计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239,CN 11-1777/TP

《计算机研究与发展》网络首发论文

题目： 两种计算命题极小模型的新方法
作者： 张丽，王以松，谢仲涛，冯仁艳
收稿日期： 2020-06-05
网络首发日期： 2021-03-02
引用格式： 张丽，王以松，谢仲涛，冯仁艳．两种计算命题极小模型的新方法．计算机研究与发展. https://kns.cnki.net/kcms/detail/11.1777.tp.20210302.1327.008.html

网络首发时间：2021-03-02 16:21:39
网络首发地址：https://kns.cnki.net/kcms/detail/11.1777.tp.20210302.1327.008.html

 DOI:10.7544/issn1000-1239.2021.20200370

两种计算命题极小模型的新方法

张丽[1] 王以松[1,2] 谢仲涛[1] 冯仁艳[1]

[1]（贵州大学计算机科学与技术学院 贵阳 550025）

[2]（公共大数据国家重点实验室(贵州大学) 贵阳 550025）

（gs.lizhang18@gzu.edu.cn）

Two New Approaches of Computing Propositional Minimal Models

Zhang Li[1], Wang Yisong[1,2], Xie Zhongtao[1], and Feng Renyan[1]

[1]（*College of Computer Science and Technology, Guizhou University, Guiyang* 550025）

[2]（*State Key Laboratory of Public Big Data* (*Guizhou University*), *Guiyang* 550025）

Abstract Computing minimal models is an essential task in many reasoning systems in artificial intelligence. However, even for a positive conjunctive normal form (CNF) formula, the tasks of computing and checking its minimal model are not tractable. To compute a minimal model of a clause theory, one of the current main methods is to convert the clause theory into a disjunction logic program and use an answer set program (ASP) solver to compute its stable model. This paper proposes a method MMSAT for computing minimal models based on (satisfiability problem) SAT solvers. In terms of the recently proposed minimal reduct based minimal checking algorithm CheckMinMR, a minimal model decomposing based minimal model computing algorithm MRSAT is proposed. Finally, the two algorithms are evaluated by a large number of randomly generated 3CNF formulas and industrial benchmarks from the SAT international competition. Experimental results show that the two methods proposed in this paper compute minimal models at about three times faster than clingo for random 3CNF formulas and slightly faster than clingo for industrial SAT benchmarks. Thus, they are effective. In addition, it also reveals that clingo sometimes incorrectly computes a minimal model. Thus, the two proposed methods in this paper are more stable than clingo. Experimental code and data address: https://github.com/zhangli-hub123/minimal-model.

Key words minimal model; SAT solver; CNF formula; minimal reduct; decomposing minimal model

摘要 计算命题公式的极小模型在人工智能推理系统中是一项必不可少的任务．然而，即使是正 CNF（conjunctive normal form）公式，其极小模型的计算和验证都不是易处理的．当前，计算 CNF 公式极小模型的主要方法之一是将其转换为析取逻辑程序后用回答集程序(answer set programming，ASP)求解器计算其稳定模型/回答集．针对计算 CNF 公式的极小模型的问题，提出一种基于可满足性(satisfiability problem，SAT)求解器的计算极小模型的方法 MMSAT；然后结合最近基于极小归约的极小模型验证算法 CheckMinMR，提出了基于极小模型分解的计算极小模型方法 MRSAT；最后对随机生成的大量的 3CNF 公式和 SAT 国际竞赛上的部分工业基准测试用例进行测试，实验结果表明，MMSAT 和 MRSAT 对随机 3CNF 公式和 SAT 工业测试用例都是有效的，且计算极小模型的速度都明显快于最新版的 clingo，并且在 SAT 工业实例上发现了 clingo 有计算出错的情况，而 MMSAT 和 MRSAT 则更稳定．实验代码及数据地址：https://github.com/zhangli-hub123/minimal-model.

关键词 极小模型；SAT 求解器；CNF 公式；极小归约；极小模型分解

中图法分类号 TP3-05

收稿日期：2020-06-05；**修回日期：**2021-01-08
基金项目：国家自然科学基金项目（61976065，U1836205）
This work was supported by the National Natural Science Foundation of China (61976065 ,U1836205).
通信作者：王以松（yswang@gzu.edu.cn）

命题可满足性（satisfiability problem，SAT）问题是计算机科学和人工智能研究的中心问题之一，在自动推理和人工智能等领域都具有非常重要的理论意义和实践价值，世界各国的相关研究人员在这方面做了大量的工作，提出了许多求解算法和大量的改进技术.

SAT 问题的模型即为 CNF（conjunctive normal form）公式可满足时，使得命题公式可满足的一组真值指派中赋值为真的原子集合.当命题公式可满足时，极小模型的计算和验证问题就成了的人们关注的重点问题.当命题公式不可满足时，人们通常对分析不可满足性感兴趣.极大可满足（maximum satisfiability problem，MaxSAT）问题[1-2]和极小不可满足子集（minimal unsatisfiable subset，MUS）问题都属于这种分析.MaxSAT 是 SAT 问题的优化版，其目标是找到一组真值指派使得 CNF 公式中满足（不满足）子句的数量极大化（极小化）.随着 MaxSAT 技术的不断发展，MaxSAT 问题在 Android 恶意软件检测[3]、排课[4]和诊断[5]等问题中都得到了很好的应用.MUS 是 SAT 问题的扩展，是计算一公式集的极小不可满足公式子集，其所有真子集均是可满足的.在现实中许多重要问题可以编码为 MUS 问题进行求解[6-8].

