

求解 MinSAT 问题的加强式格局 检测与子句加权算法

周俊萍 任雪亮 殷 茜 李睿智 殷明浩

(东北师范大学计算机科学与信息技术学院 长春 130117)

摘 要 MaxSAT 问题的研究已成为一个比较热门的研究领域,与 MaxSAT 问题相对的是 MinSAT 问题. MinSAT 是 SAT 问题的另一种优化形式,与 MaxSAT 问题不同的是 MinSAT 问题需要找到一组赋值使得可满足的子句数目最少. 在求解某些组合优化问题时,将其转化为 MinSAT 问题比转化为 MaxSAT 问题有着更快的速度,因此该文提出了一种求解 MinSAT 问题的加强式格局检测与子句加权局部搜索算法. 该算法在随机行走算法的基础上融入了子句加权策略,并根据 MinSAT 问题本身的特征对格局检测策略进行了加强. 加强式格局检测策略通过考查 MinSAT 问题中变量的环境信息可以减少局部搜索中的循环问题,以此提高局部搜索算法的性能. 加强式格局检测策略是格局检测的一种加强. 其对格局检测策略的加强主要体现在:当 MinSAT 问题的环境信息(即格局)发生变化时,仅该变量发生变化的赋值不一定能够作为候选解. 只有当格局发生了变化并且格局的变化使得当前考查的子句由不可满足变为可满足时,仅该变量发生变化的赋值(翻转该变量的赋值)才可以作为候选解并向该候选进行解搜索. 在局部搜索中,选择合适的变量翻转对提高算法的效率具有重要的意义. 如果没有加强式格局检测,对于一般的局部搜索,在变量翻转的时候选择启发值一般是最高的. 换言之当格局没有发生变化的时候也允许翻转,这就很可能直接导致上次刚刚翻转的变量又被翻转回来. 通过限制只允许格局发生变化且格局的变化使得当前考查的子句由可满足变为不可满足的变量进行翻转,这就会避免上述情况,因此加强格局检测策略在整个搜索过程中都具有重要的意义. 子句加权策略通过增加局部最优的代价,使得算法可以进一步找到隐藏在局部最优附近的更好的解,避免了在搜索过程中陷入局部最优. 子句加权策略应用在 MinSAT 问题的基本思想是:对公式 F 中的每个子句都赋予一个权值 1, MinSAT 问题求解算法在搜索过程中一旦陷入局部最优就增加在当前解下可满足子句的权值,这使得算法能跳出局部最优并向更好方向搜索. 实验结果表明,该算法可以有效求解 MinSAT 问题,并且在大规模实例中具有良好的表现,这也一定程度上说明了加强式格局检测策略和子句加权策略的有效性.

关键词 最小可满足问题;局部搜索算法;子句加权;格局检测;加强式格局检测

中图法分类号 TP319 **DOI 号** 10.11897/SP.J.1016.2018.00745

Algorithm of Strengthened Configuration Checking and Clause Weighting for Solving the Minimum Satisfiability Problem

ZHOU Jun-Ping REN Xue-Liang YIN Qian LI Rui-Zhi YIN Ming-Hao

(College of Computer Science and Information Technology, Northeast Normal University, Changchun 130117)

Abstract The maximum satisfiability problem (MaxSAT) has become a popular research field, which has a relative problem—minimum satisfiability problem (MinSAT). MinSAT is another form of satisfiability problem. Unlike MaxSAT problem, MinSAT problems need to find an assignment that can satisfy the least number of clauses. When solving some combinatorial optimization

收稿日期:2015-07-16;在线出版日期:2016-11-23. 本课题得到国家自然科学基金(61370156,61403076,61403077,61300099)、高等学校博士学科点专项科研基金(20120043120017)、新世纪优秀人才支持计划(NCET-13-0724)、吉林省大型科学仪器装备共享共用专项项目(20150623024TC-03)、吉林省青年科研基金项目(20160520104JH)资助. 周俊萍,女,1981年生,博士,讲师,主要研究方向为智能规划与自动推理. E-mail: zhoujp877@nenu.edu.cn. 任雪亮,男,1989年生,硕士,主要研究方向为算法设计与分析. 殷茜,女,1990年生,硕士,主要研究方向为智能规划与自动推理. 李睿智(通信作者),女,1989年生,博士,主要研究方向为算法设计与分析. E-mail: lirz111@nenu.edu.cn. 殷明浩(通信作者),男,1979年生,博士,教授,博士生导师,主要研究领域为自动推理和智能规划. E-mail: mhyin@nenu.edu.cn.

problems, converting them to MinSAT is faster than converting them to MaxSAT. Therefore, we propose a local search algorithm with strengthened configuration checking and clause weighting to solve MinSAT. Based on the random walk local search, the algorithm incorporates the clause weighting strategy and strengthens the configuration checking strategy according to the characteristic of MinSAT. The strengthened configuration checking strategy can reduce the loops by checking the variable environmental information of MinSAT so that the performance of local search algorithm can be improved. The strengthened configuration checking strategy is obtained by strengthening the configuration checking. The strengthening of the configuration checking is mainly embodied in the following aspects. When the environmental information of MinSAT is changed, the assignment with only changing the variable is not able to be as a candidate solution. Only when the environmental information of MinSAT is changed and the changing environmental information could make the current clause satisfied, which is not satisfied before, the assignment with only flipping the variable is able to be as a candidate solution. In the local search algorithm, selecting the appropriate variable to flip plays an important role for improving the efficiency of the algorithm. For the general local search algorithm, if the strengthened configuration checking strategy is not used, the variable with highest heuristic value is usually selected. In other words, the flipping is allowed when the configuration is not changed, which always results in the fact that the last flipping variable is flipped again. The circumstance can be avoided by restricting the variable when to be flipped. The strengthened configuration checking strategy sets a rule restricting the variable when to be flipped. The variable is allowed to be flipped when the configuration of the variable is changed and changing the configuration could make the current clause satisfied, which is not satisfied before. Thus the strengthened configuration checking strategy has important significance in the process of the local search. The clause weighting strategy can make us find a better solution near the local optima through increasing the cost of the local optima and avoid the search being trapped in the local optimum. The basic idea of the clause weighting strategy used in MinSAT is as follow. The weight of each clause in the formula F is initialized as 1. When the algorithm solving MinSAT problem falls into the local optimum, the weight of the satisfied clause will be increased by 1. In this manner, the search will proceed in the right direction and jump out of the local minimum. The experimental results show that the strengthened configuration checking and clause weighting local search algorithm can solve MinSAT effectively, and it has good performance in the large-scale instances. To a certain extent, the results show that the effectiveness of strengthened configuration checking strategy and clause weighting strategy.

