Deriving Minimal Hitting-Sets for Linear Conflict Sets*

Xiangfu Zhao
Department of Computer, Zhejiang Normal University, Jinhua, Zhejiang Province, China
Email: xiangfuzhao@gmail.com

Abstract

Efficient generation of minimal hitting-sets (MHSes) for *large* conflict sets is an important topic for classical model-based diagnosis. So far, the structures of conflict sets have not been taken into account for computing MHSes by most previous methods, although they may play an important role. In this paper, a type of linear structure of conflict sets is proposed, and along with the classical strategy "divide and conquer" for solving problems with *large* sizes, an efficient approach for deriving MHSes is presented. In theory, compared to the direct "divide and conquer" without considering the linear structure of conflict sets, our approach decreases the complexity of each merge from *quadratic* to *linear*, with the product of numbers of MHSes for each sub-family of conflict sets. Experimental results show that our strategy has higher efficiency than many other well-known approaches, with even *several orders of magnitude* time (seconds) having been saved, evidently.

1 Introduction

Timely finding the faults of a device is vital for repairing and recovering it. For novel or high-tech devices with nearly no expert experience, model-based diagnostic technique has played an important role for error checking of such devices (e.g. the Livingstone system in "Deep Space 1" of NASA [NASA, 2016]). In classical model-based diagnosis, after generating conflict sets of components of a device, by checking consistency between the prediction of device model and real observation of the device, candidate diagnoses, which are the minimal hitting-sets (MHSes) for conflict sets, will be computed. As such, efficient computation of MHSes becomes a significant step for timely fault diagnosis.

Unfortunately, it is proved that generating MHSes for conflict sets is NP-hard [Karp, 1972; Garey and Johnson, 1979; Vinterbo and Ohrn, 2000; Fijany and Vatan, 2004], with exponential complexity with the number of basic elements in conflict sets. Thus, it will be even non-solvable when the size of conflict sets is very large.

There have been a huge number of approaches for deriving MHSes, such as HS-tree [Reiter, 1987], HS-DAG [Greiner *et al.*, 1989], HST-tree [Wotawa, 2001], the binary HS-tree (BHS-tree) [Lin and Jiang, 2003], the HSSE-tree [Zhao and Ouyang, 2006], the CSSE-tree/CSISE-tree [Zhao and Ouyang, 2007], the bi-partite graph based method [de Kleer, 2011]. Besides the above tree or graph based approaches, a conflict-based A^* algorithm [Williams and Ragno, 2007], the Boolean algebra based approach (abbreviated as Bool method in this paper) [Lin and Jiang, 2003] and related optimizations [Pill and Quaritsch, 2012], and the statics-based approximation approaches SAFARI [Feldman *et al.*, 2008] and STACCATO [Abreu and van Gemund, 2009], *etc.*, have also been proposed.

Besides, many other well-known problems, including the set-covering problem [Karp, 1972], the 0/1 integer programming problem [Fijany and Vatan, 2004], the minimal unsatisfiable subset problem [Liffiton and Sakallah, 2008], and the minimal hypergraph transversal and monotone dualization problem [Eiter *et al.*, 2008; Crama and Hammer, 2011; Boros and Makino, 2009; Khachiyan *et al.*, 2007] can also be seen as a hitting-set problem.

However, all of the previous approaches can be seen as *centralized*, until in recent years a *distributed* strategy was put forward [Zhao and Ouyang, 2013; 2015], with decreasing the complexity from $O(2^n)$ into $O(2^{n/d})$, where n and d are the number of basic elements in conflict sets and the number of distributions of all conflict sets.

Although the complexity is considerably reduced by the distributed strategy, it will be still rather complex if d is very small, for instance, when d = 1, that is, the conflict sets cannot be divided into many distributions.

In addition, most previous approaches to deriving MHSes for conflict sets did not consider the *structure* of conflict sets, which may give some clues for efficiently computing MHSes, just as designing proper "data *structures*" for developing efficient algorithms. In this regard, this paper would like to consider a series of conflict sets in a simple linear structure. Furthermore, the well-known "divide and conquer" strategy provides a perfect way to solving large size problems. Accordingly, this paper would like to propose an approach for deriving MHSes for large size conflict sets in linear structure, based on the "divide and conquer" strategy.

This paper is organized as follows. Some preliminaries about MHSes are introduced in the second section. The "divide and conquer" strategy and resolved sub-MHSes for deriving MHSes are presented in the third section. The linear structure of conflict sets is formalized and the related strat-

*This work was supported in part by Zhejiang Provincial Natural Science Foundation of China (No. LY16F020004); National Natural Science Foundation of China (Nos. 61003101, 61272208, and 61472369).

egy for deriving MHSes is proposed in the fourth section. Experimental results are shown in the fifth section.

2 Preliminaries

In this section, relevant background information on MHS and model-based diagnosis is introduced.

Definition 1. [1]

Let $\mathcal{F}$ be a non-empty family of sets. A set H is a hitting-set for $\mathcal{F}$ if the following two conditions hold:

(1) $H \subseteq \bigcup \mathcal{F}$;

(2) *For any set* $S \in \mathcal{F}$, $H \cap S \neq \emptyset$.

If no proper subset of H is a hitting-set for $\mathcal{F}$, then H is called a minimal hitting-set (MHS) for $\mathcal{F}$.

For simplicity, we also use $MHS(\mathcal{F})$ to denote the family of all MHSes for $\mathcal{F}$, in other words, $MHS(\mathcal{F}) = \{S \mid S \text{ is an MHS for } \mathcal{F}\}$.

Example 1. Let $\mathcal{F}$ = {{1, 2}, {2, 3}, {3, 4}, {5, 6}}. We have $MHS(\mathcal{F})$ = {{1, 3, 5}, {1, 3, 6}, {2, 3, 5}, {2, 3, 6}, {2, 4, 5}, {2, 4, 6}}.

Consider set {1, 3, 5} as a simple example to explain the meaning of MHS. First, obviously, {1, 3, 5} is related to all four sets in $\mathcal{F}$, thus, it is a hitting-set for $\mathcal{F}$. Second, if we remove 1 from {1, 3, 5}, then set {1, 2} in $\mathcal{F}$ cannot be related to {3, 5}; if we remove 3 from {1, 3, 5}, then {3, 4} cannot be related to {1, 5}; and if we remove 5 from {1, 3, 5}, then {5, 6} cannot be related to {1, 3}. Thus, {1, 3, 5} is minimal. In summary, {1, 3, 5} is one MHS for $\mathcal{F}$. ◇

In model-based diagnosis, MHSes for conflict sets were proposed to represent candidate diagnoses [Reiter, 1987; de Kleer and Williams, 1987]. We use a simple example to show the relations between MHSes and conflict sets for diagnosis as follows.

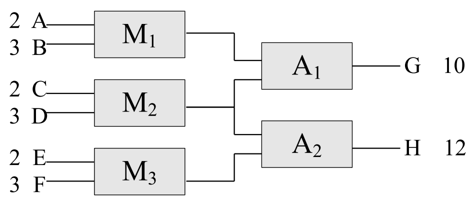

Figure 1: Multiplier-Adder example.

Example 2. Shown in Fig. 1 is a commonly used example in model-based diagnosis, where $M_1 - M_3$ are three multipliers, A_1 and A_2 are two adders. $A - F$ are six input terminals of three multipliers $M_1 - M_3$, with given input values 2, 3, 2, 3, 2, 3, respectively. G and H are two output terminals of A_1 and A_2, respectively.

Suppose we observe that the current output values of G and H are 10 and 12, respectively.

According to the topology of the system and the rules of addition and multiplication, we know that there are necessarily some components working abnormally[2].