基于极小模型的推理一直是人工智能研究的重要主题[9-11].极小模型也是回答集程序设计（answer set programming，ASP）和其他非单调知识表示和推理范式的核心[12]，例如，限制逻辑[13-16]、缺省逻辑[17]、极小诊断[18-22]和稳定模型语义下的逻辑程序[23-24]等.极小模型主要涉及 2 个任务，对于一个给定的子句理论 T，计算（任务 1）：寻找极小模型即计算出 T 的一个极小模型；判定（任务 2）：验证极小模型即检验给定的一个原子集是否是 T 的极小模型.

关于极小模型推理的研究结果表明，在一般情况下，这些问题是难以处理的.事实上，即使是正子句理论，计算其极小模型也是 $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$-hard[25].对于一个给定的理论，验证一个模型是否是其极小模型是 co-NP-complete[26].关于极小模型求解这一问题的研究也吸引了一部分专家学者的注意，他们认为挑选出能够有效解决这些问题的各个理论是有意义的[27-29].目前计算子句理论的极小模型，可将子句理论转换成逻辑程序后用 ASP 求解器计算其回答集，ASP 求解器的典型代表有 clasp[30]，clingo[31]，DLV[32]等.

2016 年 Ben-Eliyahu-Zohary 等人[33-34]提出了基于极小模型分解的计算极小模型的算法 ModuMin 和验证极小模型的算法 CheckMin，使极小模型计算和验证的艰巨任务在原始理论的子集之间进行分解，把一个任务分解成多个子任务进行计算.但是该验证算法 CheckMin 并不可靠.

2020 年 Wang 等人①对这一问题展开了进一步的研究，提出了极小归约（minimal reduct，MR）.极小归约是对 Ben-Eliyahu-Zohary 等人[33-34]的分解极小模型定理的补充，从而得到一个可靠的验证极小模型的算法 CheckMinMR.

每年可满足性理论和应用方面的国际会议都会组织 SAT 竞赛，以求能够找到一组最快的 SAT 求解器.MiniSAT[35]是一个极简并开源的 SAT 求解器，赢得了 2005 年 SAT 竞赛的所有工业类别测试.MiniSAT 的出现对于 SAT 问题的未来研究以及使用 SAT 的应用都是一个很好的起点.MiniSAT 始于 2003 年，其目的是通过小型而高效的，并且提供有良好文档的 SAT 求解器来帮助人们进入 SAT 领域.其第 1 个版本只有 600 多行 C 语言代码，同时 MiniSAT 仍包含了 2003 年最新 SAT 求解的核心算法.在之后的版本中，代码量虽有所增长，但相较于其他的 SAT 求解器而言 MiniSAT 的代码量仍然非常小，为我们的代码实现提供了非常有利的基础.

本文的主要工作有 3 个部分：首先在 2013 年 Eén 等人[36]推出的最新版本 MiniSAT2.2 求解器的基础上，对源代码进行修改实现了计算子句理论极小模型的算法 MMSAT.

其次将 MMSAT 算法与 CheckMinMR 算法结合成 MRSAT 实现快速极小模型的求解，将 2 个算法进行结合的目的是在验证一个模型是否是极小模型时，采用分解模型的思想将 1 个任务分解成多个子任务的方法来验证.即分解一个理论及其模型，当两者都变为空时，则意味着该模型确实是给定理论的极小模型.

最后使用本文的 2 个算法对大量随机生成的子句理论以及 SAT 国际竞赛上的部分基准测试用例进行极小模型的计算；本文选择 clingo 作为评估标准，将所有测试用例分别转换成逻辑程序，使用 clingo 计算其回答集，clingo 是目前计算逻辑程序回答集最有效的实现，它是 gringo 和 clasp 的组合；同时记录本文的 2 个算法计算极小模型使用的时间及 clingo 使用的时间.实验结果表明，本文提出的 2 个方法对随机子句理论和 SAT 竞赛工业测试用例十分有效，计算极小模型的速度都明显快于 clingo.而且从在计算结果正确率上的表现来看，本文的 2 个方法也更加稳定.

① 个人通信，Wang Yisong, Eiter Thomas, Zhang Yuanlin, Lin Fangzhen, 2020.

1 预备知识

本节介绍回答集程序[21-22]、极小模型、依赖图、极小模型分解[31-32]和极小归约[33]的基本思想和基础概念.假设一个命题语言 L,其原子公式集 A 是有穷原子集 $\{p_1, \cdots ,p_n\}$ ，$S\subseteq A$ ，我们记 $\overline{S}=A-S$ ，$\neg S=\{\neg p \mid p\in S\}$，.原子公式和原子公式的否定称为文字.

子句 δ 是由 1 个或多个文字通过逻辑或（$\vee$）连接组成.子句理论 Σ 是由 1 组有限子句构成.SAT 问题中的 CNF 公式由 1 个或多个子句通过逻辑与连接起来组成.若子句 δ 在模型 M 下为真，则称 M 满足 δ，记为 $M \models \delta$.

定义 1. 极小模型. 给定一个命题公式 Φ 及其一个模型 M，称 M 是 Φ 的极小模型，当且仅当不存在 $M'\subset M$ ，使得 Φ 可满足.

一个析取逻辑程序 P 是由具有以下形式的规则构成的有限集合：

$$a_1 \vee \cdots \vee a_m \leftarrow a_{m+1}, \cdots, a_n, \text{not}\, a_{n+1}, \cdots, \text{not}\, a_l \quad (1)$$

其中 a_i（$1\leqslant i\leqslant l$）是原子，$a_1 \vee \cdots \vee a_m$ 表示该规则的头（head），$a_{m+1}, \cdots, a_n, \text{not}\, a_{n+1}, \cdots, \text{not}\, a_l$ 表示该规则的体（body）.令 r 是形如式（1）的规则，记 $H(r)=\{a_1, a_2, \cdots, a_m\}$ ，$B^+(r)=\{a_{m+1}, \cdots, a_n\}$ ，$B^-(r)=\{a_{n+1}, \cdots, a_l\}$，$B(r)=B^+(r)\cup \text{not}\, B^-(r)$，通常将规则 r 表示成 $H(r)\leftarrow B(r)$.令 $A(r)$表示规则 r 中的原子的集合,$A(P)$表示逻辑程序 P 中的原子的集合.当 $l=n$ 时，规则 r 被称作是正规则，如果程序中所有规则都是正规则，则该程序为正程序.令 $M\subseteq A(P)$，P 关于 M 进行 GL-归约的结果记为 P^M，$P^M=\{H(r)\leftarrow B^+(r) \mid r\in P, B^-(r)\cap M=\varnothing\}$.若 M 是 P^M 的极小模型，则称 M 就是 P 的稳定模型（也称为回答集）.

