

Incremental complexity of a bi-objective hypergraph transversal problem

Ricardo Andrade², Etienne Birmelé¹, Arnaud Mary², Thomas Picchetti¹, and Marie-France Sagot²

¹ MAP5, UMR CNRS 8145, Université Paris Descartes

² Université de Lyon, F-69000, Lyon ; Université Lyon 1 ; CNRS, UMR5558, Laboratoire de Biométrie et Biologie Evolutive, F-69622 Villeurbanne, France / INRIA Grenoble Rhône-Alpes - ERABLE

Abstract. The hypergraph transversal problem has been intensively studied, from both a theoretical and a practical point of view. In particular, its incremental complexity is known to be quasi-polynomial in general and polynomial for bounded hypergraphs. Recent applications in computational biology however require to solve a generalization of this problem, that we call bi-objective transversal problem. The instance is in this case composed of a pair of hypergraphs $(\mathcal{A}, \mathcal{B})$, and the aim is to find minimal sets which hit all the hyperedges of \mathcal{A} while intersecting a minimal set of hyperedges of \mathcal{B} . In this paper, we formalize this problem, link it to a problem on monotone boolean $\wedge - \vee$ formulae of depth 3 and study its incremental complexity.

1 Introduction

Let $G(V, \mathcal{A})$ with $\mathcal{A} \subseteq 2^V$ be a hypergraph on a finite set V . By abuse of language, we may also from now on refer to the hypergraph more simply by \mathcal{A} only. A *transversal* of \mathcal{A} is any set $S \subseteq V$ intersecting all hyperedges of \mathcal{A} . It is straightforward to see that being a transversal is a monotone property on the subsets of V , so that the collection of minimal transversals characterizes all of them. This collection is called the *dual* or *transversal hypergraph* of \mathcal{A} , and is denoted by $tr(\mathcal{A})$.

The problem of computing the transversal hypergraph of any \mathcal{A} is equivalent to enumerating maximal independent sets in hypergraphs [3] or to solving the Boolean function dualization problem [8]. Furthermore, it has many applications, for instance in artificial intelligence [6]. This problem thus received much attention in the last decades, both from a theoretical and a practical point of view (see [8] for a review).

The first method was proposed by Berge [1], who considered the hyperedges iteratively, updating the partial solutions obtained at each step. This algorithm may however have to store a high number of partial solutions, and no final solution will be available until the algorithm stops. More recent work thus focused on methods that build the minimal transversals iteratively, that is which study the following problem [14]:

Problem 1. DUAL(\mathcal{A}, \mathcal{X}) Given a hypergraph \mathcal{A} and a set \mathcal{X} of minimal transversals of \mathcal{A} , prove that $tr(\mathcal{A}) = \mathcal{X}$ or find a new minimal transversal in $tr(\mathcal{A}) \setminus \mathcal{X}$.

The complexity of this problem remains an open question. However, Fredman and Khachiyan [9] showed that it is quasi-polynomial by proposing two algorithms of respective complexities $N^{\mathcal{O}(\log^2 N)}$ and $N^{o(\log N)}$, where $N = |\mathcal{A}| + |\mathcal{X}|$ is the size of the input. We define the dimension $dim(\mathcal{A})$ of a hypergraph \mathcal{A} as the size of its largest hyperedge, and the degree of a vertex as the number of hyperedges it belongs to. For hypergraphs of bounded dimension, the problem is polynomial [5] and parallelizable [3]. It is also polynomial for hypergraphs of bounded degree [7]. Moreover, the complexity class of the problem does not change if multiple minimal transversals or partial minimal transversals are required [4].

The performance of the algorithms in practice was also studied in several publications. Khachiyan et al. [14] introduced an algorithm of the same worst-case complexity than the one of Fredman and Khachiyan but with a better performance in practice. More recently, Toda [16] and Murakami and Uno [15] compared the existing algorithms and proposed new ones which can deal with large scale hypergraphs.

Determining the transversal of a hypergraph has also several already studied or potential applications in computational biology. It was for example proposed for elaborating knock-out strategies in metabolic networks [11], the hyperedges representing metabolic pathways whose activity should be suppressed. One may also consider the vertices as genes and a hyperedge as the set of mutated genes in a tumoral tissue. The transversal hypergraph then lists the collection of minimal mutation sets covering all the tumors. The mutation scenarios would be described by sets of genes, rather than by a single ranking of the genes based on the p-value of a statistical over-representation test.