Then, we can find two minimal conflict sets $\{M_1, M_2, A_1\}$ and $\{M_1, A_1, M_3, A_2\}$, that is, there will be a conflict if all of the components in each conflict set works normally. Thus, there is at least one component working abnormally in each conflict set.

In other words, each MHS of the conflict sets can be seen as a candidate diagnosis. For instance, MHSes $\{M_1\}$ and $\{A_1\}$ are single fault diagnoses, whereas $\{M_2, M_3\}$ and $\{M_2, A_2\}$ are multiple fault diagnoses.

Take $\{M_1\}$ as a simple example to show the result. When other components, except M_1, worked normally, it was possible that the output value of G was 10. That is, the component M_1 incidently worked abnormally and made a wrong output 4, which caused the output value of G to be 10. ◇

Therefore, in model-based diagnosis, generating MHSes for conflict sets is important for explanation of abnormality. The following question, how to improve the efficiency of computing MHSes for conflict sets, becomes a very important topic. As already known, unfortunately, the computation of hitting-sets is NP-hard, with the complexity being exponential in the number of basic elements in a family of conflict sets. Thus, dividing a large family of conflict sets into sub-families with *smaller* sizes would be a good choice for solving such a problem.

3 Divide and Conquer Strategy for Solving MHSes for A Large Family of Conflict Sets

3.1 Rationale for Direct Merge

It is well-known that "divide and conquer" is a proper strategy for solving problems with *large* sizes [Junker, 2004]. According to the strategy, the direct way is to first divide the family $\mathcal{F}$ of conflict sets into two sub-families $\mathcal{F}_1$ and $\mathcal{F}_2$; then, derive MHSes $MHS(\mathcal{F}_1)$ and $MHS(\mathcal{F}_2)$ for $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively with a smaller size recursively; finally, *directly merge* $MHS(\mathcal{F}_1)$ and $MHS(\mathcal{F}_2)$ to get $MHS(\mathcal{F})$ for $\mathcal{F}$.

Formally, the cross-product "$\times$" of two families by set union and minimization (Min) of a family by deleting proper supersets, are defined respectively as follows.

Given two families $\mathcal{F}_1$ and $\mathcal{F}_2$ of sets, then,

$$\mathcal{F}_1 \times \mathcal{F}_2 = \{S_1 \cup S_2 | S_1 \in \mathcal{F}_1, S_2 \in \mathcal{F}_2\}$$

Given a family $\mathcal{F}$ of sets, then,

$$Min(\mathcal{F}) = \{S \in \mathcal{F} | \forall S' \in \mathcal{F} (S \subseteq S' \vee (S \not\subset S' \wedge S' \not\subset S))\}$$

In fact, the rationale for the strategy is based on the following formula.

$$MHS\left(\mathcal{F}\right) = Min\left(MHS\left(\mathcal{F}_1\right) \times MHS\left(\mathcal{F}_2\right)\right) \quad (1)$$

According to Formula (1), Algorithm 1 describes the basic steps of the *direct* "divide and conquer" strategy for deriving MHSes.The main body of Algorithm 1 is the recursive function $DirectMerge$, in charge of dividing $\mathcal{F}$ (line 4) and merging MHSes for the divided $\mathcal{F}_1$ and $\mathcal{F}_2$ (line 7).

Each key merge includes two steps: computing cross-product and computing minimization with complexity $|\mathcal{M}_1||\mathcal{M}_2|$ and $|\mathcal{M}|^2$, respectively. Therefore, the complexity for each merge can be represented as $O((|\mathcal{M}_1||\mathcal{M}_2|)^2)$. Overall, it is the *minimization* operation that causes the complexity from *linear* to *quadratic* with the cardinality of cross-product of $\mathcal{M}_1$ and $\mathcal{M}_2$.

[1] In this paper, in order to discriminate different types of symbols, we usually use a lower-case letter (such as e), an upper-case letter (such as S), and a calligraphic letter (such as $\mathcal{F}$) respectively to denote a basic element of a set, a basic set including some basic elements, and a family of some basic sets.

[2] In model-based diagnosis, usually there is an assumption that the connections of components work normally.

Algorithm 1 Function $DirectMerge(\mathcal{F})$: Deriving MHSes by Direct Merge.

Input: A family $\mathcal{F}$ of (conflict) sets.
Output: $MHS(\mathcal{F})$ - the family of all MHSes for $\mathcal{F}$.

1: **if** ($\mathcal{F} = \{X\}$, i.e., $\mathcal{F}$ includes only one set X) **then**
2: **return** $\{\{c_i\} \mid c_i \in X, X \in \mathcal{F}\}$;
3: **end if**
4: Divide $\mathcal{F}$ into two disjoint sub-families $\mathcal{F}_1$ and $\mathcal{F}_2$: $\mathcal{F} = \mathcal{F}_1 \uplus \mathcal{F}_2$ (i.e., $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ and $\mathcal{F}_1 \cap \mathcal{F}_2 = \emptyset$);
5: $\mathcal{M}_1 = DirectMerge(\mathcal{F}_1)$;
6: $\mathcal{M}_2 = DirectMerge(\mathcal{F}_2)$;
7: $\mathcal{M} = Min(\mathcal{M}_1 \times \mathcal{M}_2)$;
8: **return** $\mathcal{M}$;

3.2 Resolved Sub-MHSes to Avoid Some Minimization

Actually, if one MHS $M_1 \in MHS(\mathcal{F}_1)$ is only related with $\mathcal{F}_1$ (i.e., there is no basic element in any set of $\mathcal{F}_2$ also in M_1), and similarly, if one MHS $M_2 \in MHS(\mathcal{F}_2)$ is only related with $\mathcal{F}_2$, then $(M_1 \cup M_2) \in MHS(\mathcal{F})$ holds without needing further complex minimization.

Accordingly, we resolve $\mathcal{M}_1$ $\big($shorten for $MHS(\mathcal{F}_1)\big)$ and $\mathcal{M}_2$ $\big($shorten for $MHS(\mathcal{F}_2)\big)$ into two disjoint sub-MHSes $\mathcal{M}_{11}$, $\mathcal{M}_{12}$, and $\mathcal{M}_{21}$, $\mathcal{M}_{22}$, respectively, where $\mathcal{M}_{11}$ is only related with $\mathcal{F}_1$, while $\mathcal{M}_{21}$ is only related with $\mathcal{F}_2$.

In other words, Let

$$C = \bigcup \mathcal{F}_1 \cap \bigcup \mathcal{F}_2$$

denote all common basic elements in both $\mathcal{F}_1$ and $\mathcal{F}_2$, then,

$$\mathcal{M}_{11} = \{M_{11} | M_{11} \in \mathcal{M}_1 \wedge M_{11} \cap C = \emptyset\}$$
$$\mathcal{M}_{12} = \{M_{12} | M_{12} \in \mathcal{M}_1 \wedge M_{12} \cap C \neq \emptyset\}$$
$$\mathcal{M}_{21} = \{M_{21} | M_{21} \in \mathcal{M}_2 \wedge M_{21} \cap C = \emptyset\}$$
$$\mathcal{M}_{22} = \{M_{22} | M_{22} \in \mathcal{M}_2 \wedge M_{22} \cap C \neq \emptyset\}$$

Clearly, $\mathcal{M}_{11}$ or $\mathcal{M}_{21}$ are only related with $\mathcal{F}_1$ or $\mathcal{F}_2$ respectively; while $\mathcal{M}_{12}$ and $\mathcal{M}_{22}$ are both related with $\mathcal{F}_1$ and $\mathcal{F}_2$.

