

基于奖励机制的 SAT 求解器分支策略



沈 雪 陈树伟 艾森阳

西南交通大学数学学院 成都 610031

系统可信性自动验证国家地方联合工程实验室 成都 610031

(1035475108@qq.com)

摘 要 分支决策是 CDCL(Conflict Driven Clause Learning)求解器一个十分关键的环节,一个好的分支策略可以减少分支决策次数进而提高 SAT 求解器的效率。目前,先进的分支策略大都结合了冲突分析过程,但分支策略对参与冲突分析的变量奖励方法有所不同,因此所挑选出的决策变量会有所差异。文中考虑到决策变量总是在未赋值变量中选取的这一重要事实,在 EVSIDS(Exponential Variable State Independent Decaying Sum)分支策略的基础上提出了一种新的分支策略,称为基于奖励机制的分支策略(简称 RACT 分支策略)。RACT 分支策略对冲突分析中被撤销赋值的变量再次给予奖励,以增大未赋值变量中频繁参与冲突分析的变量被选择为分支变量的可能性。最后,将所提出的分支策略嵌入到 Glucose4.1 求解器中以形成新的求解器 Glucose4.1+RACT,以 2017 年 SAT 竞赛中的 350 个实例为实验数据集来测试 RACT 分支策略的有效性。实验结果表明,求解器 Glucose4.1+RACT 比原版求解器能求解出更多的实例个数,尤其在求解可满足实例的个数上增加了 13.5%,此外在求解 350 个竞赛实例上所花费的总时间较 Glucose4.1 减少了 3.9%,以上实验数据均说明所提分支策略可以有效减少搜索树的分支决策次数并给出正确的搜索空间,进而提高了 SAT 求解器的求解能力。

关键词: 可满足性问题;完备算法;冲突驱动子句学习;回溯搜索;分支策略

中图法分类号 TP181

Reward Mechanism Based Branching Strategy for SAT Solver

SHEN Xue, CHEN Shu-wei and AI Sen-yang

School of Mathematics, Southwest Jiaotong University, Chengdu 610031, China

National-Local Joint Engineering Laboratory of System Credibility Automatic Verification, Chengdu 610031, China

Abstract Branching decision is one of the most critical parts of CDCL solver, and an excellent branching strategy can reduce the number of branching decisions and improve the SAT solver's efficiency. Most of the state-of-art branching strategies have incorporated the conflict analysis process. But the branching strategies have different reward methods on variables participating in conflict analysis, so the selected branching variables are different. In this paper, considering the important fact that decision variables are always selected in the unassigned variable set, a new branching strategy based on the EVSIDS (Exponential Variable State Independent Decaying Sum) branching strategy is proposed, which is called the branching strategy based on the reward mechanism (referred to as the RACT branching strategy). The RACT branching strategy is to reward the variables that are de-assigned in the conflict analysis again to increase the probability that the variables that are frequently involved in the conflict analysis in the unassigned variables are selected as branch variables. Finally, the proposed branching strategy is embedded into the Glucose4.1 solver to form a new solver Glucose4.1+RACT, and the effectiveness of the RACT branching strategy is tested by using 350 instances of the 2017 SAT competition as experimental data sets. The experimental comparison shows that the solver Glucose4.1+RACT can solve more instances than the original solver, especially with an increase of 13.5% in the number of satisfied instances, and the total time spent on solving 350 competition examples is also 3.9% lower than that of Glucose 4.1. The above experimental data shows that the proposed branching strategy can effectively reduce the number of branch decisions in the search tree and give the correct search space, thus improving the solving ability of the SAT solver.

Keywords Satisfiability problem, Complete algorithms, Conflict driven clause learning, Backtrack search, Branching strategy

到稿日期: 2019-07-26 返修日期: 2019-09-22 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目: 国家自然科学基金(61673320); 中央高校基本科研业务费专项资金(2682018ZT10, 2682018CX59)

This work was supported by the National Natural Science Foundation of China (61673320) and Fundamental Research Funds for the Central Universities (2682018ZT10, 2682018CX59).