Keywords MinSAT; local search algorithm; clause weighting; configuration checking; strengthened configuration checking

1 引 言

可满足性问题(Satisfiability Problem, SAT)是计算机科学和人工智能领域的重要问题,它在电路设计与调试、生物信息学、统计物理、智能规划与调度等众多研究领域都得到广泛的应用^[1-4].近年来, SAT问题的求解技术取得了较大的突破, Zchaff^[5]、Survey Propagation^[6]等高效的 SAT问题求解系统

已经可以求解含有 10^5 个以上变量 10^6 个以上子句规模的 SAT问题实例. SAT问题求解技术的成功也使得其它研究领域受益匪浅,例如在经典智能规划研究领域, SATPLAN^[7]和 BLACKBOX^[8]等经典智能规划系统通过将规划问题编译为 SAT问题,然后调用高效的 SAT问题求解系统求解智能规划问题,也在近年来的国际智能规划竞赛中屡次取得优异的成绩,并获得了多项冠军^[9-10].作为第 1 个被证明为 NP 完全的问题^[11], SAT问题只判断给定的命题公

式是否可满足^[12], 而很多实际问题更多的时候需要一个最优解, 不是仅仅“是”或“否”可以判断. 因此, 近年来, 很多学者开始研究 SAT 问题的优化问题, 如 MaxSAT 问题、MinSAT 问题、QBF 等^[13-15].

最大可满足性问题 (Maximum Satisfiability Problem, 简称 MaxSAT 问题) 是 SAT 问题的一种重要的优化形式, 该问题的目标是找到一组赋值使得可满足的子句数目最多^[16]. MaxSAT 问题在生物信息学、统计物理、智能规划与调度、组合拍卖和概率推理等领域有着广泛的应用. 图论中的最大割问题和最大团问题也可以转化为 MaxSAT 问题进行求解^[17-18]. 目前求解 MaxSAT 问题的算法主要分为精确算法和局部搜索算法两类. 精确算法可以保证得到最优解, 局部搜索算法往往只能得到近似解, 但是由于局部搜索算法可以快速地找到一个次优解, 因此对于大规模的问题实例, 局部搜索算法有其重要的实际意义. 精确算法的主要代表是分支定界算法, 当前大多数 MaxSAT 问题求解器都是基于这一算法, 如 UP、Toolbar、maxsatz、MiniMaxSAT、Lazy、PMS 和 maxsatzl-4icss 等^[19-23]. 经典的局部搜索算法有 GSAT、WSAT、TSAT、NSAT 等^[24-27].

与最大可满足性问题相对的是最小可满足性问题 (Minimum Satisfiability Problem, MinSAT), 它是 SAT 问题的另一种优化形式. 与 MaxSAT 问题不同的是 MinSAT 问题需要找到一组赋值使得可满足的子句数目最少^[28]. 法国亚眠大学教授 Li Chumin 指出 MinSAT 问题具有重要的研究意义: 首先, 很多实际问题可以转化为 MinSAT 问题进行求解, 并且效果更佳; 其次, MinSAT 问题的精确算法有一个更好的下界. 因此, 近年来 MinSAT 问题的研究受到广大研究学者的关注. 1994 年, Kohli 等人^[29]从理论上给出了第 1 个贪心算法求解近似 MinSAT 问题并证明了该算法是 N 近似算法. 1996 年, Marathe 等人^[30]通过将 MinSAT 问题近似规约到 MinVC 问题, 并结合现有的求解 MinVC 问题的启发式算法从理论上提出了一种 MinSAT 问题的近似算法, 并证明该算法是 2 近似算法. 2005 年, Avidor 和 Zwick 给出了一种求解 Min2-SAT 问题和 Min3-SAT 问题的近似求解方法——半定规划^[31]. 2010 年, Zhu 等人^[32]给出了一种将 MinSAT 问题转换成 MaxSAT 问题编码的方法, 从而可以通过求解相应的 MaxSAT 问题来求解 MinSAT 问题; 2012 年, Li 等人^[33]进一步给出了一种基于 DPLL 过程的 MinSAT 问题精确求解器 MinSatz, 该求解器通过将合取范式

(Conjunctive Normal Form, CNF) 转化成无向图, 并且计算其最大团来获取 MinSAT 问题的启发式下界.