However, due to the complexity of cellular mechanisms, in both previous cases it appears there actually are two types of hyperedges, some of them having to be intersected while some others should be avoided. Indeed, if one wants to knock-out a given set of metabolic pathways, one needs to maintain the biomass production of the cell in order to avoid cellular death. Hädicke and Klamt [12] introduced thus the notion of *constrained minimal cut sets* corresponding to vertex sets hitting all target pathways while avoiding at least n pathways among a prescribed set. An adaptation of the Berge algorithm was proposed and was compared to binary integer programming on real data sets [13].

Coming back to the tumoral mutation example, a similar bi-objective problem appears. Mutations may indeed not be related to cancer, and the goal is to discriminate driver mutations from so-called back-seat mutations. Bertrand et al. [2] show that this is equivalent to the *minimal set cover* and used a greedy approximation algorithm to solve it. An alternative way to deal with the problem would be to use other mutation data on similar but non tumoral tissues, and to look for mutation collections covering all tumors while covering as few healthy samples as possible.

We therefore propose to consider a bi-objective generalization of the hypergraph transversal problem, in which two distinct hypergraphs represent, respectively, the sets of nodes to hit and those to avoid, and to search for the minimal sets of vertices fulfilling both criteria.

2 The bi-objective transversal problem

2.1 The problem

We consider two hypergraphs on the same set of vertices V . The first hypergraph \mathcal{A} will be denoted as the red hypergraph and represents the sets of vertices that have to be intersected. The second hypergraph \mathcal{B} will be denoted as the blue hypergraph and represents the sets of vertices which should not be intersected if possible. We will represent such an instance as a tripartite graph, as shown in Figure 1.

For any $S \subset V$, we define

$$\mathcal{A}_S = \{A \in \mathcal{A}; S \cap A \neq \emptyset\}$$

and

$$\mathcal{B}_S = \{B \in \mathcal{B}; S \cap B \neq \emptyset\}$$

In particular, for $x \in V$, \mathcal{A}_x and \mathcal{B}_x denote the sets of red and blue hyperedges that contain x .

The problem then becomes:

Problem 2. Bi-objective hypergraph transversal problem Given a hypergraph $G = (V, \mathcal{H})$ and a partition $\mathcal{H} = \mathcal{A} \cup \mathcal{B}$ of its hyperedges, enumerate the sets $S \subset V$ such that:

1. $\mathcal{A}_S = \mathcal{A}$ and S is minimal for this property;
2. there exists no S' verifying condition 1 and such that $\mathcal{B}_{S'}$ is a strict subset of \mathcal{B}_S .

Such sets are called *bi-objective minimal transversals* of the couple $(\mathcal{A}, \mathcal{B})$. The collection of bi-objective minimal transversals of $(\mathcal{A}, \mathcal{B})$ is denoted by $btr(\mathcal{A}, \mathcal{B})$.

A first approach for this problem would be to enumerate all minimal transversals for \mathcal{A} and then to check for the minimality condition with respect to \mathcal{B} . However, such a procedure may spend an exponential time on enumerating minimal transversals of \mathcal{A} which will be ruled out in the second step. Indeed, consider a hypergraph \mathcal{A} on a vertex set V such that \mathcal{A} has an exponential number of minimal transversals. Let S be one of them. Consider the hypergraph \mathcal{B} having $V \setminus S$ as unique hyperedge. S is then the unique bi-objective minimal transversal of $(\mathcal{A}, \mathcal{B})$.

As the dual hypergraph problem corresponds to the special case $\mathcal{B} = \emptyset$, we propose to adopt the same strategy of an incremental search of the solutions. We therefore introduce the following problem:

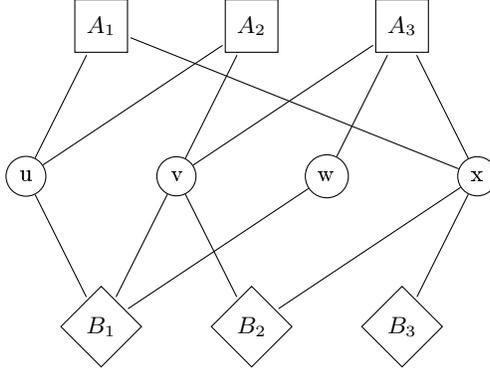


Fig. 1. Tripartite representation of an instance of the problem. The circled vertices are the vertices of the hypergraphs. The squared (resp. diamond) vertices represent the hyperedges of \mathcal{A} (resp. \mathcal{B}). Consider the sets $S = \{u, v\}$ and $T = \{u, w\}$. Both are minimal transversals for \mathcal{A} . However S is not a solution for the bi-objective problem as \mathcal{B}_T is a strict subset of \mathcal{B}_S .

