

一种基于搜索路径识别的 CDCL 命题逻辑求解器延迟重启算法

陈青山^{1,2} 徐 扬² 吴贯锋^{1,2} 何星星²

(西南交通大学信息科学与技术学院 成都 611756)¹

(系统可信性自动验证国家地方联合工程实验室 成都 610031)²

摘 要 适当的重启有助于求解器跳出局部最优,但频繁重启会严重降低效率。为解决 CDCL 求解器重启触发条件随意性大的问题,提出一种基于搜索路径识别的延迟重启算法。该算法使用 Luby 序列触发延时重启判断,将当前搜索路径和已搜索路径转换为向量空间模型,通过计算向量空间相似度来判断当前搜索过程是否会进入重复搜索空间。若向量空间相似度达到设定阈值,则触发重启,否则延迟重启。采用 SAT 国际竞赛的实例,与两个主流的求解器进行了对比实验。结果表明,所提算法能够有效规避重复搜索空间问题,并显著提高求解效率。

关键词 SAT 问题,命题逻辑,重启,路径识别,向量空间模型

中图法分类号 TP311.1 文献标识码 A DOI 10.11896/j.issn.1002-137X.2017.11.042

Path Identification Based Delaying Restart Algorithm for CDCL SAT Solver

CHEN Qing-shan^{1,2} XU Yang² WU Guan-feng^{1,2} HE Xing-xing²

(School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China)¹

(National-Local Joint Engineering Laboratory of System Credibility Automatic Verification, Chengdu 610031, China)²

Abstract An appropriate restart is helpful for a solver to jump out of the local optimization, but more frequent restarts will significantly reduce the efficiency. To address the arbitrariness of triggering conditions for the restart of CDCL solver, the delaying restart algorithm based on path identification was proposed in this paper. Specifically, the Luby sequence is utilized to trigger the delaying restart decision, which converts current path and the searched paths to vector space models (VSMs) such that the similarity of VSMs is calculated to judge whether the current search process will get into the repetitive search space. Once the underlying similarity reaches a given threshold, the restart is triggered, otherwise it will be delayed. SAT international testing example and two state-of-the-art solvers were adopted for comparison purpose. The experimental results show that the proposed algorithm can not only effectively avoid the repetitive search space, but also obviously improve the solving efficiency.

Keywords Satisfiability problem, Propositional logic, Restart, Path identification, Vector space model

1 引言

布尔可满足性(SAT)问题是首个被证明的 NP 完全问题^[1]。工业领域中很多问题都可以被转化为 SAT 问题来求解,相应的 SAT 求解器被广泛应用于集成电路验证、模型检验、软件验证和组合优化等领域。传统的 SAT 求解器大多基于 DPLL^[2] (Davis Putnam Logemann Loveland)算法框架。在过去 20 年间,学者们在 DPLL 的基础上提出了很多启发式算法和程序优化技术,如启发式分支决策^[4]、非时序回溯^[5]、重启^[6]、周期性学习子句删除、懒散数据结构^[4]等,它们逐渐发展并形成了目前最为流行的 CDCL^[3] (Conflict-Driven Clause Learning)求解框架。

尽管如此,由于 CPU、内存等硬件条件和求解算法自身的限制,在应对工业领域中的大规模复杂问题时,即使是当今最先进的求解器也很难表现出完美的性能,其主要原因与 CDCL 求解器的搜索机制相关。CDCL 求解器的搜索过程实质是基于虚拟二叉树的深度优先搜索。在此方式下,搜索过程极易频繁地进入局部冲突和回溯状态,从而陷入局部最优,进而无法跳出。文献[8]揭示了回溯搜索机制下的运行时间服从重尾分布(Heavy-Tailed Distributions),即少部分变元的正确赋值对快速得到结论起着决定性作用,并且在搜索中进行的绝大多数回溯实际上毫无意义。在此基础上,作者提出了随机重启策略,重启指通过随机重置搜索过程,即清除之前的所有变元赋值状态,并从根节点开始选择决策变元来进行

到稿日期:2016-11-28 返修日期:2017-02-09 本文受国家自然科学基金项目(61673320, 11526171, 61305074),中央高校基本科研业务费项目(2682017ZT12)资助。

