

文章编号:1006-2475(2019)07-0020-05

基于变量决策层的启发式变量选择策略

刘 姚,宋振明

(西南交通大学数学学院,四川 成都 611756)

摘要:启发式分支策略是 SAT 求解器中不可或缺的一部分,直接影响求解器的效率。早期的启发式分支决策需要遍历整个子句数据库,效率比较低。随着独立变量状态衰减和 (Variable State Independent Decaying Sum, VSIDS) 分支策略的出现, SAT 求解器的效率有所提高,但 VSIDS 策略以及它的延伸策略中变量的增量都只是与变量的冲突次数有关,没有考虑变量的决策层在分支策略中的影响。因此当发生冲突时,如果与冲突有关的变量的得分相同而决策层不同时,对于变量的选择就具有随机性。基于此,本文在阐述变量的决策层的重要性之后在 VSIDS 策略的基础上,提出一种基于变量决策层的启发式变量选择策略——HSVDL 策略。然后通过实例显示 HSVDL 策略在变量决策阶段选择决策层低的变量的可能性比选择决策层高的变量的可能性要大,而且得分比较小,减少了内存的占用。最后通过实验表明 HSVDL 策略能够求解出更多的实例,求解器的效率也有所提高,说明该策略有一定的优势。

关键词:启发式分支策略; SAT 求解器; VSIDS 策略; HSVDL 策略; 变量决策

中图分类号:TP311.1

文献标识码:A

doi: 10.3969/j.issn.1006-2475.2019.07.004

Heuristic Variable Selection Strategy Based on Variable Decision Level

LIU Yao, SONG Zhen-ming

(School of Mathematics, Southwest Jiaotong University, Chengdu 611756, China)

Abstract: The heuristic branch strategy is an integral part of the SAT solver and directly affects the efficiency of the solver. Early heuristic branch decisions require traversal of the entire clause database, which is inefficient. With the emergence of variable state independent decaying sum (VSIDS) branch strategy, the efficiency of the SAT solver is improved, but the increments of the variables in the VSIDS strategy and its extended strategy are only related to the number of conflicts with the variable, and do not consider the influence of the decision layer of the variable in the branching strategy. Therefore, when a conflict occurs, if the scores of the variables related to the conflict are the identical but the decision levels are different, the choice of variables is random. Based on this, after expounding the importance of the decision level of variables, based on the VSIDS strategy, a heuristic variable selection strategy named HSVDL strategy based on variable decision level is proposed. Then the example shows that the HSVDL strategy is more likely to choose the variable with lower decision-making level than the variable with higher decision-making level in the variable decision-making stage, and the score is smaller, which reduces the memory occupation. Finally, experiments show that the HSVDL strategy can solve more examples, and the efficiency of the solver is also improved, indicating that the strategy has certain advantages.

Key words: heuristic branch strategy; SAT solver; VSIDS strategy; HSVDL strategy; variable decision

0 引 言

命题逻辑公式的可满足性问题 (Boolean Satisfiability Problem, SAT) 是研究最广泛的 NP (NP-complete, NPC) 问题之一。在一个 SAT 问题中,其目的是找到满足合取范式 (Conjunctive Normal Form,

CNF) 公式的布尔变量的赋值,如果这样的一组赋值不存在,那么它是不可满足的。SAT 问题广泛应用于工程技术、计算机软件与理论^[1]、软件验证^[2-3]、逻辑电路等价性验证以及数学定理证明^[4] 等多个领域。一直以来, SAT 问题深受国内外学者的广泛关注,并为此做了很多研究。因而设计并实现求解 SAT 问题

收稿日期:2019-01-04; 修回日期:2019-03-04

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

作者简介:刘姚(1992-),女,河南洛阳人,硕士研究生,研究方向:智能信息处理,E-mail: yaoliu1213@163.com; 宋振明(1960-),男,四川成都人,教授,博士,研究方向:智能信息处理,E-mail: zhmsong@home.swjtu.edu.cn。