Therefore, according to Formula (1), we get

$$\begin{aligned} & MHS(\mathcal{F}) = Min(\mathcal{M}_1 \times \mathcal{M}_2) \\ = \ & Min\big((\mathcal{M}_{11} \uplus \mathcal{M}_{12}) \times (\mathcal{M}_{21} \uplus \mathcal{M}_{22})\big) \\ = \ & Min\big((\mathcal{M}_{11} \times \mathcal{M}_{21}) \cup (\mathcal{M}_{11} \times \mathcal{M}_{22}) \cup \\ & (\mathcal{M}_{12} \times \mathcal{M}_{21}) \cup (\mathcal{M}_{12} \times \mathcal{M}_{22})\big) \\ = \ & (\mathcal{M}_{11} \otimes \mathcal{M}_{21}) \cup (\mathcal{M}_{11} \otimes \mathcal{M}_{22}) \cup \\ & (\mathcal{M}_{12} \otimes \mathcal{M}_{21}) \cup (\mathcal{M}_{12} \otimes \mathcal{M}_{22}) \end{aligned}$$

where "$\otimes$" is defined as

$$\mathcal{M}_{ki} \otimes \mathcal{M}_{(3-k)j} = \{H \in MHS(\mathcal{F}) | H \in \mathcal{M}_{ki} \times \mathcal{M}_{(3-k)j}\}$$

where $i, j, k \in \{1, 2\}$.

In other words, $\mathcal{M}_{ki} \otimes \mathcal{M}_{(3-k)j}$ denotes all MHSes in the direct cross-product of $\mathcal{M}_{ki}$ and $\mathcal{M}_{(3-k)j}$.

According to definitions of $\mathcal{M}_{11}$ and $\mathcal{M}_{21}$, each of which is only related with $\mathcal{F}_1$ or $\mathcal{F}_2$, respectively, then, we get

$$\mathcal{M}_{11} \otimes \mathcal{M}_{21} = \mathcal{M}_{11} \times \mathcal{M}_{21}.$$

In other words, the direct cross-product of $\mathcal{M}_{11}$ and $\mathcal{M}_{21}$ just contains MHSes for $\mathcal{F}$ *without* needing further minimization.

In the following, we take $\mathcal{M}_{11} \otimes \mathcal{M}_{22}$ as an example, to show how to try to *avoid* final complex *minimization* and still get the only MHSes for $\mathcal{F}$ in direct product $\mathcal{M}_{11} \times \mathcal{M}_{22}$.

Given any set $M_{22} \in \mathcal{M}_{22}$, which is not only an MHS for $\mathcal{F}_2$, but also for the family $\left(\mathcal{F}_2 \cup \bigcup_{X \in \mathcal{F}_1} \{X | X \cap M_{22} \neq \emptyset\}\right)$. Then, after minimizing $\mathcal{M}_{11}$ as $\mathcal{M}_{11}^{'}$, i.e., the family of MHSes for $\left(\mathcal{F}_1 - \bigcup_{X \in \mathcal{F}_1} \{X | X \cap M_{22} \neq \emptyset\}\right)$, the direct cross-product $\{M_{22}\} \times \mathcal{M}_{11}^{'}$ will just contain some MHSes for $\mathcal{F}$.

Let

$$\begin{aligned} Min\_\mathcal{M}_{11}(M_{22}, \mathcal{F}_1) = \Big\{ & M_{11}^{'} \subseteq M_{11} | M_{11} \in \mathcal{M}_{11}, M_{11}^{'} \\ & \in MHS\left(\mathcal{F}_1 - \bigcup_{X \in \mathcal{F}_1} \{X | X \cap M_{22} \neq \emptyset\}\right)\Big\} \end{aligned} \quad (2)$$

Formula (2) just denotes the minimized family $\mathcal{M}_{11}$, each set of which is an MHS for the minimized $\mathcal{F}_1$, where all sets related with set M_{22} are removed.

Then, we consider each $M_{22} \in \mathcal{M}_{22}$ and get

$$\mathcal{M}_{11} \otimes \mathcal{M}_{22} = \bigcup_{M_{22} \in \mathcal{M}_{22}} (\{M_{22}\} \times Min\_\mathcal{M}_{11}(M_{22}, \mathcal{F}_1))$$

Similarly, we get [3]

$$\mathcal{M}_{12} \otimes \mathcal{M}_{21} = \bigcup_{M_{12} \in \mathcal{M}_{12}} (\{M_{12}\} \times Min\_\mathcal{M}_{21}(M_{12}, \mathcal{F}_2))$$

$$\begin{aligned} \mathcal{M}_{12} \otimes \mathcal{M}_{22} &= \bigcup_{M_{12} \in \mathcal{M}_{12}} (\{M_{12}\} \times Min\_\mathcal{M}_{22}(M_{12}, \mathcal{F}_2)) \\ &= \bigcup_{M_{22} \in \mathcal{M}_{22}} (\{M_{22}\} \times Min\_\mathcal{M}_{12}(M_{22}, \mathcal{F}_1)) \end{aligned}$$

In summary, although a large number of complex minimization is avoided in computing $\mathcal{M}_{11} \otimes \mathcal{M}_{21}$, we still need a smaller number of minimization (in computing $Min\_\mathcal{M}_{ij}$) on $\mathcal{M}_{11} \otimes \mathcal{M}_{22}$, $\mathcal{M}_{12} \otimes \mathcal{M}_{21}$, and $\mathcal{M}_{12} \otimes \mathcal{M}_{22}$.

4 Linear Structure of Conflict Sets

Sometimes, structure information plays an important role for solving complex problems. As very clearly known, nearly all software engineers have studied the "data structure" course before they design practical algorithms to solve complex problems, since the structure of large number of data of a complex problem usually provides important information for solving it.

Similarly, we consider the structure of large conflict sets to try to find useful information to lower complexity for computing MHSes. On the other hand, as we know, the linear structure is very simple. Thus, we will consider a type of family of conflict sets in linear structure in the following.

[3]The subsequent three families $Min\_\mathcal{M}_{21}(M_{12}, \mathcal{F}_2)$, $Min\_\mathcal{M}_{22}(M_{12}, \mathcal{F}_2)$, and $Min\_\mathcal{M}_{12}(M_{22}, \mathcal{F}_1)$ are similarly defined as Formula (2).

4.1 Formalization of A Basic Linear Structure

A basic linear structure of conflict sets is defined as follows.

Definition 2. *Let $\mathcal{F} = \{S_1, S_2, \ldots, S_m\}$ be a family of conflict sets. $\mathcal{F}$ is organized in a basic linear structure if, for each $i \in [1 \cdots m]$,*

- $S_i \cap S_{i+1} \neq \emptyset$;
- $S_i \cap S_k = \emptyset, (i + 2 \leq k \leq m)$.

In other words, if a family $\mathcal{F}$ of sets is with a basic linear structure, then only any two neighbor sets share common elements. For instance, family $\mathcal{F}_1$ = {{1, 2}, {2, 3, 4}, {3, 4, 5}, {5, 6, 7}} is with a linear structure, whereas $\mathcal{F}_2 = \{\{1,2\},\{2,3\},\{3,4\},\{5,6\}\}$ is not, although some of its sets {1, 2}, {2, 3}, and {3, 4}, can be organized in linear structure.

Given $\mathcal{F} = \{S_1, S_2, \cdots, S_m\}$ ($m \geq 2$) a family of conflict sets in linear structure, let $\mathcal{F}_1 = \{S_1, S_2, \cdots, S_{mid}\}$ ($mid = m/2$), and $\mathcal{F}_2 = \{S_{mid+1}, S_{mid+2}, \ldots, S_m\}$, denote two divided disjoint sub-families of $\mathcal{F}$.