陈青山(1982—),男,博士生,主要研究方向为智能信息处理,E-mail:qschen@home.swjtu.edu.cn(通信作者);徐 扬(1956—),男,博士,教授,博士生导师,主要研究方向为自动推理;吴贯锋(1986—),男,博士生,主要研究方向为智能信息处理;何星星(1982—),男,博士,讲师,主要研究方向为自动推理。

重新搜索。实验证明,采用随机重启策略可以有效消除重尾现象并跳出局部最优状态。变元赋值状态存储^[9]和赋值序列复用^[10]的提出显著减少了重复分支决策的次数,对求解工业领域的问题(尤其是可满足问题)颇为有效。

在 SAT 问题求解中,根据重启间隔可大致分为静态重启^[11-14]和动态重启^[15-17]两类。文献[17]详细评估了静态重启和动态重启策略的效率,证明了动态重启策略总体上优于静态重启策略。然而,从本质上讲,重启是为了避免搜索过程进入重复的冲突空间,即当前搜索路径不与已搜索路径重复。但从理论上说,识别重复路径需要惊人的匹配量,这显然是无法完成的。

本文在 Luby^[13]静态重启策略的基础上,结合动态重启策略对冲突状态的评估,通过建立搜索路径匹配的向量空间模型,提出了一种基于轻量级搜索路径匹配的延迟重启策略。当搜索冲突次数达到 Luby 序列设定的阈值时,通过搜索路径匹配,判断当前搜索与已搜索路径的相似度,即进入重复搜索空间的相似度,一旦相似度达到设定条件即重启,以规避进入相同冲突空间。对比实验表明,所提算法显著加快了求解速度。

2 基本原理

2.1 SAT 问题

给定一组有限变元集合 $X = \{x_1, x_2, \dots, x_n\}$, x_i 可以被赋值为 1(真)或 0(假)。SAT 问题可表示成以合取范式(Conjunctive Normal Form, CNF)的形式给出的公式 ϕ , ϕ 由多个子句合取而成,子句 C 由文字析取而成。文字 l_i 是对应变元 x_i 或其否定 $\neg x_i$, 有 $x_i = \neg \neg x_i$ 且 x_i 和 $\neg x_i$ 不能同时为 1 或 0。如 $\phi_0 = (\neg x_1 \vee x_3) \wedge (x_1 \vee x_2) \wedge (x_2 \vee \neg x_3 \vee x_5) \wedge (\neg x_2 \vee \neg x_4 \vee \neg x_5)$, 即是一个有 5 个变元和 4 个子句的 CNF 公式。若子句中任意文字被赋值为真,则子句值为 1,或被满足;若子句中所有文字同时被赋值为 0,则称指定赋值使得子句为假,或指定赋值使得子句不被满足。公式 ϕ 是可满足的当且仅当所有子句同时被满足。

SAT 问题指寻找 X 上的所有变元赋值 $X': X \rightarrow \{0, 1\}$, 使得 ϕ 为真。若存在这样的赋值,则 ϕ 是可满足的;否则 ϕ 是不可满足的。对于公式 ϕ_0 , 存在 $X' = \{x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 0, x_5 = 0\}$, 使得 ϕ_0 的所有子句为 1, 因此 ϕ_0 是可满足的。

2.2 CDCL 框架

CDCL^[3]的搜索本质是基于完全二叉树的非时序回溯搜索。该算法在每个决策层挑选一个未赋值变元作为决策变元,并确定一个分支方向,即将决策变元赋值为 1 或 0。用 x_i 表示决策层 $d(d \geq 0, d \in \mathbb{N})$ 的决策变元, $L(x_i)$ 表示 x_i 所在的决策层,有 $L(x_i) = d$ 。利用布尔约束传播规则(Boolean Constraint Propagation, BCP)可推导出必须赋值的变元。BCP 规则可返回推导过程是否出现冲突,即某个变元的两个分支必须同时被赋值为 1 或 0。若出现冲突,则启动冲突分析和非时序回溯,并清除当前决策层到回溯层之间的所有变元赋值,从回溯层决策变元的另一分支开始搜索。若回溯层为 0,则说明冲突发生在初始搜索位置,可以直接判定原公式是不可满足的。若公式中所有变元都被赋值且无冲突,则说