Problem 3. BIDUAL($\mathcal{A}, \mathcal{B}, \mathcal{X}$) Given hypergraphs \mathcal{A} and \mathcal{B} on the same vertices and a set \mathcal{X} of bi-objective minimal transversals of $(\mathcal{A}, \mathcal{B})$, prove that $btr(\mathcal{A}) = \mathcal{X}$ or find a new minimal transversal in $btr(\mathcal{A}) \setminus \mathcal{X}$.

2.2 A new enumeration problem

For $B \subseteq \mathcal{B}$, we denote by S_B the set such that $S_B = \{x \in V(\mathcal{B}) \mid \mathcal{B}_x \subseteq B\}$. In other words, S_B is the set of vertices belonging to no blue hyperedges except for those in B .

For instance, in Figure 1, $S_{B_1} = \{u, w\}$, $S_{B_2} = \emptyset$ and $S_{\{B_2, B_3\}} = \{x\}$.

We define a predicate $f : 2^{\mathcal{B}} \rightarrow \{0, 1\}$ with

$$f(B) = \begin{cases} 1 & \text{if } S_B \text{ is a transversal of } \mathcal{A} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

and introduce a new enumeration problem:

Problem 4. Minimal \mathcal{B} -sets enumeration problem Enumerate all minimal $B \subseteq \mathcal{B}$ such that $f(B) = 1$.

In Figure 1, B_1 is the only solution to the above problem.

The sets to be enumerated in Problems 2 and 4 are then linked through the following result.

Lemma 1. 1. Let S be a solution to Problem 2. We then have that \mathcal{B}_S is a solution to Problem 4.

2. Let B be a solution to Problem 4. We then have that every minimal hitting set S of \mathcal{A} included in S_B is a solution to Problem 2 and $\mathcal{B}_S = B$.

Proof. The first item is a direct consequence of the definition of a solution to Problem 2.

For the second item, consider S' a minimal hitting set for \mathcal{A} with $\mathcal{B}_{S'} \subset \mathcal{B}_S \subset B$. Then $f(\mathcal{B}_{S'}) = 1$. By minimality of B , $\mathcal{B}_{S'} = \mathcal{B}_S = B$. \square

Let us then consider the algorithm which enumerates the solutions of Problem 4 and which, each time a new solution B is found, enumerates all minimal sets S covering \mathcal{A} and included in S_B .

By the first item of Lemma 1, every solution of Problem 2 is enumerated by this algorithm.

By the second item, all the sets enumerated by the algorithm are solutions to Problem 2 and none of them is enumerated twice. Indeed, if that were the case, meaning that two solutions for Problem 4, say B and B' , lead to a same solution S to Problem 2. The second item then implies that $B = \mathcal{B}_S = B'$.

Moreover, given a set $B \in \mathcal{B}$, the enumeration of all minimal sets covering \mathcal{A} and included in S_B is the transversal hypergraph problem, and can therefore be solved in quasi-polynomial time, and even in polynomial time if \mathcal{A} is of bounded dimension or if \mathcal{A} is of bounded degree.

A natural question is therefore the complexity class of Problem 4. The answer to this question is already partially known. Indeed, consider a set B of hyperedges of \mathcal{B} , and for each hyperedge b_i of \mathcal{B} , the boolean variable c_i indicating if $b_i \in B$. Then, for any vertex x , $x \in S_B$ if and only if $\bigwedge_{b_i \in \mathcal{B}_x} c_i = 1$. A hyperedge a of \mathcal{A} is then hit by S_B if and only if $\bigvee_{x \in a} \bigwedge_{b_i \in \mathcal{B}_x} c_i = 1$. Finally,

$$f(B) = \bigwedge_{a \in \mathcal{A}} \bigvee_{x \in a} \bigwedge_{b_i \in \mathcal{B}_x} c_i$$

Our enumeration problem is therefore the enumeration of all minimal truth assignments satisfying a monotone boolean function of depth 3. The situation is illustrated by Figure 2. Conversely, for any monotone formula of type $\bigwedge \bigvee \bigwedge$, one can easily construct an equivalent instance of Problem 4: simply represent it as a tree of depth 3, merge leaves that are labeled with the same literal, remove the root, and the result is a tripartite graph as in Figure 1.

