

求解 SAT 问题的算法的研究进展

郭 莹^{1,2} 张长胜¹ 张 斌¹

(东北大学信息科学与工程学院 沈阳 110819)¹ (宁夏理工学院电气信息工程学院 石嘴山 753000)²

摘 要 SAT 问题是研究最广泛的 NPC 问题之一。由于 SAT 问题本身的特性,除非 $P=NP$, 否则不存在最坏情况下多项式阶时间复杂度的 SAT 求解算法。因此设计出高效快速的 SAT 求解算法至今仍是研究热点。首先简要介绍了 SAT 问题;其次从完备算法、不完备算法和组合算法 3 个角度总结了新近的研究进展,深入分析了已有算法解决 SAT 问题的基本流程,并从适用问题类别、算法特点、求解效率等方面对各类先进的求解器进行了对比分析;最后讨论了求解 SAT 问题的算法面临的挑战,并对下一步研究工作进行了展望。

关键词 SAT 问题,完备算法,不完备算法,组合算法

中图分类号 TP301 文献标识码 A DOI 10.11896/j.issn.1002-137X.2016.3.002

Research Advance of SAT Solving Algorithm

GUO Ying^{1,2} ZHANG Chang-sheng¹ ZHANG Bin¹

(Department of Information Science & Engineering, Northeastern University, Shenyang 110819, China)¹

(College of Electrical and Information Engineering, Ningxia Institute of Technology, Shizuishan 753000, China)²

Abstract SAT is one of the most widely studied NPC problems. Because of SAT characteristic, unless $P=NP$, polynomial time worst-case complexity of SAT algorithm does not exist. To design fast and efficient SAT algorithms is still a research hotspot. Firstly, the basic concept of SAT problem was introduced. Secondly, the recent research progress was summarized from three categories of complete algorithms, incomplete algorithms and combination algorithms. The general process of previous algorithms was thoroughly analyzed, and comparative analysis of some advanced solvers from the perspective of suitable problem class, algorithm character and solution efficiency was carried on. Finally, the main challenges of SAT solving algorithm was discussed, and some research issues in future were pointed out.

Keywords Satisfiability problem, Complete algorithm, Incomplete algorithm, Combination algorithm

1 引言

可满足性问题(Boolean Satisfiability Problem, SAT)^[1]是一个著名的判定问题。S. A. Cook^[2]于 1971 年证明了可满足性问题是世界上第一个 NP 完全(NP-complete, NPC)问题,也就是说,任何非确定多项式(Non-Deterministic Polynomial, NP)问题都可在多项式时间内规约到 SAT 进行求解,因此一个能高效地解决 SAT 问题的算法一定能解决所有的 NP 问题。在实际应用中存在着大量的 NP 问题,因此寻找高效快速的 SAT 求解算法并将其应用于工程实践,对提高生产率、促进社会的发展具有重要意义^[3,4]。

过去几十年中产生了大量高效且可扩展的 SAT 求解算法,目前 SAT 求解器可以高效地处理数百万变量规模的问题。然而,由于 SAT 问题本身的特性,除非 $P=NP$, 否则不存在最坏情况下多项式阶时间复杂度的 SAT 求解算法,因此设计出高效快速的 SAT 求解算法至今仍是研究热点。目前典型的 SAT 求解算法包括完备算法和不完备算法两大类,近年来两大类型算法都得到了广泛关注和研究。本文第 2 节简

单介绍了 SAT 问题的相关概念;第 3 节到第 5 节分别从完备算法、不完备算法和组合算法 3 个角度综述了 SAT 求解算法的研究进展;第 6 节分析了各类算法的实验对比结果;第 7 节讨论了 SAT 求解算法面临的挑战和研究方向;最后总结全文。

2 预备知识

由于任何一个命题逻辑公式都可在多项式时间内转化为合取范式(Conjunctive Normal Formula, CNF)^[5]形式,主流的 SAT 求解算法通常假设目标公式已被处理为 CNF 形式。假设 CNF 公式 $F_0(X)$ 如式(1)描述,下面举例并给出 SAT 问题的一般性描述。

$$F_0(X) = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (x_2 \vee \neg x_3 \vee \neg x_4) \quad (1)$$

定义 1(变量集合) 用符号 X 表示命题变量的集合,若 X 由 n 个变量 $x_1, x_2, x_3, \dots, x_n$ 组成,那么 $X = \{x_1, x_2, x_3, \dots, x_n\}$, 用 $n = |X|$ 表示变量集合的大小,问题规模一般由变量数 n 测量。对于 $F_0(X)$, $X = \{x_1, x_2, x_3, x_4\}$, $n = 4$ 。

到稿日期:2015-01-24 返修日期:2015-04-15 本文受宁夏回族自治区自然科学基金资助项目(NZ13265)资助。

郭 莹(1979—),女,博士生,副教授,主要研究方向为数据管理、智能信息处理, E-mail: yingfng@126.com; 张长胜(1980—),男,博士生,副教授,主要研究方向为智能信息处理; 张 斌(1964—),男,教授,博士生导师,主要研究方向为服务计算、数据挖掘。

定义 2(真值赋值) 给定一组布尔变量 $X = \{x_1, \dots, x_n\}$, 其真值赋值 $S(X)$ 是一个 n 元布尔函数: $S(X): X \rightarrow \{0, 1\}^n$, 在 X 上存在 2^n 种不同的真值赋值。如果 $S(x_i) = 1$, 则称 x_i 在 $S(X)$ 赋值下取真值, 否则为假值。

一个 SAT 求解器尝试求解一个问题时, 会进行完全或部分真值赋值。一个完全的真值赋值(记作 α)指给 $F(X)$ 的每一个变量赋值 0 或 1。一个部分的真值赋值(记作 β)指仅给 $F(X)$ 的变量子集赋值。例如 $F_0(X)$ 的一个完全真值赋值可能是 $\alpha = \{x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 0\}$, 而它的一个部分赋值可能是: $\beta = \{x_1 = 0, x_2 = 1\}$ 。

定义 3(文字) 对任意变量 x_i , 符号 x_i 和 $\neg x_i$ 是其文字, 称 x_i 是正文字, $\neg x_i$ 是负文字。用 L 表示文字的集合, l 表示某个文字。正文字 x_i 在真值赋值 $S(X)$ 下取真值当且仅当 $S(x_i) = 1$; 负文字 $\neg x_i$ 在真值赋值 $S(X)$ 下取真值当且仅当 $S(x_i) = 0$ 。例如有 $F_0(X)$ 的真值赋值 $S(x_1) = 0$, 则 $S(\neg x_1) = 1$ 。

定义 4(子句) X 上的子句是 X 中有限个文字的析取, 用 c 表示, $c = l_1 \vee l_2 \vee l_3 \vee \dots \vee l_k$ 。子句 c 在真值赋值 $S(X)$ 下取真值(或称子句 c 在真值赋值 $S(X)$ 下是可满足的), 当且仅当子句包含的文字中至少有一个在真值赋值 $S(X)$ 下取真值。 $k = |c|$ 表示子句 c 中的文字数, 称为子句长度。一个长度为 k 的子句是一个 k -子句。当 $k = 1$ 时, 称为单元子句。一个子句中的文字不许重复出现, 因为重复出现的文字会被子句忽略, 而不会影响公式的可满足性。例如 $F_0(X)$ 由 3 个子句组成, 其中子句 $c_1 = x_1 \vee \neg x_2 \vee x_3$, 子句长度为 3。

定义 5(子句集) 子句集 C 是由 X 上有限个子句组成的集合, $C = \{c_1, c_2, \dots, c_m\}$ 。 $m = |C|$ 表示子句集中子句的个数。以 k 为子句长度上限构成的问题称作 k -SAT。子句集 C 在真值赋值 $S(X)$ 下是可满足的, 当且仅当 C 中所有的子句 c 在真值赋值 $S(X)$ 下都是取真值的。例如 $F_0(X)$ 由 3 个子句组成, 子句最大长度为 3, 构成了一个 3-SAT 问题。