明原公式是可满足的。不难看出,CDCL 搜索算法在最坏情况下拥有与变元个数相关的指数级复杂度。

2.3 重启策略

静态重启是最简单的重启策略,该方法为总冲突次数设置一个计数器,并设定重启周期,当总冲突次数达到周期阈值时即重启。设 $T = \{t_1, t_2, \dots, t_k\} (k \geq 0, k \in \mathbb{N})$ 表示重启时的总冲突次数阈值序列, $t_i (0 \leq i \leq k)$ 表示第 i 次重启时的总冲突次数阈值,则静态重启可大致分为以下 4 类。

(1) 固定间隔重启。其序列状态 T 为等差数列,即 $t_i = t_{i-1} + \Delta t (i \geq 1)$, 其中 Δt 为间隔次数。在 Chaff^[4] 中 Δt 为 700, Berkmin^[7] 为 550, Siege^[20] 为 16000。

(2) 固定几何增长因子重启。其序列状态 T 为等比数列,即 $t_i = r \cdot t_{i-1} (i \geq 1)$, 其中 r 为几何增长倍数。如 Minisat v1.14^[18] 设置 $t_i = 1.5 t_{i-1} (i \geq 1)$, Walsh^[22] 设置 $t_i = r \cdot t_{i-1} (i \geq 1, 1 < r < 2)$ 。

(3) 嵌套几何增长因子重启。PicoSAT^[21] 首次提出了这种重启控制方式,在搜索过程中设置短周期阈值 *inner* 和长周期阈值 *outer* 嵌套的计数器,初始值均为 100,用 *inner* 控制重启开关。当 $inner \geq outer$ 时, *outer* 增大 1.1 倍,并重置 *inner* 为 100;否则 *inner* 增大 1.1 倍。

(4) 叠加几何增长因子重启。Luby^[13] 序列是该类序列中的代表,是公认的解决随机问题的最佳调度策略^[23]。其序列在几何增长的同时叠加前序所有阈值,其序列状态 $T = \{1, 1, 2, 1, 1, 2, 4, 1, 1, 2, 1, 1, 2, 4, 8, \dots\}$, 可由如下公式给出:

$$\delta_i = \begin{cases} 2^{k-1}, & \text{if } i = 2^k - 1 \\ \delta_{i-2^{k-1}-1}, & \text{if } 2^{k-1} \leq i < 2^k - 1 \end{cases}$$

由于 δ_i 间隔较小,在实际应用中无意义,因此通常需要乘以一个放大系数 r , 即 $t_i = r \cdot \delta_i$ 。在 Minisat 2.2^[11] 中, r 被设置为 100。

静态重启算法存在一个明显缺陷,即无论间隔周期设置得有多大,一旦搜索过程接近得出的结论(可满足或不可满足)时就触发重启,则前面所做的搜索将被重置,求解器不得不重新开始搜索。为弥补该缺陷,学者们提出了动态重启策略,该策略研究冲突状态并引入评估算子,当冲突状态到达临界状态时重启。

动态重启策略体现了冲突原因与重启的相关性。文献[15]提出了文字块距离(Literal Block Distance, LBD)的概念,即学习子句中包含的决策层的个数。通过监控学习子句的 LBD,设置重启触发条件:1) LBD 的短期平均值大于长期平均值 25%,且上一次重启至今冲突次数大于 50;2) 最近学习子句的 LBD 太大。文献[16]为每个决策层设置 $c(d)$ 来记录对应的总冲突次数。当搜索发生冲突并回溯到决策层 d 时,计算总冲突次数 g 与 $c(d)$ 的差值 $h = g - c(d)$ 。其中, h 反映了在决策层 d 对应决策变元的冲突规模,若 h 大于某个临界值,则重启。Glucose^[17] 首先计算区间 LBD 移动平均值 SMA 和累积移动平均值 CMA,然后比较区间移动平均与累积移动平均的偏离程度,若 $SMA > c \cdot CMA (c > 1)$, 则重启。文献[12]提出了一种基于敏捷度的延迟重启策略,即通过动态观察搜索过程的敏捷度来决定是否需要重启。若大部分变元已保存的极性都被翻转过,即很多变元的两个分支均被搜

索,说明很快得出不可满足结论的可能性在增大,则暂缓重启;若仅有少部分变元的极性被翻转,说明搜索过程可能已陷入局部最优而无法跳出,则重启。文献[24]在 Glucose 的基础上提出了一种基于平均决策高度的延迟重启策略,若平均决策高度小于(不超过 70%)平均 LBD,则跳过重启。