的高效算法具备重要的研究意义和应用前景。

解决 SAT 问题的一个重要算法是 Davis 和 Putnam 提出的方法,被称为 Davis-Putnam 算法,后来 Logemann 和 Loveland 对该算法做了进一步的改进,形成的算法被称为 DPLL (Davis Putnam Logemann Loveland) 算法^[5-6]。DPLL 的算法框架是当今许多高效的 SAT 求解器的构成基础。对所给的 CNF 公式不断地按照重言式规则、单子句规则、纯文字规则以及分裂规则进行变换,直到不能进行为止,这是 DPLL 算法的核心思想。

DPLL 求解器是求解 SAT 问题的“鼻祖”,但其算法自身的回溯机制对大规模问题的求解有一定程度的约束。为了让 DPLL 算法的求解效率有所提升,学者们优化了 DPLL 算法框架中的很多模块,形成了目前最为流行的 CDCL (conflict-driven-clause-learning)^[7] 求解框架。由于 CDCL 算法是由 DPLL 算法发展而来的,因此很多学者称其为 DPLL-Based。

对于求解器而言,有效的分支策略可以减少搜索过程中冲突的发生次数,从而提高整个求解器的求解效率。具有代表性的分支策略有: Jeroslow-Wang (JW) 策略^[8]; BOHM 策略^[9]; Maximum Occurrence in Minimization (MOM) 策略^[10]; Dynamic Largest Individual Sum (DLIS) 策略和 Dynamic Largest Combined Sum (DLCS) 策略^[11]; Variable Move-to-Front (VMTF) 策略^[12]; Conflict History-based Branching Heuristic (CHB)^[13] 策略及 Learning Rate Branching (LRB) 策略^[14] 等。这些策略的核心思想是为每个变量设置一个分数,并在每次作出分支决策时选择对应于新分支方向的最大(或最小)分数的变量。尽管已经存在很多分支策略,但由于实际问题是多种多样的,目前不存在一种分支策略适应于求解所有的问题。VSIDS 策略及其延伸策略(本文将在第 1 章对其详细介绍)在 Chaff^[15] 求解器中已经实现,并取得了一定的进展。因此,本文在 VSIDS 策略及其延伸策略的基础上充分考虑变量的决策层和冲突次数对分支策略的影响,提出一种选择策略——基于变量决策层的启发式变量选择策略。

1 预备知识

1.1 CDCL 算法

算法 1 CDCL 算法。

Input: CNF formula S , assigned variables v .

Output: the property of S .

1. Begin
2. $dl \leftarrow 0$ // dl = decision level

3. If ($BCP(S, v) == \text{CONFLICT}$) // 布尔约束传播
4. then return UNSAT
5. end
6. While (not All Variable Assigned(S, v))
7. do $p \leftarrow \text{Pick Branching Lit}(S, v)$ // 选择新的分支
8. $dl \leftarrow dl + 1$
9. $v \leftarrow v \cup \{p\}$
10. If ($BCP(S, v) == \text{CONFLICT}$) then
11. $\beta = \text{Conflict Analysis}(S, v)$
12. If ($\beta < 0$) then // 冲突发生在顶层
13. return UNSAT
14. else
15. Back Track(S, v, β) // 非同步回溯
16. $dl \leftarrow \beta$
17. end
18. end
19. end
20. Return SAT
21. End

算法 1 为经典的 CDCL 求解框架,在 CDCL 算法^[16-17]中有一个重要的函数 Pick Branching Lit,它的作用是从给定 SAT 问题的变量集中选择一个自由变量并赋值 0 或 1, Pick Branching Lit 函数可以从未赋值的变量中随机选择一个变量随机赋值,由于随机性使得目标不明确,因此它的效率不高。不同的启发式分支决策策略对应于不同的搜索空间,良好的启发式分支决策策略可以尽早去除无用的搜索空间,缩短求解时间,提高求解效率。而相对不太好的启发式分支决策策略则计算开销较大,求解速度相对缓慢。

1.2 VSIDS 策略

早期的启发式分支决策需要遍历整个子句数据库,该数据库因为添加学习子句而快速增大,计算代价比较大,所需要的时间较长。2001 年, Moskewicz 等人提出一种启发式分支决策策略——独立变量状态衰减和 (Variable State Independent Decaying Sum, VSIDS)^[18],旨在提高分支决策的效率,VSIDS 策略描述如下^[19]:

1) 每个变量的每个文字都有一个记分器,记录该文字在子句集中出现的次数,初始为 0。

2) 向数据库添加子句时,子句中相应文字的记分器将增加 1。

3) 在每个决策中,将未分配赋值和记分器最高的文字进行分配赋值。

4) 如果文字的最大得分相同,则随机选择其中的一个文字分配赋值。

5) 所有文字的记分器定期除以一个常数。

下一次选择决策变量时,选择决策层低的变量的可能性增大。

将 HSVDL 策略嵌入 CDCL 算法结构形成新的算法,HSVDL 策略如算法 2 所示。

算法 2 HSVDL 策略。

```

1. if BCP() = conflict //预处理
2. then return UNSAT
3. end if
4. dl = 0 //dl = decision level
5. conflict number = 0
6. while not All Variables Assigned() do
7. Pick Decision Variable() //分支决策
8. dl ← +1 //变量的决策层
9. if BCP() = conflict then
10. blevel = Conflict Analysis() //冲突分析
11. conflict number + = 1 //冲突次数
12.  $d_v$  = the decision level of  $v$  appeared in the Conflict Analysis()
13.  $s \leftarrow s/2 + n/(sd_v)$ 
14. end if
15. if blevel < 0 then
16. return UNSAT
17. else
18. Backtrack(blevel) //回溯
19. end if
20. end while
21. return SAT

```

求解器的预处理器通过重复使用单元子句规则来简化 CNF 公式。在变量选择阶段,函数 Pick Decision Variable() 根据 HSVDL 策略,选择具有最高分数的变量来分配赋值,并记录其决策层,第 1 个分支变量的决策层是 1。然后,布尔约束传播 (BCP) 从分支变量的赋值中推导出一些变量的赋值。由 BCP 过程推导出的变量的决策层与分支变量的决策层相同。当发生冲突时,函数 Conflict Analysis() 将分析冲突并生成一个学习子句,随后撤消与冲突相关的所有赋值并回溯到较早的决策层。如果回溯水平小于 0,那么问题就不可满足。

3 策略分析

HSVDL 策略中变量得分的增量与冲突次数成正比。即越往后,用于冲突分析的变量具有的增量越来越大。而变量越早被用于分支决策,冲突次数越少,决策层越低,变量的得分增量就会越小。而且在发生的同一次冲突中,决策层高的变量的得分增量小于决策层低的变量的得分增量,因此在下一次变量决策时决策层低的变量被选为决策变量的可能性有所增加。

下面通过例子从理论的角度来分析变量选择过程中 HSVDL 和 EVSIDS、ACIDS 的不同。

例 1 设子句集 $C = \{c_1, c_2, \dots, c_6\}$, 其中:

$$c_1 = \neg x_1 \vee \neg x_3 \vee x_5$$

$$c_2 = x_1 \vee x_4$$

$$c_3 = \neg x_2 \vee \neg x_4 \vee x_7 \vee x_8 \vee x_9$$

$$c_4 = x_1 \vee x_2 \vee \neg x_6$$

$$c_5 = x_1 \vee \neg x_9$$

$$c_6 = \neg x_4 \vee \neg x_8 \vee \neg x_{10}$$

用 S_i 表示变量 x_i 的得分,假设变量当前的得分情况如表 1,当前的部分赋值为 $\{x_3 = 1(5), x_5 = 0(2), x_6 = 1(3), x_7 = 0(4), x_{10} = 1(3)\}$,小括号内的数表示变量的决策层,如 $x_3 = 1(5)$ 表示 x_3 在第 5 决策层被赋值为 1,根据这些赋值对 c_1 进行布尔约束传播 (BCP),可以得到 $x_1 = 0(5)$,再对 c_2 和 c_5 进行布尔约束传播,可以得到 $x_4 = 1(5), x_9 = 0(5)$ 。由 $x_6 = 1(3)$ 和 c_4 可以得到 $x_2 = 1(5)$,然后由 c_3 得到 $x_8 = 1(5)$,但是由 c_6 得到 $x_8 = 0(5)$ 。由此,搜索过程遇到了冲突。假设这是整个搜索过程的第 150 次冲突,由冲突分析过程得到学习子句为 $c_L = x_1 \vee \neg x_6 \vee x_7 \vee \neg x_{10}$,得到学习子句的过程中所涉及的变量有 $x_1, x_2, x_4, x_6, x_7, x_8, x_9, x_{10}$ 。所以在下一次变量决策前只有变量 x_3 和 x_5 的得分保持不变。用 S_{A_i} 表示用 ACIDS 策略发生冲突后变量的得分,用 S_{E_i} 表示用 EVSIDS 策略发生冲突后变量的得分,取 $f = 0.99$,用 S_{H_i} 表示用 HSVDL 策略发生冲突后变量的得分,得分情况如表 1 所示。