Like before, $C = \bigcup \mathcal{F}_1 \cap \bigcup \mathcal{F}_2$ denotes all common basic elements shared by $\mathcal{F}_1$ and $\mathcal{F}_2$; and now $C = S_{mid} \cap S_{mid+1}$ according to the linear structure of $\mathcal{F}$.

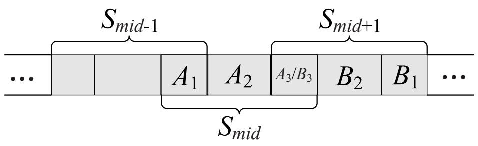

Figure 2: A general family of sets in linear structure.

Shown in Fig. 2 denotes a family $\mathcal{F}$ of sets in linear structure. Let $S_{mid} = A_1 \uplus A_2 \uplus A_3$, $S_{mid+1} = B_3 \uplus B_2 \uplus B_1$, $A_3 = B_3 = C$, $S_{mid-1} \cap S_{mid} = A_1$, $S_{mid+1} \cap S_{mid+2} = B_1$, then, $A_2 \cap \bigcup(\mathcal{F} - \{S_{mid}\}) = \emptyset$, and $B_2 \cap \bigcup(\mathcal{F} - \{S_{mid+1}\}) = \emptyset$. In other words, all basic elements in A_2 and B_2 are the exclusive ones in S_{mid} and S_{mid+1}, respectively.

According to Definition 2 and related properties, Formula (2) for computing $Min\_\mathcal{M}_{11}(M_{22}, \mathcal{F}_1)$ has the following simpler formalization if $\mathcal{F}$ is with a basic linear structure.

Proposition 1.

$$\begin{aligned} & Min\_\mathcal{M}_{11}(M_{22}, \mathcal{F}_1) \\ = \; & \{M_{11} - A_2 | M_{11} \in \mathcal{M}_{11}, M_{11} \cap A_2 \neq \emptyset\} \\ & \uplus \; \{M_{11} | M_{11} \in \mathcal{M}_{11}, M_{11} \cap A_1 \neq \emptyset, \\ & \quad (M_{11} - A_1) \cap S_{mid-1} = \emptyset\} \end{aligned} \tag{3}$$

Proof (sketch).
In Formula (2),

$$\bigcup_{X \in \mathcal{F}_1} \{X | X \cap M_{22} \neq \emptyset\} = \{S_{mid}\}.$$

Because in $\mathcal{F}_1$, only set S_{mid} is related with M_{22} with exactly one common element in C (or A_3 equivalently).

Then, in Formula (2), any M'_{11} is a subset of some $M_{11} \in \mathcal{M}_{11}$, also an MHS for $(\mathcal{F}_1 - \{S_{mid}\})$. Since any $M_{11} \in \mathcal{M}_{11}$ is an MHS for $\mathcal{F}_1$, which is also $(\mathcal{F}_1 - \{S_{mid}\}) \uplus \{S_{mid}\}$, then, there are two cases to consider as follows.

(a) $M'_{11} \cap A_1 = \emptyset$, then $\left(M_{11} - M'_{11}\right) \subseteq A_2$, that is, M_{11} can be seen as M'_{11} extended with exactly *one* element in A_2 (because $M_{11} \cap A_3 = \emptyset$, the extended element is not in A_3 here). Thus, M'_{11} = M_{11} - A_2.

(b) $M'_{11} \cap A_1 \neq \emptyset$, then there are two cases to consider.

- $(M_{11} - A_1) \cap S_{mid-1} = \emptyset$, then $M'_{11} = M_{11}$;
- $(M_{11} - A_1) \cap S_{mid-1} \neq \emptyset$, then $M'_{11} = M_{11} - A_1$; but all M'_{11}s in this case are contained in Case (a).

□

Similar as Proposition 1, we get the following formulas.

$$\begin{aligned} & Min\_\mathcal{M}_{12}(M_{22}, \mathcal{F}_1) \\ = \; & \{M_{12} - A_3 | M_{12} \in \mathcal{M}_{12}\} \quad (4) \\ = \; & \{M_{11} - A_2 | M_{11} \in \mathcal{M}_{11}, M_{11} \cap A_2 \neq \emptyset\} \quad (5) \end{aligned}$$

$$\begin{aligned} & Min\_\mathcal{M}_{21}(M_{12}, \mathcal{F}_2) \\ = \; & \{M_{21} - B_2 | M_{21} \in \mathcal{M}_{21}, M_{21} \cap B_2 \neq \emptyset\} \\ & \uplus \; \{M_{21} | M_{21} \in \mathcal{M}_{21}, M_{21} \cap B_1 \neq \emptyset, \\ & \quad (M_{21} - B_1) \cap S_{mid+2} = \emptyset\} \end{aligned} \tag{6}$$

$$\begin{aligned} & Min\_\mathcal{M}_{22}(M_{12}, \mathcal{F}_2) \\ = \; & \{M_{22} - B_3 | M_{22} \in \mathcal{M}_{22}\} \quad (7) \\ = \; & \{M_{21} - B_2 | M_{21} \in \mathcal{M}_{21}, M_{21} \cap B_2 \neq \emptyset\} \quad (8) \end{aligned}$$

Note: Formulas (4) and (5) are equal, since by simply speaking, they both denote all MHSes for $(\mathcal{F}_1 - \{S_{mid}\})$ without including any element in A_1. For the similar reason, Formulas (7) and (8) are equal.

Above formulas (3)-(8) show that we can easily implement the *minimization* operation in *linear* complexity, with the cardinality of considered family of sets, while *quadratic* in basic minimization by deleting proper supersets.

Accordingly, we get the following simpler formulas for computing $\mathcal{M}_{11} \otimes \mathcal{M}_{22}$, $\mathcal{M}_{12} \otimes \mathcal{M}_{21}$, and $\mathcal{M}_{12} \otimes \mathcal{M}_{22}$.

$$\mathcal{M}_{11} \otimes \mathcal{M}_{22} = \mathcal{M}_{22} \times Min\_\mathcal{M}_{11}(M_{22}, \mathcal{F}_1) \tag{9}$$
$$\mathcal{M}_{12} \otimes \mathcal{M}_{21} = \mathcal{M}_{12} \times Min\_\mathcal{M}_{21}(M_{12}, \mathcal{F}_2) \tag{10}$$
$$\mathcal{M}_{12} \otimes \mathcal{M}_{22} = \mathcal{M}_{22} \times Min\_\mathcal{M}_{12}(M_{22}, \mathcal{F}_1) \tag{11}$$
$$= \mathcal{M}_{12} \times Min\_\mathcal{M}_{22}(M_{12}, \mathcal{F}_2) \tag{12}$$

Based on above formulas (6), (8), (10), and (12), we get the following lemma about the relation between $\mathcal{M}_{12} \otimes \mathcal{M}_{22}$ and $\mathcal{M}_{12} \otimes \mathcal{M}_{21}$.

Lemma 1. $\mathcal{M}_{12} \otimes \mathcal{M}_{22} \subseteq \mathcal{M}_{12} \otimes \mathcal{M}_{21}$.

Similarly, according to (3), (5), (9), and (11), we get

Lemma 2. $\mathcal{M}_{12} \otimes \mathcal{M}_{22} \subseteq \mathcal{M}_{11} \otimes \mathcal{M}_{22}$.