通信作者: 陈树伟(swchen@swjtu.edu.cn)

1 引言

可满足性问题(Satisfiability Problem, SAT 问题)是经典的 NP(Non-deterministic Polynomial)完全问题^[1],由于所有的 NP 问题都能在多项式时间内进行转化,因此 SAT 问题的快速求解具有重要的理论意义。此外, SAT 问题求解在人工智能、规划、定理证明、软件和硬件验证等许多领域都有重要的应用价值^[2-3]。现有 SAT 问题求解算法按照完备性可分为两类:完备算法和不完备算法。现今求解器主要在完备算法的基础上进行发展,因为对于可满足的问题,完备算法可以找到问题的可满足解;而对于不可满足的问题,算法能够给出不可满足的证明过程。虽然不完备算法在可满足性问题求解效率方面优于完备算法,但无法证明问题的不可满足性,而许多应用领域都需要证明一个实例的不可满足性。因此,完备算法的应用更加广泛,发展完备算法也显得十分必要。

目前,绝大多数完备算法是基于深度遍历二叉树的 DPLL(Davis Putnam Logemann Loveland) 算法^[4]。2009 年, Marques-silva 等^[5]在 DPLL 算法的基础上提出了冲突驱动子句学习(CDCL),使得求解规模在合理时间内从数百个变量扩大到数万个变量。后续提出的 SAT 完备算法都是基于 CDCL 算法,对不同的环节进行改进,CDCL 算法主要分为 5 个环节:单元传播、分支决策、子句学习、重启和学习子句删除。在求解 SAT 问题时,求解器将花费 70%~80% 的时间来进行单元传播,而好的分支决策可以减少冲突分析次数和传播次数,因此优秀的分支决策对提高求解器的性能至关重要。

早期的分支策略在选择决策变量时主要从子句所固有的一些结构信息出发,如子句的长度、变量个数、子句数目、文字在子句中出现的频数等^[6]。直到 GRASP 求解器的作者提出冲突分析和非时序回溯^[7],分支启发式算法在选择分支变量时便不再需要精确找到所有包含指定文字的子句,它很好地结合了冲突分析过程,使求解效率得到显著提升。例如,在求解器 Chaff 中提出的 VSIDS(Variable State Independent Decaying Sum)分支策略只需要考虑学习子句中出现的变量^[8]。之后提出的 VSIDS 的有效变体 NVSIDS(Normalized VSIDS)^[9], EVSIDS(Exponential VSIDS)^[10]和 ACIDS(Average Conflict-Index Decision Score)^[11]分支策略在 VSIDS 分支策略的基础上,额外考虑了与冲突分析过程有关的变量,但上述分支策略对每个参与冲突分析的变量的活性增长值是一样的,忽略了一个重要的事实,即决策变量总是在未赋值变量中选取。因此,本文提出的新策略是在 EVSIDS 分支策略的基础上,对回溯撤销赋值的变量再赋予一次奖励,使得未赋值变量比已赋值的变量更容易被选为分支变量。实验结果表明,所提分支策略可以有效提高求解器的求解能力。

本文第 2 节介绍相关术语和技术背景;第 3 节回顾目前已经被提出的分支启发式算法,分析当前分支策略存在的问题;第 4 节提出一种新的分支策略并分析其主要思想;第 5 节在所提算法的基础上实现了一个 SAT 求解器,并通过实验比较它和 Glucose4.1 的性能;最后总结全文。

2 预备知识

2.1 相关定义

本节将给出文中所用的概念和符号。

定义 1(真值指派)^[12] 给定一个命题变量集合 $V = \{x_1, x_2, \dots, x_n\}$, 其真值指派 A 是一个映射: $V \rightarrow \{0, 1\}$, 0 和 1 分别代表变量在真值指派 A 下取真值和取假值。

定义 2(子句)^[12] 子句 c 是 V 上有限个文字的析取, 记作 $c = l_1 \vee l_2 \vee \dots \vee l_n$ 。 V 上有限个子句组成的集合称作子句集 C , 记 $C = \{c_1, c_2, \dots, c_n\}$ 。子句集 C 在真值指派 A 下是可满足的, 当且仅当 C 中所有的子句 c 在真值指派 A 下都取真值。

定义 3(合取范式, CNF 公式)^[12-13] 合取范式 $F(V)$ 是 V 上有限个子句的合取, 记作 $F(V) = c_1 \wedge c_2 \wedge \dots \wedge c_n$ 。合取范式 $F(V)$ 在真值指派 A 下取真值 (或称 $F(V)$ 在真值指派 A 下是可满足的), 当且仅当 $F(V)$ 中包含的所有子句在真值指派 A 下取真值。