一个子句 c 可转换析取逻辑程序的规则 r，且 $B^-(r)=\varnothing$,即由 CNF 公式转换的逻辑程序为正程序.因为对正程序 P，$P^M=P$，故正程序的回答集就是其极小模型.

1. 1 依赖图

令 P 是逻辑程序,P 的依赖图(dependency graph)是一个有向图，记为 $G_P=(V, E)$.G_P 的定义为:

1) P 中的所有原子 $A(P)$和子句 δ 都是 G_P 中的节点;

2) 如果 $a\in B^+(\delta)$，则对应的边 e 表示为(a, δ);如果 $a\in H(\delta)$，则 e 表示为(δ, a);

逻辑程序 P 的强依赖图 SG_P 是由依赖图 G_P 构造的有向无环图，对于 G_P 中的每一个强连通组件，在 SG_P 中都将其折叠成为一个节点.若 G_P 中存在一条边 e，在 G_P 中是由强连通组件 sc_1 中的一个节点指向强连通组件 sc_2 中的节点; 在 SG_P 中这条边表示为由 sc_1 指向 sc_2 即可.一个有向图中入度为 0 的节点被称为源（source）.若 SG_P 中源中不包含原子，则称该源为空.

1.2 极小模型分解

CheckMin 算法是 Ben-Eliyahu-Zohary 等人[33-34]提出对给定子句理论的一个模型进行验证，检验该模型是否为该子句理论的极小模型.

对于任意一个子句理论 T,令 X 和 Y 为原子集合，且 $X\cap Y=\varnothing$,则 $Reduce(T, X, Y)$是将 X 在 T 中的原子全部赋值为真，Y 在 T 中的原子全部赋值为假，从而得到约简的子句理论.

对于给定 T 的一个源 S，T_S 表示在 T 中只包含 S 的子句集.

定理 1. 极小模型分解定理[33-34]. 给定一个子句理论 T 和 T 的一个模型 M，令图 G 表示 T 的强依赖图 SG，如果 G 中存在一个源 S 使得 $X=S\cap M$ 是 T_S 的极小模型，令 $T'\leftarrow Reduce(T, X, S-X)$，则 $M-X$ 是 T'的极小模型.

根据极小模型分解定理（定理 1）Ben-Eliyahu-Zohary 等人[33-34]提出 CheckMin 算法(算法 1）.然而 Ben-Eliyahu-Zohary 等人[33-34]提出的 CheckMin 算法并不是完备的，因此他们又提出了完备性的充分条件——模块化性质.

定义 2. 模块化性质.

1) 如果子句理论 T 的强依赖图 SG_T 中只有一个强连通组件，则 T 的一个极小模型 M 相对于 T 具有模块化性质;

2) 给定一个子句理论 T 和 T 的一个模型 M,如果 T 中存在源 S 使得 $X=S\cap M$ 是 T_S 的一个极小模型，并且 $M-X$ 是 $T'\leftarrow Reduce(T, X, S-X)$的极小模型，则 $M-X$ 相对于 T'具有模块化性质.

算法 1. CheckMin(T, M).

输入：一个子句理论 T 和 T 的一个模型 M;

输出：true 或 false.

```
G←SG_T;
迭代地删除 G 中所有为空的源;
While G 中存在源 S，且 S 是 T_S 的极小模型 do
    X←M∩S;
    M←M−X;
    T←Reduce(T, X, S−X);
    G←SG_T;
    迭代地删除 G 中所有为空的源;
end while
if M = ∅ then return true;
else return false.
```

如果 T 的极小模型 M 关于 T 具有模块化性质，

则 CheckMin(T, M)返回 true.

1.3 极小归约

对于极小模型分解定理的不完备，Wang 等人①对此提出了极小归约，对极小模型分解定理进行补充. 同时对算法 CheckMin 也进行了修改得到新的完备算法 CheckMinMR（算法 2）.

算法 2. CheckMinMR(T, M).

输入：子句理论 T 和 T 的一个模型 M;

输出：true 或 false.

return CheckMin($MR(T, M)$, M).

定义 3. Minimal Reduct. 给定一个逻辑程序 P 和原子集 S，$S\subseteq A(P)$则 P 关于 S 的极小归约表示为 $MR(P, S)$，$MR(P, S)$是正逻辑程序：

$$\{H(r)\cap S\leftarrow B^{+}(r) \mid r\in P, B^{+}(r)\subseteq S \,\&\, B^{-}(r)\cap S=\varnothing\}.\quad(2)$$

从极小归约的定义可以得到一个非常明显的结论 $A(MR(P, S))\subseteq S$.极小归约是根据闭区间假设来约简一个逻辑程序，即 S 中的原子被假定为真，而其它原子被假定为假.

引理 1[35]. 令 δ 表示一条子句，M 表示一个模型，有：

1) $M\models\delta$ 当且仅当 $M\models MR(\{\delta\}, M)$；

2) $\neg\overline{M}\cup\{\delta\}\equiv\neg\overline{M}\cup MR(\{\delta\}, M)$.

