3J1-OS7-9

一般双対化問題における冗長節生成の抑止法とその評価

On Reducing Non-Monotone Dualization to Monotone Dualization

山本 泰生[*1] 鍋島 英知[*1] 岩沼 宏治[*1]
Yoshitaka Yamamoto Hidetomo Nabeshima Koji Iwanuma

[*1]山梨大学大学院医学工学総合研究部
Department of Research Interdisciplinary, Graduate School of Medicine and Engineering, University of Yamanashi

This paper investigates non-monotone dualization (NMD) of general Boolean functions from the viewpoint of monotone dualization (MD). MD is one of the few problems whose tractability status is still unknown, and thus has received much attention that yields many remarkable algorithms. In contrast, NMD has not been much worked as yet, since an easy reduction from SAT problems gives NP-hardness. In this paper, we show that any NMD can be reducing to two equivalent MD problems. This feature enables us to provide a new solution for NMD based on the state-of-the-art MD computation.

1. Introduction

The problem of *non-monotone dualization* (NMD) is to generate an irredundant prime CNF formula ψ of the dual f^d where f is a *general* Boolean function represented by CNF [Eiter 03]. The DNF formula ϕ of f^d is easily obtained by De Morgan's laws interchanging the connectives of the CNF formula. Hence, the main task of NMD is to convert the DNF ϕ to an equivalent CNF ψ.

NMD has been continuously studied in computer science [Miltersen 05] and is used in several application domains, such as learning theory [DeRaedt 97] and logical design [Friedman 86], in order to seek an alternative representation of the input form. For instance, by converting a given CNF formula into DNF, we obtain the models satisfying the CNF formula. This fact shows an easy reduction from SAT problems to NMD, and also conjectures the hardness of it [Eiter 02]. In this context, the research has been focused on some restricted classes of Boolean functions.

Monotone dualization (MD) is one such class that deals with *monotone* functions for which CNF formulas are negation-free [Eiter 08, Hagen 08]. MD is one of the few problems whose tractability status with respect to polynomial-total time is still unknown. Besides, it is known that MD has many equivalent problems in discrete mathematics, such as the minimal hitting set enumeration. Thus, this class has received much attention that yields remarkable algorithms: in terms of complexity, the literature [Fredman 96] shows that this is solvable in a quasi-polynomial-total time (i.e., $(n+m)^{O(log\ (n+m))}$ where n and m denote the input and output size, respectively). Uno [Uno 02] shows a practical fast algorithm whose average computation time is experimentally $O(n)$ per output, for randomly generated instances.

This paper aims at clarifying whether or not NMD can be solved using these techniques of MD, and if it can be then how it is realized. In general, it is not straightforward to use them because of the following two problems in NMD:

- NMD has to treat *redundant clauses* like resolvents and tautologies.

連絡先: 山本 泰生，yyamamoto@yamanashi.ac.jp

Example 1 Let a CNF formula ϕ be $(x_1 \vee x_2) \wedge (\overline{x_2} \vee x_3)$. If we treat negated variables as regular variables, we can apply MD to ϕ and obtain the CNF formula $\psi = (x_1 \vee \overline{x_2}) \wedge (x_1 \vee x_3) \wedge (x_2 \vee \overline{x_2}) \wedge (x_2 \vee x_3)$. However, ψ contains the tautology $x_2 \vee \overline{x_2}$ and the resolvent $x_1 \vee x_3$ of $x_1 \vee \overline{x_2}$ and $x_2 \vee x_3$, which are to be removed.

- Unlike MD, the output of NMD is *not* necessarily *unique*. The literature [Rymon 94] shows that the output of MD uniquely corresponds to the set of all the prime implicates of f^d. In contrast, some prime implicates can be redundant in NMD problems. Thus, the output of NMD corresponds to an irredundant subset of the prime implicates. However, such a subset is not unique in general.

For the first problem, we show a technique to prohibit any resolvents from being generated in MD computation. This is done by simply adding some tautologies to the input CNF formula ϕ in advance. We denote by ϕ_t and ψ_t the extended input formula and its output by MD, respectively. Then, ψ_t contains no resolvents.