The associated incremental problem is then the following, introduced in [10]:

Problem 5. GEN(\mathcal{S}) Consider a monotone boolean function of depth 3 in the shape $\bigwedge \bigvee \bigwedge$. Given a set \mathcal{S} of minimal solutions, determine if \mathcal{S} is the set of all minimal solutions or find a new one.

This problem was shown in [10] to be coNP-complete. More precisely, the reduction in [10] shows that in order to determine if a DNF formula d is a tautology, it suffices to define a monotone formula \tilde{d} by replacing in d every negative literal \bar{x}_i with a new variable y_i , and ask whether the monotone formula

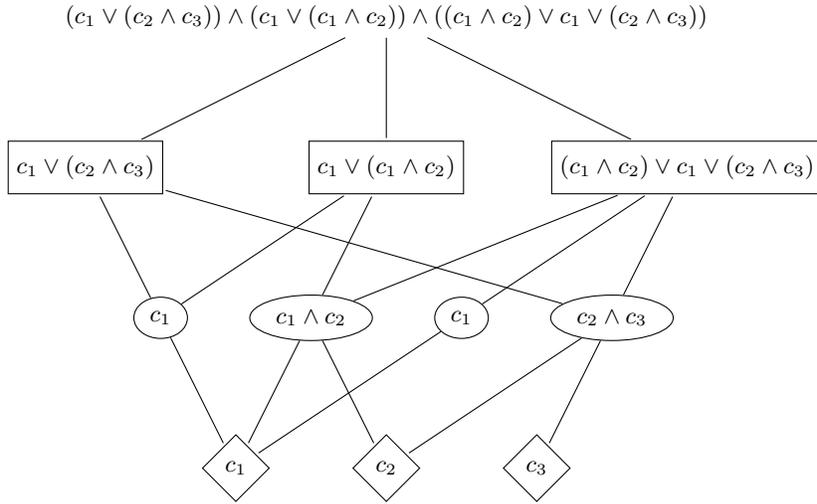


Fig. 2. Representation of the instance of Figure 1 in terms of a monotone boolean function. The formula of depth 3 is not true for the truth assignment $c_1 = c_2 = c_3 = 0$, indicating that it is not possible to hit all hyperedges of \mathcal{A} avoiding every hyperedge of \mathcal{B} . However, the truth assignment $c_1 = 1, c_2 = c_3 = 0$ satisfies the formula, indicating that there exists a solution hitting only the the first blue (diamond) hyperedge.

$f_0 = \tilde{d} \vee [(x_1 \vee y_1) \wedge (x_2 \vee y_2) \wedge \cdots \wedge (x_n \vee y_n)]$ has other prime implicants than the terms in \tilde{d} .

By distributivity $f_0 \equiv f_1 = (x_1 \vee y_2 \vee \tilde{d}) \wedge (x_1 \vee y_2 \vee \tilde{d}) \wedge \cdots \wedge (x_1 \vee y_2 \vee \tilde{d})$ which is in the shape $\bigwedge \bigvee \bigwedge$, hence the coNP-completeness of GEN .

Since in the instance of Problem 4 associated to f_1 the hypergraphs \mathcal{A} and \mathcal{B} have maximum degree 1 and 3 (if d is in 3-DNF), this reduction implies that the corresponding decision problem is coNP-complete even with these degree constraints.

However, the question remains open if the dimension of \mathcal{A} or \mathcal{B} is bounded that is if the red (square) or the blue (diamond) vertices in Figure 2 are of bounded degree.

When only one hypergraph is considered, bounding the degree or the dimension reduces the complexity of the transversal enumeration from quasi-polynomial to polynomial. In the following, we prove in Section 3 that bounding the dimension of \mathcal{B} or the degree of the vertices is not enough in the bi-objective case as then 3-SAT can be reduced to both *BIDUAL* and *GEN*. However, we show in Section 4 that both problems become polynomial if the dimension of \mathcal{A} is bounded.

3 Bounding the dimension or degree of \mathcal{B}

3.1 Results for \mathcal{B} of bounded dimension

Theorem 1. *3-SAT can be reduced to BIDUAL, even in the case of a bounded value for $\dim(\mathcal{B})$.*

Proof. Let us consider a 3-SAT instance with boolean variables $\{x_1, \dots, x_n\}$ and clauses C_1, \dots, C_m . We can consider, without loss of generality, that there exists no i such that all clauses contain either x_i or \bar{x}_i .