定义 6(合取范式) X 上的合取范式 $F(X)$ 是 X 上的一些子句的合取, $F(X) = c_1 \wedge c_2 \wedge \dots \wedge c_m$ 。合取范式 $F(X)$ 在真值赋值 $S(X)$ 下取真值(或称 $F(X)$ 在真值赋值 $S(X)$ 下是可满足的), 当且仅当 $F(X)$ 中包含的所有子句 c 在真值赋值 $S(X)$ 下都是取真值的。CNF 公式可简单描述为式(2)。

$$F(X) = \bigwedge_{i=1}^m \left(\bigvee_{j=1}^k l_{i,j} \right) \quad (2)$$

定义 7(SAT 问题) SAT 问题的基本形式指给定一个命题变量的集合 X 和一个 X 上的合取范式 $F(X)$, 判断是否存在一个关于 X 的真值赋值 $S(X)$, 使得 $F(X)$ 为真, 如果存在则称 $F(X)$ 是可满足的, 否则称 $F(X)$ 是不可满足的。例如, 若有 $S_1(X) = \{0, 0, 1, 1\}$, 该真值赋值使得 $F_0(X)$ 为 0, 那么 $S_1(X)$ 不是 $F_0(X)$ 的一个解; 但若 $S_2(X) = \{0, 0, 0, 1\}$, 该真值赋值使得 $F_0(X)$ 为 1, 那么 $S_2(X)$ 是 $F_0(X)$ 的一个解, 则 $F_0(X)$ 是可满足的; 若所有的真值赋值都不能使得公式真值为 1, 则 $F_0(X)$ 是不可满足的。

定义 8(解空间) 给定一个 SAT 问题公式 $F(X)$, 它的所有真值赋值称为它的解空间, 其中真值赋值的个数 2^n 为解空间的大小。解空间中的两个有且仅有一位不同的真值赋值称为相邻解。使得公式满足的一个真值赋值称为一个解。

定义 9(SAT 求解算法) 给定一个 SAT 问题公式 $F(X)$, 在有限的时间内判定其是否可满足的算法称为 SAT 求解算法。在 SAT 问题可满足的情况下, 算法往往会给出公式

的一个解。由于问题的 NP 特性, 算法可能没有判断结果。

SAT 问题按照可满足性可划分为可满足的 SAT 类和不可满足的 UNSAT 类; 按照问题产生的来源可划分为应用类(application)、随机生成类(random)和难以满足的组合类(hard combination)。不同的 SAT 求解算法在相同问题上或同一求解算法在不同问题上的表现会有很大不同。典型的 SAT 求解算法包括完备算法和不完备算法两大类, 目前很多研究人员尝试成功混合不同求解策略或并行多个求解器(本文统称为组合算法), 使其逐渐发展为一个新的研究热点。本文按照完备性和组合性将算法划分为完备算法、不完备算法和组合算法, 下文将从各类算法的求解过程、主要策略和典型算法等方面进行深入分析。

3 完备求解算法

完备算法比不完备算法的出现时间早。由于采取穷举和回溯的思想, 完备算法从理论上能保证判定给定命题公式的可满足性, 在实例无解的情况下可以给出完备证明, 因此也被称为系统搜索方法。该方法针对应用类实例很有效, 优势明显。

3.1 DPLL 算法

标准完备算法 DPLL (Davis Putnam Logemann Loveland, DPLL)^[7]起源于 20 世纪 60 年代初, 由于受到算法的指数级复杂度和计算机硬件性能的限制, 求解效率很低, 只能求解规模很小的 SAT 问题。DPLL 的重要思想在于穷举、分支回溯和布尔约束传播(Boolean Constraint Propagation, BCP), 对所有变量进行深度搜索以形成一棵完全二叉树的遍历过程, 如果问题是可满足的, 则给出所有解; 如果问题是不可满足的, 则给出完备性证明。若公式中所有变量都被赋值, 且没有发现一个冲突子句, 那么该布尔公式是可满足的, 否则是不可满足的。

算法 1 DPLL 算法

输入: CNF 公式 φ

输出: φ 的可满足性

```
while true do
    if !decide( $\varphi$ ) then
        return SATISFIABLE
    end if
    if !booleanConstraintPropagate( $\varphi$ ) then
        if !resolveConflict( $\varphi$ ) then
            return UNSATISFIABLE
        end if
    end if
end while
```

该算法的关键技术主要包括:

(1) 变量决策。DPLL 算法在尚未赋值的变量中选择某个变量并赋予某个值, 称为变量决策。变量决策决定了搜索树的形状。

(2) BCP。BCP 过程指根据选择的变量决策, 检查只包含为 0 的文字和一个未赋值文字的单元子句, 给此文字赋值为 1, BCP 过程一步步进行下去, 直到没有单元子句, 此时返回真; 若不同的单元子句要求一个变量取相反的值, 则称出现了冲突, 此时返回假。

(3) 回溯。当决策产生冲突时, 如果最后一个被赋值的变

量只尝试一个值(1 或 0),则翻转它的赋值(1 翻转为 0,0 翻转为 1);若仍有冲突发生,则翻转更早时候只被赋值 1 次的变量,取消在它之后所有的赋值操作,并返回真;如果不存在这样的变量,则说明所有搜索空间都被尝试过了,问题为不可满足的,返回假。

例如上文 $F_0(X)$ 的变量集合 $X=\{x_1, x_2, x_3, x_4\}$, 其所有可能的真值赋值将形成一个搜索空间,如图 1 所示。图中的每一条通路就表示一组真值赋值。DPLL 对这棵二叉树进行深度优先搜索,直到找到一个可满足的解或将所有通路遍历完成而没有找到可满足的解。同时也容易看出,随着问题规模变量数 n 的增大,搜索空间也呈指数级增长,甚至产生组合爆炸问题。

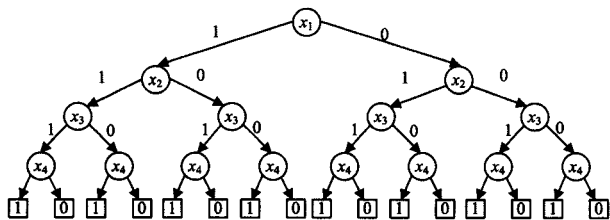


图 1 DPLL 搜索空间

由于 SAT 问题是一个 NP 问题,为了提高算法的求解效率,近年来提出了多种启发式策略。当前流行的完备求解算法大多都是基于 DPLL 这类框架的,如 SATO^[8]、BerkMin^[9]、PicoSAT^[10]等。按照 Heule 等人的观点^[6],基于 DPLL 的 SAT 求解器主要发展为两种形式,即冲突驱动子句学习(Conflict Driven Clause Learning, CDCL)和向前搜索(Look Ahead, LA),前者较适合大规模验证问题,后者则更适合不可满足的难解随机问题实例,二者都得到了广泛应用。

3.2 CDCL 算法

CDCL 算法是目前完备算法中最重要的一类,主要框架基于 DPLL,但在预处理、分支策略、冲突分析、子句学习、非时序回溯、重启、数据结构等方面做了一系列改进。CDCL 与 DPLL 的 3 个重要区别是^[11]:搜索过程不再递归,回溯使用一个显示的赋值栈(称为 trail);CDCL 在搜索过程中每次发现一个冲突子句时,都通过一个学习机制推导并增加新的子句,似乎增加的子句是多余的,但在剩余的整个搜索中它们能帮助 BCP 确定文字取值;回溯不再限制返回之前的决策点,子句学习的结果是双重的,在生成一个学习子句的同时分析哪些决策导致冲突,如果最近的 k 个决策与冲突不相干,程序不仅会撤销最近的决策,也会撤销这 k 次决策以及相应的 BCP 蕴涵。

算法 2 CDCL 算法

```

输入: CNF 公式  $\varphi$ 
输出:  $\varphi$  的可满足性
status = Preprocess( )
if (status != UNKNOWN) then
    return status
end if
blevel = 0
while not AllVariablesAssigned do
    Decide( $\varphi$ )
    blevel = blevel + 1
    while true do
        if booleanConstraintPropagate( $\varphi$ ) == CONFLICT then