表 1 变量得分情况 ($n = 150$)

得分	变量									
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
S_i	210	280	320	300	510	400	370	250	200	370
S_{E_i}	214.516	284.516	320	304.516	510	404.516	374.516	254.516	204.516	374.516
S_{A_i}	180	215	320	225	510	275	260	200	175	260
S_{H_i}	105.143	140.107	320	150.1	510	200.25	185.101	125.12	100.15	185.135

若采用 EVSIDS 策略,首先对 x_5 赋值为 0,然后对 x_6 赋值为 0,再随机对 x_7 (或 x_{10}) 赋值为 0。再对

x_{10} (或 x_7) 赋值为 0,然后依次对 x_3, x_4 都赋值为 0,最后由 BCP 得到 $x_1 = 1$ 。

若采用 ACIDS 策略,则赋值过程与 EVSIDS 策略的赋值过程相同。

若采用 HSVDL 策略,则首先对 x_5 赋值为 0,再对 x_3 赋值为 0,再依次对 x_6 、 x_{10} 、 x_7 、 x_4 都赋值为 0,则通过 BCP 得到 $x_1=0$,最后对 x_2 赋值为 0。

对于上述实例,3 种策略在冲突之后都成功回溯并得到部分赋值,使得子句可满足。但策略不同,对于搜索结果就会有不同。相比较而言,由于 EVSIDS 策略中变量的增量与冲突次数成指数关系,所以变量的得分比其他 2 种策略的得分要大,这对内存的占用有很大的影响。后 2 种策略的不同在于,当参与冲突的子句的得分相同,而变量所在的决策层不同时, EVSIDS 策略和 ACIDS 策略在选择变量 x_7 和 x_{10} 时都具有随机性,而 HSVDL 策略选择了决策层较低的变量 x_{10} ,不再具有随机性,因此在后面发生冲突后的变量选择时,变量 x_{10} 被选为决策变量的可能性比变量 x_7 要大。而且 HSVDL 策略在冲突之后,变量的得分是 3 种策略中最小的,这对防止内存的溢出有很大的意义。

4 实验结果

为了评估 HSVDL 策略对 CDCL 求解器求解效率的影响,本文采用 2018 年 SAT 竞赛例进行对比测试,并给出实验结果。具体的测试方法如下:将本文所提出的基于变量决策层的启发式变量选择策略嵌入 Glucose-4.1 求解器中,求解器中其他诸如回溯、子句删除、重启等组成部分不作任何改变。所有测试均在硬件条件为 Intel(R) Core 64-bit(TM) i5-7200U CPU @ 2.50 GHz CPU、4.00 GB RAM、Windows 10 的计算机上进行。表 2 给出了 2 个求解器对 2018 年 SAT 竞赛中 400 个测试例的测试结果,每个测试例限时 5000 s。“Glucose-HSVDL 策略”表示将 HSVDL 策略嵌入求解器 Glucose-4.1 中形成的求解器。

表 2 2018 年 SAT 竞赛求解实例个数和时间

求解器	实例数/个			平均用时/s
	SAT	UNSAT	Total	
Glucose-4.1	82	88	170	1083.66
Glucose-HSVDL	89	93	182	1070.4

表 2 中:SAT 表示成功求解可满足测试例的个数;UNSAT 表示成功得到的不可满足例子个数;TOTAL 表示求解测试例的总和;平均用时表示求解器成功求解的测试例的平均用时。

从表 2 可以看出,求解器 Glucose-4.1 在加入了 HSVDL 策略后能求解出更多的测试例,求解个数提高了 7.0588%,同时求解用时也缩短了,因此 HSVDL