定理 2. 极小模型性质[35].令 T 是可满足的子句理论，且 $S\subseteq A(T)$，则以下 3 种情况是等价的：

1) S 是 T 的一个极小模型；

2) S 是 $MR(T, S)$的最小模型；

3) $S=\{p \mid MR(T, S)\models p\}$.

2 计算极小模型的新方法 MMSAT 和 MRSAT

本节将详细介绍本文提出的 2 个计算子句理论的极小模型算法，分别是基于 SAT 求解器的算法 MMSAT 和基于极小归约的算法 MRSAT.

2.1 基于 SAT 的极小模型算法 MMSAT

MMSAT 算法的主要思想是：当子句理论 T 是可满足时，SAT 求解器可计算出 T 的一个模型 M; 将该模型取反得到一子句 $\vee\neg M$，并且将不属于 M 的其余原子也分别取反得到子句集 $\neg\overline{M}$，将这些子句添加到 T 中得到一个新的子句集 T'，再用 SAT 求解器迭代该过程计算.若 M 是 T 的极小模型，则新的子句集 T'是不可满足的；反之，则说明 M 不是 T 的极小模型.根据这个思想我们设计出 MMSAT 算法（算法 3）.

算法 3. MMSAT(T).

输入：子句理论 T;

输出：T 的一个极小模型 M 或无模型.

① if T 是不可满足的

② return 无模型;

③ end if

④ while T 是可满足的 do

⑤ $M\leftarrow$MiniSAT(T);

⑥ $T\leftarrow T\cup\{\vee\neg M\}\cup\neg\overline{M}$;

⑦ end while

⑧ return M.

引理 2. 给定一个子句理论 T 及其一个模型 M，M 是 T 的极小模型当且仅当 $T\cup\{\vee\neg M\}\cup\neg\overline{M}$ 不可满足.

引理 2 显然成立.MMSAT 算法（算法 3）的⑤行计算 T 的一个模型，其 while 循环（④~⑦行）每次迭代生成的模型是其前一次生成的模型的真子集，因 M 是有穷的，故循环一定会在有限步内终止，引理 2 保证终止时计算出来的 M 是输入子句理论 T 的极小模型.

2.2 基于极小归约的极小模型算法 MRSAT

MRSAT 算法（算法 4）是 MMSAT 与基于极小归约的极小模型验证算法 CheckMinMR 的结合.首先由 MiniSAT 计算子句理论的模型，然后由 CheckMinMR 检验计算出的模型是否是其极小模型.

下面的引理 3 可保证算法 MRSAT 的可靠性.

引理 3. 给定一个子句理论 T，及其一个模型 M，若 $M'\subseteq M$，$M'\models T$，则 $MR(T', M')\equiv MR(T, M')$，其中 $T'=MR(T, M)$.

证明.

基始：当 $T=T'$时，$MR(T, M')\equiv MR(T', M')$成立.

步骤：令 $\delta\in T$，$\{\delta'\}=MR(\{\delta\}, M)$，由极小归约的定义可得 $MR(\{\delta\}, M')=\{H(\delta)\cap M'\leftarrow B^{+}(\delta) \mid B^{+}(\delta)\subseteq M'\}$. $MR(\{\delta'\}, M')=MR(MR(\{\delta\}, M), M')=\{H(\delta)\cap M\cap M' \leftarrow B^{+}(\delta) \mid B^{+}(\delta)\subseteq M'\subseteq M\}$. 已知 $M'\subseteq M$，综上可得 $MR(\{\delta\}, M')\equiv MR(\{\delta'\}, M')$.又因 $\delta\in T$，所以 $MR(T, M')\equiv MR(T', M')$，$T'=MR(T, M)$. 证毕.

算法 4. MRSAT(T).

输入：子句理论 T;

输出：T 的一个极小模型 M 或无模型.

① if T 是不可满足的

② return 无模型;

③ end if

④ while T 是可满足的 do

⑤ $M\leftarrow$MiniSAT(T);

⑥ if CheckMinMR($MR(T, M)$, M) then return M;

⑦ $T\leftarrow T\cup\{\vee\neg M\}\cup\neg\overline{M}$;

⑧ end while

⑨ return M.

① 个人通信，Wang Yisong, Eiter Thomas, Zhang Yuanlin, Lin Fangzhen, 2020.

3 实验结果与分析

我们在 MiniSAT 的基础上实现了上述 2 个算法，在随机 3CNF 公式和 SAT 国际竞赛上的部分基准测试用例上测试了.在实验中我们使用的 clingo 是目前的最新版本 clingo5.4①，它是基化器 gringo[36]和 ASP 求解器 clasp 的结合.本实验的工作环境是 Linux5.1.11、8 核 3.50GHz 的 CPU 和 32GB 内存.

3.1 随机 3CNF 公式

实验中的3CNF公式是通过设置原子数量和子句数量随机生成的子句长度为 3 的 CNF 公式，其中原子数量 n 的范围设置为 50 到 1000，增幅为 50；子句数量 m 的范围是 $3.0\times n$ 到 $5.0\times n$，增幅为 $0.1\times n$；对于其中的每种情况都分别有 10 个不同的 3CNF 文件，一个文件即一个 CNF 公式.我们使用 MMSAT，MRSAT，clingo 分别计算这些3CNF公式的极小模型，并统一设置计算时间上限为 1800s.文件的计算结果的输出类型有 SATISFIABLE（可满足）、UNSATISFIABLE（不可满足）和 UNKNOWN（计算超时被终止）.我们按照输出类型分别统计了所有计算的平均结果.

