | 3 | 8 | 5 | 7 | 2 | 4 | 9 |
| $s_{22}$ | 1 | 2 | ③ | 4 | 5 | 6 | 7 | 8 | 9 |
| $s_{23}$ | 1 | ⑨ | 3 | 4 | 5 | 6 | 8 | 2 | 7 |
| $s_{24}$ | ⑧ | 4 | 3 | 2 | 5 | 6 | 7 | 9 | 1 |
| $s_{25}$ | 1 | ⑨ | 3 | 4 | 5 | 2 | 7 | 8 | 6 |
| $s_{26}$ | 1 | 3 | ⑤ | 4 | 2 | 6 | 8 | 7 | 9 |
| $s_{27}$ | ⑧ | 2 | 7 | 1 | 5 | 6 | 3 | 4 | 9 |
| $s_{28}$ | ⑦ | 1 | 3 | 4 | 8 | 6 | 2 | 5 | 9 |
| $s_{29}$ | ⑦ | 2 | 5 | 4 | 9 | 6 | 1 | 8 | 3 |
| $s_{30}$ | 1 | 2 | 3 | ⑦ | 5 | 6 | 9 | 8 | 4 |
| $s_{31}$ | ⑤ | 2 | 4 | 3 | 1 | 7 | 6 | 8 | 9 |
| $s_{32}$ | ④ | 2 | 8 | 1 | 5 | 7 | 6 | 3 | 9 |
| $s_{33}$ | 1 | 7 | 3 | ⑥ | 5 | 8 | 2 | 4 | 9 |
| $s_{34}$ | ④ | 2 | 3 | 7 | 1 | 6 | 5 | 8 | 9 |
| $s_{35}$ | 2 | 1 | 3 | 4 | ⑤ | 8 | 9 | 6 | 7 |
| $s_{36}$ | 1 | 2 | 8 | 5 | 7 | 6 | 4 | 3 | ⑨ |

表 3: 成績優先アルゴリズムの結果（例題 7）

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $s_1$ | ① | 2 | 3 | 4 | 5 | 8 | 6 | 9 | 7 |
| $s_2$ | ① | 2 | 9 | 3 | 5 | 7 | 6 | 8 | 4 |
| $s_3$ | ① | 8 | 3 | 4 | 5 | 2 | 7 | 6 | 9 |
| $s_4$ | ① | 9 | 3 | 6 | 5 | 4 | 2 | 8 | 7 |
| $s_5$ | ② | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| $s_6$ | ② | 5 | 3 | 4 | 1 | 6 | 7 | 8 | 9 |
| $s_7$ | ④ | 1 | 2 | 3 | 5 | 6 | 7 | 8 | 9 |
| $s_8$ | 1 | ② | 3 | 4 | 5 | 7 | 6 | 8 | 9 |
| $s_9$ | ⑧ | 2 | 3 | 4 | 1 | 6 | 7 | 9 | 5 |
| $s_{10}$ | 1 | 2 | ⑥ | 4 | 5 | 8 | 7 | 3 | 9 |
| $s_{11}$ | ⑥ | 2 | 3 | 1 | 8 | 4 | 7 | 5 | 9 |
| $s_{12}$ | 1 | ⑧ | 9 | 4 | 2 | 6 | 7 | 5 | 3 |
| $s_{13}$ | 1 | 2 | ④ | 7 | 5 | 6 | 9 | 8 | 3 |
| $s_{14}$ | 1 | 2 | ⑦ | 4 | 5 | 6 | 3 | 8 | 9 |
| $s_{15}$ | ③ | 5 | 1 | 6 | 2 | 4 | 7 | 8 | 9 |
| $s_{16}$ | 1 | 2 | ③ | 4 | 6 | 5 | 7 | 8 | 9 |
| $s_{17}$ | 1 | ⑥ | 2 | 3 | 5 | 4 | 7 | 8 | 9 |
| $s_{18}$ | 1 | 2 | ⑤ | 3 | 4 | 6 | 8 | 7 | 9 |
| $s_{19}$ | 1 | 2 | ③ | 9 | 5 | 4 | 7 | 6 | 8 |
| $s_{20}$ | ⑨ | 2 | 6 | 8 | 5 | 3 | 7 | 4 | 1 |
| $s_{21}$ | 1 | ⑥ | 3 | 8 | 5 | 7 | 2 | 4 | 9 |
| $s_{22}$ | 1 | 2 | 3 | 4 | ⑤ | 6 | 7 | 8 | 9 |
| $s_{23}$ | 1 | ⑨ | 3 | 4 | 5 | 6 | 8 | 2 | 7 |
| $s_{24}$ | ⑧ | 4 | 3 | 2 | 5 | 6 | 7 | 9 | 1 |
| $s_{25}$ | 1 | ⑨ | 3 | 4 | 5 | 2 | 7 | 8 | 6 |
| $s_{26}$ | 1 | ③ | 5 | 4 | 2 | 6 | 8 | 7 | 9 |
| $s_{27}$ | ⑧ | 2 | 7 | 1 | 5 | 6 | 3 | 4 | 9 |
| $s_{28}$ | ⑦ | 1 | 3 | 4 | 8 | 6 | 2 | 5 | 9 |
| $s_{29}$ | ⑦ | 2 | 5 | 4 | 9 | 6 | 1 | 8 | 3 |
| $s_{30}$ | 1 | 2 | 3 | 7 | 5 | 6 | ⑨ | 8 | 4 |
| $s_{31}$ | ⑤ | 2 | 4 | 3 | 1 | 7 | 6 | 8 | 9 |
| $s_{32}$ | ④ | 2 | 8 | 1 | 5 | 7 | 6 | 3 | 9 |
| $s_{33}$ | 1 | ⑦ | 3 | 6 | 5 | 8 | 2 | 4 | 9 |
| $s_{34}$ | ④ | 2 | 3 | 7 | 1 | 6 | 5 | 8 | 9 |
| $s_{35}$ | ② | 1 | 3 | 4 | 5 | 8 | 9 | 6 | 7 |
| $s_{36}$ | 1 | 2 | 8 | ⑤ | 7 | 6 | 4 | 3 | 9 |