需要指出的是, 现有的 MinSAT 问题的精确求解器只能求解变量规模在 100 左右的 MinSAT 问题实例, 求解的实例规模有限. 现有的局部搜索算法中, 文献[29]采用的是贪心算法求解 MinSAT 问题, 其在求解问题时总是做出在当前看来最好的选择, 而不从整体最优上加以考虑, 因此算法容易陷入局部最优; 文献[30]和文献[32]采用的方法都是将 MinSAT 问题转换为其它问题, 再利用其它问题的求解方法进行解决, 其求解方法未结合 MinSAT 问题本身的信息, 因此其求解效率有限; 文献[31]给出的求解方法只能求解子句中最多包含 2 个和 3 个文字 (literals) 的 MinSAT 问题实例, 其求解范围有限. 基于此, 本文提出了加强式格局检测与子句加权局部搜索算法求解近似 MinSAT 问题. 该算法在随机行走算法的基础上融入了子句加权策略, 并根据 MinSAT 问题本身的特征对格局检测策略进行了加强. 子句加权策略通过增加局部最优的代价, 使得算法可以进一步找到隐藏在局部最优附近的更好的解, 避免了在搜索过程中陷入局部最优; 加强式格局检测策略通过考查 MinSAT 问题中变量的环境信息 (即变量的格局), 减少了局部搜索中的循环问题, 以此提高局部搜索算法的性能. 实验结果表明, 该算法在大规模实例中具有良好的表现, 这也在一定程度上说明了加强式格局检测策略和子句加权策略的有效性.

本文第 2 节给出本文用到的相关概念; 第 3 节介绍加强式格局检测与子句加权局部搜索算法; 第 4 节给出完备搜索算法、随机行走局部搜索算法、子句加权局部搜索算法、加强式格局检测与子句加权局部搜索算法的对比实验; 第 5 节为本文的结论.

2 相关概念

首先对文中出现的符号做出约定: v_1, v_2, \dots 表示布尔变量; l_1, l_2, \dots 表示文字 (literals); c_1, c_2, \dots 表示子句; F_1, F_2, \dots 表示公式; s, s' 表示解.

定义 1 (解的邻域). 设 $V = \{v_1, v_2, \dots, v_n\}$ 是一个布尔变量 (简称为变量) 集合, 定义在 V 上的解是一个函数 $\Phi: V \rightarrow \{\text{true}, \text{false}\}$. 每个解可以用一个 n 元布尔向量表示, 那么在 V 上存在 2^n 个不同的解, 所有的解构成了局部搜索算法的解空间. 解 s 的邻

域是只改变候选解 s 中任意一个布尔变量真值所形成解的集合。

定义 2(文字(literals)). 布尔变量及其否定统称为文字(literals). 设 v 是一个布尔变量, 则称 v 和 $\neg v$ 是文字(literals). 当布尔变量 v 的真值为 true 时, 文字(literals) v 的赋值为 true, 文字(literals) $\neg v$ 的赋值为 false; 反之, 当布尔变量 v 的真值为 false 时, 文字(literals) v 的赋值为 false, 文字(literals) $\neg v$ 的赋值为 true^[34].

定义 3(子句). 子句 $c = l_1 \vee l_2 \vee \dots \vee l_k$, 其中 l_1, l_2, \dots, l_k 是文字(literals). 子句 c 可满足当且仅当在解 s 下至少有一个文字(literals) $l_i (1 \leq i \leq k)$ 赋值为 true.

定义 4(CNF 范式). CNF 范式 $F = c_1 \wedge c_2 \wedge \dots \wedge c_i$, 其中 c_1, c_2, \dots, c_i 是子句^[35].

定义 5(SAT 问题). SAT 问题是指给定一个 CNF 范式 F , 判断 F 是否存在一个解 s 使得其真值为真, 或者证明它永为假. 如果存在一个解 s 使得 F 的真值为真, 则称公式 F 是可满足的, 否则是不可满足的^[36].

定义 6(MinSAT 问题). MinSAT 问题是指给定一个 CNF 范式 F , 寻找 F 的一个解 s^* , 使其满足 $f(s^*) \leq f(s)$, 其中 s 表示解空间的任意一个解, 函数 $f(s)$ 表示在解 s 下 CNF 范式 F 中可满足的子句数目^[37-38].

定义 7(邻居变量). 给定一个 CNF 范式 F , 变量 v 的邻居变量 v' 是与变量 v 存在于 F 中同一个子句 c 的变量, 即 $N_F(v) = \{v' \mid (v \in c \vee \neg v \in c) \wedge (v' \in c \vee \neg v' \in c) \wedge c \in F\}$, 其中 $N_F(v)$ 称为变量 v 的邻居变量集合.

定义 8(格局). 给定一个 CNF 范式 F 和 F 的一个解 s , 设 F 中变量 v 有 k 个邻居变量, 变量 v 的格局是变量 v 的所有邻居变量的真值, 它可以用一个 k 维向量 $\alpha(v) = \{\alpha_1, \alpha_2, \dots, \alpha_k\}$ 表示, 其中 α_i 是在解 s 下变量 v 的第 i 个邻居变量的真值^[30].

3 求解 MinSAT 问题的加强式格局检测与子句加权局部搜索算法

本节给出了加强式格局检测与子句加权局部搜索算法近似求解 MinSAT 问题. 算法在随机行走的框架基础上融入了子句加权策略, 并根据 MinSAT 问题本身的特征在格局检测策略基础上进行了加强. 它从一个解开始搜索, 每一步根据子句加权策略

和加强式格局检测策略获得的信息从当前解移动到其邻域中的一个解, 以寻找一个更好的解. 对于现有的 MinSAT 问题的局部搜索算法来说, 因为其在搜索过程中只能保留当前局部信息, 因此经常会重复地移动到最近已经移动过的解, 使得算法陷入局部最优, 即 $f(s) \leq f(s')$ (解 s' 是解 s 的邻域中局部搜索算法已经移动过的一个解, 函数 $f(s)$ 是在解 s 下 CNF 范式 F 中可满足的子句数目), 这不仅增加了算法的执行时间, 也影响了算法的性能, 因此, 减少局部搜索中的循环问题对提高算法的效率具有重要的作用. 本算法所采用的加强式格局检测策略就可以减少局部搜索中的循环问题; 子句加权策略通过增加局部最优的代价, 使得算法可以进一步找到隐藏在局部最优附近的更好的解. 接下来将分别介绍这两个策略.