定义 4(SAT 问题)^[12] 给定一个命题变量集合 V 和一个 V 上的合取范式 $F(V)$, 判断是否存在一个真值指派 A , 使得 $F(V)$ 为真, 如果存在则称 $F(V)$ 是可满足的, 否则称 $F(V)$ 是不可满足的。

2.2 技术背景

CDCL 算法在 DPLL 算法原有的框架上引入了冲突分析和子句学习机制以及非时序回溯机制。CDCL 算法的伪代码见 OSID 码算法 1。 *Decide()* 函数用于分支决策, 在搜索的每个阶段, 使用该函数选择变量进行真值指派 (所选变量处于未赋值状态), 赋值后的变量会对应一个决策层 d 。 *Deduce()* 函数用于识别蕴含的变量赋值, 该函数对应于单元传播过程。每当一个子句在某一赋值下不可满足时, *Deduce()* 函数就会返回一个冲突, 再使用 *Diagnose()* 函数进行冲突分析得出学习子句, 并计算回溯层。若回溯层不为 0, 则进行非时序回溯, 使用 *Erase()* 函数撤销 d 层的变量以及其蕴含变量的所有赋值。本文主要研究 *Diagnose()* 函数中的冲突分析环节, 并且重点关注冲突分析后需要撤销赋值的变量。

3 已有的分支策略

3.1 VSIDS 分支策略

VSIDS 分支策略首次应用于 Chaff 求解器, 是目前使用得最多、影响最大的分支决策算法, 该策略更多地体现了求解过程中的冲突相关性。该策略的具体描述如下:

- 1) 每一个文字都附有一个计数器 s , 且初始值设置为 0;
- 2) 当学习子句加入到公式中时, 子句中有关文字的计数器增加 f , 即该文字的活性值更新为 $s' = s + f$;
- 3) 每次决策时, 选择未赋值的变量中具有最大活性值的变量为分支决策文字;
- 4) 当两个变量所对应的计数器的值相同时, 随机选择一个变量;
- 5) 所有变量的计数器会周期性地除以一个大于 1 的常数, 以避免求解器陷入局部最优。

该算法的主要思想是为每个文字设置一个得分 s , 每次选择得分最高的文字对应的变量作为决策变量。在发生冲突时, 通过对 s 增加一个增量 f 来改变与冲突分析有关的得分。相应地, 这个增量与目前发生的冲突次数 n 有关, 越往后发生的冲突对应的 n 越大, s 的增量也越多, 这样就能优先满足最新的学习子句。

3.2 EVSIDS 分支策略

EVSIDS^[10] 分支策略是对 VSIDS 分支策略的进一步发

展,作为 VSIDS 的有效变体,EVSIDS 策略是现代 CDCL 求解器分支的重要组成部分,被广泛用于现代求解器中,如 Minisat^[8], Lingeling^[14], Glucose^[15] 等。EVSIDS 策略与 VSIDS 策略的不同之处主要有 3 点:1)EVSIDS 分支策略为每个变量仅设置一个活跃度计数器,减少了计数器的个数;2)EVSIDS 分支策略更新的变量不仅是学习子句中的变量,而是在每次发生冲突时将与冲突有关的变量都更新;3)更新的变量活性值的增量为 g^i ,其中 $g = \frac{1}{f}$, f 为 VSIDS 策略中的增量因子, $0 < f < 1$, i 为冲突次数。值得注意的是, g^i 是一个与冲突有关的变量,并且随着冲突次数的增加而逐渐增加,这样就能优先考虑最新的学习子句。Minisat 中, $f = 0.95$ 。

上述两种分支启发式算法会更新两种变量,即学习子句中的变量和与冲突有关的变量,但忽略了一个事实,即分支策略会选择最新产生的、未满足的、得分最高的自由变量作为下一个决策变量^[6]。也就是说,通过冲突分析确定回溯层后,回溯之后某些变量会被撤销赋值,这样的变量比已赋值的变量更加重要,因为决策变量总是在未赋值的变量集中选取,它相比已赋值的变量更有可能被选择作为决策变量。因此,在回溯之后,我们可以设计一种新的分支启发式算法来再次给予这些撤销赋值的变量一个奖励,使得它更有可能被选择为决策变量。