表 4: 希望優先アルゴリズムの結果（例題 7）

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $s_1$ | ① | 2 | 3 | 4 | 5 | 8 | 6 | 9 | 7 |
| $s_2$ | ① | 2 | 9 | 3 | 5 | 7 | 6 | 8 | 4 |
| $s_3$ | ① | 8 | 3 | 4 | 5 | 2 | 7 | 6 | 9 |
| $s_4$ | ① | 9 | 3 | 6 | 5 | 4 | 2 | 8 | 7 |
| $s_5$ | ② | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| $s_6$ | ② | 5 | 3 | 4 | 1 | 6 | 7 | 8 | 9 |
| $s_7$ | ④ | 1 | 2 | 3 | 5 | 6 | 7 | 8 | 9 |
| $s_8$ | 1 | ② | 3 | 4 | 5 | 7 | 6 | 8 | 9 |
| $s_9$ | ⑧ | 2 | 3 | 4 | 1 | 6 | 7 | 9 | 5 |
| $s_{10}$ | 1 | 2 | ⑥ | 4 | 5 | 8 | 7 | 3 | 9 |
| $s_{11}$ | ⑥ | 2 | 3 | 1 | 8 | 4 | 7 | 5 | 9 |
| $s_{12}$ | 1 | ⑧ | 9 | 4 | 2 | 6 | 7 | 5 | 3 |
| $s_{13}$ | 1 | 2 | ④ | 7 | 5 | 6 | 9 | 8 | 3 |
| $s_{14}$ | 1 | 2 | ⑦ | 4 | 5 | 6 | 3 | 8 | 9 |
| $s_{15}$ | ③ | 5 | 1 | 6 | 2 | 4 | 7 | 8 | 9 |
| $s_{16}$ | 1 | 2 | ③ | 4 | 6 | 5 | 7 | 8 | 9 |
| $s_{17}$ | 1 | ⑥ | 2 | 3 | 5 | 4 | 7 | 8 | 9 |
| $s_{18}$ | 1 | 2 | ⑤ | 3 | 4 | 6 | 8 | 7 | 9 |
| $s_{19}$ | 1 | 2 | ③ | 9 | 5 | 4 | 7 | 6 | 8 |
| $s_{20}$ | ⑨ | 2 | 6 | 8 | 5 | 3 | 7 | 4 | 1 |
| $s_{21}$ | 1 | ⑥ | 3 | 8 | 5 | 7 | 2 | 4 | 9 |
| $s_{22}$ | 1 | 2 | ③ | 4 | 5 | 6 | 7 | 8 | 9 |
| $s_{23}$ | 1 | ⑨ | 3 | 4 | 5 | 6 | 8 | 2 | 7 |
| $s_{24}$ | ⑧ | 4 | 3 | 2 | 5 | 6 | 7 | 9 | 1 |
| $s_{25}$ | 1 | ⑨ | 3 | 4 | 5 | 2 | 7 | 8 | 6 |
| $s_{26}$ | 1 | 3 | ⑤ | 4 | 2 | 6 | 8 | 7 | 9 |
| $s_{27}$ | ⑧ | 2 | 7 | 1 | 5 | 6 | 3 | 4 | 9 |
| $s_{28}$ | ⑦ | 1 | 3 | 4 | 8 | 6 | 2 | 5 | 9 |
| $s_{29}$ | ⑦ | 2 | 5 | 4 | 9 | 6 | 1 | 8 | 3 |
| $s_{30}$ | 1 | 2 | 3 | ⑦ | 5 | 6 | 9 | 8 | 4 |
| $s_{31}$ | ⑤ | 2 | 4 | 3 | 1 | 7 | 6 | 8 | 9 |
| $s_{32}$ | ④ | 2 | 8 | 1 | 5 | 7 | 6 | 3 | 9 |
| $s_{33}$ | 1 | 7 | 3 | 6 | ⑤ | 8 | 2 | 4 | 9 |
| $s_{34}$ | ④ | 2 | 3 | 7 | 1 | 6 | 5 | 8 | 9 |
| $s_{35}$ | ② | 1 | 3 | 4 | 5 | 8 | 9 | 6 | 7 |
| $s_{36}$ | 1 | 2 | 8 | 5 | 7 | 6 | 4 | 3 | ⑨ |