3.1 加强式格局检测策略

局部搜索算法在搜索中常重复地访问一些解, 即会出现循环现象, 这不仅浪费了时间, 也降低了算法的性能. 另一方面, 如果将之前访问过的解都记录下来, 虽然可以避免循环现象的发生, 但这不仅需要指数级的存储空间, 也需要大量的时间去匹配检查. 因此, 如何在占用较少的空间和时间前提下避免循环现象对提高算法性能具有重要的作用. 格局检测(Configuration Checking, CC)策略和加强式格局检测(Strengthened Configuration Checking, STCC)策略就是采用有限的空间和时间有效地解决循环现象的方法.

格局检测策略是在 2011 年由 Cai 等人^[39]提出的, 该策略在求解问题时考虑到了问题的环境信息(即格局). 如果格局未发生变化(图问题中的格局定义为点的环境, 可满足性等问题中的格局定义为变量的环境), 则仅该点或变量发生变化的赋值不能作为解; 反之, 如果格局发生了变化, 则仅该点或变量发生变化的赋值可以作为解, 算法可以向该解执行搜索. 该策略避免了算法再次访问已经搜索过的解, 因此可以有效减少循环现象的发生. 该策略最初用于求解近似最小顶点覆盖问题^[39]; 随后应用于求解近似 SAT 问题, 基于该策略的 SAT 求解器获得了 2012 年国际 SAT 比赛随机组冠军, 这也是迄今为止国内在该项比赛上获得的最好成绩^[40]; 之后该策略又成功应用于求解带有长子句的 SAT 问题^[41]、MaxSAT 问题^[24]、部分 MaxSAT 问题上^[42]、集合覆盖问题^[43]、加权团问题^[44]、泛化顶点覆盖问题^[45], 基于该策略的 MaxSAT 求解器在 2013 年国

际 MaxSAT 比赛随机组的 4 个问题类中获得了第 1 名;另外,基于该策略的 SAT 求解器也获得了 2014 国际 SAT 比赛 Hard-combinatorial 组亚军.因此,格局检测策略是一种可操作性强、实现简单、额外开销小的策略,它在提高算法性能方面发挥了重要的作用.

加强式格局检测策略是格局检测的一种加强.其对格局检测策略的加强主要体现在:当 MinSAT 问题的环境信息(即格局)发生变化时,仅该变量发生变化的赋值不一定能够作为候选解.只有当格局发生了变化并且格局的变化使得当前考查的子句由不可满足变为可满足时,仅该变量发生变化的赋值才可以作为候选解并向该候选解搜索.因此,加强式格局检测策略是格局检测的加强,其具体实现方法是用格局表示变量的环境,并为每个变量设置一个格局变量,我们用 $confChange[v]$ 表示变量 v 的格局变量. $confChange[v]=0$ 表示发生了以下两种情况之一:(1) 变量 v 从上一次翻转(即变量 v 的真值由 true 变为 false 或者由 false 变为 true)后它的格局未发生变化;(2) 变量 v 从上一次翻转后它的格局发生了变化,但格局的变化使得当前考查的子句由可满足变为不可满足. $confChange[v]=1$ 表示变量 v 从上一次翻转后它的格局发生了变化并且格局的变化使得当前考查的子句由不可满足变为可满足.加强式格局检测策略作为求解 MinSAT 问题的一种启发式,其基本原理是:如果变量 v 的格局在上次翻转后没有发生变化,即 $confChange[v]=0$,那么就代表变量 v 的环境没有变化,算法不会选择变量 v 进行翻转,否则就很容易跳转到曾经搜索过的某个节点,从而在搜索中出现循环问题.如果变量 v 从上一次翻转后它的格局发生了变化,但格局的变化使得当前考查的子句由可满足变为不可满足,即 $confChange[v]=0$,那么就代表变量 v 环境的变化很有可能增加了不可满足的子句的数目,这使得算法能进一步找到更好的解,因此算法不会选择变量 v 进行翻转.如果变量 v 从上一次翻转后它的格局发生了变化并且格局的变化使得当前考查的子句由不可满足变为可满足,即 $confChange[v]=1$,那么就代表变量 v 环境的变化很有可能减少了不可满足的子句的数目,因此算法会选择变量 v 进行翻转.在局部搜索中,选择合适的变量翻转对提高算法的效率具有重要的意义.如果没有加强式格局检测,对于一般的局部搜索,在变量翻转的时候一般就是选择启发值最高的.换言之当格局没有发生变化的时候

也允许翻转,这就很可能直接导致上次刚刚翻转的变量又被翻转回来.通过限制只允许格局发生变化且格局的变化使得当前考查的子句由可满足变为不可满足的变量进行翻转,这就会避免上述情况,因此加强格局检测策略在整个搜索过程中都具有重要的意义,由于格局的定义是在一般变量上定义的,因此对所有的变量翻转都具有意义.根据格局的定义,我们的加强格局检测策略是定义在任意变量上的,对变量没有任何约束要求,因此可以普遍应用于以后的任意 MinSAT 问题随机搜索算法中,从这个意义上讲,该策略对求解 MinSAT 问题具有一般性和完整性.