图 1~3 分别展示了 clingo，MMSAT，MRSAT 计算随机 3CNF 公式的极小模型的平均 CPU 时间.如图 1 所示，clingo 的峰值在原子数量为 900，子句数量为 3.8×900 时，CPU 时间为 1711s，MMSAT 和 MRSAT 在相同情况下的 CPU 时间分别为 499s 和 505s.此时 MMSAT 用时最少，而 clingo 所用时间几乎是 MMSAT 和 MRSAT 的 3 倍.如图 2 和图 3 所示，MMSAT 和 MRSAT 耗时最多的情况均是原子数量为 800，子句数量为 3.9×800，此时它们的 CPU 时间分别是 760s 和 758s，而 clingo 在此情况下的 CPU 时间为 842s.此时也是 clingo 用时较长，而 MRSAT 用时最少.

实验结果表明，计算随机 3CNF 公式的极小模型时，MMSAT 和 MRSAT 在计算速度上的优势十分显著，从整体来看，几乎所有的情况下 MMSAT 和 MRSAT 的用时均比 clingo 少很多.而 MMSAT 和 MRSAT 之间则无十分明显的差别.

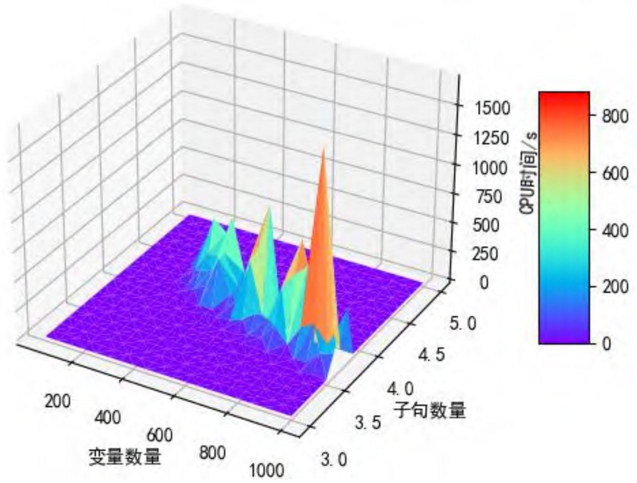

Fig.1 The average CPU time by clingo on random 3CNF formulas

图 1 clingo 计算随机 3CNF 公式极小模型的平均 CPU 时间

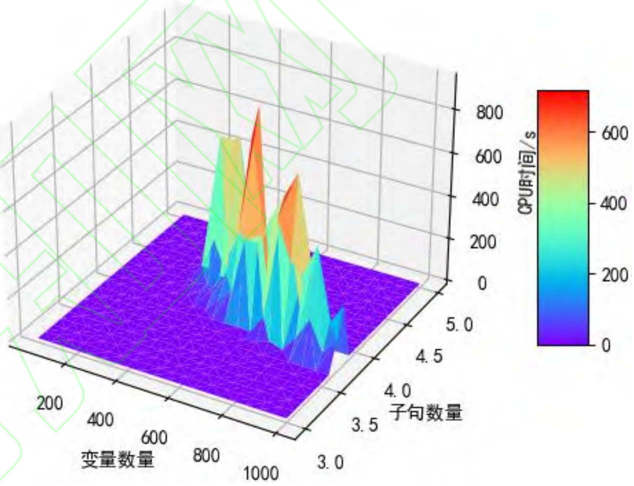

Fig. 2 The average CPU time by MMSAT on random 3CNF formulas

图 2 MMSAT 计算随机 3CNF 公式极小模型的平均 CPU 时间

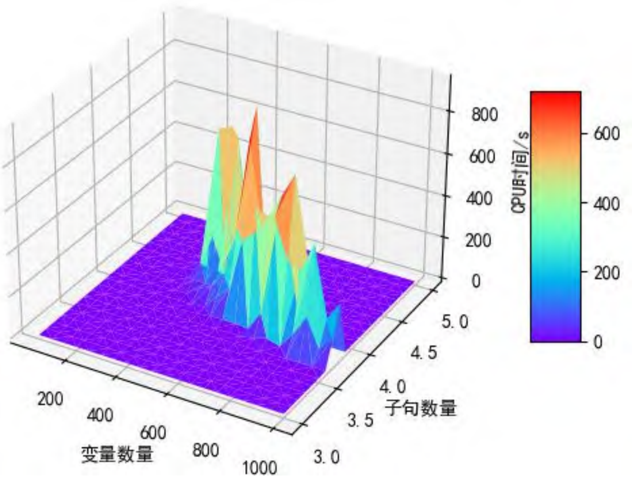

Fig. 3 The average CPU time by MRSAT on random 3CNF formulas

图 3 MRSAT 计算随机 3CNF 公式极小模型的平均 CPU 时间

3.2 SAT 竞赛基准测试实例

本实验所用的基准测试实例均获取自 SAT 国际

① https://potassco.org

竞赛，分别是 2009 年工业上的 SAT 实际测试用例①和 2020 年的测试实例②.我们从 2009 年的工业实例中随机挑选了 7 类实例，共计 109 个测试文件；从 2020 年的实例中随机选出 43 个实例，记为 SAT2020，本文实验所用实例共计 152 个，即 152 个 CNF 公式.分别使用 clingo5.4，MMSAT，MRSAT 计算这些工业实例对应逻辑程序的回答集.由于 clingo5.4 在工业实例中出现了计算出错情况，因此我们在评估比较中加入 gringo3.0.5+claspD1.1(其中 gringo 是 claspD 的前端). 由于实例比较复杂，计算实例模型所需时间较长，并且本文算法是在计算模型的基础上更进一步计算极小模型，此时可能需要迭代多次，因此本实验将所有实例的计算时间上限设置为 7200s.