```

```

        blevel = AnalyzeConflict( $\varphi$ )
        if blevel == 0 then
            return UNSATISFIABLE
        else
            Backtrack(blevel)
        end if
    end while
end while
return SATISFIABLE

```

该算法的关键技术主要包括:

(1) 预处理。CDCL 算法增强了预处理,即在求解之前对实例进行化简,例如删除在一个子句内重复出现的文字、包含同一个变量的正文字和负文字的子句,以及一些推理消除技术。

(2) 分支策略。分支决策是为了在执行一次赋值计算后从自由变量中找到下一个计算的分支节点。好的分支决策策略可以快速找到决策变量,加速 BCP 过程,减少回溯次数。CDCL 算法有许多著名的分支策略,如最小子句出现次数最多的变量优先(Maximum Occurrences in Clauses of Minimum, MOM)、半动态分支决策策略(Variable State Independent Decaying Sum Heuristic, VSIDS)、最短正子句优先等策略。

(3) 冲突分析。最早的 DPLL 算法并没有冲突分析,CDCL 算法引入了冲突分析和子句学习机制,在冲突发生时,将冲突子句加入子句集而不改变问题的可满足性,防止以后发生同样的冲突。

(4) 非时序回溯。相比早期的 DPLL 算法简单回跳到当前节点的父节点,CDCL 算法根据冲突分析和学习到的子句,回跳某一个或几个回溯层次,回溯到除蕴涵点变量之外其他变量对应的最大决策深度,称为非时序回溯。

(5) 重启。对于规模比较大的问题,如果早期决策时给某个变量赋值出错,那么子问题是不可满足的,算法对子问题仍在不停选择决策变量赋值,遇到冲突回退,直到算法回退到早期的那个错误分支,而当决策深度较深时,算法很难回退到初始的分支高度。重启指在算法运行期间,终止算法并重新启动,而重启之前学习到的冲突子句仍然保留。重启之前学习到的信息有利于调整全局搜索顺序,从而增加算法的稳健性。

(6) lazy 数据结构。数据结构不仅直接影响内存的使用,也间接影响了计算效率。CDCL 算法使每个子句保持两个观察文字,同时监测子句的可满足性。

近年来 CDCL 算法研究取得了丰富的成果^[12]。例如预处理方面,Eén 等人提出的 SatElite^[13]首次实现了预处理简化问题的规模和复杂性,Balint 等人提出了预处理剪枝技术^[14],Surynek 使用 $(2-k)$ 一致性边界对难以约束的区域进行设置^[15],熊伟等人利用对称扩展的一元子句推导得到更多的一元子句^[16]等;分支策略方面,一些新的智能启发式搜索技术先后被引入到 SAT 算法中,采用了一系列精巧的技术来减小搜索空间和问题规模,如变量无关性递减和启发算法^[17]、基于子句消减和附加推理过程的处理规则^[18]等;冲突分析和子句学习方面,Marques-Silva 等人基于 DPLL 的基础上提出的 GRASP 算法首次引入了冲突学习回溯策略^[19],并系统介绍了完善的 CDCL 求解器^[20],Hölldobler 进一步扩展了 CDCL,

提出了通用的形式化描述^[21]；重启策略方面，Nieuwenhuis 等人分析了随机重启在求解过程中的重要作用^[22]，Wu 研究了 SAT 求解的随机化和重启策略^[23]，Huang 分析了基于子句学习效率的重启效果^[24]，Audemard 等人研究了预测学习子句的质量评价方法^[25]，Biere 基于观察文字和积极的重启时间表等策略提出了 PicoSAT 求解算法，并在文献^[26]中系统阐述了基于冲突驱动的 SAT 求解器的自适应重启控制，Ramos 等人分析了 CDCL 求解器的重启和回溯问题^[27]，Haim 等人认为相比一组回溯间隔，重启应更加频繁^[28]；数据结构方面，多数研究采用堆栈或链表结构，也出现了双文字子句表示^[29]、改进存储访问模式^[30]等新技术。当前 CDCL 求解器的代表作有 Chaff^[20]、MiniSat^[31]、Lingeling^[32]等。

3.3 LA 算法

DPLL 算法和 CDCL 算法绝大多数时间都用来尝试各种可能的情况，可以看作蛮力求解算法。LA 算法^[33]的框架基于 DPLL，但侧重于通过大量的附加推理以及选择有效决策变量构建一个小巧平衡的搜索树来求解问题，在某些问题上优于前两种算法。LA 算法简化公式的方式可以通过观察进而强制给变量指定值，也可以通过附加分解得到进一步的公式约束。LA 算法整体是在前向搜索过程和 DPLL 搜索过程之间不断转换的，其在每一步会选择一决策变量，并递归调用 DPLL 简化公式，正如其名字所示，当一个自由变量被设置确定布尔值时，它会执行一决策启发式前向计算可能的消减，从而测量该变量的重要性，然后使用导向启发式测量选择该变量的真值赋值。在变量选择和真值选择过程，该算法主要增强了决策启发、方向启发和附加推理等策略，主要目的在于通过赋值推理尽可能地降低问题的复杂性。

算法 3 LA 算法

```

输入: CNF 公式  $\varphi$ 
输出:  $\varphi$  的可满足性
 $\varphi = \text{Simplify}(\varphi)$ 
while true do
    if  $\varphi = \text{NULL}$  then
        return SATISFIABLE
    else if  $\varphi$  contains empty clause then
        return UNSATISFIABLE
    end if
end if
 $\langle \varphi, x_{\text{decision}} \rangle = \text{LookAhead}(\varphi)$ 
 $B = \text{GetDirection}(x_{\text{decision}})$ 
if  $\text{DPLL}(x_{\text{decision}} \leftarrow B) = \text{SATISFIABLE}$  then
    return SATISFIABLE
end if
return  $\text{DPLL}(\varphi(x_{\text{decision}} \leftarrow \neg B))$ 
end while

```

该算法的关键技术主要包括：

(1) 决策启发。LA 选择判定变量，包括差异启发和混合差异启发两部分，差异启发在前向搜索过程中先量化变量取值对公式的简化程度，混合差异启发则综合考虑变量一对互补取值的前向差异值，进而选择最有效的自由变量。

(2) 方向启发。LA 从概率的角度在 GetDirection 程序中应用了方向启发，好的方向启发可以有效地缩小搜索空间。

(3) 附加推理。LA 在搜索过程中会检查哪些文字不会导致冲突出现，只对决策变量选择有用，而附加推理中利用一些失败文字简化公式，剩余的文字也可以通过生成学习子句进一步约束公式。

Posit 首次实现了 LA 算法^[34]，其包含了当今流行 LA 求解器的主要启发式，如 DIFF 启发式、导向启发式、变量预选择启发式等；Satz^[35]增强了在难随机 3-SAT 问题上的求解性能，进一步优化了决策启发式，并增加了 DoubleLook 过程，进一步消减搜索空间；OKsolver^[36]增加了局部学习、自给推理、回跳等推理过程；kcdfs^[37]与 Satz 很相似，在求解难随机 3-SAT 问题上进行了很多重要的改进，如主干搜索启发式 (Backbone Search Heuristic, BSH)、实施优化等；March^[38,39]是目前最为成功的 LA 求解器之一，主要特征是将 SAT 公式预处理为 3-SAT、分别存储二元子句和非二元子句、执行 LA 之前首先移除所有的已满足子句、控制失败文字数和预先设置自由变量数为最优值等。

4 不完备求解算法

除了极个别的不完备算法被用来尝试解决不可满足问题^[40,41]外，绝大多数不完备算法都不能判断 SAT 问题的不可满足性。它尽管是不完备的，但在处理可满足的大规模随机生成问题时往往比完备算法快，因此受到广泛的应用和研究。不完备算法主要基于局部搜索的思想，将不满足的子句数量看成优化目标，在解空间随机搜索，但并非是单纯随机地搜索，而是在目标函数的引导下逐步逼近最终解。