图 1 给出了 MinSAT 问题变量的格局变量的更新函数.该函数的输入参数为 CNF 范式 F , F 中所有变量的集合 V 和待翻转的变量 v .对于一个合取范式 F ,最初所有变量的格局变量 $confChange[v]$ 都赋值为 1.当变量 v 的真值发生了翻转(即 $flip(v)$ 的返回值为真),则将变量 v 的格局变量 $confChange[v]$ 赋值为 0 并在变量集合 V 中删除变量 v (2 行~4 行);然后检查 F 中每个子句真值的变化(5 行):如果变量 v 真值的改变使得某一个子句 c 由可满足变为不可满足,那么同时存在于该子句 c 和变量集合 V 中的变量的格局变量赋值为 0 并在变量集合 V 中删除这些变量(6 行~9 行);如果变量 v 真值的改变使得某一个子句 c 由不可满足变为可满足,那么同时存在于该子句 c 和变量集合 V 中的变量的格局变量赋值为 1 并在变量集合 V 中删除这些变量(10 行~13 行).

```

Function UpdateconfChange( $F, V, v$ )
1. begin
2.   if ( $flip(v)$ )
3.      $confChange[v]=0$ ;
4.     remove  $v$  from  $V$ ;
5.     for each clause  $c \in F$ 
6.       if ( $SATtoUnSAT(c)$ )
7.         for each  $v' \in c$  and  $v' \in V$ 
8.            $confChange[v']=0$ ;
9.           remove  $v'$  from  $V$ ;
10.      if ( $UnSATtoSAT(c)$ )
11.        for each  $v' \in c$  and  $v' \in V$ 
12.           $confChange[v']=1$ ;
13.          remove  $v'$  from  $V$ ;
14.   end

```

图 1 格局变量更新函数 UpdateconfChange

加强式格局检测策略对于 MinSAT 问题的局部搜索求解算法来说很有必要并且效果显著,由于格局变量的取值是由以前的搜索路径决定的,因此

它可以指导搜索状态的转移,并且能够从备选搜索状态中去除一些没有必要进行的搜索,进而减小搜索空间,缩短算法的执行时间。

3.2 子句加权策略

子句加权策略是一种多样性搜索策略,其在 SAT、MaxSAT 等问题的局部搜索算法中被广泛应用.它实际上是将问题的自身结构信息量化为权值,搜索过程中对权值进行调整,从而防止算法陷入局部最优.子句加权策略应用在 MinSAT 问题的基本思想是:对公式 F 中的每个子句都赋予一个权值 1, MinSAT 问题求解算法在搜索过程中一旦陷入局部最优就增加在当前解下可满足子句的权值,这使得算法能进一步找到隐藏在局部最优附近更好的解.为了估计待搜索的解 s 与其邻域解的重要程度,我们给出了解的评价函数 $cost(F, s)$, 定义它为在解 s 下所有可满足子句的权值之和,对于 MinSAT 问题来说解的评价函数越小则表示该解越好.为了判断解 s 与其邻域解的好坏,算法引入函数 $dscore(v)$, 定义它为解 s 与其邻域中一个解 s' 的评价函数之差,如果它们的差值大于 0,则说明解 s' 优于 s ;反之,则说明搜索过程陷入局部最优.接下来我们分别给出评价函数 $cost(F, s)$ 和函数 $dscore(v)$ 的计算公式.

(1) 评价函数 $cost(F, s)$:

$$cost(F, s) = \sum_{c \in satisfied(C(s))} w(c),$$

其中: $satisfied(C(s))$ 表示在解 s 下所有满足子句的集合; $w(c)$ 表示子句 c 的权值.

(2) 函数 $dscore(v)$:

$$dscore(v) = cost(F, s) - cost(F, s'),$$

其中: $cost(F, s)$ 表示解 s 的评价函数;解 s' 是指翻转解 s 中变量 v 的真值后所形成的解.

显然,对公式 F 中的每个变量 v 都对应一个 $dscore(v)$.如果在公式 F 中至少存在一个变量 v 的 $dscore(v) > 0$,则说明在解 s 的邻域中至少有一个解 s' 要优于 s .如果 $dscore(v)$ 越大(其中 $dscore(v) > 0$),则在解 s' (只改变解 s 中变量 v 的真值后形成的解)下可满足子句的权值之和越小,也在一定程度上说明了在解 s' 下可满足子句的个数越少,则表示在搜索空间中解 s' 优于解 s 的程度越高.如果 $dscore(v) < 0$,则表示解 s 邻域中的所有解均差于解 s ,此时对解 s 下的可满足子句的权值加 1,然后重新计算每个变量 v 的 $dscore(v)$.这时我们发现增加可满足子句的权值后,局部最优的代价也发生了改变,这使得算法