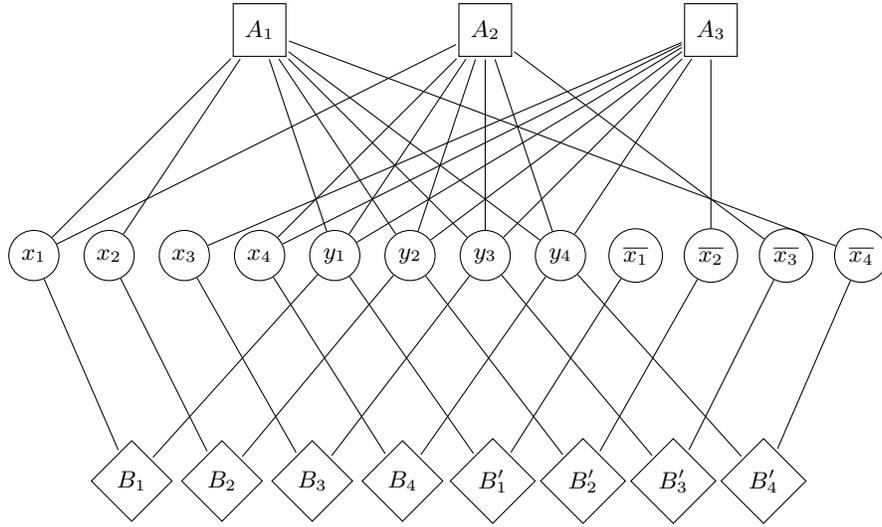


Fig. 3. Instance for the reduction of 3-SAT to *BIDUAL*. The considered clauses are $C_1 = x_1 \vee x_2 \vee \bar{x}_4$, $C_2 = x_1 \vee \bar{x}_3 \vee x_4$ and $C_3 = \bar{x}_2 \vee x_3 \vee x_4$.

Construct the following hypergraph (see Figure 3):

1. Consider $3n$ vertices $V = \{x_1, \bar{x}_1, \dots, x_n, \bar{x}_n, y_1, \dots, y_n\}$.
2. For every $1 \leq j \leq m$, define a red hyperedge A_j including the x_i 's and \bar{x}_i 's defining C_j as well as $\{y_1, \dots, y_n\}$.
3. For every $1 \leq i \leq n$, define a blue hyperedge $B_i = \{x_i, y_i\}$ and a blue hyperedge $B'_i = \{\bar{x}_i, y_i\}$. Observe that \mathcal{B} is of dimension 2.

For every $1 \leq i \leq n$, consider $S_i = \{y_i\}$. It covers all the red hyperedges as well as B_i and B'_i . As neither x_i nor \bar{x}_i is contained in all the clauses, it is a minimal solution to the bi-objective problem.

Consider the problem $BIDUAL(\mathcal{A}, \mathcal{B}, \mathcal{X})$ for $\mathcal{X} = \{\{y_1\}, \dots, \{y_n\}\}$.

Suppose that there exists a minimal bi-objective transversal S in $btr(\mathcal{A}, \mathcal{B}) \setminus \mathcal{X}$. For any $1 \leq i \leq n$, S cannot contain both vertices x_i and \bar{x}_i . Indeed, it would

then cover B_i and B'_i , implying $\mathcal{B}_{S_i} \subset \mathcal{B}_S$. As at least one clause contains neither x_i nor $\overline{x_i}$, this inclusion would be strict and contradict the minimality of S with respect to the second bi-objective criterion.

The former implies that S corresponds to a truth assignment of the boolean variables x_i . The fact that it covers \mathcal{A} is then equivalent to the fact that all clauses are satisfied. Deciding if there is such S is therefore NP-Complete, i.e. deciding whether $btr(\mathcal{A}, \mathcal{B}) = \mathcal{X}$ is coNP-complete. \square

Remark 1. The same reduction is an alternative proof to [10] of the coNP-hardness of *GEN* by considering $\mathcal{S} = \bigcup_{1 \leq i \leq n} (B_i, B'_i)$. A same reasoning can then be applied to show that finding a new solution would be equivalent to finding a truth assignment satisfying all clauses.

3.2 Results for vertices of bounded degree

The reduction of the former paragraph can be slightly adapted to prove that both *BIDUAL* and *GEN* remain hard if the degree of the vertices is bounded rather than the dimension of \mathcal{B} . This is a strong difference with the traditional transversal problem as *DUAL* becomes polynomial in this case.

Theorem 2. 3 – *SAT* can be reduced to *BIDUAL* and *GEN* in the case of a bounded degree of the vertices of V .