Lemma 3. *Let $\mathcal{M}$ and $\mathcal{M}_1$ be the families of MHSes for $(\mathcal{F}_1 - \{S_{mid}\})$ and $\mathcal{F}_1$, respectively, where S_{mid} is defined* *as before, such that $S_{mid} = A_1 \uplus A_2 \uplus A_3$. Then,*

$$\begin{aligned}
&\mathcal{M}_{11}\\
=\ &\{M|M\cap A_1\neq\emptyset, M\in\mathcal{M}\}\uplus\{M\uplus\\
&\{e\}|e\in A_2, M\cap A_1=\emptyset, M\in\mathcal{M}\} \quad (13)\\
&\mathcal{M}_{12}\\
=\ &\{M\uplus\{e\}|e\in A_3, M\cap A_1=\emptyset,\\
&M\in\mathcal{M}\} \quad (14)\\
&\mathcal{M}_{12}\otimes\mathcal{M}_{22}\\
=\ &\{M|M\cap A_1=\emptyset, M\in\mathcal{M}\}\times\mathcal{M}_{22} \quad (15)\\
&and,\\
&\mathcal{M}_{11}\otimes\mathcal{M}_{22}\\
=\ &\{M|M\cap A_1=\emptyset, M\in\mathcal{M}\}\times\mathcal{M}_{22}\uplus\\
&\{M|M\cap A_1\neq\emptyset, M\in\mathcal{M}\}\times\mathcal{M}_{22} \quad (16)
\end{aligned}$$

Proof.(sketch)

(1) For each $M \in \mathcal{M}$, M is an MHS for $\mathcal{F}_1 - \{S_{mid}\}$, then,

- (i) If $M \cap A_1 = \{e\} \neq \emptyset$, then $M \in \mathcal{M}_1$, i.e., M is an MHS for $\mathcal{F}_1$. Meanwhile, $M \cap A_3 = \emptyset$ (thus, $M \in \mathcal{M}_{11}$). Since otherwise M is not minimal for $\mathcal{F}_1$. Because in $\mathcal{F}_1$, all elements in A_3 are only in S_{mid}. e is also in S_{mid}. So, if $M \cap A_3 = \{e'\} \neq \emptyset$, then, e' is redundant.
- (ii) If $M \cap A_1 = \emptyset$, then for each $M' \in \mathcal{M}_1 - \mathcal{M}$, $M' = M \uplus \{e\}, e \in A_2 \cup A_3$.
  - (a) If $M' = M \uplus \{e\}, e \in A_2$, then, $M' \cap A_3 = \emptyset$, thus, $M' \in \mathcal{M}_{11}$.
  - (b) If $M' = M \uplus \{e\}, e \in A_3$, then, $M' \cap A_3 \neq \emptyset$, thus, $M' \in \mathcal{M}_{12}$.

All of MHSes $\mathcal{M}_1$ for $\mathcal{F}_1$ can be obtained by MHSes $\mathcal{M}$ for $\mathcal{F}_1 - \{S_{mid}\}$ by the previous analysis.

Then, we get

$$\begin{aligned}
\mathcal{M}_{11} &= \{M|M\cap A_1\neq\emptyset, M\in\mathcal{M}\}\uplus\{M\uplus\{e\}|\\
&\quad e\in A_2, M\cap A_1=\emptyset, M\in\mathcal{M}\}\\
&\quad and,\\
\mathcal{M}_{12} &= \{M\uplus\{e\}|e\in A_3, M\cap A_1=\emptyset, M\in\mathcal{M}\}
\end{aligned}$$

(2)

$$\begin{aligned}
\mathcal{M}_{12}\otimes\mathcal{M}_{22} &= Min_\mathcal{M}_{12}\times\mathcal{M}_{22}\\
&= \{M-A_3|M\cap A_3\neq\emptyset,\\
&\quad M\cap A_1=\emptyset, M\in\mathcal{M}_{12}\}\times\mathcal{M}_{22}\\
&= \{M|M\cap A_1=\emptyset, M\in\mathcal{M}\}\times\mathcal{M}_{22}
\end{aligned}$$

$$\begin{aligned}
&\mathcal{M}_{11}\otimes\mathcal{M}_{22}\\
=\ &Min_\mathcal{M}_{11}\times\mathcal{M}_{22}\\
=\ &\{M-A_2|M\in\mathcal{M}_{11}\}\times\mathcal{M}_{22}\\
=\ &\{M-A_2|M\cap A_1=\emptyset, M\cap A_2\neq\emptyset, M\in\mathcal{M}\}\\
&\times\mathcal{M}_{22}\uplus\{M-A_2|M\cap A_1\neq\emptyset, M\cap A_2=\emptyset,\\
&M\in\mathcal{M}\}\times\mathcal{M}_{22}\\
=\ &\{M|M\cap A_1=\emptyset, M\in\mathcal{M}\}\times\mathcal{M}_{22}\uplus\\
&\{M|\mathcal{M}\cap A_1\neq\emptyset, M\in\mathcal{M}\}\times\mathcal{M}_{22}
\end{aligned}$$

□

Corollary 1. *Let $\mathcal{M}'$ and $\mathcal{M}_2$ be the families of MHSes for $(\mathcal{F}_2 - \{S_{mid+1}\})$ and $\mathcal{F}_2$, respectively, where S_{mid+1} is defined as before, such that $S_{mid+1} = B_1 \uplus B_2 \uplus B_3$. Then,*

$$\begin{aligned}
&\mathcal{M}_{21}\\
=\ &\{M|M\cap B_1\neq\emptyset, M\in\mathcal{M}'\}\uplus\{M\uplus\\
&\{e\}|e\in B_2, M\cap B_1=\emptyset, M\in\mathcal{M}'\} \quad (17)\\
&\mathcal{M}_{22}\\
=\ &\{M\uplus\{e\}|e\in B_3, M\cap B_1=\emptyset,\\
&M\in\mathcal{M}'\} \quad (18)\\
&\mathcal{M}_{12}\otimes\mathcal{M}_{22}\\
=\ &\{M|M\cap B_1=\emptyset, M\in\mathcal{M}'\}\times\mathcal{M}_{12} \quad (19)\\
&and,\\
&\mathcal{M}_{21}\otimes\mathcal{M}_{12}\\
=\ &\{M|M\cap B_1=\emptyset, M\in\mathcal{M}'\}\times\mathcal{M}_{12}\uplus\\
&\{M|M\cap B_1\neq\emptyset, M\in\mathcal{M}'\}\times\mathcal{M}_{12} \quad (20)
\end{aligned}$$

Based on the above formulas and two lemmas, we get the following interesting proposition.

Proposition 2.

$$\mathcal{M}_{12}\otimes\mathcal{M}_{22} = (\mathcal{M}_{12}\otimes\mathcal{M}_{21})\cap(\mathcal{M}_{11}\otimes\mathcal{M}_{22}).$$

Proof.(sketch)

Let

$$\mathcal{X}_1 = \{M|M\cap A_1\neq\emptyset, M\in\mathcal{M}\}, \text{ and}$$

$$\mathcal{X}_2 = \{M'|M'\cap B_1\neq\emptyset, M'\in\mathcal{M}'\}.$$

Let $M \uplus M_{22} \in \mathcal{X}_1 \times \mathcal{M}_{22}$, where $M \in \mathcal{X}_1$ and $M_{22} \in \mathcal{M}_{22}$, there exists $a_1 \in A_1$ such that $M = X \uplus \{a_1\}$, where $X \subset M$ and $a_1 \in A_1$.

Let $M' \uplus M_{12} \in \mathcal{X}_2 \times \mathcal{M}_{12}$, where $M' \in \mathcal{X}_2$ and $M_{12} \in \mathcal{M}_{12}$, there exists $b_1 \in B_1$ such that $M' = X' \uplus \{b_1\}$, where $X' \subset M'$ and $b_1 \in B_1$.

Since $a_1 \notin X'$, $a_1 \notin M_{12}$, and $b_1 \neq a_1$, then $a_1 \notin (X' \uplus \{b_1\} \uplus M_{12}) (= M' \uplus M_{12})$.

