

一种基于加权决策变量决策层的分支策略

王 萌 何星星
(西南交通大学数学学院 成都 610031)

摘 要 为了提高 CDCL 求解器的求解效率,针对可满足性(SAT)问题算法中决策变量的选择问题,提出了一种基于加权决策变量决策层的分支策略。这个新策略的主要思想是:基于布尔约束传播(BCP)过程中的回溯以及重启机制,首先考虑变量作为决策变量的次数以及所在决策层;其次,由于被选择的次数以及所在决策层不同,即占权重不同,对此进行加权;最后,结合冲突分析过程,对不同的变量给予不同的奖励得分。将不同变量在新策略与 VSIDS 和 EVIDS 策略中的得分进行对比,采用 SATLIB(SAT Little Information Bank)中的大量例子进行实验测试,结果表明,新策略能够减少冲突次数以及求解时间(cpu),提高了求解器的求解效率。

关键词 决策层,加权,重启,决策变量,冲突

中图法分类号 TP301 **文献标识码** A

Branching Strategy Based on Weighted Decision Variable Level
WANG Meng HE Xing-xing
(School of Mathematics, Southwest Jiaotong University, Chengdu 610031, China)

Abstract In order to improve the solution efficiency of CDCL solver, for the choice of decision variable problem of the satisfiability (SAT) problem algorithm, a kind of branching strategy based on weighted decision variable level was proposed. The main idea of the new strategy is based on the boolean constraint propagation (BCP) back track and restart mechanism in the process. Firstly, the number of variables used as decision variables and the decision-making level are considered. Secondly, due to the selected number of variables and the difference in the decision-making level, the weight of variables is considered to be different. Finally, in combination with the conflict analysis process, the variables are rewarded and scored. The scores of different variables in the new strategy are compared with those in the VSIDS and EVIDS strategies. A large number of examples in SATLIB (SAT Little Information Bank) are used for experimental testing, and the results show that the new strategy can reduce the number of conflicts and the solution time (CPU), and improve the solving efficiency of the solver.

Keywords Decision level, Weighted, Restart, Decision variables, Conflicts

1 引言

布尔可满足问题(SAT)是指将给定的布尔变量 $\{x_1, x_2, \dots, x_n\}$ 描述为合取范式 $F^{[1]}$,判断是否存在一组布尔变量赋值,使得 F 上所有子句的值为真。SAT问题是人工智能、计算机科学以及软件和硬件验证等领域的一个核心问题,也是第一个被证明的NP完全问题^[2]。因此,研究SAT问题并且提高求解器的求解速度很有意义。

目前,最为流行的求解算法是冲突驱动子句学习(Conflict Driven Clause Learning, CDCL)^[3]算法。选择决策变元是该算法中的一个重要问题。数十年来,国内外学者们提出了各种启发式分支算法,如JW^[4], MOM^[5], DLIS^[6], 该类算法可被看作是由结构信息本身提出的算法,并没有结合冲突分析机制,故求解效率较低。而另一类启发式算法,如VSIDS^[7]及其变体、CHB^[8]、LRB^[9],不仅利用了子句本身的一些信息特征,还考虑了冲突分析过程,比第一类算法有优势。但是,此类算法中的一些策略主要依据冲突分析机制,考虑因素单一,未能提升求解效率,在求解时间等方面还有优化进步的空间。基于此,本文在第二类算法的基础上进行优化,基于冲突分析、回溯以及重启机制,提出一种基于加权决策变量决策层的分支策略,并将该改进策略算法嵌入求解器中,采用基本例子进行实验测试和对比分析,最后得出结论:新策略的加入能减少冲突、决策等指标的次数,尤为明显地减少了求解例子的时间,提高了求解效率。

2 基础知识

SAT问题的算法主要分为不完备算法和完备算法^[10]。而SAT完备求解算法几乎都基于DPLL(Davis Putnam Longmann Loveland)算法^[11],其主要利用单子句规则^[1]、重言式规则^[1]和纯文字规则^[1]以及分裂规则对问题进行化简及分析。CDCL算法是DPLL算法的扩展,其主要在变量决策^[12-15]、冲突分析、子句学习、非时序回溯^[16]以及重启^[17]

方面做出了改进。多数的 CDCL 求解器通过变量独立状态衰减和 (Variable State Independent Decaying Sum, VSIDS) 启发式及其各种变体如 EVSIDS^[18], ACIDS^[19] 等来确定每一决策层中决策变量的顺序。下面介绍 VSIDS 策略和 EVSIDS 策略。