4 RACT 分支策略

本节提出了一个新的分支策略来有效地更新被撤销赋值变量的活性值,称其为基于奖励机制(RACT)的分支启发式算法。

在求解 SAT 问题中,随着求解的深入,冲突次数不断增大。早期的分支启发式算法,仅依赖于子句所固有的一些结构信息,如子句的长度、变量个数、子句数目以及文字在子句中出现的频数等,没有很好地与冲突分析相结合。后期的分支启发式策略大都是基于 VSIDS 分支策略的改进,如 3.1 节中 VSIDS 分支策略的更新方式,该分支策略首次有效地结合了冲突过程,直接使用与冲突次数相关的增量 f 来更新变量活性值。这种结合冲突过程的分支策略不仅使变量活性值更新过程由静态变为了动态,而且根据不同变量参与冲突分析次数的差异性将变量的“重要性”区分开来。因此,在现代的分支策略中,结合冲突过程的分支策略是具有深远意义的。

在现代的分支策略中,结合冲突过程的分支策略,通过变量参与冲突分析的次数来更新变量活性值^[16-20]。一方面,变量参与冲突的次数越多,代表其越活跃,其对应的活性值就越大,越容易被挑选为决策变量;另一方面,冲突次数是一个递增值,代表越往后发生的冲突,其活性值增加得越大,也就使搜索过程越重视最近发生的冲突。

本文在以上的基础上,充分考虑变量参与冲突次数对变量“重要性”的影响,提出了基于现代 VSIDS 分支策略的 RACT 分支启发式算法,该算法不改变原有策略的活性增加方式,而是在其基础上,主要针对回溯期间撤销赋值的变量进行更深入的研究。RACT 分支启发式算法在原有策略的基础上,针对执行非时序性回溯期间需要撤销赋值的变量,引入了变量上次参与冲突分析时的总冲突次数,文中将变量上次参与冲突分析时的总冲突次数与当前的冲突次数的比值作为奖

励值,该值有效地量化变量最近参与冲突的频率,比值越大,代表该变量最近越活跃,对应的活性值就越大。具体增量方式如下:

奖励值用 $R(v)$ 表示,被撤销赋值变量的增量方式如下:
$$s'' = s' + R(v) \cdot g^i \tag{1}$$

其中, f 为增量因子, $R(v)$ 与变量参与冲突的次数有关,定义如下:

$$R(v) = \frac{lastconflict(v)}{NBconflict(v)} \tag{2}$$

其中, $lastconflict(v)$ 表示变量最近一次参与冲突时的冲突次数, $NBconflict(v)$ 表示当前冲突总次数。

由 $R(v)$ 的定义可知,当某个变量最近一次参与冲突的次数与当前冲突总次数较为接近时, $R(v)$ 趋近于 1,这说明频繁参与冲突的变量具有更高的活性值,更有可能被挑选为下一次决策变量。其次, $R(v) \in [0, 1]$, 则 $R(v) \cdot g^i \leq g^i$, 说明新提出的基于奖励机制的分支策略是仅对当前参与冲突未赋值的变量一个适当的奖励,对于在此之后参与冲突的变量的活性不会有太大的影响,这不会改变 EVSIDS 分支策略的总体趋势,因此所改进的分支策略是通过保留 EVSIDS 分支策略筛选得到的变量,然后对这些变量再作进一步的筛选。具体实现过程见 OSID 码中算法 2。

基于奖励机制的分支启发式算法将分支过程分为了两个环节,procedureOne 环节为原版求解器的分支环节,当变量参与冲突分析增加变量的活性值时,活性值的奖励方式保持不变。procedureTwo 为本文的改进环节,即为 RACT 分支策略的具体实现过程。当搜索过程发生回溯后,我们再一次增加撤销赋值的变量的活性,使其更有可能成为决策变量,减少求解器搜索的次数。综上,本文的 RACT 分支启发式算法在考虑当前冲突分析中变量参与冲突的情况的基础上,加入了撤销赋值变量在最近一次用于冲突分析时对应的冲突次数在全局冲突中的作用,从总体上考虑了冲突过程对搜索过程的影响。基于此,下一节将通过实验来进一步说明此策略的可行性。