给定公式 $F(X)$ 和一个可能解 $S(X)$ ，目标函数 $fit(S)$ 一般定义为在 $S(X)$ 赋值下 $F(X)$ 中不可满足的子句数，如式 (3)。其中 $card(A)$ 表示 A 的基数； $sat(S, C)$ 表示在赋值 $S(X)$ 下子句 C 是否可满足，满足时值为 1，不满足时值为 0； $\neg$ 表示取反操作。 $fit(S)$ 函数的最小值 0 表示在该 $S(X)$ 赋值下，公式 $F(X)$ 是可满足的，即 $S(X)$ 为问题的一个解。

$$fit(S) = card(\{C | \neg sat(S, C) \wedge C \in F\}) \quad (3)$$

不完备算法的基本思想主要可分为 3 类，即随机局部搜索 (Stochastic Local Search, SLS)、消息传送 (Message Passing, MP) 和演化计算 (Evolutionary Algorithms, EA) 算法。

4.1 SLS 算法

大多数不完备算法是基于 SLS 的，其搜索过程为：给定一个命题公式，首先随机地生成一个真值赋值，如果该赋值使得公式可满足，则搜索结束；否则选择其中一个变量翻转其真值，该过程一直重复，直到找到使得公式满足的真值赋值或迭代次数达到预定上限 (公式的可满足性不能确定)。尽管 SLS 求解器会因附加不同的启发式而使复杂性提高，但整体来说，SLS 求解器都基于局部搜索非常简单的规则，同时支持运行过程中相当严谨的分析^[42]。

算法 4 SLS 算法

```

输入: CNF 公式  $\varphi$ , maxTries, maxSteps
输出: 使  $\varphi$  满足的一个解或者  $\varphi$  是未知的
for  $i = 1$  to maxTries do
     $s = \text{initAssign}(\varphi)$ 
    for  $j = 1$  to maxSteps do
        if  $s$  satisfies  $\varphi$  then
            return  $s$ 

```

```

else
  x=chooseVariable( $\varphi$ , s)
  s=s with truth value of x flipped
end if
end for
end for
return no satisfying assignment found

```

该算法的关键技术主要包括：

(1)搜索方式。常用的搜索方式分为贪心策略和最速下降策略，贪心策略只要获得比当前可行解好的解就进一步向前搜索，而最速下降策略要找到邻域中下降幅度最大的点。

(2)如何避免局部最优。如果一点的邻域中的所有点的目标函数值都比此点大，那么这个点就可以称为局部最优解。为了避免陷入局部最优，局部搜索算法通常采取 Escaping 策略，通常的做法是引入一个概率 p ，在某些情况下，以概率 p 选择非最优变量来改变其赋值。

Selman 等人提出了 SLS 思想的贪心局部搜索算法 GSAT^[43]，之后又对 GSAT 陷在局部最优解的情况作了改进，提出了 WalkSAT^[44]。早期的不完备求解器如 Unit-Walk^[45]、AdaptNovelty^[46] 等都是基于 WalkSAT 框架为基础的。另一个产生重要影响的 SLS 算法是离散拉格朗日方法 (Discrete Lagrangian Method, DLM)^[47]。国内许多对 SLS 的研究也得到了广泛关注，如顾钧等人提出的局部搜索算法^[48]、杨晋吉等人对局部搜索算法的改进^[49]、刘涛等人提出的分级重排算法^[50]等。近年来，SLS 算法的研究取得了进一步进展，如 Duong 等人研究了局部搜索过程中避免重复翻转的方法^[51]，Balint 提出了一种工程随机搜索算法^[52]，Luo 研究了随机局部搜索的双重配置检查方法^[53]，Brys 等人研究了 SLS 和重启策略的应用^[54]，Wang 等人进一步改进了 WalkSAT^[55]等。

4.2 MP 算法

MP 算法是相对较新的一种启发式算法。MP 起源于无序系统的统计物理领域，通过计算全局信息来引导搜索过程，尽管消息传送方法并不适合每一类 SAT 问题，但它发展出了一种计算所有解的统计特性的有效方法。文献^[56]进行了不同消息传送算法的综述，相关研究也可参考文献^[57-59]等。

消息传送算法工作在 SAT 公式的因子图表示^[60]之上，其变量和子句被表示为图中的顶点，变量节点与它出现过的子句相连，子句与它包含的变量相连，每个变量有一个介于 $[-1, 1]$ 之间的值，表示它选择某个极性 (即真或假) 的趋向。消息传送算法基于以下相同的原理：从变量到子句和从子句到变量进行消息传送，直到消息聚集到一点为止 (即它们的值不再有重大变化)，一旦如此，就开始大量地设置这些最具极性的变量，取它们的聚集值，并通过删除这些变量来简化问题，然后在简化后的公式上重新开始消息传送过程，直到公式简单到易于被其他更快的方法求解，如在此处使用一个局部搜索求解器。

算法 5 MP 算法 (以 BP 为例)

输入: CNF 公式 φ , 因子图, 迭代次数上限 $t_{\max}$, 精确度参数 ϵ

输出: 使 φ 满足的一个解或者 φ 是未知的

```

while true do
  BP-UPDATE( )

```

for each variable x_i do

compute $u_i(x_i)$

$x_i = x_i^*$ (using maximal-probability BP decimation)

for each variable x_i in a random order do

if exists a constraint b such that $f_b(x_b) = 0$ then

flip(x_i)

end if

if the total energy cost vanishes then

return the values of all the variables as a solution

else

return no satisfying assignment found

end if

end for

end while

该算法的关键技术主要包括：

(1)空腔偏量和空腔场。从子句节点到变量节点的基本消息称为空腔变量，从变量节点到子句节点的基本消息称为空腔场。在迭代搜索过程中将二者作为基本消息不断更新。

(2)消解。通过消息传送估计每个变量在所有解中的统计特性，在消解过程中固定尽可能多的变量，同时不除去过多的簇。

此类最著名且最成功的求解器有可信传播 (Belief Propagation, BP)、纵览传播 (Survey Propagation, SP)^[62]、模拟退火算法^[61]等。基于 SP 的求解器能在可接受的时间内解决数以百万计变量的大规模随机问题。

4.3 EA 算法

EA 算法是近年来发展起来的一种模仿自然界生物进化过程中“物竞天择，适者生存”原理的优化方法，研究人员在 EA 算法求解 SAT 问题方面也做了许多成功尝试，开辟了解决 SAT 问题的途径。其基本思想是将 SAT 问题转化为一个求响应目标函数最小值 (或最大值) 的优化问题，采取一系列启发式初始化种群并不断寻优，直到达到终止条件或找到问题的解。

算法 6 EA 算法 (以 GASAT 为例)

输入: CNF 公式 φ , Maxflip, MaxNbCrossovers

输出: 使 φ 满足的一个解或者 φ 是未知的

CreatePopulation (P)

NbCrossovers $\leftarrow$ 0

while no $x \in P$ satisfies φ and NbCrossovers $<$ MaxNbCrossovers do

$P' \leftarrow$ Select (P, NbInd)

ChooseX, Y $\in P'$

Z $\leftarrow$ Crossover(X, Y)

Z $\leftarrow$ TS(Z)

P $\leftarrow$ Replace (Z, P)

NbCrossovers $\leftarrow$ NbCrossovers + 1

if there exists X $\in P$ satisfying φ then

return X as a solution

else return no satisfying assignment found

end if

end while

该算法的关键技术主要包括：

(1)适应度函数。为了更好地体现不同解之间的差异，用

真值指派能满足公式中子句的数目作为适应度函数,问题转化为能使适应度函数取最大值的优化问题。

(2)选择算子。目的是从当前种群及新繁殖的个体中选择生命力强的若干个体形成新一代种群。一般为每个个体指定一个与其适应度函数值成比例的选择概率。

(3)变异算子。改变个体的部分信息,产生新的个体。

(4)交叉算子。交换父代中两个个体部分信息,产生出新的个体。

演化计算产生了丰富的成果,如遗传算法 GASAT^[63,64]、拟物拟人算法^[65]、量子免疫克隆算法^[66]、粒子群算法^[67]、模拟 DNA 算法^[68]、量子算法^[69]、人工蜂群算法^[70]、组织进化算法^[71,72]等,逐渐成为一个热门的研究领域。