VSIDS(Chaff 求解器^[7]) 策略的计分方式为:

- (1) 每一个文字都带有一个计数器, 当文字在子句集中出现时, 记录其出现的次数, 初值是 0;
- (2) 当 BCP 过程发生冲突产生学习时, 构成学习子句的文字的得分都增加;
- (3) 进行分支时, 选择计数器上分数最高的并且未被赋值的文字进行赋值;
- (4) 当计数器上最高分数相同时, 随机选择其一进行赋值;
- (5) 周期性将计数器上的得分除以某一个常数。

EVSIDS(Minisat 求解器^[18]) 策略的计分方式为:

由于 EVSIDS 为 VSIDS 的变体, 因此其计分方法与 VSIDS 类似, 不同点在于 BCP 过程中每一次发生冲突时, 是将参与冲突过程的相关变量的得分都增加 g^n (其中, $g=1/f$, $0<f<1$, n 为到目前为止发生的总冲突次数。

3 DVT 分支策略

从上述策略中可知, 不论 VSIDS 还是 EVSIDS 都只考虑了学习子句或者参与冲突的变量, 并对其增加得分, 而我们知道在 BCP 过程中, 非时序回溯与重启是必不可少的两部分, 同时每一个决策层中能作为决策变量的变量很重要, 并且越低决策层的决策变量越重要。基于此, 新策略考虑变量作为决策变量出现的次数 (Decide Variable Time, DVT), 并考虑此决策变量每次出现所在的决策层, 对决策层进行加权, 即不同决策层的决策变量的活性不同, 得分不同, 然后利用公式计算其得分。

- (1) 变量作为决策变量, 每次出现在同一决策层时:

$$S_n = s + DVT[V] \times \frac{1}{d}$$

- (2) 变量作为决策变量, 出现在不同决策层:

$$S_n = s + \sum (DVT[v]_i \times \frac{1}{d_j})$$

其中, $i \in [0, n], j \in [1, n], s_n$ 为新的得分, s 为在此次冲突发生之前变量的得分, $DVT[v]$ 为到目前为止变量作为决策变量的次数, d_j 为变量作为决策变量所在的决策层。

算法 1 DVT 策略算法

- 1. if (UnitPropagation() == conflict)
- 2. then return UNSAT;
- 3. end if
- 4. decision level of every variable = 0
- 5. conflict number n = 0;
- 6. while (not All Variables Assigned())

- 7. Pick Decision Variable();
- 8. blevel = blevel + 1;
- 9. dvt[v] = the number of variable is used as a decision variable
- 10. if (UnitPropagation() == conflict)
- 11. then blevel = ConflictAnalysis();
- 12. conflict number + = 1;
- 13. dj = the decision level;
- 14. The score of every variables is s
- 15. $S_n = S + \sum dvt[v]_i \times d_j$;
- 16. end if
- 17. If (blevel < 0)
- 18. then return UNSAT;
- 19. else Backtrack(blevel)
- 20. end while
- 21. return SAT

4 举例分析

给出一个子句集: $C_1 = X_1 \vee \neg X_2, C_2 = X_2 \vee \neg X_3 \vee X_4, C_3 = X_1 \vee X_3, C_4 = X_4 \vee \neg X_5 \vee X_8, C_5 = X_4 \vee \neg X_6 \vee \neg X_9, C_6 = X_6 \vee X_7, C_7 = X_2 \vee X_5 \vee \neg X_7 \vee \neg X_{10}$ 。部分赋值为 $X_1 = 0@, 1X_8 = 0@2, X_9 = 1@3, X_{10} = 0@4$, 其中 $X = 0@a$ 表示为变量 x 在第 a 决策层赋值 0。蕴含图^[20]为:

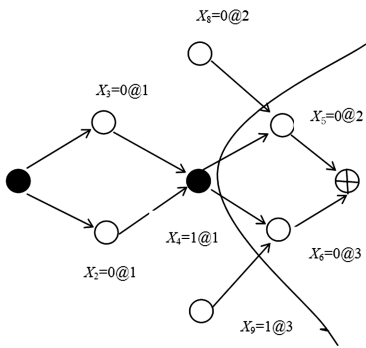


图 1 蕴含图

由此得到学习子句 $C_8 = X_4 \vee \neg X_8 \vee X_9 \vee \neg X_{10}$ 。

假设经过 100 次重启之后, 在此次冲突之前总共发生了 50 次冲突, 即此次为第 51 次冲突, 变量在前 50 次冲突之后得分为 s , 分别用 EVSIDS (在 Chaff 求解器) 方法和提出的 DVT 方法计算 10 个变量新的得分 s_1 以及 s_n 。分成两种情况讨论。