如表 1 所示，我们对 4 种方法分别按照输出类型可满足（SAT）、不可满足（UNSAT）和超时（TO）统计了实例的计算结果③.在规定时间内 clingo 和 claspD 计算完成（即输出类型为可满足或不可满足）的实例的数量均明显少于 MMSAT 和 MRSAT，其中，clingo 在 bioinfo，c32sat，sat08 实例类中共有 11 个实例发生了内存不足（而被 killed）；在 SAT2020 中有 8 个实例发生了内存不足.gringo+claspD 在 c32sat 中也有一个实例发生了内存不足（而出现 std::bad_alloc)；且在 SAT2020 中有 8 个实例发生了内存不足.MMSAT 在 SAT2020 中有 1 个实例发生了内存不足（而出现 INDETERMINATE）.

值得注意的是，如表 1 所示，bioinfo 实例类中 clingo 在规定时间内完成计算的实例数量为 12，全部为不可满足.而 MMSAT 和 MRSAT 完成计算的实例数量均为 20 个，其中可满足的实例数量为 9 个，不可满足的为 11 个.通过对实验记录数据对比，我们发现在 clingo 计算的结果为不可满足的 12 个实例中，有 7 个实例在 claspD，MMSAT，MRSAT 中的结果都为可满足.表 1 中括号内的数字表示 clingo 计算出错④的实例个数.因此，我们将 MMSAT 和 MRSAT 计算的极小模型分别与文件中的子句进行了验证，验证结果均表明这 7 个实例确实是可满足的.同时，我们对所有测试实例的结果都进行了对比，在用例类 sat08 中和 SAT2020 也分别发现了一个实例为 clingo 计算出错的情况.

我们还统计了计算完成的实例的平均 CPU 时间，如图 4 所示，MMSAT 在 crypto/md5gen 实例类下计算极小模型的用时最多，除此之外，均是 claspD 完成计算的所需平均时间最长，其次便是 clingo 用时较长.整体来看 MMSAT 计算极小模型的平均时间最短，效果最好，MRSAT 次之.对计算结果为可满足的实例的平均 CPU 时间我们也做了统计，如图 5 所示，在 c32sat 用例类下是 MRSAT 用时最多，在此实例类下 clingo 和 claspD 均没有算出实例的极小模型；其余均是 claspD 耗时较长，而 MMSAT 用时最短.

综上，在 MiniSAT2.2 基础上实现的计算极小模型的算法 MMSAT 和 MRSAT 是非常有效的，其计算极小模型的速度都明显快于最新版的 clingo 和 claspD，并且 clingo 发生了计算错误，clingo 和 claspD 均出现了内存不足，而我们提出的算法则更加稳定.

Table 1 The Results on Industrial SAT Benchmarks in 7200s

表 1 计算 SAT 工业实例的结果对比

实例类	实例数	clingo			gringo+claspD			MMSAT			MRSAT		
		SAT	UNSAT	TO	SAT	UNSAT	TO	SAT	UNSAT	TO	SAT	UNSAT	TO
aprove09	18	16	0	2	18	0	0	**18**	0	0	**18**	0	0
bioinfo	20	0	12(7)	0	8	6	6	**9**	11	0	**9**	11	0
c32sat	9	0	2	6	0	2	6	**2**	3	4	**2**	3	4
sat08	27	8	11(1)	6	3	5	19	**9**	12	6	8	12	7
satcom09	6	5	0	1	3	0	3	**5**	0	1	**5**	0	1
crypto/desgen	19	7	0	12	5	0	14	**7**	0	12	**7**	0	12
crypto/md5gen	10	0	4	6	0	4	6	0	7	3	0	7	3
SAT2020	43	8	1(1)	26	3	0	32	**24**	0	18	**18**	0	25
总计	152	44	30	59	40	17	86	74	33	44	67	33	52

① http://satcompetition.org/

② https://satcompetition.github.io/2020/downloads.html

③ SAT，UNSAT，TO 之外的其他实例均为内存不足. MRSAT 在所有的实例类中均未出现内存不足.

④ 可满足的实例由 clingo 计算出的结果为不可满足.

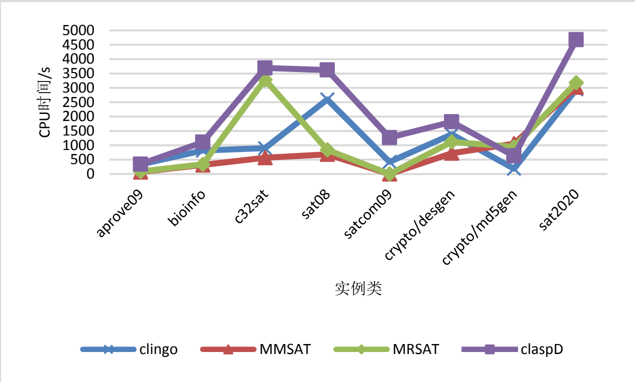

Fig. 4 The average CPU time on industrial SAT benchmarks correctly completed

图 4 正确完成计算 SAT 工业实例平均 CPU 时间

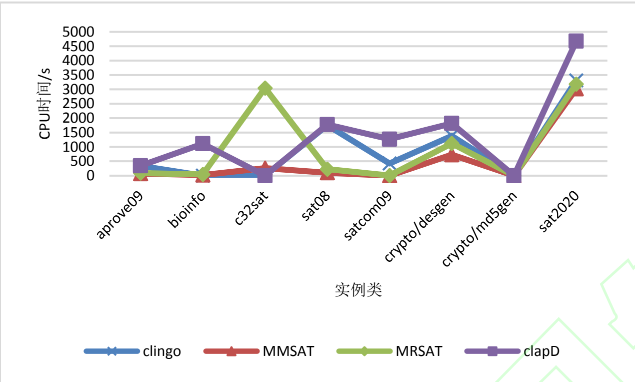

Fig.5 The average CPU time of computing a minimal model on industrial SAT benchmarks

图 5 计算出 SAT 工业实例极小模型的平均 CPU 时间

4 总结

本文提出了基于 SAT 和基于极小归约的计算命题极小模型的算法 MMSAT 和 MRSAT，首先证明了算法的可靠性，其次对算法进行了实现.最后，分别使用 MMSAT，MRSAT，clingo5.4 在随机 3CNF 公式和 SAT 国际竞赛上的工业实例上做了实验，在 SAT 国际竞赛上的工业实例上还使用 gringo3.0.5+claspD1.1 进行了测试.实验表明，使用 MMSAT 算法和 MRSAT 算法计算命题的极小模型，在时间上明显快于 claspD 和 clingo，在正确率上也明显优于 clingo.