5 组合求解算法

5.1 混合求解

某个求解器往往适用于一些特定问题类型,例如随机生成问题可能很容易利用不完备算法求解,但利用完备算法可能花费更长时间甚至不能求解;反之,对于完备算法能快速解决的应用和组合问题,不完备算法可能很慢或无能为力。因此很多研究者尝试混合不同求解策略的优势,来增强算法的适用性和求解效率。混合可能发生在不同的完备算法之间^[73,74]或在不同的不完备算法之间^[75,76],更多研究将完备算法和不完备算法进行了混合^[77-81],取得了成功进展,有效解决了许多应用类、随机类和组合类问题。

一个混合求解器包含两个或更多求解器,基本目标是创建能继承各个组件优点而回避缺点的一个求解器。一个理想的混合求解器能快速、完备、低成本地求解各类 SAT 问题。不完备算法擅长使用较少内存处理随机类问题,可以把全局信息传送给完备算法,从而指导完备算法决策变量的选择;而完备算法占用较多内存,擅长处理应用类和加工类问题,在随机类问题上表现平平,但它逻辑性较强,可以给不完备算法提供逻辑指导,二者有一定的互补性。

著名的混合求解器如 hybridGM^[82]以 SLS 求解器 gNovelty+为主导,阶段性调用 DPLL 求解器 March_ks 辅助搜索过程,是一个不完备的求解器;SatHys^[83]是一个保留了完备和不完备同等优势的完备求解器,目标是证明不完备算法可给完备算法提供重要信息;HBISAT^[84]使用 WalkSAT 进行局部搜索,与 DPLL 组合,能够加速增量搜索和不可满足性证明。这些求解器或许不能适合所有的 SAT 问题实例,然而已经呈现出接近最好的性能。

5.2 并行求解

单纯的序列(非并行)求解算法不可避免地受到 CPU 速度限制,并行求解算法日益受到广泛关注。最初的并行求解基于网络通信下的多核 CPU、共享存储空间等技术实现,目前并行 SAT 求解主要采取两种方法,一种是采取分治思想(Divide and Conquer,DC)将搜索空间划分为多个子空间进而分别调用序列算法求解^[85];另一种更为突出的方法是并行组合方法,它利用不同序列算法的互补性使之在同一公式上相互竞争和合作,如 Plingeling^[86]、动态子句共享策略^[87]、并行局部搜索^[88]、BMC 并行求解^[89]等。

其主要形式包括:

(1)网络通信。代表性求解器如 PMSat^[90]。网络通信形式为主从方式。主机进行假设并将问题划分为多个实例,主机存储这些分离的变量,向空闲的终端发送部分假设推理;终端可动态增加,如果某个终端证明某个子公式是不可确定的,它将返回 50 个最活跃的学习子句给主机。

(2)共享存储。代表性求解器如 PaMiraXT^[91]。共享存储形式在多个单核或多核 CPU 基础上建立,每个 CPU 有各自的存储总线和主存。

(3)单纯并行。代表性求解器如 ManySAT^[92]。序列 SAT 求解器中频繁重启产生的效益使得很多研究者转向求解器的并行组合,将搜索空间分离为多个,采取竞争并行的方式,允许多个求解器为同一 SAT 问题实例寻找解,因此也常被称为组合求解器。

6 几种典型 SAT 求解算法比较

通过以上分析可以看出,不同类别的 SAT 求解算法存在各自的求解特点和适用性,也存在各自的优点和局限性。表 1 通过 SAT Competition 2014(SC2014)结果^[93]分析了各类算法的适用问题类别。表 2 从算法特点、求解效率等方面对各类算法进行了定性对比分析。结果明显表明,完备算法特别是 CDCL 算法适用于应用类问题,在难以满足的组合类问题上组合类算法显示了明显优势,而随机性强的不完备算法更适合求解随机类问题实例。

表 1 SAT Competition 2014 结果分析

问题类别	应用类		难组合类		随机类	
	金奖	银奖	金奖	银奖	金奖	银奖
核心求解器						
SAT+UNSAT	CDCL	CDCL	混合算法	CDCL	/	/
SAT	CDCL	混合算法	混合算法	混合算法	SLS+MP	SLS
Certified UNSAT	CDCL	CDCL	混合算法	CDCL	/	/
核心求解器,并行						
SAT+UNSAT	CDCL 为主	CDCL 为主	CDCL 为主	CDCL 为主	/	/
SAT	/	/	/	/	SLS 为主	CDCL 为主

表 2 各类求解算法的定性比较

算法	随机性	逻辑性	全局性信息	速度	内存要求	主要障碍
DPLL	弱	强	少	慢	较大	决策变量选择
CDCL	弱	强	少	较慢	较大	决策变量选择
LA	弱	很强	少	慢	大	决策变量选择
SLS	强	弱	多	快	小	局部最优值判断和跳出
MP	强	弱	多	快	较小	局部最优值判断和跳出
智能优化	强	弱	多	快	小	局部最优值判断和跳出
混合求解	较强	较强	较多	快	大	共享信息
并行求解	较强	较强	较多	快	很大	共享资源

注:SLS、MP 和智能优化算法的求解速度快只限于问题有解时。

为了更准确地对比各类算法之间的求解优势和性能区别,选取 SC2014 获胜求解器为分析对象,其代表了当前各类求解器最先进的水平。下面从求解时间的角度分析不同算法。

图2显示了在序列应用类、难组合类和随机类问题实例上的获胜求解器的CPU时间分布。由于不完备算法在不可满足问题上不停机,未参与 UNSAT 类问题求解,鉴于横向比较需要,图2仅分析了 SAT 类的运行结果。结果显示,不同类别的问题适用于对不同的算法进行求解,例如 CDCL 算法 (minisat_bldb, RissBlackBOX, SWDiA5BY) 在应用类问题上表现突出;而对于难组合类问题,由 CDCL 和 SLS 混合组成的算法 (CCAnr+glucose, SGSeq, SparrowToRiss) 包揽了前三甲;随机类问题则主要由不完备算法求解, SLS 算法 (BalancedZ, CSCCSat) 名列前茅,其中 SLS 混合 MP 的算法 (Dimetheus) 更胜一筹。

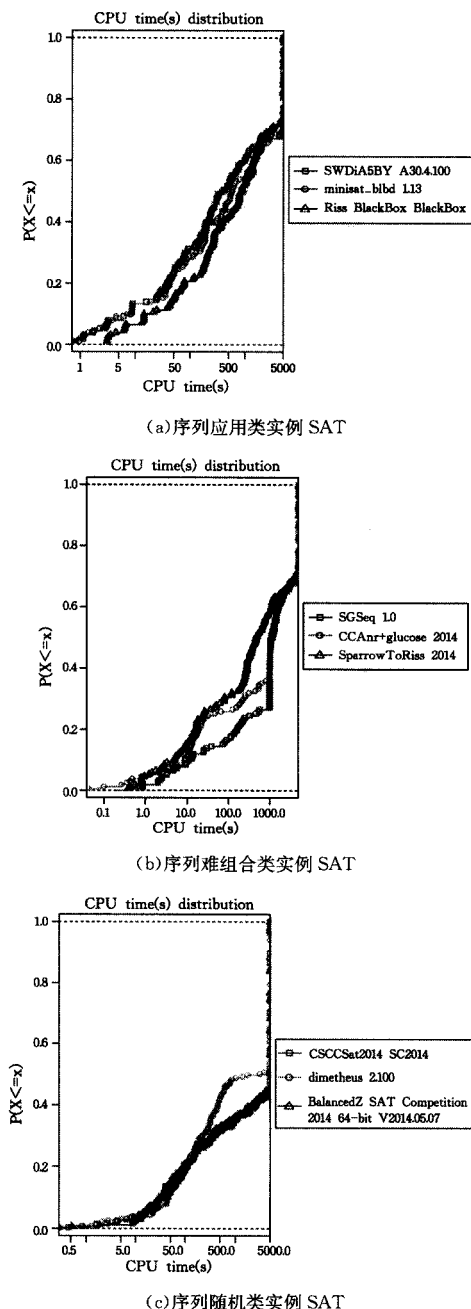


图2 各类求解算法适用问题类别比较

图3显示了在并行应用类、难组合类和随机类问题实例上获胜求解器的CPU时间分布。可以看出应用类和难组合类问题仍以完备算法为主 (Treengeling, Plingeling, Penelope, pmsat); 随机类问题主要采用不完备算法 (pprobSAT,