情况 1 $f=0.999, g=1/f$, 在前 50 次冲突中, 10 个变量都未作过决策变量, 且由蕴含图知参与冲突的变量有 $X_4, X_5, X_6, X_7, X_8, X_9, X_{10}$ 。各变量的得分情况如表 1 所列。

此情况下, 新方法 X_1, X_8, X_9, X_{10} 的得分比 EVSIDS 的方法高, 即作为一次决策变量之后, 活性增强, 使得后续的过程分支更快, 效率更高。

表 1 3 种不同策略下变量的得分

得分	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
S (VSIDS)	100	150	200	250	300	350	400	450	500	550
S_1 (EVSIDS)	100	150	200	251.051	301.051	351.051	401.051	451.051	501.051	551.051
S_n	101	150	200	251.051	301.051	351.051	401.051	451.551	501.351	551.301

情况 2 $f=0.999, g=1/f$, 假设在经过 100 次重启之后, 在前 50 次冲突中, 变量 X_1 作过 30 次决策变量, 并且决策层都是第一层, 变量 X_8 作过决策变量 20 次, 并且决策层都

在第 2 层, 其他变量并未作过决策变量, 另外由蕴含图可知参与冲突的变量有 $X_4, X_5, X_6, X_7, X_8, X_9, X_{10}$ 。各变量的得分情况如表 2 所列。

表 2 3 种不同策略下变量的得分

得分	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
S(VSIDS)	100	101	102	103	104	105	106	107	108	109
S_1 (EVIDS)	100	101	102	104.051	105.051	106.051	107.051	108.051	109.051	110.051
S_n	130	101	102	104.051	105.051	106.051	107.051	118.551	109.351	110.301

此情况下,新方法 X_1, X_8, X_9, X_{10} 的得分依然比 EVSIDS 的方法高,并且由此可以简单看出经过多次冲突、回溯以及重启之后,当一个变量作过多次的决策变量,并且决策层越低时,其得分就会改变越大。例如对于表 2 中的变量 X_1 ,我们可以看到其得分已经超过其他变量,由此可使后续的过程分支更快,效率更高。

5 实验分析

本文以 glucose4.1 为基础工具来实现 DVT 分支启发式策略,并与其进行对比分析。

本文测试的运行主机为 Intel (R) Core (TM) i5-4200U

CPU@1.60GHz 2.30GHz 4.00GB RAM Windows 8+cygwin Terminal,x64 处理器,64 位操作系统,设置每个测试例子的时限为 1800s。

实验测试实例来自于 SATLIB 中的 10 组(2000 个)基础例子,经过多次测量,将 10 组数据的结果展示在表 3 中,将 CBS_k3_n100_m449_b50,uuf250 组的部分数据展示在表 4 中,将 CBS_k3_n100_m403_b10 组的结果展示在图 1 中。在表中,V 代表变量个数,C 代表子句个数,Res 表示重启次数,Conf 表示冲突次数,CPU 代表求解时间,Ave CPU 代表求解的平均时间。

表 3 10 组例子的测试结果

组别名称	V	C	glucose		glucose+DVT	
			CPU	Ave CPU/s	CPU	Ave CPU/s
m403_b10	100	403	10.169	0.0203	6.679	0.0134
uf75	75	325	1.918	0.0192	1.375	0.0137
m449_b50	100	449	1.547	0.0155	1.449	0.0145
uuf250	200	860	603.024	6.030	590.936	5.909
uuf100	100	430	1.915	0.0192	1.791	0.0179
m403_b30	100	403	3.472	0.0174	3.053	0.0153
m411_b10	100	411	2.229	0.0223	1.1979	0.012
m411_b30	100	411	2.424	0.0242	1.639	0.0164
uuf50	50	218	1.25	0.013	0.852	0.009
m449_b30	100	449	7.904	0.016	6.958	0.014

表 4 CBS_k3_n100_m449_b50,uuf250 组的部分测试结果

组别名称	glucose					glucose+DVT				
	V	C	Res	Conf	CPU/s	V	C	Res	Conf	CPU/s
b50_0	100	449	1	222	0.031	100	449	1	178	0
b50_2	100	449	1	457	0.031	100	449	1	300	0
b50_3	100	449	1	61	0.046	100	449	1	55	0.015
b50_5	100	449	1	312	0.031	100	449	1	312	0.015
b50-8	100	449	2	472	0.062	100	449	1	402	0
250-04	250	1065	35	117775	6.093	250	1065	30	117700	5.656
250-09	250	1065	83	241044	12.125	250	1065	83	241021	11.593
250-20	250	1065	17	174603	8.453	250	1065	17	174508	7.921
250-22	250	1065	41	204881	10.125	250	1065	36	202734	9.156
250-24	250	1065	24	90600	4.234	250	1065	20	90600	3.718