Proof. The reduction is similar to the one used to prove Theorem 1. Two modifications are done:

1. The vertices of the hypergraph are split: each y_i is substituted by m vertices y_i^j , $1 \leq j \leq m$, y_i^j belonging to the hyperedges B_i , B'_i and A_j . Similarly, each x_i (resp. $\overline{x_i}$) is split in as many vertices as the number of red hyperedges it belongs to, and each of them belongs to B_i (resp. B'_i) and one of the copies.
2. For each $1 \leq i \leq n$, a new blue hyperedge B''_i is created, containing all x_i^j and $\overline{x_i^j}$.

Figure 4 gives an example of this reduction.

It is easy to see that each vertex of V is of degree exactly 3. Moreover, for each i , $S_i = \{y_i^1, \dots, y_i^m\}$ is a solution to the bi-objective problem and $\mathcal{B}_{S_i} = \{B_i, B'_i\}$. Consider then the problem *BIDUAL*($\mathcal{A}, \mathcal{B}, \mathcal{X}$) where \mathcal{X} is the set of all those solutions.

Let S be a new solution of the bi-objective problem. The set S cannot, for any indices i, j, k , contain both y_i^k and a vertex of index j . Indeed, \mathcal{B}_S would then contain B_i , B'_i and one set among B_j , B'_j and B''_j , and therefore also contain \mathcal{B}_{S_i} as a proper subset. S thus contains no vertex y .

The same argument holds if S contains both x_i^k and $\overline{x_i^l}$ for some indices i, k, l . Indeed, \mathcal{B}_S then contains B_i , B'_i and B''_i and consequently contains \mathcal{B}_{S_i} as a proper subset.

For every index i , S therefore contains either only vertices of the form x_i^k or only vertices of the form $\overline{x_i^k}$. As S covers all the red hyperedges, assigning a

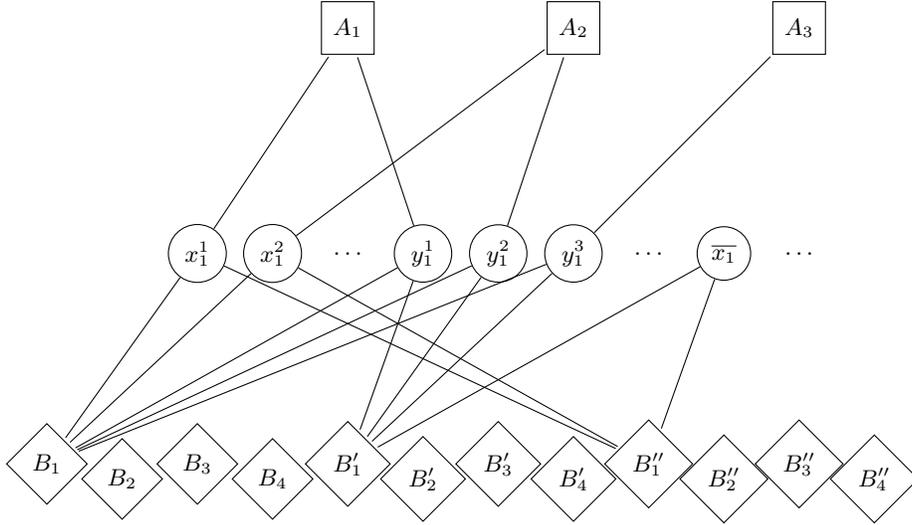


Fig. 4. Reduction of 3-SAT to *BIDUAL*. The considered clauses are $C_1 = x_1 \vee x_2 \vee \overline{x_4}$, $C_2 = x_1 \vee \overline{x_3} \vee x_4$ and $C_3 = \overline{x_2} \vee x_3 \vee x_4$. For readability reasons, only the vertices corresponding to the index $i = 1$ are represented in the intermediate layer.

value to the variable x_i in the clauses according to this criterion yields a truth assignment satisfying all the clauses.

The reduction from 3-SAT to *GEN* can be done with exactly the same arguments, considering as already known the set \mathcal{S} of solutions of the form $\{B_i, B'_i\}$.

□

4 Bounding the dimension of \mathcal{A}

We show in this section that one can reduce Problem 2 to the transversal hypergraph problem in a hypergraph of bounded dimension whenever \mathcal{A} is of bounded dimension.