CSCCSat), 不同程度地使用了混合策略和并行求解技术。分析同一求解器在不同类别问题上的表现可以得到: Treengeling 和 Plingeling 在应用类和难组合类问题上都取得了名次, 然而其时间分布截然相反, 进一步验证了算法的适用性特点。

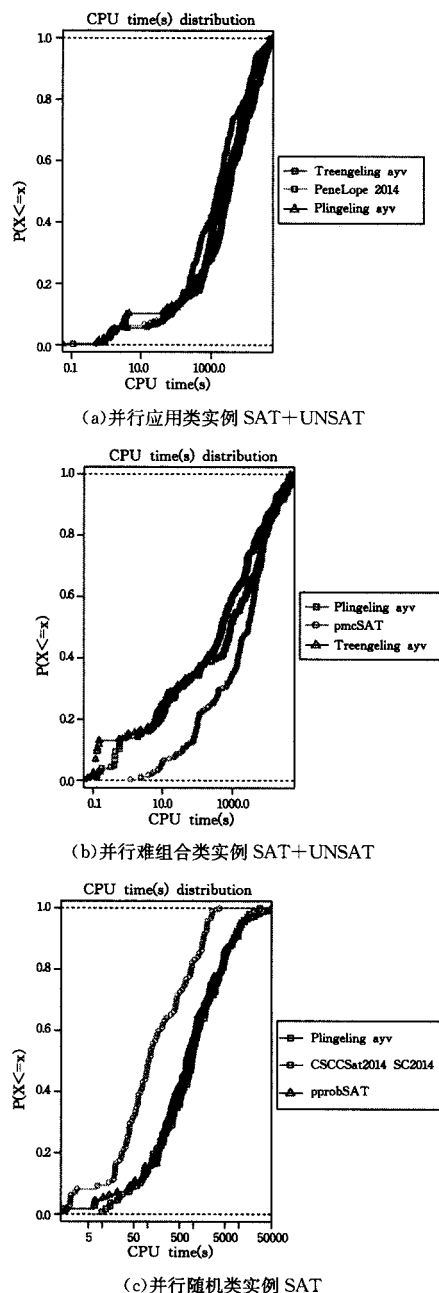


图3 各类求解算法求解速度比较

7 研究挑战和展望

Selman 和 Kautz 等人先后于 1997 年^[93]、2003 年^[94]、2007 年^[95]对当时的 SAT 问题研究现状进行了全面的阐述和总结, 提出了 SAT 问题面临的十大挑战性问题, 这十大挑战性问题的提出对于 SAT 基准问题的理论研究和算法改进都起到了强有力的推动作用。现有的 SAT 求解算法虽然已取得巨大成功, 但仍有一些问题没有得到高效解决, 已经解决的问题可能还存在更好的求解算法, 一些高效的求解器忽视了算法的正确性和完备性, 因此研究并实现高效率的求解算法仍是当前要解决的中心问题之一。未来的研究方向包括但不限于以下方面:

(1)问题应用和编码方面。加强实例和工业应用研究,将生产实际中更多 NP 难题转化为 SAT 问题进行求解,研究问题规约和表示,并研究特定领域的问题求解算法。为不同问题类型设计不同的算法是求解 SAT 问题的有效途径。

(2)预处理方面。进一步研究应用预处理技术降低 SAT 的问题规模和求解空间,并将有效的预处理技术融入其他求解算法。

(3)完备算法方面。深入研究完备算法的 BCP 过程、变量决策策略、冲突分析、子句学习机制和重启等关键技术点,提出优化搜索和启发式策略。

(4)不完备算法方面。深入理解不完备算法内部工作原理,探索新的求解算法或启发式策略,改善不完备算法在难解区域不易收敛的问题,设计有效的不完备算法来证明公式的不可满足性。

(5)混合求解方面。结合完备算法能够进行完备求解和局部搜索算法能够以较快速度进行求解的优点,研究不同问题上的混合求解策略,例如构建一个基于局域搜索的高效完全型求解器仍然任重道远。

(6)并行求解方面。针对大规模难组合类问题,研究实现各类并行 SAT 求解器,使得 SAT 求解器在多核结构上达到更佳效率和适应性。

(7)当前难解问题方面。针对目前多数求解器无能为力的难满足实例、大规模问题实例等,加强问题内部逻辑结构的研究,将一些应用的特定知识作为启发式信息,提高解决实际问题的效率。

(8)求解器实现方面。研究求解算法的设计与实现,降低求解时间和空间代价。

(9)理论研究方面。研究各类典型 SAT 求解算法的证明系统、证明复杂性及算法分析等。

(10)算法应用方面。研究基于 SAT 算法求解 Max-SAT、Mini-SAT、All-SAT、QBF、增量 SAT 及其他组合优化问题。

结束语 研究 SAT 问题在计算机科学、人工智能等领域具有重要的理论和应用意义,自 20 世纪 90 年代以来得到了充分的关注。本文重点考察 SAT 求解算法的特点和研究现状,将求解算法划分为完备算法、不完备算法和组合算法 3 类,并分析了各类算法的求解思路,同时总结了 SAT 求解算法研究历程的主要工作,概括介绍了著名的求解器,对比分析了各类求解算法的特点和性能,最后讨论了 SAT 求解算法面临的挑战,展望了未来的研究方向。

参考文献

- [1] Wikipedia. BooleanSatisfiabilityProblem[EB/OL]. http://en.wikipedia.org/wiki/Boolean_satisfiability_problem
- [2] Cook S A. The complexity of theorem proving procedures[C]// Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing. New York, 1971:151-158
- [3] Marques-Silva J. Practical applications of Boolean satisfiability [C]// 9th International Workshop on Discrete Event Systems (WODES 2008). IEEE, 2008:74-80
- [4] Xu L, Yu J P. Improved Bounded Model Checking on Verification of Valid ACTL Properties[J]. Computer Science, 2013, 40 (6A):99-102(in Chinese)

徐亮,余建平.改进的验证正确性 ACTL 性质的限界模型检测方法[J]. 计算机科学, 2013, 40(6A):99-102

- [5] Eén N, Mishchenko A, Sörensson N. Applying logic synthesis for speeding up SAT[M]// Theory and Applications of Satisfiability Testing(SAT2007). Springer Berlin Heidelberg, 2007:272-286
- [6] Heule M, Van Maaren H. March_dl: Adding Adaptive Heuristics and a New Branching Strategy[J]. JSAT, 2006, 2(1/4):47-59
- [7] Davis M, Putnam H. A computing procedure for quantification theory[J]. Journal of the ACM (JACM), 1960, 7(3):201-215
- [8] Zhang H. SATO: Anefficient prepositional prover[M]// Automated Deduction—CADE-14. Springer Berlin Heidelberg, 1997: 272-275
- [9] Goldberg E, Novikov Y. BerkMin: A fast and robust SAT-solver [J]. Discrete Applied Mathematics, 2007, 155(12):1549-1561
- [10] Biere A. PicoSATEssentials[J]. JSAT, 2008, 4(2-4):75-97
- [11] Claessen K, Eén N, Sheeran M, et al. SAT-solving in practice [C]// 9th International Workshop on Discrete Event Systems (WODES 2008). IEEE, 2008:61-67
- [12] Kullmann O. Present and future of practical SAT solving[M]// Complexity of Constraints. Springer Berlin Heidelberg, 2008: 283-319
- [13] Eén N, Biere A. Effective preprocessing in SAT through variable and clause elimination[C]// Theory and Applications of Satisfiability Testing. Springer Berlin Heidelberg, 2005:61-75
- [14] Balint A, Manthey N. Boosting the performance of SLS and CDCL solvers by preprocessor tuning[C]// Pragmatics of SAT. 2013, 29:1-14
- [15] Surynek P. Preprocessing in Propositional Satisfiability Using Bounded (2,k)-Consistency on Regions with a Locally Difficult Constraint Setup[J]. International Journal on Artificial Intelligence Tools, 2014, 23(1):1-27
- [16] Xiong W, Tang P S. A New Algorithm for SAT Preprocessing [J]. Microelectronics & Computer, 2007, 24(10):193-196 (in Chinese)
熊伟,唐璞山.一种新的 SAT 问题预处理算法[J]. 微电子学与计算机, 2007, 24(10):193-196
- [17] 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. ACM, 2001:530-535
- [18] Järvisalo M, Heule M J H, Biere A. Inprocessing rules[M]// Automated Reasoning. Springer Berlin Heidelberg, 2012:355-370
- [19] Marques-Silva J P, Sakallah K A. GRASP: A search algorithm for propositional satisfiability[J]. IEEE Transactions on Computers, 1999, 48(5):506-521
- [20] Marques-Silva J, Lynce I, Malik S. Conflict-driven clause learning SAT solvers[M]// Handbook of Satisfiability. 2009:131-153
- [21] Hölldobler S, Manthey N, Philipp T, et al. Generic CDCL-A Formalization of Modern Propositional Satisfiability Solvers[C]// Pragmatics of SAT 2014. 2014, 27:89-102
- [22] Nieuwenhuis R, Oliveras A, Tinelli C. Solving SAT and SAT modulo theories: From an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL[J]. Journal of the ACM, 2006, 53 (6):937-977