3 重复搜索路径识别

重复搜索路径识别指当前搜索出现冲突并达到 Luby 序列设定的重启阈值时,将当前搜索路径与已搜索路径进行匹配,若当前搜索路径与某条已搜索路径达到一定的相似度,则停止当前搜索(即不再在本条路径上回溯),并调用重启程序,开始新的搜索。

定义 1(搜索路径) 设 ϕ 是 CNF 公式, p_i 是 ϕ 的对应搜索树中的任意节点(即决策变元及其分支方向),则从根节点到任意叶节点之间的决策分支构成一条搜索路径 $r = \{p_1, p_2, \dots, p_n\}$, 其中 n 为 ϕ 中的变元个数。

定义 2(搜索路径集合) 设 ϕ 是 CNF 公式, r_k 是 ϕ 的一条搜索路径,则求解 ϕ 的所有可能搜索路径可表示为 $R = \{r_1, r_2, \dots, r_k\} (1 \leq k \leq 2^n)$, 其中 n 为 ϕ 中的变元个数。

定义 3(路径相似度) 设 ϕ 是 CNF 公式, R 是 ϕ 的可能搜索路径集合,对于 $\forall r_1 \in R, \forall r_2 \in R$, r_1 和 r_2 的路径相似度可表示为 $S(r_1, r_2)$ 。

3.1 精确路径匹配

精确路径匹配是较简单的一种匹配方法,即逐一比较两条路径上对应的决策变元,一旦出现不一致即停止比较,并返回相似度。图 1 给出了两条搜索路径,其中, r_1 为当前搜索路径, r_2 为待匹配的已搜索路径,每一个决策层的第一个变元为决策变元,决策变元及其推导变元所在决策层用 *level* 表示。

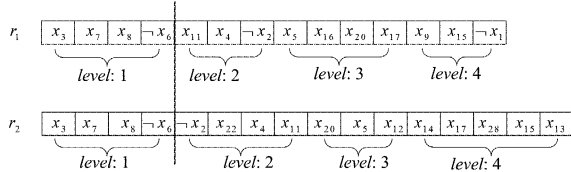


图 1 搜索路径示例

用 $H(r)$ 表示路径 r 的长度,有 $H(r_1) = 14, H(r_2) = 16$; 用 $C(r_1, r_2)$ 表示 r_1 和 r_2 中连续相同的决策变元个数, r_1 和 r_2 在第一决策层的决策分支一致,因此 $C(r_1, r_2) = 4$ 。 r_1 与 r_2 的精确相似度为:

$$S(r_1, r_2) = \frac{C(r_1, r_2)}{H(r_1)} = \frac{4}{14} \approx 0.29$$

但在大多数情况下,重启后的搜索路径从根节点处即选择了不同的决策变元,相应的路径相似度为 0。因此,在实际应用中这种匹配模式的意义不大。

3.2 基于向量空间模型的路径匹配

回顾图 1 所示的搜索路径,虽然 r_1 与 r_2 的精确匹配度不高,但在相同的决策层存在重合的决策分支,因此实质上它们是具有关联性的。

定理 1 设 Φ 为 CNF 公式 ϕ 的搜索空间,存在搜索路径 $r_1 = \{x_1, x_2, x_3\}, r_2 = \{x_3, x_2, x_1\}$, 分别对应搜索空间 $\Phi|_{r_1}$ 和 $\Phi|_{r_2}$, 假定 Φ_1 和 Φ_2 无冲突,则 Φ_1 和 Φ_2 是等价的,记为 $\Phi_1 \Leftrightarrow \Phi_2$ 。

证明:由 CDCL 决策过程可知,搜索路径 r_1 对应的决策分支为 $x_1 = 1 \wedge x_2 = 1 \wedge x_3 = 1$, 等价于 $x_3 = 1 \wedge x_2 = 1 \wedge x_1 = 1$, 即等价于 r_2 对应的决策分支。假定 Φ_1 和 Φ_2 无冲突,则说明 r_1 和 r_2 为有效赋值路径,有 $\Phi|_{x_1=1 \wedge x_2=1 \wedge x_3=1} \Leftrightarrow \Phi|_{x_3=1 \wedge x_2=1 \wedge x_1=1}$, 即 $\Phi_1 \Leftrightarrow \Phi_2$ 。