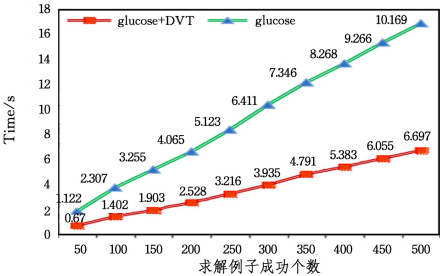


图 1 CBS_k3_n100_m403_b10 组的测试结果

由表 3 可以看出,经过实验测试,10 组例子即所有测试例子,在加入 DVT 策略后的总求解时间即 CPU 用时都是少于原求解器的,并且 10 组实验的平均求解时间也都少于原求解器,说明加入新策略,优化了求解过程,减少了求解时间,进而说明此策略是有效的。

由表 4 可以看出,两组测试例子在加入 DVT 策略的求解器后由于迭代次数减少,因此重启和冲突次数等各项指标均

优于原求解器,最终使得求解时间明显减少,说明 DVT 策略的确具有优化作用,使得求解效果更好。

从本组测试成功求解的 500 个例子的结果来看,glucose+DVT 策略的求解器的求解时间少于 glucose4.1,并且随着求解例子个数的增加,两种求解器的求解时间差也越来越大,说明加入 DVT 策略后求解器的求解效果更好。

结束语 本文介绍了 SAT 问题中的 CDCL 算法,对此算法中 VSIDS 及其变体分支策略进行分析,并以此为基础,提出了一种新的分支策略——DVT 分支策略,其改变了变量的活性。然后通过举例进行分析,得到在两种不同情况下,新策略均可使变量得分改变,为下一次分支提供更加有利的方向。最终,通过实验对比发现,加入 DVT 分支策略的求解器能够使得多数例子在冲突次数和求解时间等多项指标上明显优于原求解器,说明 DVT 分支策略更具有优势。

在接下来的研究中,计划将 DVT 策略嵌入不同的求解器,并增加试时间,测试更多的例子,以进行更深入的分析。

参 考 文 献

[1] 王国俊. 数理逻辑引论与归结原理[M]. 北京: 科学出版社, 2007:23-24.

[2] COOK S A. The complexity of theorem proving procedures [C] // Proceedings of the ACM SymPosium on Theory of Computin. Shaker Heights,1971:151-158.

[3] SILVA J P M,SAKALLAH K A. GRASP:a new search algorithm for satisfiability[C]//Proceedings of the IEEE/ACM International Conference on Computer-Aided Design. Los Alamitos,2002:220-227.

[4] JEROSLOW R G,WANG J. Solving propositional satisfiability problems[J]. Annals of Mathematics and Artificial Intelligence, 1990,1(1/2/3/4):167-187.

[5] FREEMAN J W.Improvements to propositional satisfiability search algorithms[D]. Philadelphia:University of Pennsylvania, 1995.

[6] MARQUES-SILVA J P,SAKALLAH K A. GRASP:A new search algorithm for satisfiability[C]// International Conference on Computer Aided Design. 1996:220-227.

[7] MOSKEWICZ M W,MADIGAN C F,ZHAO Y,et al. Chaff:engineering an efficient SAT solver[C]// Proceedings of the 38th Design Automation Conference. New York:ACM,2001:530-535.

[8] LIANG J H,GANESH V,POUPART P,et al. Learning rate based branching heuristic for SAT solvers[C]// Proceedings of the 19th International Conference on Theory and Applications of Satisfiability Testing(SAT 2016). 2016:123-140.

[9] AUDEMARD G,SIMON L. Predicting learnt clauses quality in modern SAT solvers[C]//Proceedings of the 21st International Joint Conference on Artificial Intelligence(IJCAI 2009). 2009: 399-404.

[10] MARQUES-SILVA J P,SAKALLAH K A. GRASP: A search algorithm for propositional satisfiability[J]. IEEE Transactions on Computers,1999,48(5):506-521.

[11] DAVIS M,LOGEMANN G,LOVELAND D. A machine program for theorem proving[J]. Communications of the ACM, 1962,59(5):394-397.