In other words, since $a_1 \in M \uplus M_{22}$ and $a_1 \notin M' \uplus M_{12}$, then $M \uplus M_{22} \neq M' \uplus M_{12}$.

Therefore, there is no common element between $\mathcal{X}_1 \times \mathcal{M}_{22}$ and $\mathcal{X}_2 \times \mathcal{M}_{12}$, that is, $(\mathcal{X}_1 \times \mathcal{M}_{22}) \cap (\mathcal{X}_2 \times \mathcal{M}_{12})$ = $\emptyset$.

According to formulas (15) and (16) in Lemma 3, and formulas (19) and (20) in Corollary 1, we get the conclusion. □

According to formulas (9), (11) and above propositions, we have the following proposition.

Proposition 3.

$$\begin{aligned}
&(\mathcal{M}_{11}\otimes\mathcal{M}_{22}) - (\mathcal{M}_{12}\otimes\mathcal{M}_{22})\\
=\ &\mathcal{M}_{22}\times\{M_{11}|M_{11}\in\mathcal{M}_{11}, M_{11}\cap A_1\neq\emptyset,\\
&(M_{11}-A_1)\cap S_{mid-1}=\emptyset\}. \quad (21)
\end{aligned}$$

Based on above formulas and propositions, we have the following proposition for final computation of MHSes for $\mathcal{F}$ in linear structure.

Proposition 4.

$$
\begin{aligned}
& MHS(\mathcal{F}) \\
= \ & (\mathcal{M}_{11} \times \mathcal{M}_{21}) \uplus \\
& (\mathcal{M}_{12} \times Min\_\mathcal{M}_{21}(M_{12}, \mathcal{F}_1)) \uplus \\
& ((\mathcal{M}_{11} \otimes \mathcal{M}_{22}) - (\mathcal{M}_{12} \otimes \mathcal{M}_{22})) \\
= \ & (\mathcal{M}_{11} \times \mathcal{M}_{21}) \uplus \\
& \Big(\mathcal{M}_{12} \times (\{M_{21} - B_2 | M_{21} \in \mathcal{M}_{21}, M_{21} \cap B_2 \neq \emptyset\} \\
& \uplus \{M_{21} | M_{21} \in \mathcal{M}_{21}, M_{21} \cap B_1 \neq \emptyset, \\
& (M_{21} - B_1) \cap S_{mid+2} = \emptyset\})\Big) \uplus \\
& (\mathcal{M}_{22} \times \{M_{11} | M_{11} \in \mathcal{M}_{11}, M_{11} \cap A_1 \neq \emptyset, \\
& (M_{11} - A_1) \cap S_{mid-1} = \emptyset\}). \qquad (22)
\end{aligned}
$$

4.2 Theoretical Complexity Analysis

For simplicity, on average, we assume that $|\mathcal{M}_{11}| = |\mathcal{M}_{12}| = \frac{1}{2}|\mathcal{M}_1|$, and similarly $|\mathcal{M}_{21}| = |\mathcal{M}_{22}| = \frac{1}{2}|\mathcal{M}_2|$. From Proposition 4, we can see that the complexity for computing each *merge* of MHSes for $\mathcal{F}$ in linear structure is about $\frac{3}{4}|\mathcal{M}_1||\mathcal{M}_2|$, in contrast, as previously stated that the complexity of direct merge without considering linear structure information is $(|\mathcal{M}_1||\mathcal{M}_2|)^2$.

In other words, the complexity for each key merge by our strategy has been decreased from *quadratic* into *linear*, with the number of the product of the two cardinalities of sub-MHS families for corresponding divided conflict sets.

4.3 Example Analysis

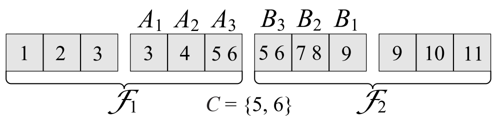

Figure 3: A linear family $\mathcal{F}$ of conflict sets.

Example 3. Shown in Fig. 3, $\mathcal{F}$ = {{1, 2, 3}, {3, 4, 5, 6}, {5, 6, 7, 8, 9}, {9, 10, 11}}, and $\mathcal{F}_1, \mathcal{F}_2, C, A_1, A_2, A_3, B_1, B_2, B_3$ are marked in Fig. 3. Now, for simplicity, we only show the last key *merge* operation for deriving $MHS(\mathcal{F})$ as follows, according to Proposition 4 and related formulas.

- Suppose we get MHSes $\mathcal{M}_1$ = {{3}, {1, 4}, {1, 5}, {1, 6}, {2, 4}, {2, 5}, {2, 6}} for $\mathcal{F}_1$; and $\mathcal{M}_2$ = {{9}, {5, 10}, {5, 11}, {6, 10}, {6, 11}, {7, 10}, {7, 11}, {8, 10}, {8, 11}} for $\mathcal{F}_2$.
- Divide $\mathcal{M}_1$ into $\mathcal{M}_{11}$ = {{3}, {1, 4}, {2, 4}}, and $\mathcal{M}_{12}$ = {{1, 5}, {1, 6}, {2, 5}, {2, 6}}; and divide $\mathcal{M}_2$ into $\mathcal{M}_{21}$ = {{9}, {7, 10}, {7, 11}, {8, 10}, {8, 11}}, and $\mathcal{M}_{22}$ = {{5, 10}, {5, 11}, {6, 10}, {6, 11}}.
- $\mathcal{M}_{11} \otimes \mathcal{M}_{21} = \mathcal{M}_{11} \times \mathcal{M}_{21}$ = { {3, 9}, {3, 7, 10}, {3, 7, 11}, {3, 8, 10},{3, 8, 11}, {1, 4, 9}, {1, 4, 7, 10}, {1, 4, 7, 11}, {1, 4, 8, 10}, {1, 4, 8, 11}, {2, 4, 9}, {2, 4, 7, 10}, {2, 4, 7, 11}, {2, 4, 8, 10}, {2, 4, 8, 11}}.
- $\mathcal{M}_{12} \otimes \mathcal{M}_{21} = \mathcal{M}_{12} \times Min\_\mathcal{M}_{21}(M_{12}, \mathcal{F}_2) = \mathcal{M}_{12} \times$ {{9}, {10}, {11}} = {{1, 5, 9}, {1, 5, 10}, {1, 5, 11}, {1, 6, 9}, {1, 6, 10}, {1, 6, 11}, {2, 5, 9}, {2, 5, 10}, {2, 5, 11}, {2, 6, 9}, {2, 6, 10}, {2, 6, 11}}
- $(\mathcal{M}_{11} \otimes \mathcal{M}_{22}) - (\mathcal{M}_{12} \otimes \mathcal{M}_{22}) = \mathcal{M}_{22} \times \{\{3\}\}$ = {{3, 5, 10}, {3, 5, 11}, {3, 6, 10}, {3, 6, 11}}.
- Finally, all MHSes generated by above steps are composed into $MHS(\mathcal{F})$. ◇

4.4 Algorithm Description

Algorithm 2 Function $LinearMerge(\mathcal{F})$: Deriving MHSes for Family $\mathcal{F}$ of Conflict Sets in Linear Structure.

Input: A family $\mathcal{F}$ of conflict sets in linear structure.
Output: $MHS(\mathcal{F})$ - the family of MHSes for $\mathcal{F}$.