定理 1 体现了决策分支的无序性,因此可建立向量空间模型^[19] (Vector Space Model, VSM) 来测算 r_1 与 r_2 的相似度。VSM 是一种用于评估文本信息相似度的代数模型,因其简洁、高效而被广泛应用于文本信息检索中。在 VSM 中,文档 D 由互不相关的特征项(即词条) $T_i (1 \leq i \leq n)$ 构成,其中 n 为 D 中的特征项总数,每个特征项 T_i 按照重要程度被赋予一定权重 W_i , 有 $\sum_{i=1}^n W_i = 100$ 。文档 D 可表示为 n 维向量 $D(T_1, W_1; T_2, W_2; \dots; T_n, W_n)$ 。给定两个文档 $A(T_{A1}, W_{A1}; T_{A2}, W_{A2}; \dots; T_{An}, W_{An})$ 和 $B(T_{B1}, W_{B1}; T_{B2}, W_{B2}; \dots; T_{Bn}, W_{Bn})$, T_{Ai} 必须与 T_{Bi} 逐一对应,即 $T_{Ai} = T_{Bi} (1 \leq i \leq n)$, 则 A 和 B 的内容相似度可表示为 A 和 B 空间夹角 θ 的余弦值:

$$S(A, B) = \cos(\theta) = \frac{\sum_{k=1}^n W_{Ak} W_{Bk}}{\sqrt{\sum_{k=1}^n W_{Ak}^2} \cdot \sqrt{\sum_{k=1}^n W_{Bk}^2}} \quad (1)$$

CDCL 搜索路径可转化为向量空间,由定理 1 可知,搜索路径 r_1 与 r_2 中包含的决策变元的并集构成向量空间特征项,即 $T = r_1 \cup r_2$ 。对于 $\forall p_i \in T_{r_1, r_2}$, 其权重由 p_i 在搜索路径中对应的决策层决定。从深度优先搜索过程来看,搜索过程越深入,剩余搜索子空间越小,即越接近根部的决策分支,对搜索空间的剪枝能力越强。设 r_1 和 r_2 对应的向量空间分别为 R_1 和 R_2 , 则 p_i 在 R_1 中的权重 $W_{R_1 i}$ 可由下式给出:

$$W_{R_1 i} = \frac{n - L(p_i) + 1}{C(p_i) \cdot \sum_{k=1}^n k} * 100 = \frac{200(n - L(p_i) + 1)}{C(p_i) \cdot (n^2 + n)} \quad (2)$$

其中, $L(p_i)$ 为决策变元 p_i 在 r_1 中所属的决策层, n 为 r_1 中的总决策层数, $C(p_i)$ 为 r_1 中与 p_i 处于同一决策层的变元个数。因此,搜索路径 r_1 与 r_2 的相似度为:

$$S(r_1, r_2) = \frac{\sum_{k=1}^n W_{R_1 k} W_{R_2 k}}{\sqrt{\sum_{k=1}^n W_{R_1 k}^2} \cdot \sqrt{\sum_{k=1}^n W_{R_2 k}^2}} \quad (3)$$

由 $W_{R_1 i}$ 的定义可知,同一搜索路径中具有相同决策层的变元的权重相同。

对于图 1 所示搜索路径 r_1 和 r_2 , 其特征项 T 为 r_1 和 r_2 中所有出现的决策变元,即:

$$T(\neg x_1, \neg x_2, x_3, x_4, x_5, \neg x_6, x_7, x_8, x_9, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{20}, x_{22}, x_{28})$$

由式(3)可知, r_1 中第一决策层中变元 x_3 对应的权重为:

$$W_{R_1 3} = \frac{200(n - L(p_i) + 1)}{c(n^2 + n)} = \frac{200(4 - 1 + 1)}{4(4^2 + 4)} = 10$$

以此类推, r_1 的空间向量为:

$$R_1(\neg x_1, 3.3; \neg x_2, 10; x_3, 10; x_4, 10; x_5, 5; \neg x_6, 10; x_7, 10; x_8, 10; x_9, 3.3; x_{11}, 10; x_{12}, 0; x_{13}, 0; x_{14}, 0; x_{15}, 3.3; x_{16}, 5; x_{17}, 5; x_{20}, 5; x_{22}, 0; x_{28}, 0)$$

r_2 的空间向量为:

$R_2 (\neg x_1, 0; \neg x_2, 7.5; x_3, 10; x_4, 7.5; x_5, 6.6; \neg x_6, 10;$
 $x_7, 10; x_8, 10; x_9, 0; x_{11}, 7.5; x_{12}, 6.6; x_{13}, 2; x_{14}, 2;$
 $x_{15}, 2; x_{16}, 0; x_{17}, 2; x_{20}, 6.6; x_{22}, 7.5; x_{28}, 2)$

r_1 与 r_2 的相似度为:

$$S(r_1, r_2) = \frac{\sum_{k=1}^{19} W_{R_1 k} W_{R_2 k}}{\sqrt{\sum_{k=1}^{19} W_{R_1 k}^2} \cdot \sqrt{\sum_{k=1}^{19} W_{R_2 k}^2}}$$

$$= \frac{707.6}{28.86 \times 27.85} \approx 0.88$$

计算结果表明, r_1 与 r_2 具有高度相似性, 若 r_1 继续在本路径上回溯搜索, 则可能会进入与 r_2 相同的搜索空间。如果此时重启搜索过程, 则在很大程度上可避免重复搜索, 从而加快求解速度。

4 实验结果

本文在 Minisat 和 MapleCOMSPS 求解器的基础上分别实现了基于路径识别的延迟重启策略, 相应的改进版分别称为 Minisat_PI 和 MapleCOMSPS_PI。Minisat 是国际上知名的 CDCL SAT 求解器, 近几年 SAT 国际竞赛中很多具有优异表现的求解器都是 Minisat 的改进版, 且 Minisat 中未集成更多的优化技术, 在对比测试中容易看出改进算法的性能差异。MapleCOMSPS 是 2016 年 SAT 国际竞赛 Main Track 组的冠军, 与之对比能够更好地体现本文算法的能力。实验测试实例来自于 2015 年 SAT-Race 的 main track 组 (300 个实

例) 和 2016 年 SAT Competition 的 main track (Application) 组 (300 个实例), 涉及密码验证、多机器人路径规划等领域。每个测试实例无预处理且限时 2000s。硬件环境为 Intel i7-6700 3.4GHz CPU, 16G 物理内存, Red Hat Enterprise 7.3 操作系统。

动态重启间隔阈值采用 Luby 序列, 放大系数为 100, 即 $T = \{100, 100, 200, 100, 100, 200, 400, 100, 100, 200, 100, 100, 200, 400, 800, \dots\}$ 。为减少路径匹配量, 实验中设置当前搜索路径 r_1 与已搜索路径 r_2 的长度比大于 0.8 时才开始匹配, 即 $\frac{\min(H(r_1), H(r_2))}{\max(H(r_1), H(r_2))} \geq 0.8$, 否则继续选择下一条已搜索路径进行匹配。为比较不同相似度阈值对求解效率的影响, 分别设置了 r_1 与 r_2 的相似度阈值 $S(r_1, r_2) = 0.5 + 0.05i (i \in \mathbb{N}, 0 \leq i \leq 9)$, 对应 Minisat_PI 和 MapleCOMSPS 的不同版本。

对比测试结果分别如表 1 和表 2 所列, Minisat_PI 和 MapleCOMSPS_PI 均能正常运行, 且求解结果正确。对于 2015 年的竞赛例, Minisat_PI 的 10 个改进版中有 8 个改进版的求解总数优于 Minisat 原版, 尤其是对不可满足实例, 10 个改进版本的求解个数均不少于原版。MapleCOMSPS 的 10 个改进版中, 有 7 个改进版优于原版, 对于不可满足实例, 有 8 个版本的求解个数不少于原版。对于 2016 年的竞赛例, Minisat 的 10 个改进版中有 9 个版本的求解实例数不少于原版; MapleCOMSPS 的 10 个改进版中有 7 个版本的求解实例数不少于原版。

表 1 Minisat 嵌入路径识别延迟重启策略前后求解个数的对比

测试集	类别	Minisat	Minisat_PI(使用不同相似度阈值)									
			0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
SAT Race 2015 Main Track	SAT	111	120	111	105	119	117	112	115	110	120	106
	UNSAT	59	65	61	62	63	61	61	60	61	65	59
	TOTAL	170	185	172	167	182	178	173	175	171	185	165
SAT Competition 2016 Application of Main Track	SAT	49	59	59	57	57	57	56	54	58	55	53
	UNSAT	58	58	58	60	58	57	59	59	59	56	52
	TOTAL	107	117	117	117	115	114	115	113	117	111	105