能够进一步找到隐藏在局部最优附近更好的解,这也体现了一种搜索的多样性。

3.3 加强式格局检测与子句加权局部搜索算法

本节将详细介绍求解 MinSAT 问题的加强式格局检测与子句加权局部搜索算法(Strengthened Configuration Checking and Clause Weighting Local Search, STCCCWLS),该算法在随机行走的框架基础上融入了子句加权策略,并根据 MinSAT 问题本身的特征对格局检测策略进行了加强.加强式格局检测策略可以减少局部搜索中的循环问题;子句加权策略通过增加局部最优的代价,使得算法可以进一步找到隐藏在局部最优附近的更好的解.图 2 给出了 STCCCWLS 算法的基本框架。

```

Algorithm STCCCWLS( $F$ ).
Input: a CNF-formula  $F$ .
Output: a solution  $s$ , and the number of satisfied clauses  $cla$  in  $s$ .
1.  $cla \leftarrow +\infty$ ;
2. initialize the weight of each clause as 1;
3. initialize  $confChange[v]$  as 1 for each variable  $v$ ;
4.  $S \leftarrow$  randomly generate a solution;
5.  $csc \leftarrow CSC(s)$ ;
6. if  $csc < cla$ , then
7.    $cla \leftarrow csc$ ;
8. while ( $t < t_{max}$ ) do
9.   compute  $dscore(v)$  for each variable  $v$ ;
10.  if ( $(G = \{v | dscore(v) > 0 \& confChange[v] = 1\}) \neq \emptyset$ ), then
11.    choose the variable  $v$  with the largest  $dscore(v)$  from  $G$ ;
12.    flip the truth value of variable  $v$  in  $s$  as the  $p$ ;
13.    flip the truth value of a random variable in  $s$  as the  $1-p$ ;
14.    update the solution  $s$ ;
15.     $csc \leftarrow CSC(s)$ ;
16.    if  $csc < cla$ , then
17.       $cla \leftarrow csc$ ;
18.    Update  $confChange(F, V, v)$ ;
19.  else
20.    for each satisfied clause  $c_i$ ;
21.       $w(c_i) \leftarrow w(c_i) + 1$ ;
22.  endwhile
23. return  $cla$  and  $s$ ;

```

图 2 STCCCWLS 算法的基本框架

首先介绍算法 STCCCWLS 中的符号: cla 表示当前最小满足子句数; $CSC(s)$ 表示一个函数,该函数计算在解 s 下可满足的子句数; csc 表示函数 $CSC(s)$ 的返回结果; $confChange[v]$ 表示变量 v 的格局变量; t_{max} 表示算法最大运行时间.算法首先初始化变量 cla (1 行)、所有子句的权值(2 行)以及所有变量的格局变量 $confChange[v]$ (3 行);其次以等概率赋值的方式随机产生一个解 s (4 行);然后调用 $CSC(s)$ 函数计算在当前解下可满足的子句数(5 行),如果

当前最小满足子句数 cla 小于解 s 下的可满足的子句数 csc , 那么更新当前最小满足子句数 cla (6 行~7 行). 接下来算法进入一个循环过程 (8 行), 在每次循环中首先更新所有变量的 $dscore(v)$ 函数 (9 行), 如果存在一个不为空的变量集合 G , 其中该变量集合中任意变量的 $dscore(v)$ 函数的值都大于 0 并且该变量的格局变量 $confChange[v]=1$, 则以 $p(0 < p < 1)$ 的概率翻转变量集合 G 中具有最大 $dscore(v)$ 函数值的变量 v , 以 $1-p$ 的概率随机选择一个变量进行翻转 (10 行~13 行), 更新解并计算当前解下可满足子句的数目 (14 行~15 行), 如果当前最小满足子句数 cla 小于解 s 下的可满足的子句数 csc , 那么更新当前最小满足子句数 cla (16 行~17 行), 然后调用格局变量更新函数 $UpdateconfChange(F, V, v)$ (如图 1 所示) (18 行). 如果没有找到格局变量 $confChange[v]=1$ 并且 $dscore(v)$ 值为正数的变量 (19 行), 则将所有可满足的子句的权值都增加 1 (20 行~21 行), 停止此次循环 (22 行). 最后算法返回解 s 和在解 s 下最小可满足子句数 cla (23 行).

3.4 相关算法的比较

近年来, 格局检测和子句加权策略已经被广泛应用于 SAT 问题的近似求解中, 其求解器也在国际 SAT 比赛中获得了优异的成绩. 这里对这些算法与本文算法做简单的比较.

参考文献[46]中提出了局部搜索算法用于求解 SAT 问题, 其中使用的方法与本文提出的有相近之处. 两者算法的区别主要体现在 3 个方面: (1) 格局检测策略的使用. 文献[46]中使用了格局检测策略, 其基本思想在于: 当点或变量的环境信息未发生变化时, 则仅该点或变量发生变化的赋值不能作为解; 反之, 当点或变量的环境信息发生了变化时, 则仅该点或变量发生变化的赋值就作为解算法向其进行搜索. 而本文中使用的加强式格局检测策略, 其基本思想在于: 当点或变量的环境信息未发生变化时, 则仅该点或变量发生变化的赋值不能作为解 (此与格局检测的思想相同); 反之, 当点或变量的环境信息发生了变化时, 仅该变量发生变化的赋值不一定能够作为解. 只有当格局发生了变化并且格局的变化使得当前考查的子句由不可满足变为可满足时, 仅该变量发生变化的赋值才可以作为解并向该解搜索; (2) 子句加权策略的使用. 文献[46]中给出了 3 种近似求解 SAT 问题的算法, 这 3 种算法实际上都使用了子句加权策略, 其子句权值的初始设置都与本文相同, 但由于 SAT 问题本身的求解特征, 其权值

的调整方法与本文的算法不同, 在文献[46]中算法 Swcca 是对解 s 下的不可满足子句的权值加 1; 算法 Swcca 是当所有子句权值的平均值大于一个临界值时, 根据公式 $w(c_i) = \lfloor \rho \cdot w(c_i) \rfloor + \lfloor (1-\rho) \cdot w(c_i) \rfloor$ (其中 c_i 表示第 i 个子句, $w(c_i)$ 表示子句的权值, ρ 是一个 $0 < \rho < 1$ 的常数) 更改各子句的权值; 否则, 对解 s 下的不可满足子句的权值加 1; 算法 CCAsubscore 是当所有子句权值的平均值大于一个临界值时, 根据公式 $w(c_i) = \lfloor \rho \cdot w(c_i) \rfloor + \lfloor (1-\rho) \cdot w(c_i) \rfloor$ (其中 c_i 表示第 i 个子句, $w(c_i)$ 表示子句的权值, ρ 是一个 $0 < \rho < 1$ 的常数) 更改各子句的权值; 否则, 对解 s 下的所有权值大于 1 的可满足子句的权值以概率 sp 减小 1, 对解 s 下的不可满足子句的权值加 1. 而本文所使用的子句加权策略根据 MinSAT 问题近似求解的特征进行权值调整的方法与之不同, 本文的调整方法仅是对解 s 下的可满足子句的权值加 1; (3) 算法框架的使用. 文献[46]的 3 个算法都是在基本的局部搜索算法基础上给出的, 而本文的算法是在随机行走算法的基础上给出的.