- [23] Wu H. Randomization and restart strategies[D]. Waterloo: University of Waterloo, 2006
- [24] Huang J. The Effect of Restarts on the Efficiency of Clause Learning[C]// IJCAI. 2007, 7: 2318-2323
- [25] Audemard G, Simon L. Predicting Learnt Clauses Quality in Modern SAT Solvers[C]// IJCAI. 2009, 9: 399-404
- [26] Biere A. Adaptive restart control for conflict driven SAT solvers [M]// Theory and Applications of Satisfiability Testing (SAT 2008). Springer Berlin Heidelberg, 2008: 8-13
- [27] Ramos A, Van Der Tak P, Heule M J H. Between restarts and backjumps[M]// Theory and Applications of Satisfiability Testing (SAT 2011). Springer Berlin Heidelberg, 2011: 216-229
- [28] Haim S, Heule M. Towards ultra rapid restarts[J]. arXiv preprint arxiv:1402.4413, 2014
- [29] Sörensson N, Eén N. Minisat v1. 13-a SAT solver with conflict-clause minimization[C]// SAT 2005. 2005
- [30] Biere A. A short history on sat solver technology and what is next [EB/OL]. <http://fmv.jku.at/biere/talks/biere-sat07-talk.pdf>
- [31] Eén N, Sörensson N. Translating pseudo-Boolean constraints into SAT[J]. Journal on Satisfiability, Boolean Modeling and Computation, 2006, 2(1/4): 1-26
- [32] Biere A. Lingeling, plingeling and treengeling entering the sat competition 2013[C]// Proceedings of SAT Competition. 2013: 51-52
- [33] Giunchiglia E, Maratea M, Tacchella A. Effectiveness of Look-Ahead Techniques in a Modern SAT Solver[M]// Principles and Practice of Constraint Programming-CP 2003. Springer Berlin Heidelberg, 2003: 842-846
- [34] Crawford J M, Kearns M J, Shapire R E. The minimal disagreement parity problem as a hard satisfiability problem[R]. Computational Intell. Research Lab and AT&T Bell Labs TR, 1994
- [35] Li C M. Integrating equivalency reasoning into Davis-Putnam procedure[C]// AAAI/IAAI. 2000: 291-296
- [36] Oliver K. The OKlibrary [EB/OL]. <http://www.ok-sat-library.org>
- [37] Dequen G, Olivier Dubois. Knfs: An efficient solver for random k-SAT formulae [M]// Theory and Applications of Satisfiability Testing. Springer Berlin Heidelberg, 2004: 486-501
- [38] Heule M, Van Maaren H. March_dl: Adding Adaptive Heuristics and a New Branching Strategy[J]. JSAT, 2006, 2(1-4): 47-59
- [39] Marijn H. Marchbr [M]// Proceedings of SAT Competition. Springer Berlin Heidelberg, 2013: 53
- [40] Audemard G, Simon L. GUNSAT: A Greedy Local Search Algorithm for Unsatisfiability[C]// IJCAI. 2007: 2256-2261
- [41] Pereira D, Lynce I, Prestwich S. On improving local search for unsatisfiability[J]. arXiv preprint arXiv:0910.1244, 2009
- [42] Hoos H H, Stützle T. Local search algorithms for SAT: An empirical evaluation[J]. Journal of Automated Reasoning, 2000, 24(4): 421-481
- [43] Selman B, Levesque H J, Mitchell D G. A new method for solving hard satisfiability problems [C]// Proceedings of the 12th National Conference on Artificial Intelligence. Cambridge: MIT Press, 1992: 440-446
- [44] Selman B, Kautz H A, Cohen B. Noise strategies for improving local search[C]// Proceedings of the 12th National Conference on Artificial Intelligence. 1994: 337-343
- [45] Hirsch E A, Kojevnikov A. UnitWalk: A new SAT solver that uses local search guided by unit clause elimination[J]. Annals of Mathematics and Artificial Intelligence, 2005, 43(1-4): 91-111
- [46] Hoos H H. An adaptive noise mechanism for WalkSAT[C]// AAAI/IAAI. 2002: 655-660
- [47] Shang Y, Wah B W. A discrete Lagrangian-based global-search method for solving satisfiability problems[J]. Journal of Global Optimization, 1998, 12(1): 61-99
- [48] Gu J. Local search for satisfiability (SAT) problem[J]. Trans. Systems, Man, and Cybernetics, 1993, 23(4): 1108-1129
- [49] Yang J J, Su K L. Improvement of Local Research in SAT Problem[J]. Journal of Computer Research and Development, 2005, 42(1): 60-65 (in Chinese)
杨晋吉, 苏开乐. SAT 问题中局部搜索法的改进[J]. 计算机研究与发展, 2005, 42(1): 60-65
- [50] Liu T, Li G J. Multi-stage search rearrangement algorithm for solving SAT problem[J]. Journal of Software, 1996, 7(4): 201-210 (in Chinese)
刘涛, 李国杰. 求解 SAT 问题的分级重排搜索算法[J]. 软件学报, 1996, 7(4): 201-210
- [51] Duong T T, Pham D N, Sattar A. A method to avoid duplicative flipping in local search for SAT[M]// AI 2012: Advances in Artificial Intelligence. Springer Berlin Heidelberg, 2012: 218-229
- [52] Balint A. Engineering stochastic local search for the satisfiability problem[D]. Universität Ulm; Fakultät für Ingenieurwissenschaften und Informatik, 2014
- [53] Luo C, Cai S, Wu W, et al. Double configuration checking in stochastic local search for satisfiability[C]// Twenty-Eighth AAAI Conference on Artificial Intelligence. 2014
- [54] Brys T, Drugan M M, Bosman P A N, et al. Local search and restart strategies for satisfiability solving in fuzzy logics[C]// Proceedings of the IEEE Symposium Series on Computational Intelligence-SSCI-2013. IEEE, 2013: 52-59
- [55] Wang X. A novel approach of solving the CNF-SAT problem [J]. arXiv preprint arXiv:1307.6291, 2013
- [56] Gableske O. On the interpolation between product-based message passing heuristics for SAT[M]// Theory and Applications of Satisfiability Testing (SAT 2013). Springer Berlin Heidelberg, 2013: 293-308
- [57] Zhao C, Zhou H, Zheng Z, et al. A message-passing approach to random constraint satisfaction problems with growing domains [J]. Journal of Statistical Mechanics: Theory and Experiment, 2011(2): P02019
- [58] Zhao C, Zhang P, Zheng Z, et al. Analytical and belief propagation studies of random constraint satisfaction problems with growing domains[J]. Physical Review E, 2012, 85(1/2): 016106
- [59] Mertens S, Mézard M, Zecchina R. Threshold values of random K-SAT from the cavity method[J]. Random Structures & Algorithms, 2006, 28(3): 340-373
- [60] Mézard M, Zecchina R. Random k-satisfiability problem: From an analytic solution to an efficient algorithm[J]. Physical Review E, 2002, 66(5): 56-126
- [61] Braunstein A, Mézard M, Zecchina R. Survey propagation: An algorithm for satisfiability [J]. Random Structures & Algorithms, 2005, 27(2): 201-226