Example 2 Recall Example 1. We consider the CNF formula $\phi_t = (x_1 \vee x_2) \wedge (\overline{x_2} \vee x_3) \wedge (x_2 \vee \overline{x_2})$ obtained by adding one tautology which consists of two complementary literals x_2 and $\overline{x_2}$ that appear in ϕ. Then, MD generates the CNF formula $\psi_t = (x_1 \vee \overline{x_2}) \wedge (x_2 \vee \overline{x_2}) \wedge (x_2 \vee x_3)$. Indeed, ψ_t does not contain the resolvent $x_1 \vee x_3$, unlike ψ.

By removing all tautologies from ψ_t, we obtain an irredundant CNF formula, denoted by ψ_{ir}. Note that in Example 2, ψ_{ir} is $(x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_3)$.

We next address the second problem using a good property of ψ_{ir}: a subset of the prime implicates is irredundant (i.e., an output by NMD) if and only if it subsumes ψ_{ir} but never subsumes ψ_{ir} if any clause is removed from it. This particular relation is called *minimal subsumption*. We then show that the task of computing those subsets which satisfy the minimal subsumption is also a MD problem. In this way, we reduce a given NMD problem into two MD problems: the one for computing ψ_{ir}, and the other for computing those subsets corresponding to the outputs by NMD. This reduction technique enables us to provide a new solution for NMD based on the state-of-the-art MD computation.

Due to space limitations, full proofs are omitted in this paper.

2. Background

2.1 Preliminaries

A *Boolean function* is a mapping $f : \{0,1\}^n \rightarrow \{0,1\}$. We write $g \models f$ if f and g satisfy $g(v) \leq f(v)$ for all $v \in \{0,1\}^n$. g is (*logically*) *equivalent* to f, denoted by $g \equiv f$, if $g \models f$ and $f \models g$. A function f is *monotone* if $v \leq w$ implies $f(v) \leq f(w)$ for all $v, w \in \{0,1\}^n$; otherwise it is *non-monotone*. Boolean variables $x_1, x_2, \ldots, x_n$ and their negations $\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}$ are called *literals*. The *dual* of a function f, denoted by f^d, is defined as $\overline{f}(\overline{x})$ where $\overline{f}$ and $\overline{x}$ is the negation of f and x, respectively.

A *clause* (*resp. term*) is a disjunction (*resp.* conjunction) of literals which is often identified with the set of its literals. It is known that a clause is *tautology* if it contains complementary literals. Let C_1 and C_2 be clauses, and L_1 and L_2 literals in C_1 and C_2, respectively. If L_1 and L_2 are complementary literals, then the clause $(C_1 - \{L_1\}) \cup (C_2 - \{L_2\})$ is called the *resolvent* of C_1 and C_2. A clause C is an *implicate* of a function f if $f \models C$. An implicate C is *prime* if there is no implicate C' such that $C' \subset C$.

A *conjunctive normal form* (CNF) (*resp. disjunctive normal form* (DNF)) formula is a conjunction of clauses (*resp.* disjunction of terms) which is often identified with the set of clauses in it. In the following, we represent CNF or DNF formulas by the set notation for simplicity, if no confusion arises. A CNF formula ϕ is *irredundant* if $\phi \not\equiv \phi - \{C\}$ for every clause C in ϕ; otherwise it is *redundant*. ϕ is *prime* if every clause in ϕ is a prime implicate of ϕ; otherwise it is *non-prime*. Let ϕ_1 be $\{\{x_1,\ x_2\},\ \{\overline{x_2},\ x_3\},\ \{x_1,\ x_3\}\}$ and ϕ_2 be $\{\{x_1,\ x_2\},\ \{\overline{x_2},\ x_1\}\}$. On the one hand, ϕ_1 is prime but redundant, since the last clause is the resolvent of the others. On the other hand, ϕ_2 is irredundant but non-prime, since there is an implicate $\{x_1\}$ of ϕ_2 that is a subset of $\{x_1,\ x_2\}$.

Let ϕ_1 and ϕ_2 be two CNF formulas. ϕ_1 *subsumes* ϕ_2, denoted by $\phi_1 \succeq \phi_2$, if there is a clause $C \in \phi_1$ such that $C \subseteq D$ for every clause $D \in \phi_2$. In turn, ϕ_1 *minimally subsumes* ϕ_2, denoted by $\phi_1 \succeq^{\natural} \phi_2$, if ϕ_1 subsumes ϕ_2 but $\phi_1 - \{C\}$ does not subsume ϕ_2 for every clause $C \in \phi_1$.