[12] BURO M,KLEINE-BÜNING H. Report on a SAT competition [J]. Bulletin of the European Association for Theoretical Computer Science,1993,5(2):49.

[13] MALIK S,ZHAO Y,MADIGAN C F. Chaff:an efficient SAT solver[C]//Proceedings of the Design Automation Conference. LasVegas,2001:530-535.

[14] SELMAN B,KAUTZ H,MCALLESTER D. Ten challenges in propositional reasoning and search[C]// Proceedings of the 15 the International Conference on Artificial Intelligence. Aichi, 1997:50-54.

[15] JEROSLOW R G,WANG J. Solving propositional satisfiabilityproblems[J]. Annals of Mathematics & Artificial Intelligence,1990,1(1/2/3/4):167-187.

[16] MARQUES-SILVA JP,SAKALLAH K A. Grasp:a new search algorithm for satisfiability[C]// Proceedings of the 1996 IEEE/ ACM International Conference on Computeraided Design. Los Alamitos,IEEE Computer Society Press,1996:220-227.

[17] GOMES C P, SELMAN B,CRATO N. Heavy-tailed distributions in combinatorial search[C]// Proceedings of the International Conference on Principles and Practice of Constraint Programming. Linz,1997:121-135.

[18] EËN N,SÖRENSSON N. An Extensible SAT-solver[M]// Theory and Applications of Satisfiability Testing. Santa Margherita Ligure,2003:502-518.

[19] BIERE A,FRÖHLICH A. Evaluating CDCL Variable Scoring Schemes[C]// International Conference on Theory and Applications of Satisfiability Testing. Cham:Springer,2015:405-422.

[20] ZHANG L,MADIGAN C,MOSKEWICZ M,et al. Efficient Conflict Driven Lening in a Boolean Satisfiability Slover[C]// International Conferenceon Computer Aided Design (ICCAD). SanJose CA,2001.

(上接第 18 页)

[6] 全昌勤,何婷婷,姬东鸿,等. 基于多分类器决策的词义消歧方法[J]. 计算机研究与发展,2006,43(5):933-939.

[7] 曹鸿霞. 统计与神经网络相结合的词义消歧模型[J]. 武汉理工大学学报,2006,28(8):131-134.

[8] 张雄,陈福才,黄瑞阳. 基于双词主题模型的半监督实体消歧方法研究[J]. 电子学报,2018,46(3):607-613.

[9] JIMENO Y A. Word embeddings and recurrent neural networks based on Long-Short Term Memory nodes in supervised biomedical word sense disambiguation[J]. Journal of biomedical informatics,2017,73:137-147.

[10] CORREA JR E A,LOPES A A,AMANCIO D R. Word sense disambiguation:A complex network approach[J]. Information Science,2018,442:103-113.

[11] 杨安,李素建,李芸. 基于领域知识和词向量的词义消歧方法[J]. 北京大学学报(自然科学版),2017,53(2):204-210.

[12] 杨陟卓,黄河燕. 基于语言模型的有监督词义消歧模型优化研究[J]. 中文信息学报,2014,28(1):19-25.

[13] 苏新春,洪桂治,唐师瑶. 再论义类词典的分类原则与方法[J]. 世界汉语教学,2010(2):16-27.

[14] WEN L,LI J,JIN Y,et al. A method for Word Sense Disambiguation combining contextual semantic features[C]//International Conference on Asian Language Processing. IEEE Press, 2017:283-287.

[15] LEI X,CAI Y,LI Q,et al. Combining Local and Global Features in Supervised Word Sense Disambiguation[C]// Web Information Systems Engineering-WISE 2017. Puschino, Russia: Springer,2017:117-131.

[16] PESARANGHADER A,PESARANGHADER A,MATWIN S, et al. One Single Deep Bidirectional LSTM Network for Word Sense Disambiguation of Text Data[C]// Advances in Artificial Intelligence. Canadian AI 2018. Toronto, Canada: Springer, 2018:96-107.

[17] LOPEZ-AREVALO I,SOSA-SOSA V J,ROJAS-LOPEZ F, et al. Improving selection of synsets from WordNet for domain-specific word sense disambiguation[J]. Computer Speech & Language,2016,41(C):128-145.

[18] 李安. 多义词义项的语义关系及其对词义消歧的影响[J]. 语言文字应用,2014(1):29-37.

[19] 廖光蓉. 多义词意义关系模式研究[J]. 外语教学,2005,26(3): 56-59.

[20] UNGERER F,SCHMID H J. An Introduction to Cognitive Linguistics[M]. London:Addison Welsey,1996.