在未来的工作中，根据分解极小模型的主要思想，在基于极小归约的算法 MRSAT 中可以做进一步的改进.通过采用并行计算的方式，使得对极小模型的验证更高效，从而提高极小模型计算的速度.此外，根据极小归约的性质，我们考虑将其直接应用在 MMSAT 算法中，将求解子句理论 T 的极小模型转化为求解极小归约之后子句理论的极小模型.我们将对这些问题做更进一步的研究.

参 考 文 献

[1] Lei Zhendong, Cai Shaowei. Solving (weighted) partial MaxSAT by dynamic local search for SAT[C]//Proc of the 27th Int Joint Conf on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann, 2018: 1346-1352

[2] Lei Zhendong, Cai Shaowei. Solving set cover and dominating set via maximum satisfiability[C]//Proc of the 34th AAAI Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2020: 1569-1576

[3] Feng Yu, Bastani O, Martins R, et al. Automated synthesis of semantic malware signatures using maximum satisfiability[C]//Proc of the 24th Annual Network and Distributed System Security Symp. Reston, VA: The Internet Society, 2017

[4] Asín Achá R, Nieuwenhuis R. Curriculum-based course timetabling with SAT and MaxSAT[J]. Annals of Operations Research, 2014, 218(1): 71－91

[5] Zhou Huisi, Ouyang Dantong, Liu Meng, et al. Partial maximum satisfiability problem method combined with structure characteristics for diagnostic problems[J]. Scientia Sinica Informationis, 2019, 49(6): 685-697 (in Chinese)

（周慧思，欧阳丹彤，刘梦，等. 一种结合结构特征求解诊断问题的 PMS 方法[J]. 中国科学：信息科学, 2019, 49(6): 685-697

[6] Xiao Guohui, Ma Yue. Inconsistency measurement based on variables in minimal unsatisfiable subsets[C]//Proc of the 20th European Conf on Artificial Intelligence. Ohmsha: IOS Press, 2012, 242: 864-869

[7] Torlak E, Vaziri M, Dolby J. MemSAT: Checking axiomatic specifications of memory models [J]. ACM SIGPLAN Notices, 2010, 45(6): 341-350

[8] Ouyang Dantong, Gao Han, Tian Naiyu, et al. MUS enumeration based on double-model[J]. Journal of Computer Research and Development, 2019, 56(12): 2623-2631 (in Chinese)

（欧阳丹彤，高菡，田乃予，等. 基于双模型的 MUS 求解方法[J]. 计算机研究与发展, 2019, 56(12): 2623-2631）

[9] Giunchiglia E, Maratea M. Sat-based planning with minimal-#actions plans and “soft” goals[G]//LNAI 4733: Proc of the Artificial Intelligence and Human-Oriented Computing. Berlin, Springer, 2007: 422-433

[10] Kirousis L M, Kolaitis P. The complexity of minimal satisfiability problems[G]//LNCS 2010: Proc of the 18th Annual Symp on Theoretical Aspects of Computer Science Dresden(STACS). Berlin, Springer, 2001: 407-418

[11] Leone N, Rullo P, Scarcello F. Disjunctive stable models: Unfounded sets, fixpoint semantics, and computation[C]//Proc of the Int Conf on Database and Expert Systems Applications, vol 1134. Berlin, Springer, 1996: 654-666

[12] Lifschitz V, Morgenstern L, Plaisted D. Knowledge representation and classical logic[J]. Foundations of Artificial Intelligence, 2008, 3: 3-88(只有

卷)

[13] Lifschitz V. Computing circumscription[C]//Proc of the 9th Int Joint Conf on Artificial Intelligence, vol 1. San Francisco, CA: Morgan Kaufmann, 1985, 121-127

[14] McCarthy J. Circumscription—A form of non-monotonic reasoning[J]. Artificial Intelligence, 1980, 13(1/2): 27-39

[15] Lin Zuoquan, Li Wei. Open logic and its relationship to restriction logic[J]. Scientia Sinica Technologica, 1998, 28(6): 550-558 (in Chinese)
（林作铨, 李未. 开放逻辑及其与限制逻辑的关系[J]. 中国科学：技术科学, 1998, 28(6): 550-558）

[16] McCarthy J. Applications of circumscription to formalizing commonsense knowledge[J]. Artificial Intelligence, 1986, 28(1): 89-116

[17] Reiter R. A logic for default reasoning[J]. Artificial Intelligence, 1980, 13 (l/2): 81-132

[18] Ouyang Dantong, Chen Xiaoyan, Ye Jing, et al. Test pattern set reduction based on the method of computing minimal hitting set[J]. Journal of Computer Research and Development, 2019, 56(11): 2448-2457 (in Chinese)
（欧阳丹彤, 陈晓艳, 叶靖, 等. 基于极小碰集求解算法的测试向量集约简[J]. 计算机研究与发展, 2019, 56(11): 2448-2457）

[19] Tian Naiyu, Ouyang Dantong, Liu Meng, et al. A method of minimality-checking of diagnosis based on subset consistency detection[J]. Journal of Computer Research and Development, 2019, 56(7): 1396-1407 (in Chinese)
（田乃予, 欧阳丹彤, 刘梦, 等. 基于子集一致性检测的诊断解极小性判定方法[J]. 计算机研究与发展, 2019, 56(7): 1396-1407）