策略有一定的优势。

5 结束语

本文首先介绍了 VSIDS 策略及其 2 种变体 EVSIDS 策略和 ACIDS 策略,然后在 VSIDS 延伸策略的基础上提出了一种基于变量决策层的启发式变量选择策略——HSVDL 策略,并生成了算法。随后,通过实例从理论上说明了变量选择阶段 EVSIDS 策略、ACIDS 策略和 HSVDL 策略的区别,并分析了 3 种策略的特点以及 HSVDL 策略的优势。最后通过实验结果说明了 HSVDL 策略的优势。但启发式分支决策过程不是独立存在的,它与变量所在的决策层以及冲突次数都有一定的联系。因此接下来的工作是对变量的决策层、变量得分以及冲突次数是如何相互影响的进行更深入的研究,并形成求解器,使得分支策略的选择更加高效。

参考文献:

- [1] 陈稳. 基于 DPLL 的 SAT 算法的研究及应用[D]. 成都:电子科技大学, 2011.
- [2] JACKSON D, VAZIRI M. Finding bugs with a constraint solver[C]// Proceedings of the 2000 ACM SIGSOFT International Symposium on Software Testing and Analysis. 2000:14-25.
- [3] VELEV M N, BRYANT R E. Effective use of Boolean satisfiability procedures in the formal verification of superscalar and VLIW microprocessors[J]. Journal of Symbolic Computation, 2003,35(2):73-106.
- [4] KAUTZ H, SELMAN B. The state of SAT[J]. Discrete Applied Mathematics, 2007,155(12):1514-1524.
- [5] 张健. 逻辑公式的可满足性判定:方法工具及应用[M]. 北京:科学出版社, 2000:8-20.
- [6] DAVIS M, LOGEMANN G, LOVELAND D. A machine program for theorem-proving[J]. Communications of the ACM, 1962,5(7):394-397.
- [7] CALABRO C, IMPAGLIAZZO R, PATURI R. A duality between clause width and clause density for SAT[C]// Proceedings of the 21st Annual IEEE Conference on Computational Complexity. 2006:252-260.
- [8] JEROSLOW R G, WANG J C. Solving propositional satisfiability problems[J]. Annals of Mathematics and Artificial Intelligence, 1990,1(14):167-187.
- [9] Buro M, Buning H K. Report on a SAT Competition[R]. University of Paderborn, 1992.
- [10] DUBOIS O, ANDRE P, BOUFKHAD Y, et al. SAT versus UNSAT[M]// DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 1996,26:415-434. (下转第 96 页)