表 2 MapleCOMSPS 嵌入路径识别延迟重启策略前后求解效率的对比

测试集	类别	MapleCOMSPS	MapleCOMSPS_PI(使用不同相似度阈值)									
			0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
SAT Race 2015 Main Track	SAT	147	142	143	149	148	152	152	151	150	149	136
	UNSAT	75	75	74	76	75	75	78	76	75	78	71
	TOTAL	222	217	217	225	223	227	230	227	225	227	207
SAT Competition 2016 Application of Main Track	SAT	63	61	61	63	58	64	64	64	63	64	62
	UNSAT	68	63	65	68	75	68	71	67	70	70	66
	TOTAL	131	124	126	131	133	132	135	131	133	134	128

对比测试结果表明, 基于路径识别的延迟重启策略在规避重复搜索空间中发挥了作用, 减少了重复冲突。但从表中也可看出, 相似度越高并不代表可求解的实例越多, 这也说明设置过高的延迟重启条件也可能导致搜索陷入局部最优, 从而无法跳出。在实际应用中, 建议路径相似度 $S(r_1, r_2)$ 取 $[0.65, 0.90]$, 可得到较好的优化性能。

结束语 本文对 SAT 问题求解中的重启技术进行了深入分析, 指出了现有重启策略的不足; 对于静态重启方式, 无论间隔周期设置得有多大, 一旦求解器在趋近于得出的结论时便会触发重启过程, 则已有的搜索将前功尽弃。动态重启虽

不考虑间隔周期, 但反复评估当前搜索状态, 容易陷入局部最优。本文引入信息检索中的向量空间模型相似度刻画方法, 在 Luby 静态重启策略的基础上, 结合动态重启策略对冲突状态的评估, 提出了一种基于轻量级搜索路径匹配的延迟重启策略。实验结果表明, 延迟重启策略可有效规避重复搜索空间, 并显著提高求解效率。

参考文献

- [1] COOK S A. The complexity of theorem-proving procedures [C] // Proceedings of 3rd Annual ACM Symposium on Theory of