Let ϕ be a CNF formula. $\mu(\phi)$ denotes the CNF formula obtained by removing every redundant clause in ϕ that is included in another clause. $\tau(\phi)$ denotes the CNF formula obtained by removing all tautologies from ϕ. We say ϕ is *tautology-free* if $\phi = \tau(\phi)$.

Now, we formally define the dualization problem as follows.

Definition 1 (Dualization problem)
Input: A tautology-free CNF formula ϕ
Output: An irredundant prime CNF formula ψ such that ψ is logically equivalent to ϕ^d.

We call it *monotone dualization* (MD) if ϕ is negation-free; otherwise it is called *non-monotone dualization* (NMD). As well known [Eiter 08], the task of MD is equivalent to enumerating the *minimal hitting sets* (MHSs) of a family of sets, as described next.

2.2 MD as MHS enumeration

We first introduce the notion of minimal hitting sets.

Definition 2 ((Minimal) Hitting set) Let Π be a finite set and $\mathcal{F}$ be a subset family of Π. A finite set E is a *hitting set* of $\mathcal{F}$ if for every $F \in \mathcal{F}$, $E \cap F \neq \emptyset$. A finite set E is a *minimal hitting set* (MHS) of $\mathcal{F}$ if E satisfies the following two conditions:

1. E is a hitting set of $\mathcal{F}$;
2. For every subset $E' \subseteq E$, if E' is a hitting set of $\mathcal{F}$, then $E' = E$.

Note here that any CNF formula ϕ can be identified with the family of clauses in ϕ. Accordingly, we can consider the CNF formula that is the conjunction of all the MHSs of the family ϕ. We denote it by $M(\phi)$. The literature [Rymon 94] shows a property of $M(\phi)$, which describes the relation between MD and MHS computation.

Theorem 1 [Rymon 94] Let ϕ be a tautology-free CNF formula. A clause C is in $\tau(M(\phi))$ if and only if C is a non-tautological prime implicate of ϕ^d.

In the case of MD, we do not need to consider any redundant clauses like tautologies and resolvents, since the input formula ϕ contains no negations. Thus, the output of MD is the CNF formula consisting of all the prime implicates of ϕ^d, which corresponds to $\tau(M(\phi))$ by Theorem 1.

We next introduce a practical fast algorithm for computing $\tau(M(\phi))$ [Uno 02]. This algorithm is based on *inverse search*, which uses so-called *parent-child relationship* to structure the search space as a rooted-tree. This tree is called an *enumeration tree*. Using the enumeration tree, the algorithm searches for solutions (i.e., the non-tautological minimal hitting sets of ϕ) with the depth-first search strategy. The following figure[*1] sketches it briefly [Satoh 02].

Global $\phi_n = \{C_1, \ldots, C_n\}$
compute(i, mhs, S)
/*mhs is an MHS of ϕ_i ($1 \leq i \leq n$).
S is the family of MHSs of ϕ_n.*/
Begin
if i = n then add mhs to S and return;
else if mhs is an MHS of ϕ_{i+1}
  do *compute*($i+1$, mhs, S);
else $\forall e \in C_{i+1}$ s.t. $mhs \cup \{e\}$ is a non tautological (1)
  MHS of ϕ_{i+1} (1) do *compute*($i+1$, $mhs \cup \{e\}$, S);
output S and return;
End

図 1: Uno's algorithm for computing $\tau(M(\phi_n))$

In other words, this algorithm incrementally searches for an MHS of the next family ϕ_{i+1} from the current MHS obtained for the family ϕ_i. We once again emphasize that its average computation for randomly generated instances is experimentally $O(n)$ per output, where n is the input size.

2.3 NMD as MHS enumeration

Our motivation is to clarify whether or not NMD can be solved using MD techniques. While MD is done by the state-of-the-art enumeration algorithm, it is not straightforward to use this for

*1 Since the original version is used for computing $M(\phi)$, we modify it so as to remove the tautologies in $M(\phi)$ by way of the condition (1).

NMD. Here, we review the two problems explained before in the context of MHS enumeration.
1. *Appearance of redundant clauses*: $\tau(M(\phi))$ is prime but not necessarily irredundant.