5 实验评估

5.1 实验环境

实验的测试环境为 Win x64 操作系统、Intel i3-3240 3.40 GHz CPU、8GB 内存。对比测试例来自于 2017 年 SAT 竞赛 Main Track 组的 350 个测试例 (SAT Competition 2017)。根据测试实例的可满足性,这些测试实例被分为可满足实例集和不可满足实例集,记为 SAT 和 UNSAT。实验采用的限定时间为 3600 s,若超出,则该测试实例结果记为“—”。

5.2 实验设置

实验选择与目前最先进的 Glucose 版本的求解器 Glucose4.1 进行比较。Glucose 是国际上知名的 CDCL SAT 求解器,近年来基于 Glucose 求解器改进的版本在国际竞赛上发挥着重要作用,2016 年 SAT 竞赛在 Main Track 组外还专门设置了最佳的 Glucose 改进组奖项。我们将 RACT 分支策略嵌入目前最新的 Glucose 版本 Glucose4.1,生成新的求解器版本 Glucose4.1+RACT,两者仅分支策略不同,其余部分保持不变。通过实验对比新生成的求解器 Glucose4.1+RACT 和 Glucose4.1 的求解个数以及求解所需的时间的变

化,这对检验 RACT 分支策略的有效性有十分重要的作用。

5.3 实验结果及对比

首先,表 1 列出了 Glucose4.1+RACT 和 Glucose4.1 关于 2017 年 SAT 竞赛 Main Track 组的 350 个测试例的总体求解结果,主要从求解个数和平均求解时间这两个方面进行了比较,其中加粗字体为本文提出的求解器效果优于原版求解器。

表 1 Glucose4.1 嵌入 RACT 分支策略前后求解实例对比
Table 1 Numerical comparison of Glucose4.1 and Glucose4.1+RACT for solving SAT and UNSAT instances

SAT Competition 2017	Glucose4.1		Glucose4.1+RACT	
	求解个数	平均用时/s	求解个数	平均用时/s
SAT	74	921.57	84	766.88
UNSAT	88	873.46	83	960.83
Total	162	816.03	167	784.26

从表 1 可以看出,对于求解 SAT Competition 2017 的实例来说,新提出的求解器在求解实例个数方面比 Glucose4.1 更有效,其主要表现在求解可满足的实例上。对于可满足的实例,Glucose4.1+RACT 的求解个数较 Glucose4.1 增加了 13.5%。虽然对于不可满足实例来说,Glucose4.1+RACT 求解出了较少的实例,但是就总的求解个数而言,较求解器 Glucose4.1 仍增加了 5 个实例数。对于求解 SAT Competition 2017 的实例求解效率而言,可满足性实例求解时间明显领先于原版求解器,但不可满足性问题能力稍弱,总体求解 350 个实例花费的时间较原版 Glucose4.1 减少了 3.9%。这说明所提出的分支策略可以提高求解器的求解能力。

图 1 给出了 Glucose4.1+RACT 和 Glucose4.1 关于 2017 年的 350 个竞赛实例的求解运行时间对比, x 轴表示求解的实例个数, y 轴表示每个实例求解的时间,图中每一个点都代表一个 SAT 竞赛实例。如果图中的曲线更靠近 x 轴,表明其代表的求解器在求解实例时所花费的时间更少。图 1 中的曲线表明在求解前 100 个简单实例时,两个求解器的求解效率相当;在求解第 100~150 个较为复杂的实例时,Glu-

cose4.1+RACT 的求解表现稍弱;在求解第 150 以后的复杂实例时,Glucose4.1+RACT 的求解效率更高,这同样说明 Glucose4.1+RACT 在求解较为复杂的实例时更具有优势。

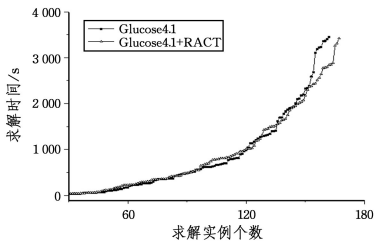


图 1 Glucose4.1 嵌入 RACT 策略前后的求解效率对比
Fig. 1 Comparison diagram of Glucose4.1 with and without RACT