文献[47]与本文的算法都使用了格局检测策略并且都是在随机行走的算法框架基础上生成的. 文献[47]的算法与本文的算法的区别体现在以下两个方面: (1) 格局检测策略的使用. 文献[47]格局检测策略中格局的定义是在子句的取值基础上给出的, 本文中所使用的格局的定义是在变量的取值基础上给出的; 文献[47]使用格局检测策略的目的是调整 $ConfTime(x)$ 的取值, 本文使用加强式格局检测策略的目的是直接用作算法的启发式; (2) 子句加权策略的使用. 文献[47]的算法未使用子句加权策略而采用打分函数用于实现贪心式搜索; 本文的算法通过使用子句加权策略以实现多样性搜索.

文献[48]与本文的算法都使用了格局检测策略和子句加权策略. 两者的区别体现在: (1) 格局检测策略的使用. 文献[48]的算法采用的是标准的格局检测策略; 而本文采用的是加强式格局检测策略; (2) 子句加权策略的使用. 文献[48]的算法实际上也是使用了子句加权策略, 其子句权值的初始设置与本文相同, 但由于 SAT 问题本身的求解特征, 其权值的调整方法与本文的算法不同; 文献[48]的算法对解 s 下的所有权值大于 1 的可满足子句的权值以 sp 概率减小 1, 对解 s 下的不可满足子句的权值以 $(1-sp)$ 概率加 1; 而本文的调整方法仅是对解 s 下的可满足子句的权值加 1.

由于文献[46-48]与本文处理的是两种不同的问题,文献[46-48]处理的 SAT 问题,当算法找到一组使得所有子句均满足的赋值时即可停止;而本文处理的 MinSAT 问题,该问题是 SAT 问题的优化,当算法找到一组使得不满足的子句数目尽可能最多的赋值时才可停止.文献[46-48]和本文处理的问题虽然具有相似性,这也就说明了为什么算法采用的策略大体相同,但是由于问题本身的求解具有不同的特征,因此,算法采用的具体策略又存在区别.

4 实验比较

本节将实验分成两个部分:第一部分比较了加强式格局检测与子句加权局部搜索算法(STCCCWLS 算法)和精确算法(MinSatz 算法)在随机实例、RB 模型实例以及工业化实例上的性能,从而说明 STCCCWLS 算法的有效性.第二部分比较了加强式格局检测与子句加权局部搜索算法(Strengthened Configuration Checking and Clause Weighting Local Search, STCCCWLS)、随机行走局部搜索算法(Random Walk Local Search, RWLS)、子句加权局部搜索算法(Clause Weighted Local Search, CWLS)、加强式格局检测局部搜索算法(Strengthened Configuration Check Local Search, STCCLS)、标准格局检测与子句加权局部搜索算法(Standard Configuration Check and Clause Weighted Local Search, SCCCWLS)在随机实例、RB 模型实例以及工业化实例上的性能,从而说明子句加权策略以及加强式格局检测策略的有效性.随机行走局部搜索算法(RWLS)是在 1992 年由 Mitchell 等人^[13]提出的,该算法在求解 SAT、MaxSAT 等问题上产生了深远的影响,之后出现了许多基于该算法的改进策略,例如目前在近似 SAT 问题难解区域求解效率最高的调查传播算法就是基于随机行走局部搜索算法的改进算法^[49].子句加权局部搜索算法(CWLS)是一种以 SAT、MinSAT 等问题的自身结构信息作为启发信息的近似求解算法,它将问题自身的结构信息量化为子句权值并采用子句加权来优化决策策略.目前该类算法已经成功求解了近似 SAT 和 MaxSAT 等问题^[42,46].本节实验所用的计算机为 Intel(R) Xeon(R) CPU, 2.4 GHz, 2 GB 内存,操作系统为 Linux(版本号为 2.6.18-164.el5PAE).

4.1 实验数据集

本文的实验使用了 3 种数据集,分别是随机实

例、RB 模型实例和工业化实例.

(1) 随机实例

本文使用的随机实例中每个子句都包含 3 个文字(literals),并采用了子句数与变量数比值分别为 4.0、4.3、5.0 的 3 组实例,实例采用的变量个数分别为 50、75、100、125、150、175、200、250.实例中变量出现在每个子句中的概率是相同的,变量取真或假的概率也是一样的.因为子句数与变量数比值在 4.0 和 5.0 之间的实例是 3-SAT 问题的相变区域,因此这些实例的求解是困难的.

(2) RB 模型实例

2000 年,北京航空航天大学的许可教授在 JAIR 上首次提出了 RB 模型, RB 模型具有精确的相变点,可以生成难解的实例.本文所有的 RB 模型的数据实例均下载于 Ke Xu 主页^①,这些实例都在相变点附近,具有一定的求解难度,可以有效验证算法的性能.

(3) 工业化实例

由于求解 SAT 问题、MinSAT 问题和 MaxSAT 问题的输入文件格式是完全相同的,所以,本文还采用了 2009 年 MaxSAT 问题比赛中的工业化实例数据,这些工业化数据都是结构化的难解实例.