Example 3 Recall the CNF $\phi_2 = \{\{x_1,\ x_2\}, \{\overline{x_2},\ x_3\}\}$ of Example 1. Figure 2 describes the enumeration tree for ϕ_2 where each node is labeled by a pair (i, E_i). This pair means that a set E_i is an MHS of ϕ_i. By the enumeration tree, we obtain $\tau(M(\phi_2)) = \{\{x_1,\ \overline{x_2}\},\ \{x_1,\ x_3\},\ \{x_2,\ x_3\}\}$. However, this contains the redundant clause $\{x_1,\ x_3\}$.

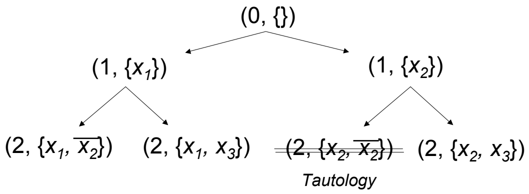

図 2: Enumeration tree for ϕ_2

2. *Non-uniqueness of NMD solutions*: there are many subsets of $\tau(M(\phi))$ that are prime and irredundant.

Example 4 Let the input CNF formula ϕ be as follows:

$$\phi = \{\{x_1,\ \overline{x_2},\ \overline{x_3}\},\ \{\overline{x_1},\ x_2,\ x_3\}\}.$$

$\tau(M(\phi))$ consists of the non-tautological prime implicates:

$$\begin{aligned}\tau(M(\phi)) = \quad & \{\{x_1,\ x_2\},\ \{\overline{x_1},\ \overline{x_3}\},\ \{\overline{x_2},\ x_3\}, \\ & \{x_1,\ x_3\},\ \{\overline{x_1},\ \overline{x_2}\},\ \{\overline{x_3},\ x_2\}\}.\end{aligned}$$

Then, we may notice that there are at least two irredundant subsets of $\tau(M(\phi))$:

$$\psi_1 = \{\{x_1,\ x_2\},\ \{\overline{x_1},\ \overline{x_3}\},\ \{\overline{x_2},\ x_3\}\}.$$
$$\psi_2 = \{\{x_1,\ x_3\},\ \{\overline{x_1},\ \overline{x_2}\},\ \{\overline{x_3},\ x_2\}\}.$$

Note that ψ_1 is logically equivalent to ψ_2, and thus both are also equivalent to $\tau(M(\phi))$ itself.

To address the two problems, this paper focuses on the following CNF formula.

Definition 3 (Bottom formula) Let ϕ be a tautology-free CNF formula and $Taut(\phi)$ the following set of tautologies:

$$Taut(\phi) = \{\ x \vee \overline{x}\ \mid\ \phi \textit{ contains both } x \textit{ and } \overline{x}\ \}.$$

Then, the *bottom formula wrt* ϕ (in short, bottom formula) is defined as the CNF formula $\tau(M(\phi \cup Taut(\phi)))$.

3. Properties of bottom formulas

Now, we show two properties of bottom formulas.

Theorem 2 Let ϕ be a tautology-free CNF formula. Then, the bottom formula wrt ϕ is irredundant.

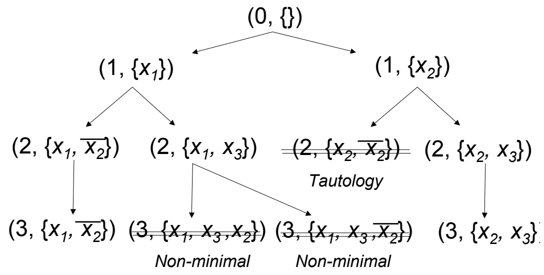

図 3: Enumeration tree for $\phi_2 \cup Taut(\phi_2)$

Example 5 Recall $\phi_2 = \{\{x_1,\ x_2\}, \{\overline{x_2},\ x_3\}\}$ in Example 3. $Taut(\phi_2)$ is $\{x_2 \vee \overline{x_2}\}$. Figure 3 describes the enumeration tree for $\phi_2 \cup Taut(\phi_2)$. From this tree, the bottom formula is $\{\{x_1,\ \overline{x_2}\},\ \{x_2,\ x_3\}\}$. Indeed, it is irredundant, since it does not contain the resolvent $\{x_1,\ x_3\}$.

In terms of the first problem, Theorem 2 shows a remarkable role of adding tautologies that prohibits any redundant clauses from being generated in MD computation. Note here that Theorem 2 ensures that the bottom formula is irredundant, but it does not ensure it is prime.