为了更深入地探究本文策略对求解过程的影响,表 2 列出了部分 Glucose4.1+RACT 可求解出,而 Glucose4.1 未求解出的实例。从表 2 可以看出,求解这 6 个实例时,Glucose4.1 用时 3600s 未求解出,而 Glucose4.1+RACT 却以较快的时间求解出了,平均耗时仅为 688.65s,这充分说明新提出的策略在求解复杂的可满足实例时更有优势,也就对应了图 1 的求解时间趋势。

表 2 Glucose4.1 嵌入 RACT 分支策略前后求解部分可满足实例的用时对比

(单位:s)		
可满足实例	Glucose4.1	Glucose4.1+RACT
g2-hwmccl5deep-oski04s	—	853.22
mpl-22.8	—	416.73
mpl-22.9	—	158.89
mpl-9_19	—	249.46
mpl-blockpuzzle_9x9_s8_free7	—	758.39
mpl-blockpuzzle_9x9_s7	—	1 668.2

表 3 列出了两个求解器在解决可满足性问题(SAT)和不可满足性问题(UNSAT)时的不同,也可以进一步说明产生图 1 的原因。

表 3 Glucose4.1 和 Glucose4.1+RACT 的不同求解性能指标对比
Table 3 Comparison of different performance indexes:Glucose4.1 vs Glucose4.1+RACT

SAT Competition 2017	实例名	Glucose4.1			Glucose4.1+RACT		
		冲突次数	决策数	用时/s	冲突次数	决策数	用时/s
SAT	g2-ak128boothbg2asisc	8499745	37558919	3187.16	3235085	16522683	1503.93
	g2-Sz512_15128_1.smt2-cvc4	—	—	—	10674106	17140617	2426.95
	g2-gss-22-s100	—	—	—	1530227	1607723	2019.18
	mpl-klieber2017s-0300-035-t12	4783696	5189335	1898.5	5465500	5923961	2302.11
	mpl-22.7	23230606	78887863	3363.91	14953	8993856	916.65
	mpl-blockpuzzle_9x9_s7_free8	7226848	24815415	3626.95	27338	4456600	1668.20
UNSAT	g2-hwmccl5deep-6s341r-k19	3817753	13568573	3448.78	14889	2268092	2434.22
	g2-T135.1.1.cnf	150042	54768943	630.48	190790	61941082	1029.61
	g2-hwmccl5deep-intel065-k11	22740668	43318399	1245.01	21376917	41484841	1209.13
	mpl-blockpuzzle_5x10_s3-free4	1706299	3732180	471.31	1976020	4214056	526.78
	mpl-klieber2017s-1000-024-eq	14166463	15211488	1404.43	10232576	11366594	1008.11
	mpl-rubikcube401	3458430	3845813	365.744	3909109	4351796	504.086

好的分支策略可以有效地减少决策次数,针对列举的 12 个有针对性的竞赛实例,我们对比了其冲突次数和决策次数。从本文的求解过程来看,对于耗时较多且复杂的 6 个可满足性问题(SAT),新的分支策略相对于原版策略,有 5 个实例的

决策次数明显减少,求解时间提高了 50.8%,在一定程度上说明了新的分支策略在求解可满足性问题时,对变量活性赋值的正确性有明显提高,有效地减少了选取错误的决策变量的概率,相应地,发生冲突的次数也随之减少。对于较为复杂

的 6 个不可满足性问题,其求解效果稍微弱于原版策略,这也解释了图 1 展示的第 100~150 的实例中,某些实例的耗时高于原版。

综上,本文提出的基于奖励机制的分支策略在一些问题上能有效减小搜索空间,给出正确的决策路径,整体可以提高求解器的求解性能。

结束语 本文主要基于决策变量总是在未赋值变量中选取这一重要事实,有效地结合撤销赋值的变量参与冲突分析的具体情况,提出了一种新的分支启发式策略。将该策略嵌入 Glucose4.1 中,生成新的求解器 Glucose4.1+RACT。通过实验对比分析发现,对于撤销赋值的变量,依据其参与冲突分析的频繁程度再次给予其一定的奖励值的方法是有效的,实验结果显示新求解器的求解能力优于原版求解器。

后续计划将此策略应用于更多的实例,针对不同问题类型探究其冲突次数与变量活性的关系。进一步地,我们也会将该策略与相似的分支策略进行对比,争取在后续工作中进一步提高求解器的性能。

参 考 文 献

[1] COOK S. The Complexity of Theorem Proving Procedures[C]// Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing. New York:1971:151-158.