- Computing. New York: ACM, 1971: 151-158.
- [2] DAVIS M, LOGEMANN G, LOVELAND D. A machine program for theorem proving[J]. Communications of the ACM, 1962, 5(7): 394-397.
 - [3] SILVA J P M, LYNCE I, MALIK S. Conflict-driven clause learning SAT solvers [M]// Handbook of Satisfiability. Amsterdam, IOS Press, 2009: 127-149.
 - [4] MOSKEWICZ M W, MADIGAN C F, ZHAO Y. Chaff: Engineering an efficient SAT solver [C]// Proceedings of the 38th Annual Design Automation Conference. New York: ACM, 2001: 530-535.
 - [5] MARQUES-SILVA J P, SAKALLAH K A. Grasp: a new search algorithm for satisfiability [C]// Proceedings of the 1996 IEEE/ACM International Conference on Computer-aided Design. Los Alamitos: IEEE Computer Society Press, 1996: 220-227.
 - [6] GOMES C P, SELMAN B, KAUTZ H. Boosting combinatorial search through randomization[C]// Proceedings of AAAI'98. 1998: 431-437.
 - [7] GOLDBERG E, NOVIKOV Y. BerkMin: a fast and robust SAT solver [C]// Proceedings of the symposium on Design, Automation and Test in Europe Conference. 2002: 142-149.
 - [8] GOMES C, SELMAN B, CRATAO N, et al. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems [J]. Journal of Automated Reasoning, 2000, 24(1/2): 67-100.
 - [9] PIPATSRISAWAT P, DARWICHE A. A lightweight component caching scheme for satisfiability solvers [C]// Proceedings of the International Conference on Theory and Applications of Satisfiability Testing-SAT 2007. Heidelberg: Springer, 2007: 294-299.
 - [10] VANDER TAK P, RAMOS A, HEULE M. Reusing the assignment trail in CDCL solvers [J]. Journal on Satisfiability, Boolean Modeling and Computation, 2011, 7(4): 133-138.
 - [11] EÉN N, SÖRENSON N. An extensible SAT solver [C]// Proceedings of the International Conference on Theory and Applications of Satisfiability Testing-SAT 2003. Heidelberg: Springer, 2004: 502-518.
 - [12] BIERE A. Adaptive restart strategies for conflict driven SAT solvers [C]// Proceedings of the International Conference on Theory and Applications of Satisfiability Testing-SAT 2008. Heidelberg: Springer, 2008: 28-33.
 - [13] LUBY M, SINCLAIR A, ZUCKERMAN D. Optimal speedup of las vegas algorithms [J]. Information Processing Letters, 1993, 47(4): 173-180.
 - [14] HUANG J B. The effect of restarts on the efficiency of clause learning [C]// Proceedings of the 20th International Joint Conference on Artificial Intelligence. Menlo Park: AAAI, 2007: 2318-2323.
 - [15] AUDEMARD G, SIMON L. Refining restarts strategies for SAT and UNSAT [C]// Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming. Heidelberg: Springer, 2012: 118-126.
 - [16] RYVCHIN V, STRICHMAN O. Local restarts [C]// Proceedings of the International Conference on Theory and Applications of Satisfiability Testing-SAT 2008. Heidelberg: Springer, 2008: 271-276.
 - [17] BIERE A, FRÖHLICH A. Evaluating CDCL Restart Schemes [C]// Proceedings of the International Workshop on Pragmatics of SAT. 2015.
 - [18] EÉN N, SÖRENSON N. MiniSat: A SAT solver with conflict-clause minimization [C]// SAT 2005 Competition. 2005.
 - [19] SALTON G, WONG A, YANG C S. A vector space model for automatic indexing [J]. Communications of the ACM, 1975, 18(11): 613-620.
 - [20] RYAN L. Efficient algorithms for clause learning SAT solvers [D]. Simon Fraser University, 2004.
 - [21] BIERE A. PicoSAT essentials [J]. Journal on Satisfiability Boolean Modeling and Computation, 2008, 4(2-4): 75-97.
 - [22] WALSH T. Search in a small world [C]// Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence. Stockholm, 1999: 1172-1177.
 - [23] BIERE A, HEULE M, MAAREN H V, et al. Handbook of Satisfiability [M]. IOS Press. 2009.
 - [24] BIERE A. Yet another local search solver and lingeling and friends entering the sat competition [C]// Proceedings of SAT Competition 2014. 2014: 39-40.

(上接第 272 页)

- 孟令奎, 夏辉宇, 陈子丹, 等. 基于 P2P 的海量水利遥感影像优化传输策略 [J]. 计算机应用研究, 2011, 28(4): 1382-1384.
- [7] XU S M, FENG J, TANG Z X. Research on remote sensing data distribution strategy based on P2P technology [J]. Computer and Digital Engineering, 2012, 40(10): 47-50. (in Chinese)
 - 胥世民, 冯钧, 唐志贤. 基于 P2P 技术的遥感数据分发策略研究 [J]. 计算机与数字工程, 2012, 40(10): 47-50.
 - [8] LV Z H, YIN T F, ZHANG X L, et al. Virtual Reality Smart City Based on WebVRGIS [J]. IEEE Internet of Things Journal, 2016, 3(6): 1015-1024.
 - [9] LIU B, CUI Y, LU Y S. Locality-awareness in BitTorrent-like P2P applications [J]. IEEE Transactions on Multimedia, 2009, 11(3): 361-371.
 - [10] DAI Q, LIU J B, LIU S B. Key technologies of mass remote sensing data sharing [J]. Computer Engineering, 2008, 34(6): 283-285. (in Chinese)
 - 戴芹, 刘建波, 刘士彬. 海量卫星遥感数据共享的关键技术 [J]. 计算机工程, 2008, 34(6): 283-285.
 - [11] 聂荣. BitTorrent 类型对等网络的位置知晓性研究 [M]. 北京: 中国科学技术出版社, 2014: 120-121.
 - [12] CHEN J. Tracking system, tracking server and tracker; WO 2013044539 A1 [P]. 2013.
 - [13] P2P Protocol Specifications Multitracker [EB/OL]. <http://wiki.depthstrike.com/wiki/P2P:Protocol:Specifications;Multitracker>.