- [62] Zhang D F, Huang W Q, Wang H X. Personification Annealing Algorithm for Solving SAT Problem[J]. Chinese Journal of Computers, 2002, 25(2): 148-152 (in Chinese)
张德富, 黄文奇, 王厚祥. 求解 SAT 问题的拟人退火算法[J]. 计算机学报, 2002, 25(2): 148-152
- [63] Lardeux F, Saubion F, Hao J K. GASAT: A genetic local search algorithm for the satisfiability problem[J]. Evolutionary Computation, 2006, 14(2): 223-253
- [64] Ling Y B, Wu X J, Jiang Y F. Genetic Algorithm for Solving SAT Problems Based on Learning Clause Weights [J]. Chinese Journal of Computers, 2005, 28(9): 1476-1482 (in Chinese)
凌应标, 吴向军, 姜云飞. 基于子句权重学习的求解 SAT 问题的遗传算法[J]. 计算机学报, 2005, 28(9): 1476-1482
- [65] Huang W Q, Jin R C. Quasi-physical and quasi-sociological algorithm for solving SAT problem[J]. Science in China (Series E), 1997, 27(2): 179-186 (in Chinese)
黄文奇, 金人超. 求解 SAT 问题的拟物拟人算法—Solar[J]. 中国科学: E 辑, 1997, 27(2): 179-186
- [66] Li Y Y, Jiao L C. Quantum-Inspired Immune Clonal Algorithm for SAT Problem [J]. Chinese Journal of Computers, 2007, 30(2): 176-183 (in Chinese)
李阳阳, 焦李成. 求解 SAT 问题的量子免疫克隆算法[J]. 计算机学报, 2007, 30(2): 176-183
- [67] Wang E Z. PSO and its application on SAT problem and multi-object problem[D]. Changchun: Jilin University, 2004 (in Chinese)
王恩重. 粒子群优化算法及其在 SAT 问题和多目标规划问题上的应用[D]. 长春: 吉林大学, 2004
- [68] Dai P, Zhou K, Wei Z, et al. Simulation DNA Algorithm[M]// Bio-Inspired Computing-Theories and Applications. Springer Berlin Heidelberg, 2014: 83-87
- [69] Wang H, Fan H, Li F. Quantum algorithm for solving some discrete mathematical problems by probing their energy spectra [J]. Physical Review A, 2014, 89(1): 012306
- [70] Guo Y, Zhang C S, Zhang B. An Artificial Bee Colony Algorithm for Solving SAT Problem [J]. Journal of Northeastern University (Natural Science), 2014, 35(1): 29-32 (in Chinese)
郭莹, 张长胜, 张斌. 一种求解 SAT 问题的人工蜂群算法[J]. 东北大学学报(自然科学版), 2014, 35(1): 29-32
- [71] Gottlieb J, Marchiori E, Rossi C. Evolutionary algorithms for the satisfiability problem [J]. Evolutionary Computation, 2002, 10(1): 35-50
- [72] Hauschild M W, Pelikan M, Sastry K, et al. Using previous models to bias structural learning in the hierarchical BOA[J]. Evolutionary Computation, 2012, 20(1): 135-160
- [73] Mpekass D, Michiel V V, Siert W. The first steps to a hybrid SAT solver[R]. Delft: Delft University of Technology, 2006
- [74] Chen J. Building a hybrid SAT solver via conflict-driven, look-ahead and XOR reasoning techniques[M]// Theory and Applications of Satisfiability Testing (SAT 2009). Springer Berlin Heidelberg, 2009: 298-311
- [75] Wang F, Zhou Y R, Ye L. Ant Colony Algorithm Combined with Survey Propagation for Satisfiability Problem[J]. Computer Science, 2012, 39(4): 227-231 (in Chinese)
王芙, 周育人, 叶立. 调查传播算法和蚁群算法相结合求解可满足性问题[J]. 计算机科学, 2012, 39(4): 227-231
- [76] Tseng L, Lin Y. A Hybrid Genetic Algorithm for the Maximum Satisfiability Problem[M]// Recent Trends in Applied Artificial Intelligence. Springer Berlin Heidelberg, 2013: 112-120
- [77] Li C, Wei W, Zhang H. Combining adaptive noise and look-ahead in local search for SAT[M]// Theory and Applications of Satisfiability Testing (SAT 2007). Springer Berlin Heidelberg, 2007: 121-133
- [78] Nelson, Nicole Ann. Hybrid Solvers for the Boolean Satisfiability Problem: An Exploration[D]. Diss. Rowan University, 2012
- [79] Layeb A, Saidouni D E. A Hybrid Quantum Genetic Algorithm and Local Search based DPLL for Max 3-SAT Problems[J]. Appl. Math, 2014, 8(1): 77-87
- [80] Letombe F, Marques-Silva J. Improvements to hybrid incremental SAT algorithms[M]// Theory and Applications of Satisfiability Testing (SAT 2008). Springer Berlin Heidelberg, 2008: 168-181
- [81] Inoue K, Soh T, et al. A competitive and cooperative approach to proposition satisfiability [J]. Discrete Applied Mathematics, 2006, 154(16): 2291-2306
- [82] Balint A, Henn M, Gableske O. A novel approach to combine a SLs and a DPLL-solver for the satisfiability problem[M]// Theory and Applications of Satisfiability Testing (SAT 2009). Springer Berlin Heidelberg, 2009: 284-297
- [83] Audemard, Gilles, et al. Boosting local search thanks to CDCL [M]// Logic for Programming, Artificial Intelligence and Reasoning. Springer Berlin Heidelberg, 2010: 474-488
- [84] Fang L, Hsiao M S. A new hybrid solution to boost SAT solver performance[C]// Proceedings of the conference on Design, automation and test in Europe. EDA Consortium, 2007: 1307-1313
- [85] Chrabakh W, Wolski R. GrADSAT: A parallel sat solver for the grid[C]// Proceedings of IEEE SC03. 2003: 53
- [86] Biere A. Lingeling, plingeling, picosat and precosat at SAT race 2010[R]. FMV Report Series Technical Report, 2010
- [87] Hamadi Y, Jabbour S, Sais J. Control-based clause sharing in parallel SAT solving[M]// Autonomous Search. Springer Berlin Heidelberg, 2012: 245-267
- [88] Arbelaez A, Hamadi Y. Improving parallel local search for SAT [M]// Learning and Intelligent Optimization. Springer Berlin Heidelberg, 2011: 46-60
- [89] Ábrahám E, Schubert T, Becker B, et al. Parallel SAT solving in bounded model checking[J]. Journal of Logic and Computation, 2011, 21(1): 5-21
- [90] Chu G, Stuckey P J, Harwood A. Pminisat: a parallelization of minisat 2.0[C]// SAT Race. 2008
- [91] Schubert T, Lewis M D T, Becker B. PaMiraXT: Parallel SAT Solving with Threads and Message Passing[J]. JSAT, 2009, 6(4): 203-222
- [92] Hamadi Y, Jabbour S, Sais L. ManySAT: a Parallel SAT Solver [J]. JSAT, 2009, 6(4): 245-262
- [93] Satlive. The International SAT Competitions Web Page [EB/OL]. <http://satcompetition.org>
- [94] Selman B, Kautz H, McAllester D. Ten challenges in propositional reasoning and search[C]// IJCAI. 1997: 50-54
- [95] Kautz H, Selman B. Ten challenges redux: Recent progress in propositional reasoning and search[M]// Principles and Practice of Constraint Programming (CP 2003). Springer Berlin Heidelberg, 2003: 1-18
- [96] Kautz H, Selman B. The state of SAT [J]. Discrete Applied Mathematics, 2007, 155(12): 1514-1524