1: **if** ($\mathcal{F} == \{X\}$, i.e., $\mathcal{F}$ includes only one set X) **then**
2:     **return** $\{\{c_i\} \mid c_i \in X, X \in \mathcal{F}\}$;
3: **end if**
4: Divide $\mathcal{F}$ into two disjoint sub-families $\mathcal{F}_1$ and $\mathcal{F}_2$;
5: $C = B_3 = A_3 = S_{mid} \cap S_{mid+1}$;
6: $A_1 = S_{mid-1} \cap S_{mid}$; $B_1 = S_{mid+1} \cap S_{mid+2}$;
7: $A_2 = S_{mid} - A_1 - A_3$; $B_2 = S_{mid+1} - B_1 - B_3$;
8: $\mathcal{M}_1 = LinearMerge(\mathcal{F}_1)$;
9: $\mathcal{M}_2 = LinearMerge(\mathcal{F}_2)$;
10: Divide $\mathcal{M}_1$ into two disjoint sub-families $\mathcal{M}_{11}$ and $\mathcal{M}_{12}$, according to they are unrelated with C or not;
11: Divide $\mathcal{M}_2$ into two disjoint sub-families $\mathcal{M}_{21}$ and $\mathcal{M}_{22}$, according to whether they are unrelated with C or not;
12: $\mathcal{M}_{1121} = \mathcal{M}_{11} \times \mathcal{M}_{21}$;
13: $\mathcal{M}_{1221} = \mathcal{M}_{12} \times Min\_\mathcal{M}_{21}(M_{12}, \mathcal{F}_2)$;
14: $\mathcal{M}_{1122-1222} = \mathcal{M}_{22} \times \{M_{11} | M_{11} \in \mathcal{M}_{11}, M_{11} \cap A_1 \neq \emptyset, (M_{11} - A_1) \cap S_{mid-1} = \emptyset\}$;
15: $\mathcal{M} = \mathcal{M}_{1121} \uplus \mathcal{M}_{1221} \uplus \mathcal{M}_{1122-1222}$;
16: **return** $\mathcal{M}$;

Shown in Algorithm 2 describes the main steps for computing MHSes for a family of conflict sets in linear structure. In this algorithm, lines 12-15 just denote the detailed process for computing formula (22). Clearly, we can see that we do not call complex Function Min, which is used in Algorithm 1, and then the complexity for each key *merge* is reduced by *one order of magnitude*, with the value of $(|\mathcal{M}_1||\mathcal{M}_2|)$.

5 Implementation

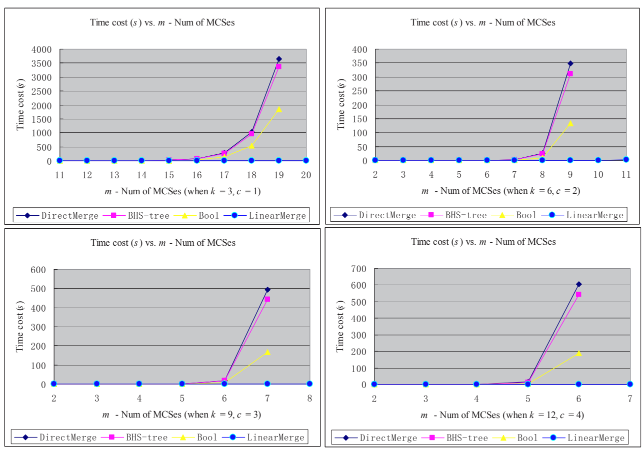

Figure 4: Comparisons of efficiency (time cost) among the *LinearMerge*, *Bool method*, *BHS-tree*, and *DirectMerge*.

Table 1: Time cost (s) for Computation of MHSes with four methods, where MCS is shorten for *minimal conflict set*, *L-Merge* for *LinearMerge*, *D-Merge* for *DirectMerge*, and **Num-MHS** for the number of generated MHSes.

MCS-Family	***L-Merge***	*Bool*	*BHS-tree*	*D-Merge*	**Num-MHS**
$\mathcal{D}_{8\_1\_6}$	**0.37**	392.08	520.99	702.26	**73,207**
$\mathcal{D}_{9\_1\_6}$	**0.81**	2,212.25	2,875.01	3,812.02	**170,625**
$\mathcal{D}_{10\_1\_5}$	**0.24**	133.07	171.10	240.58	**44,173**
$\mathcal{D}_{11\_1\_5}$	**0.44**	441.78	557.52	753.72	**76,631**
$\mathcal{D}_{12\_2\_5}$	**0.32**	247.60	392.10	534.97	**58,128**
$\mathcal{D}_{12\_4\_6}$	**0.28**	188.12	541.25	605.43	**51,264**
$\mathcal{D}_{13\_3\_5}$	**0.34**	134.68	276.24	380.88	**43,897**
$\mathcal{D}_{14\_2\_5}$	**0.78**	2,556.21	2,818.65	3,803.81	**156,248**
$\mathcal{D}_{14\_3\_5}$	**0.38**	432.40	810.05	1,090.05	**74,981**
$\mathcal{D}_{15\_3\_5}$	**0.59**	1,193.14	2,277.27	2,826.19	**122,175**

The proposed algorithm $LinearMerge$ was implemented. The experiments were carried out on a PC with Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz, 4096MB RAM, *Windows 7*, and *C++*. We tested four algorithms *LinearMerge*, *Bool method*, *BHS-tree*, and the *DirectMerge* (Algorithm 1), on large number of linear families of conflict sets.

Synthetic examples of *linear* conflict sets were designed as follows.We assume that each element in a conflict set is denoted by a natural number 0, 1, $\cdots$. Each family $\mathcal{D}_{k\_c\_m}$ of conflict sets is with m conflict sets, where each conflict set is of cardinality k, and the number of common elements (the set C) shared by two neighbor sets is c. Specifically, the m conflict sets with the same cardinality k ($k >= 3$) are like: $\{0, 1, 2, \cdots, k-1\}$, $\{k-c, k-c+1, \cdots, (k-c)+k-1\}$, $\{2*(k-c), 2*(k-c)+1, \cdots, 2*(k-c)+k-1\}, \cdots$, $\{m*(k-c), m*(k-c)+1, \cdots, m*(k-c)+k-1\}$. [4]

For a simple example, $\mathcal{D}_{6\_2\_4}$ (i.e., there are 4 conflict sets with the cardinality being 6, and the number of common elements between two neighbor sets is 2) is the family of sets: { {0, 1, 2, 3, 4, 5}, {4, 5, 6, 7, 8, 9}, {8, 9, 10, 11, 12, 13}, {12, 13, 14, 15, 16, 17} }.

Some typical implemented results[5] are shown in Fig. 4 and Table 1, from which we can see:

- The time cost is increasing considerably with the number of conflict sets.
- The time cost is increasing considerably with the cardinality of a conflict set in an MCS family.
- The time cost is decreasing with the number of common elements shared by two neighbor conflict sets in an MCS family.
- More important, our algorithm $LinearMerge$ has much obviously higher efficiency than the *DirectMerge*, *BHS-tree*, and the well-known *Bool method*[6].
- Specifically, for instance, for typical MCS families in Table 1, we can easily get all the large number (Num-MHS, some are more than 100,000) of MHSes by our algorithm $LinearMerge$ only in *less than one second*, however, all other three algorithms need *several hundred/thousand seconds*, or even more than 3,600 seconds (1 hour).

Thus, our algorithm $LinearMerge$ is much suitable to deriving MHSes for large conflict sets in linear structure.

6 Conclusion

In this paper, the *structure* information of conflict sets was studied to accelerating computation of MHSes. More specifically, a basic linear structure of conflict sets was defined, related properties were formalized, and an algorithm based on the "divide and conquer" strategy for deriving MHSes were presented. Compared to the direct "divide and conquer" strategy without considering the linear structure, our algorithm decreases the complexity of each key *merge* from ***quadratic*** ($O^2(|M_1||M_2|)$ to ***linear*** ($O(|M_1||M_2|)$, with the number ($|M_1|$ or $|M_2|$) of MHSes for each divided subfamily ($\mathcal{F}_1$ or $\mathcal{F}_2$) of conflict sets.