表 5: 第 1 希望優先アルゴリズムの結果（例題 7）

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $s_1$ | ① | 2 | 3 | 4 | 5 | 8 | 6 | 9 | 7 |
| $s_2$ | ① | 2 | 9 | 3 | 5 | 7 | 6 | 8 | 4 |
| $s_3$ | ① | 8 | 3 | 4 | 5 | 2 | 7 | 6 | 9 |
| $s_4$ | ① | 9 | 3 | 6 | 5 | 4 | 2 | 8 | 7 |
| $s_5$ | ② | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| $s_6$ | ② | 5 | 3 | 4 | 1 | 6 | 7 | 8 | 9 |
| $s_7$ | ④ | 1 | 2 | 3 | 5 | 6 | 7 | 8 | 9 |
| $s_8$ | 1 | ② | 3 | 4 | 5 | 7 | 6 | 8 | 9 |
| $s_9$ | ⑧ | 2 | 3 | 4 | 1 | 6 | 7 | 9 | 5 |
| $s_{10}$ | 1 | 2 | ⑥ | 4 | 5 | 8 | 7 | 3 | 9 |
| $s_{11}$ | ⑥ | 2 | 3 | 1 | 8 | 4 | 7 | 5 | 9 |
| $s_{12}$ | 1 | ⑧ | 9 | 4 | 2 | 6 | 7 | 5 | 3 |
| $s_{13}$ | 1 | 2 | ④ | 7 | 5 | 6 | 9 | 8 | 3 |
| $s_{14}$ | 1 | 2 | ⑦ | 4 | 5 | 6 | 3 | 8 | 9 |
| $s_{15}$ | ③ | 5 | 1 | 6 | 2 | 4 | 7 | 8 | 9 |
| $s_{16}$ | 1 | 2 | ③ | 4 | 6 | 5 | 7 | 8 | 9 |
| $s_{17}$ | 1 | ⑥ | 2 | 3 | 5 | 4 | 7 | 8 | 9 |
| $s_{18}$ | 1 | 2 | ⑤ | 3 | 4 | 6 | 8 | 7 | 9 |
| $s_{19}$ | 1 | 2 | ③ | 9 | 5 | 4 | 7 | 6 | 8 |
| $s_{20}$ | ⑨ | 2 | 6 | 8 | 5 | 3 | 7 | 4 | 1 |
| $s_{21}$ | 1 | ⑥ | 3 | 8 | 5 | 7 | 2 | 4 | 9 |
| $s_{22}$ | 1 | 2 | ③ | 4 | 5 | 6 | 7 | 8 | 9 |
| $s_{23}$ | 1 | ⑨ | 3 | 4 | 5 | 6 | 8 | 2 | 7 |
| $s_{24}$ | ⑧ | 4 | 3 | 2 | 5 | 6 | 7 | 9 | 1 |
| $s_{25}$ | 1 | ⑨ | 3 | 4 | 5 | 2 | 7 | 8 | 6 |
| $s_{26}$ | 1 | 3 | ⑤ | 4 | 2 | 6 | 8 | 7 | 9 |
| $s_{27}$ | ⑧ | 2 | 7 | 1 | 5 | 6 | 3 | 4 | 9 |
| $s_{28}$ | ⑦ | 1 | 3 | 4 | 8 | 6 | 2 | 5 | 9 |
| $s_{29}$ | ⑦ | 2 | 5 | 4 | 9 | 6 | 1 | 8 | 3 |
| $s_{30}$ | 1 | 2 | 3 | 7 | 5 | 6 | ⑨ | 8 | 4 |
| $s_{31}$ | ⑤ | 2 | 4 | 3 | 1 | 7 | 6 | 8 | 9 |
| $s_{32}$ | ④ | 2 | 8 | 1 | 5 | 7 | 6 | 3 | 9 |
| $s_{33}$ | 1 | ⑦ | 3 | 6 | 5 | 8 | 2 | 4 | 9 |
| $s_{34}$ | ④ | 2 | 3 | 7 | 1 | 6 | 5 | 8 | 9 |
| $s_{35}$ | ② | 1 | 3 | 4 | 5 | 8 | 9 | 6 | 7 |
| $s_{36}$ | 1 | 2 | 8 | ⑤ | 7 | 6 | 4 | 3 | 9 |

表 6: SAT を利用した手法の結果（例題 7）