Example 6 Recall the CNF formula ϕ in Example 4. $Taut(\phi)$ is $\{\{x_1,\ \overline{x_1}\},\ \{x_2,\ \overline{x_2}\},\ \{x_3,\ \overline{x_3}\}\}$. The bottom formula is as follows:

$$\begin{aligned}&\{\{x_1,\ x_2,\ x_3\},\ \{\overline{x_3},\ x_2,\ x_1\}, \{\overline{x_3},\ x_2,\ \overline{x_1}\}, \\ &\{\overline{x_2},\ x_3,\ x_1\},\ \{\overline{x_2},\ x_3, \overline{x_1}\},\ \{\overline{x_2},\ \overline{x_3},\ \overline{x_1}\}\}.\end{aligned}$$

We write $C_1, C_2, \ldots, C_6$ for the above clauses in turn (i.e., C_4 is $\{\overline{x_2},\ x_3,\ x_1\}$). We then notice that the bottom formula is non-prime, because it contains a non-prime implicate C_1 whose subset $\{x_1,\ x_2\}$ is an implicate of ϕ^d.

As shown in Example 6, the bottom formula itself is not necessarily an output by NMD. However, every NMD output is logically connected with this formula.

Theorem 3 Let ϕ be a tautology-free CNF formula. Then, ψ is an output by NMD for ϕ if and only if $\psi \subseteq \tau(M(\phi))$ and ψ minimally subsumes the bottom formula wrt ϕ.

Example 7 Recall Example 4 and Example 6. Figure 4 describes the subsumption lattice bounded by two irredundant prime outputs ψ_1 and ψ_2 and the bottom formula $\{C_1, C_2, \ldots, C_6\}$. The solid (*resp.* dotted) lines show the subsumption relation between ψ_1 (*resp.* ψ_2) and the bottom formula. We then notice that both outputs ψ_1 and ϕ_2 minimally subsume the bottom formula.

4. Reducing NMD to MD

Theorem 3 shows that every NMD output ψ can be generated by selecting a subset ψ of $\tau(M(\phi))$ that minimally subsumes the bottom formula. Now, we show that the task of this selection is done by MD computation.

Let the bottom formula be $\{C_1,\ C_2,\ \ldots,\ C_n\}$. We then denote by S_i ($1 \leq i \leq n$) the set of clauses in $\tau(M(\phi))$ each of which is a subset of C_i. $\mathcal{F}_\phi$ denotes the family of those sets $\{S_1, S_2, \ldots, S_n\}$.

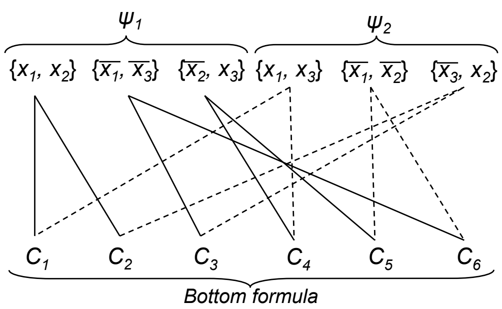

図 4: Subsumption lattice bounded by NMD outputs and the bottom formula

Theorem 4 Let ϕ be a tautology-free CNF formula. ψ is an output by NMD for ϕ if and only if ψ is an MHS of $\mathcal{F}_\phi$.

Example 8 Recall Example 7. We denote each clause of ψ_1 and ψ_2 in Figure 4 by $D_1, \ldots, D_6$, starting from left to right (i.e., D_4 is $\{x_1,\ x_3\}$). Then, $\mathcal{F}_\phi$ is as follows:

$$\mathcal{F}_\phi = \{\{D_1,\ D_4\},\ \{D_1,\ D_6\},\ \{D_2,\ D_6\}, \{D_3,\ D_4\},\ \{D_3,\ D_5\},\ \{D_2,\ D_5\}\}.$$

By MHS computation, we have the five MHSs of $\mathcal{F}_\phi$:

$$\{D_1,\ D_2,\ D_3\}, \{D_4,\ D_5,\ D_6\}, \{D_1,\ D_2,\ D_4,\ D_5\}, \{D_1,\ D_3,\ D_5,\ D_6\},\ \{D_2,\ D_3,\ D_4,\ D_6\}.$$

They contain two MHSs $\{D_1,\ D_2,\ D_3\}$ and $\{D_4,\ D_5,\ D_6\}$ that correspond to NMD outputs ψ_1 and ψ_2, respectively.