Experiments on large number of regular synthetic examples of linear conflict sets also showed that our algorithm got the best efficiency, compared to the direct merge, BHS-tree, and the well-known Boolean algebra based algorithm.

This paper is different from [Zhao and Ouyang, 2015], as follows.

- [Zhao and Ouyang, 2015] divides the whole family into many sub-families. But an implicit problem is that there may exist some sub-families still very large, so, it is still complex. In this paper, actually, there is an implicit assumption that the family cannot be divided into many sub-families (if it can then we just use the approach in [Zhao and Ouyang, 2015] as the first choice). In this case, we tried to exploit structure to efficiently solve the problem.
- [Zhao and Ouyang, 2015] uses the join relation to divide the family, which generally is more complex than linear relation, although, actually, linear relation can be seen as a special join relation.
- In [Zhao and Ouyang, 2015], each sub-family shares no basic element with each other. This paper divides $\mathcal{F}$ into $\mathcal{F}_1$ and $\mathcal{F}_2$ but with common basic elements between them.

Although a basic linear structure of conflict sets was studied in this paper, just like variable linear data structures (such as circular linear structure, etc.), diversity of linear structures of conflict sets with diversity of properties and algorithms are still worthwhile for future work.

Besides the linear structure, other structures, such as the tree structure, the graph structure, or hybrid structure of conflict sets may also be worthwhile studying for deriving all MHSes, which are also one of our future work.

Of course, given a family of conflict sets, how to determine its structure and choose a proper algorithm is another interesting research topic in the future.

A lot of benchmark data (e.g. the ISCAS-85 conflict sets) are needed to be tested in future, to verify how often the sets have a linear structure.

In addition, similar as distributed [Cardoso and Abreu, 2013] or parallel [Jannach *et al.*, 2016] environment for fast model-based diagnosis, MHS generation algorithms for large conflict sets in some structures in such configuration are also to be studied in future.

[4] The rationale for designing such artificial examples is mainly the fact that, they are properly regular in basic linear structure and can easily be extended to a very larger size in the similar way.

[5] Here the divide-and-conquer algorithm is not run in a parallel computer. The *LinearMerge* algorithm does not run merging in parallel. We will get much higher efficiency if we implement merging in parallel.

[6] From [Lin and Jiang, 2003], in many cases, the $Bool\ method$ has better efficiency than HS-tree and BHS-tree methods.

References

[Abreu and van Gemund, 2009] R. Abreu and A. van Gemund. A low-cost approximate minimal hitting set algorithm and its application to model-based diagnosis. In *Proceedings of the 8th Symposium on Abstraction, Reformulation, and Approximation (SARA-09)*, pages 2–9. AAAI Press, 2009.

[Boros and Makino, 2009] E. Boros and K. Makino. A fast and simple parallel algorithm for the monotone duality problem. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP-09)*, pages 183–194, Berlin, Heidelberg, 2009. Springer-Verlag.

[Cardoso and Abreu, 2013] N. Cardoso and R. Abreu. MHS^2: a map-reduce heuristic-driven minimal hitting set search algorithm. In *Proceedings of the 2nd International Conference on Multicore Software Engineering, Performance, and Tools (MUSEPAT-13)*, pages 25–36, 2013.

[Crama and Hammer, 2011] Y. Crama and P. L. Hammer. *Boolean Functions - Theory, Algorithms, and Applications*, volume 142 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 2011.

[de Kleer and Williams, 1987] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987.

[de Kleer, 2011] J. de Kleer. Hitting set algorithms for model-based diagnosis. In *Proceedings of the 22nd International Workshop on Principles of Diagnosis (DX-11)*, pages 100–105, 2011.

[Eiter *et al.*, 2008] T. Eiter, K. Makino, and G. Gottlob. Computational aspects of monotone dualization: a brief survey. *Discrete Applied Mathematics*, 156(11):2035–2049, 2008.

[Feldman *et al.*, 2008] A. Feldman, G. M. Provan, and A. J. C. van Gemund. Computing minimal diagnoses by greedy stochastic search. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-08)*, pages 911–918. AAAI Press, 2008.

[Fijany and Vatan, 2004] A. Fijany and F. Vatan. New approaches for efficient solution of hitting set problem. In *Proceedings of the Winter International Symposium on Information and Communication Technologies (WISICT-04)*, pages 1–10, 2004.

[Garey and Johnson, 1979] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[Greiner *et al.*, 1989] R. Greiner, B. A. Smith, and R. W. Wilkerson. A correction to the algorithm in reiter's theory of diagnosis. *Artificial Intelligence*, 41(1):79–88, 1989.

[Jannach *et al.*, 2016] D. Jannach, T. Schmitz, and K. Shchekotykhin. Parallel model-based diagnosis on multi-core computers. *Journal of Artificial Intelligence Research*, 55:835–887, 2016.

[Junker, 2004] U. Junker. Quickxplain: preferred explanations and relaxations for over-constrained problems. In *Proceedings of the 19th national conference on Artifical intelligence (AAAI-04)*, pages 167–172. AAAI Press, 2004.

[Karp, 1972] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*. Plenum Press, New York, 1972.

[Khachiyan *et al.*, 2007] L. Khachiyan, E. Boros, K. Elbassioni, and V. Gurvich. A global parallel algorithm for the hypergraph transversal problem. *Information Processing Letters*, 101(4):148–155, 2007.

[Liffiton and Sakallah, 2008] M. H. Liffiton and K. A. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. *Journal of Automated Reasoning*, 40(1):1–33, 2008.

[Lin and Jiang, 2003] L. Lin and Y. Jiang. The computation of hitting sets: review and new algorithms. *Information Processing Letters*, 86(4):177–184, 2003.

[NASA, 2016] NASA. The livingstone model-based diagnosis system. `https://ti.arc.nasa.gov/opensource/projects/livingstone2/`, 2016.

[Pill and Quaritsch, 2012] I. Pill and T. Quaritsch. Optimizations for the boolean approach to computing minimal hitting sets. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI-12)*, pages 648–653. IOS Press, 2012.

[Reiter, 1987] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.

[Vinterbo and Ohrn, 2000] S. Vinterbo and A. Ohrn. Minimal approximate hitting sets and rule templates. *International Journal of Approximate Reasoning*, 25(2):123–143, 2000.

[Williams and Ragno, 2007] B. C. Williams and R. J. Ragno. Conflict-directed A* and its role in model-based embedded systems. *Discrete Applied Mathmatics*, 155(12):1562–1595, 2007.

[Wotawa, 2001] F. Wotawa. A variant of reiter's hitting-set algorithm. *Information Processing Letters*, 79(1):45–51, 2001.

[Zhao and Ouyang, 2006] X. Zhao and D. Ouyang. A method of combining SE-tree to compute all minimal hitting sets. *Progress in Natural Science*, 16(2):169–174, 2006.

[Zhao and Ouyang, 2007] X. Zhao and D. Ouyang. Improved algorithms for deriving all minimal conflict sets in model-based diagnosis. In *Proceedings of the 3rd International Conference on Intelligent Computing (ICIC-07)*, volume 4681 of *Lecture Notes in Computer Science*, pages 157–166. Springer, 2007.

[Zhao and Ouyang, 2013] X. Zhao and D. Ouyang. A distributed strategy for deriving minimal hitting-sets. In *Proceedings of the 24th International Workshop on Principles of Diagnosis (DX-13)*, pages 33–38, 2013.

[Zhao and Ouyang, 2015] X. Zhao and D. Ouyang. Deriving all minimal hitting sets based on join relation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(7):1063–1076, 2015.