- structure within organizations[J]. The Information Society, 2005,21(2):143-153.
- [7] NEWMAN M E J, GIRVAN M. Finding and evaluating community structure in networks[J]. Physical Review E, 2004,69(2), DOI: 10.1103/PhysRevE.69.026113.
- [8] NEWMAN M E J. Fast algorithm for detecting community structure in networks[J]. Physical Review E, 2004,69(6), DOI: 10.1103/PhysRevE.69.066133.
- [9] GUIMERA R, NUNES AMARAL L A. Functional cartography of complex metabolic networks[J]. Nature, 2005, 433(7028):895-900.
- [10] NEWMAN M E J. Modularity and community structure in networks[J]. Proceedings of the National Academy of Sciences, 2006,103(23):8577-8582.
- [11] YANG B, CHEUNG W K, LIU J M. Community mining from signed social networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2007,19(10):1333-1348.
- [12] BARBER M J, CLARK J W. Detecting network communities by propagating labels under constraints[J]. Physical Review E, 2009,80(2), DOI: 10.1103/PhysRevE.80.026129.
- [13] 柴变芳,贾彩燕,于剑. 基于统计推理的社区发现模型综述[J]. 计算机科学, 2012,39(8):1-7.
- [14] HASTINGS M B. Community detection as an inference problem[J]. Physical Review E, 2006,74(3), DOI: 10.1103/PhysRevE.74.035102.
- [15] NEWMAN M E J, LEICHT E A. Mixture models and exploratory analysis in networks[J]. Proceedings of the National Academy of Sciences, 2007,104(23):9564-9569.
- [16] BLEI D M, NG A Y, JORDAN M I. Latent Dirichlet allocation[J]. Journal of Machine Learning Research, 2003, 3:993-1022.
- [17] AIROLDI E M, BLEI D M, FIENBERG S E, et al. Mixed membership stochastic blockmodels[J]. Journal of Machine Learning Research, 2008,9:1981-2014.
- [18] AIROLDI E M, BLEI D M, FIENBERG S E, et al. Mixed membership stochastic block models for relational data with application to protein-protein interactions[C]// Proceedings of the 2006 International Biometrics Society Annual Meeting. 2006:1-34.
- [19] REN W, YAN G Y, LIAO X P, et al. Simple probabilistic algorithm for detecting community structure[J]. Physical Review E, 2009,79(3), DOI: 10.1103/PhysRevE.79.036111.
- [20] BALL B, KARRER B, NEWMAN M E J. Efficient and principled method for detecting communities in networks[J]. Physical Review E, 2011,84(3), DOI: 10.1103/PhysRevE.84.036103.
- [21] DUAN D S, LI Y H, LI R X, et al. MEI: Mutual enhanced infinite generative model for simultaneous community and topic detection[C]// Proceedings of the 14th International Conference on Discovery Science. 2011:91-106.
- [22] SHEN H W, CHENG X Q, GUO J F. Exploring the structural regularities in networks[J]. Physical Review E, 2011,84(5), DOI: 10.1103/PhysRevE.84.056111.
- [23] ROWE R, CREAMER G, HERSHKOP S, et al. Automated social hierarchy detection through email network analysis[C]// Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis. 2007:109-117.
- [24] GUPTA M, SHANKAR P, LI J, et al. Finding hierarchy in directed online social networks[C]// Proceedings of the 20th International Conference on World Wide Web. 2011:557-566.
- [25] KURT S E. Finding Hidden Hierarchy in Social Networks[D]. Bilkent University, 2016.
- [26] FORTUNATO S, HRIC D. Community detection in networks: A user guide[J]. Physics Reports, 2016,659:1-44.
-
- (上接第 24 页)
- [11] MARQUES-SILVA J. The impact of branching heuristics in propositional satisfiability algorithms[C]// Proceedings of the 9th Portuguese Conference on Artificial Intelligence. 1999:62-74.
- [12] 杜忠和. 基于 CDCL 结构的 SAT 问题优化策略的研究[D]. 成都:西南交通大学, 2018.
- [13] BIERE A. Lingeling, Plingeling and Treengeling entering the SAT competition 2013[C]// Proceedings of SAT Competition 2013. 2013:51-52.
- [14] 陈青山,徐扬,何星星. 利用逻辑演绎求解 SAT 问题的启发式完全算法[J]. 西南交通大学学报, 2017,52(6):1224-1232.
- [15] MOSKEWICZ M W, MADIGAN C F, ZHAO Y, et al. Chaff: Engineering an efficient SAT solver[C]// Proceedings of the 38th Annual Design Automation Conference. 2001:530-535.
- [16] MARQUES-SILVA J, LYNCE I, MALIK S. Conflict-driven clause learning SAT solvers[M]// Frontiers in Artificial Intelligence and Applications: Handbook of Satisfiability. IOS Press, 2009,185:131-153.
- [17] HEULE M J H, KULLMANN O, WIERINGA S, et al. Cube and conquer: Guiding CDCL SAT solvers by lookahead[C]// Proceedings of the 7th International Haifa Verification Conference on Hardware and Software: Verification and Testing. 2011:50-65.
- [18] MAHAJAN Y S, FU Z H, MALIK S. Zchaff2004: An efficient SAT solver[C]// Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing. 2004:360-375.
- [19] 胡容. 基于 CDCL 的 SAT 问题的分支启发式策略研究[D]. 成都:西南交通大学, 2018.
- [20] EEN N, SORENSSON N. An extensible SAT-solver[C]// Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing. 2003:502-518.
- [21] BIERE A. Adaptive restart strategies for conflict driven SAT solvers[C]// Proceedings of the 11th International Conference on Theory and Applications of Satisfiability Testing. 2008:28-33.
- [22] BIERE A, FROHLICH A. Evaluating CDCL variable scoring schemes[C]// Proceedings of the 18th International Conference on Theory and Applications of Satisfiability Testing. 2015:405-422.
- [23] 陈青山. 基于矛盾体分离的命题逻辑动态自动演绎推理求解系统研究[D]. 成都:西南交通大学, 2018.