Both the bottom theory and $\tau(M(\phi))$ are obtained by one MD computation. Furthermore, Theorem 4 shows that the task of selecting irredundant subsets is also done by another MD computation. In summary, the NMD problem of a tautology-free CNF formula ϕ can be reconstructed into two MD problems: one for computing the bottom theory wrt ϕ and $\tau(M\phi)$, and the other for computing an MHS of $\mathcal{F}_\phi$.

5. Conclusion and future work

This paper have presented a technique for dualizing nonmonotone Boolean functions by monotone dualization computation. Previous works revealed that monotone dualization is solvable in a quasi-polynomial-total time, and efficient algorithms for it and its related problems have been proposed. In this context, our result gives an insight to use those efficient algorithms of monotone dualization for non-monotone cases. The main result is described in Theorem 3 that the bottom formula minimally subsumes every output. Based on this result, we reduce any non-monotone dualization problem to two monotone dualization problems. We emphasize that the result enables us to generate *every* output that makes possible to find the most compact solution. Our result also enables us to investigate the complexity of NMD from the viewpoint of MD computation. For instance, the complexity of generating every NMD output can be described as follows:

$$(n+k)^{O(log\ (n+k))} + (k+m)^{O(log\ (k+m))},$$

where n, k and m are the sizes of the input formula ϕ, the bottom formula wrt ϕ and all the NMD outputs. This is simply derived from the complexity of MD computation [Fredman 96].

A further investigation on previously proposed methods of NMD is an important future work. Whereas this paper provides a solution for NMD using MD computation, it is necessary to clarify the significance of our technique with respect to improvement of previous bounds and applicability to practical problems. Summaries of the state-of-the-art NMD and comparisons with them will make our contribution clearer.

Acknowledgement

This research is supported by 2008-2011 JSPS Grant-in-Aid Scientific Research (A) (No. 20240003) and by 2010-2012 JSPS Grant-in-Aid Scientific Research (B) (No. 22700141).

参考文献

[DeRaedt 97] DeRaedt, L.: Logical settings for concept-learning, *Artificial Intelligence*, Vol. 95, pp. 187–201 (1997)

[Eiter 02] Eiter, T., Ibaraki, T., and Makino, K.: Recognition and dualization of disguised bidual Horn functions, *Information Processing Letters*, Vol. 82, pp. 283–291 (2002)

[Eiter 03] Eiter, T. and Makino, K.: Abduction and the dualization problem, in *Proceedings of the 6th Int. Conf. on Discovery Science*, Vol. 2843 of *LNCS*, pp. 1–20 (2003)

[Eiter 08] Eiter, T., Makino, K., and Gottlob, G.: Computational aspects of monotone dualization: A brief survey, *Discrete Applied Mathematics*, Vol. 156, pp. 2035–2049 (2008)

[Fredman 96] Fredman, M. and Khachiyan, L.: On the complexity of dualization of monotone disjunctive normal forms, *Algorithms*, Vol. 21, pp. 618–628 (1996)

[Friedman 86] Friedman, A. D.: *Fundamentals of logic design and switching theory* , Computer Science Press (1986)

[Hagen 08] Hagen, M.: *Algorithmic and computational complexity issues of MONET*, PhD thesis, Friedrich Schiller University Jena (2008)

[Miltersen 05] Miltersen, P. B., Radhakrishnan, J., and Wegener, I.: On converting CNF to DNF, *Theoretical Computer Science*, Vol. 347, pp. 325–335 (2005)

[Rymon 94] Rymon, R.: An SE-tree based prime implicant generation algorithm, *Annals of Mathematics and Artificial Intelligence*, pp. 351–366 (1994)

[Satoh 02] Satoh, K. and Uno, T.: Enumerating maximal frequent sets using irredundant dualization, in *Proceedings of the 6th Int. Conf. on Discovery Science*, Vol. 2843 of *LNCS*, pp. 256–268 (2002)

[Uno 02] Uno, T.: A practical fast algorithm for enumerating minimal set coverings, in *Proceedings of the 83rd SIGAL Conf. of the Information Processing Society of Japan*, pp. 9–16 (2002)