[2] MARQUES-SILVA J. Practical Applications of Boolean Satisfiability[C]// 9th International Workshop on Discrete Event Systems (WODES 2008). IEEE,2008:74-80.

[3] XU L, YU J. Improved Bounded Model Checking on Verification of Valid ACTL Properties [J]. Computer Science,2013,40(6): 99-102.

[4] DAVIS M, LOGEMANN G, LOVELAND D. A Machine Program for Theorem Proving [J]. Communications of the ACM, 1962,5(5):394-397.

[5] MARQUES-SILVA J, LYNCE I, MALIK S. Conflict Driven Clause Learning SAT Solvers[M]// Handbook of Satisfiability. Amsterdam:IOS Press,2009:127-149.

[6] MARQUES-SILVA J. The Impact of Branching Heuristics on Propositional Satisfiability Algorithms[C]// Portuguese Conference on Artificial Intelligence. Heidelberg:Springer,1999:62-74.

[7] MARQUES-SILVA J, SAKALLAH K. GRASP: A New Search Algorithm for Satisfiability[M]// The Best of ICCAD. New York:Springer,2003:73-89.

[8] MALIK S, ZHAO Y, MADIGAN C. Chaff: Engineering an Efficient SAT Solver[C]// Proceedings of the 38th Annual Design Automation Conference. 2001:530-535.

[9] BIERE A. Adaptive Restart Strategies for Conflict Driven SAT Solvers[C]// Theory and Applications of Satisfiability Testing-SAT 2008. Heidelberg:Springer,2008:28-33.

[10] EEN N, SORENSSON N. An Extensible SAT-solver [C]// Theory and Applications of Satisfiability Testing-SAT 2003. Berlin: Springer,2003:502-518.

[11] BIERE A, FROHLICH A. Evaluating CDCL Variable Scoring Schemes[C]// Theory and Applications of Satisfiability Testing-SAT 2015. Switzerland:Springer,2015:405-422.

[12] 王国俊. 数理逻辑引论与归结原理[M]. 北京:科学出版社, 2003.

[13] 张建. 逻辑公式的可满足性判断——方法、工具及应用[M]. 北京:科学出版社,2000:3-28.

[14] BIERE A. Lingeling,Plingrling and Treengeling Entering the SAT Competition 2013[C]// Proceedings of SAT Competition. 2013:51-52.

[15] AUDEMARD G, SIMON L. On the Glucose SAT Solver [J]. International Journal on Artificial Intelligence Tools,2018, 27(1):18400.

[16] LIANG J, GANESH V, POUPAR P, et al. Exponential Recency Weighted Average Branching Heuristic for SAT Solvers[C]// Proceedings of the 13th AAAI Conference on Artificial Intelligence. AAAI Press,2016:3434-3440.

[17] LIANG J, GANESH V, POUPAR P, et al. Learning Rate Based Branching Heuristic for SAT Solvers[C]// Theory and Applications of Satisfiability Testing-SAT 2016. Switzerland:Springer, 2016:123-140.

[18] LIANG J, GANESH V, POUPAR P, et al. An Empirical Study of Branching Heuristics through the Lens of Global Learning Rate[C]// Proceedings of the 20th International Conference on Theory and Applications of Satisfiability Testing. Melbourne, 2017:119-135.

[19] CHEN Q S, XU Y X, HE X. Heuristic Complete Algorithm for SAT Problem by Using Logical Deduction [J]. Journal of Southwest Jiaotong University,2017,52(6):1224-1232.

[20] XIAO F, LI C, LUO M, et al. A Branching Heuristic for SAT Solvers Based on Complete Implication Graphs [J]. Science China Information Sciences,2019,62(7):72103.



SHEN Xue, born in 1994, postgraduate. Her main research interests include intelligent information processing and automated reasoning.



CHEN Shu-wei, born in 1977, Ph.D, associate professor. His main research interests include logic based automated reasoning and decision analysis.