[20] Wang Rongquan, Ouyang Dantong, Wang Yiyuan, et al. Solving minimal hitting sets method with SAT based on DOEC minimization [J]．Journal of Computer Rearch and Development, 2018, 55(6): 1273-1281 (in Chinese)
(王荣全, 欧阳丹彤, 王艺源, 等. 结合 DOEC 极小化策略的 SAT 求解极小碰集方法[J]. 计算机研究与发展, 2018, 55(6): 1273-1281）

[21] Liu Siguang, Ouyang Dantong, Zhang Liming, et al. A method of minimality-checking of candidate solution for minimal hitting set algorithm [J]．Journal of Software, 2018, 29(12): 3733-3746 (in Chinese)
（刘思光, 欧阳丹彤, 张立明, 等. 一种极小碰集求解中候选解极小性判定方法[J]. 软件学报, 2018, 29(12): 3733-3746）

[22] Ouyang Dantong, Zhou Jianhua, Liu Bowen, et al. A new algorithm combining with the characteristic of the problem for model-based diagnosis[J]. Journal of Computer Research and Development, 2017, 54(3): 502-513 (in Chinese)
（欧阳丹彤, 周建华, 刘伯文, 等. 基于模型诊断中结合问题特征的新方法[J]. 计算机研究与发展, 2017, 54(3): 502-513）

[23] Gelfond M, Lifschitz V. Classical negation in logic programs and disjunctive databases[J]. New Generation Computing, 1991, 9: 365-385（只有卷）

[24] Gelfond M, Lifschitz V. The stable model semantics for logic programming[C]//Proc of the 5th Int Logic Programming Conf and Symp, vol 50. Cambridge, MA: MIT Press, 1988: 1070-1080

[25] Cadoli M. On the complexity of model finding for nonmonotonic propositional logics[C]//Proc of the 4th Italian Conf on Theoretical Computer Science(ICTCS). Singapore: World Scientific, 1992, 125-139

[26] Cadoli M. The complexity of model checking for circumscriptive formulae [J]. Information Processing Letters, 1992, 44(3): 113-118

[27] Angiulli F, Ben-Eliyahu-Zohary R, Fassetti F, et al. On the tractability of minimal model computation for some CNF theories[J]. Artificial Intelligence, 2014, 210: 56-77

[28] Ben-Eliyahu-Zohary R, Palopoli L. Reasoning with minimal models: Efficient algorithms and applications[J]. Artificial Intelligence, 1997, 96(2): 421-449

[29] Ben-Eliyahu-Zohary R. A hierarchy of tractable subsets for computing stable models[J]. Artificial Intelligence, 1996, 5: 27-52

[30] Gebser M, Kaufmann B, Neumann A, et al. clasp: A conflict-driven answer set solver[G]//LNAI 4483: Proc of the Int Conf on Logic Programming and Nonmonotonic Reasoning (LPNMR). Berlin: Springer, 2007, 260-265

[31] Gebser M, Kaminski R, König A, et al. Advances in gringo series 3[G]///LNAI 6645: Proc of the Int Conf on Logic Programming and Nonmonotonic Reasoning (LPNMR). Berlin: Springer, 2011: 345-351

[32] Leone N, Pfeifer G, Faber W, et al. The DLV system for knowledge representation and reasoning[J]. ACM Transactions on Computational Logic, 2006, 7(3): 499-562

[33] Ben-Eliyahu-Zohary R, Angiulli F, Fassetti F, et al. Decomposing minimal models[C/OL]//Proc of the Workshop on Knowledge-Based Techniques for Problem Solving and Reasoning Co-Located with 25th Int Joint Conf on Artificial Intelligence (IJCAI). 2016[2019-09-15]. http://ceur-ws.org/Vol-1648/paper1.pdf

[34] Ben-Eliyahu-Zohary R, Angiulli F, Fassetti F, et al. Modular construction of minimal models[G]//LNAI 10377: Proc of the Int Conf on Logic Programming and Nonmonotonic Reasoning (LPNMR). Berlin: Springer, 2017: 43-48

[35] Eén N, Sörensson N. An extensible SAT-solver[G]//LNCS 2919: Proc of the Int Conf on Theory and Applications of Satisfiability Testing. Berlin: Springer, 2004: 502-518

[36] Gebser M, Kaminski R, Ostrowski M, et al. On the input language of ASP grounder gringo[G]//LNAI 5753: Proc of the Inter Conf on Logic Programming and Nonmonotonic Reasoning (LPNMR). Berlin: Springer, 2009: 502-508

Zhang Li, born in 1995. Master candidate. Her main research interest include knowledge representation and reasoning, artificial intelligence.

张丽，1995 年生. 硕士研究生. 主要研究方向为知识表示与推理、人工智能.

Wang Yisong, born in 1975. Professor at the College of Computer Science and Technology, Guizhou University. His research interest covers knowledge representation and reasoning, machine learning and artificial intelligence.

王以松，1975 年生. 贵州大学计算机科学与技术学院教授. 主要研究方向为知识表示与推理、机器学习和人工智能.

Xie Zhongtao, born in 1994. Master candidate. His main research interest include knowledge representation and reasoning, artificial intelligence. (franciscn@qq.com)

谢仲涛，1994 年生. 硕士研究生. 主要研究方向为知识表示与推理、人工智能.

Feng Renyan, born in 1991. PhD candidate. Rceived her BS degree from North China University of Technology in July 2015 and MS degree from Guizhou University in July 2018. Her main research interests include knowledge representation and reasoning, artificial intelligent. (827220951@qq.com)

冯仁艳，1991 年生.博士研究生.主要研究方向为知识表示与推理和人工智能.