Theorem 3. *GEN can be solved in polynomial time if \mathcal{A} is of bounded dimension. As a consequence, Problem 2 can be solved in incremental polynomial time.*

When the dimension of \mathcal{A} is bounded, given a set $B \in \mathcal{B}$, the enumeration of all minimal sets covering \mathcal{A} and included in S_B corresponds to the enumeration of minimal transversals of a hypergraph of bounded dimension, and therefore can be done in incremental polynomial time. To show that Problem 2 is polynomial, it is sufficient to show that Problem 4 is polynomial. Actually we will show that Problem 4 can be also reduced to the enumeration of all minimal transversals of a hypergraph of bounded dimension.

Definition 1. For $A \in \mathcal{A}$, let us denote by \mathcal{H}_A the hypergraph such that:

- $V(\mathcal{H}_A) = \bigcup_{x \in A} \mathcal{B}_x$
- $\mathcal{E}(\mathcal{H}_A) = \{\mathcal{B}_x \mid x \in A\}$

The following proposition gives a characterisation of the subsets B of \mathcal{B} for which S_B covers a given hyperedge $A \in \mathcal{A}$. Given $A \in \mathcal{A}$, by construction of the hypergraph \mathcal{H}_A , S_B intersects \mathcal{H}_A if and only if B contains a hyperedge of \mathcal{H}_A .

For our purpose, we need to reformulate this simple fact. The formulation given in Lemma 2 can be seen as a direct consequence of the following observation. A subset of vertices X of a hypergraph \mathcal{H} contains a hyperedge if and only if X is a transversal of the hypergraph $tr(\mathcal{H})$. Indeed, by the duality property between a hypergraph and its transversal hypergraph (see [1]), the minimal transversals of $tr(\mathcal{H})$ are exactly the minimal hyperedges of \mathcal{H} . Thus, a subset of vertices contains a hyperedge of \mathcal{H} if and only if it contains a minimal transversal of $tr(\mathcal{H})$, i.e. if it is a transversal of $tr(\mathcal{H})$.

Lemma 2. Let $B \subseteq \mathcal{B}$ and $A \in \mathcal{A}$. We then have that $S_B \cap A \neq \emptyset$ if and only if B is a transversal of $tr(\mathcal{H}_A)$.

Proof. (\Rightarrow) Assume that there exists $x \in S_B \cap A$. Then by definition of S_B , $\mathcal{B}_x \subseteq B$. Let $t \in tr(\mathcal{H}_A)$. Since \mathcal{B}_x is a hyperedge of \mathcal{H}_A , t must intersect \mathcal{B}_x and then $t \cap B \neq \emptyset$. We conclude that B is a transversal of $tr(\mathcal{H}_A)$.

(\Leftarrow) Assume now that B is a transversal of $tr(\mathcal{H}_A)$ and that $S_B \cap A = \emptyset$. By definition of S_B , for all $x \in A$, there exists $b_x \in \mathcal{B}_x$ such that $b_x \notin B$. Let $t \subseteq \mathcal{B}$ be the set formed by all b_x for all $x \in A$ i.e. $t := \bigcup_{x \in A} \mathcal{B}_x \setminus B$. Since for all $x \in A$, $\mathcal{B}_x \setminus B \neq \emptyset$, t is a transversal of \mathcal{H}_A and then contains a minimal transversal t' of \mathcal{H}_A . However by construction of t , we have $t' \cap B = \emptyset$, contradicting the fact that B is a transversal of $tr(\mathcal{H}_A)$. \square

Now since we require that S_B covers all hyperedges of \mathcal{A} , B must be a transversal of \mathcal{H}_A for every $A \in \mathcal{A}$.

Proposition 1. $\min_{\subseteq} \{B \subseteq \mathcal{B} \mid f(B) = 1\} = tr(\bigcup_{A \in \mathcal{A}} tr(\mathcal{H}_A))$.

Proof. Let $\mathcal{H} = \bigcup_{A \in \mathcal{A}} tr(\mathcal{H}_A)$.

$$\begin{aligned} f(B) = 1 &\iff \forall A \in \mathcal{A}, S_B \cap A \neq \emptyset && \text{by definition} \\ &\iff \forall A \in \mathcal{A}, B \text{ is a transversal of } tr(\mathcal{H}_A) && \text{by Lemma 2} \\ &\iff B \text{ is a transversal of } \mathcal{H} \end{aligned}$$

Thus, the set $\{B \subseteq \mathcal{B} \mid f(B) = 1\}$ is exactly the set of transversals of \mathcal{H} . Therefore, $\min_{\subseteq} \{B \subseteq \mathcal{B} \mid f(B) = 1\} = tr(\bigcup_{A \in \mathcal{A}} tr(\mathcal{H}_A))$. \square

If $\dim(\mathcal{A}) = C$ is bounded, for all $A \in \mathcal{A}$, \mathcal{H}_A has at most C hyperedges and then each minimal transversal t of \mathcal{H}_A is of size at most C . Then $\bigcup_{A \in \mathcal{A}} \text{tr}(\mathcal{H}_A)$ is a hypergraph of dimension at most C having at most $|\mathcal{A}||\mathcal{B}|^C$ hyperedges. We can then construct it in polynomial time and enumerate its minimal transversals in incremental polynomial time using the result for the dualization problem [5].

This implies that if \mathcal{A} is of bounded dimension, $\min_{\subseteq} \{B \subseteq \mathcal{B} \mid f(B) = 1\}$ can be enumerated in incremental polynomial time, thus proving Theorem 3.

References

1. BERGE, C. *Hypergraphs: Combinatorics of Finite Sets*. North-Holland, 1989.
2. BERTRAND, D., CHNG, K. R., SHERBAF, F. G., KIESEL, A., CHIA, B. K., SIA, Y. Y., HUANG, S. K., HOON, D. S., LIU, E. T., HILLMER, A., AND NAGARAJAN, N. Patient-specific driver gene prediction and risk assessment through integrated network analysis of cancer omics profiles. *Nucleic Acids Research* (2015).
3. BOROS, E., ELBASSIONI, K., GURVICH, V., AND KHACHYAN, L. An efficient incremental algorithm for generating all maximal independent sets in hypergraphs of bounded dimension. *Parallel Processing Letters* 10 (2000), 253–266.
4. BOROS, E., GURVICH, V., KHACHYAN, L., AND MAKINO, K. Generating partial and multiple transversals of a hypergraph. In *Automata, Languages and Programming*, U. Montanari, J. Rolim, and E. Welzl, Eds., vol. 1853 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2000, pp. 588–599.
5. EITER, T., AND GOTTLÖB, G. Identifying the minimal transversals of a hypergraph and related problems. *SIAM J. Comput.* 24, 6 (Dec. 1995), 1278–1304.
6. EITER, T., AND GOTTLÖB, G. Hypergraph transversal computation and related problems in logic and ai. In *Logics in Artificial Intelligence*, S. Flesca, S. Greco, G. Ianni, and N. Leone, Eds., vol. 2424 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2002, pp. 549–564.
7. EITER, T., GOTTLÖB, G., AND MAKINO, K. New results on monotone dualization and generating hypergraph transversals. *SIAM Journal on Computing* 32, 2 (2003), 514–537.
8. EITER, T., MAKINO, K., AND GOTTLÖB, G. Computational aspects of monotone dualization: A brief survey. *Discrete Appl. Math.* 156, 11 (June 2008), 2035–2049.
9. FREDMAN, M. L., AND KHACHYAN, L. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms* 21, 3 (1996), 618 – 628.
10. GURVICH, V., AND KHACHYAN, L. On generating the irredundant conjunctive and disjunctive normal forms of monotone boolean functions. *Discrete Applied Mathematics* 96–97, 0 (1999), 363 – 373.
11. HAUS, U.-U., KLAMT, S., AND STEPHEN, T. Computing knock-out strategies in metabolic networks. *Journal of Computational Biology* 15, 3 (2008), 259–268.
12. HÄDICKE, O., AND KLAMT, S. Computing complex metabolic intervention strategies using constrained minimal cut sets. *Metabolic Engineering* 13, 2 (2011), 204 – 213.
13. JUNGREUTHMAYER, C., NAIR, G., KLAMT, S., AND ZANGHELLINI, J. Comparison and improvement of algorithms for computing minimal cut sets. *BMC Bioinformatics* 14, 1 (2013), 318.

14. KHACHIYAN, L., BOROS, E., ELBASSIONI, K., AND GURVICH, V. An efficient implementation of a quasi-polynomial algorithm for generating hypergraph transversals and its application in joint generation. *Discrete Applied Mathematics* 154, 16 (2006), 2350 – 2372.
15. MURAKAMI, K., AND UNO, T. Efficient algorithms for dualizing large-scale hypergraphs. *Discrete Applied Mathematics* 170, 0 (2014), 83 – 94.
16. TODA, T. Hypergraph transversal computation with binary decision diagrams. In *Experimental Algorithms*, V. Bonifaci, C. Demetrescu, and A. Marchetti-Spaccamela, Eds., vol. 7933 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013, pp. 91–102.