4.2 STCCCWLS 算法与 MinSatz 算法的比较

为了说明 STCCCWLS 算法的准确性,本小节比较了 STCCCWLS 算法和 MinSatz 算法在随机实例、RB 模型实例及工业化实例上的性能. MinSatz 算法是 2010 年由法国学者 Li Chumin 提出的,它是一种基于 DPLL 过程的精确算法,能够给出 MinSAT 实例准确的最小不可满足子句的数目,其求解器来源于 Li Chumin 教授. STCCCWLS 算法是本文给出的一种基于加强式格局检测与子句加权的局部搜索算法,其求解器使用 C++ 语言进行编码实现.表 1 给出了 STCCCWLS 算法和 MinSatz 算法在随机实例上的比较结果.表 1 中 #var 表示实例中变量的个数;C/V 表示实例中子句数量与变量数量的比值.在表 1 的每组实验中,采用了 50 小组随机实例,然后将这些结果的平均值作为实验结果.表 1 中 #cla 表示该算法计算出来的可满足子句的最小数量;“—”表示算法在程序规定的时间内没有计算出对应实例的结果.实验中算法 MinSatz 的最大运行时间为 6 h,算法 STCCCWLS 的运行时间为近似解在 1/6 h 内不发生变化时停止.对于 RB 模型实例以

① <http://www.nlsde.buaa.edu.cn/~kexu/>

及工业化实例, MinSatz 算法在给定的时间无法得到问题的解, 而本文所提算法均能得到所求问题的近似最优解, 实验结果将在 4.3 节介绍。

表 1 STCCWLS 算法与 MinSatz 算法在随机实例上的比较

实例 #var C/V	MinSatz #cla	STCCWLS #cla	实例 #var C/V	MinSatz #cla	STCCWLS #cla
50 4.0	139.33	139.33	175 4.3	—	552.84
75 4.0	208.73	210.67	200 4.3	—	642.30
100 4.0	279.28	285.55	225 4.3	—	712.26
125 4.0	348.80	360.40	250 4.3	—	794.58
150 4.0	—	435.60	50 5.0	172.68	180.02
175 4.0	—	512.03	75 5.0	261.60	271.66
200 4.0	—	588.10	100 5.0	357.84	365.90
225 4.0	—	667.13	125 5.0	—	459.48
250 4.0	—	741.93	150 5.0	—	556.60
50 4.3	154.00	154.22	175 5.0	—	650.90
75 4.3	229.44	231.94	200 5.0	—	750.20
100 4.3	302.90	309.45	225 5.0	—	844.52
125 4.3	372.00	390.38	250 5.0	—	942.40
150 4.3	—	474.15			

从表 1 可以看出, 对于子句数与变量数比值分别为 4.0、变量数量在 100 个以内的随机数据实例, 精确求解器 MinSatz 在最大运行时间 6 h 内计算出了准确的结果。当变量的个数增长到 150 个及以上时, 精确求解器 MinSatz 不能够在规定时间内计算出结果, 这是由于精确算法的时间复杂度是随着实例中变量个数的增加呈指数级增长的。对于 STCCWLS 算法, 随着实例中变量个数的增加 STCCWLS 算法仍然可以有效地求解, 并且 STCCWLS 算法求得的结果很接近 MinSatz 求解器求得的结果, 对于变量个数为 50、子句个数与变量个数比值 C/V 为 4.0 的实例, STCCWLS 算法求得的结果与 MinSatz 求解器求得的结果仅相差 0.1%, 这也说明了 STCCWLS 算法的有效性。

4.3 局部搜索算法的比较

为了说明子句加权策略和加强式格局检测策略的有效性, 本小节比较了随机行走局部搜索算法 (RWLS)、子句加权局部搜索算法 (CWLS)、加强式格局检测局部搜索算法 (STCCLS)、标准格局检测与子句加权局部搜索算法 (Standard Configuration Check and Clause Weighted Local Search, SCCCWLS) 和加强式格局检测与子句加权局部搜索算法 (STCCWLS) 在随机实例、RB 模型实例以及工业化实例上的性能。随机行走局部搜索算法 (RWLS) 是以概率 $p(0 < p < 1)$ 选择一个最优的变量进行翻转, 使得不满足的子句数量增多; 以 $1 - p(0 < p < 1)$

的概率随机地选择一个变量进行翻转。子句加权局部搜索算法 (CWLS) 是在随机行走局部搜索算法的基础上加入了子句加权策略, 使得算法能够进一步找到隐藏在局部最优附近更好的解。加强式格局检测局部搜索算法 (STCCLS) 是在随机行走局部搜索算法的基础上加入了加强式格局检测策略, 使得算法能够尽可能地减少循环搜索现象。标准格局检测与子句加权局部搜索算法 (SCCCWLS) 是在子句加权局部搜索算法基础上加上标准的格局检测策略, 使得算法能够尽可能地减少循环搜索现象。加强式格局检测与子句加权局部搜索算法 (STCCWLS) 即为本文给出的算法。这几种算法所对应的求解器都是采用 C++ 语言进行编码实现的。

4.3.1 局部搜索算法在随机实例上的比较

本小节在随机实例上做了 4 个对比实验, 第 1 个实验比较了随机行走局部搜索算法 (RWLS) 和子句加权局部搜索算法 (CWLS), 该实验用于说明子句加权策略的有效性; 第 2 个实验比较了随机行走局部搜索算法 (RWLS) 和加强式格局检测局部搜索算法 (STCCLS), 该实验用于说明加强式格局检测策略的有效性; 第 3 个实验比较了子句加权局部搜索算法 (CWLS)、加强式格局检测局部搜索算法 (STCCLS) 和加强式格局检测与子句加权局部搜索算法 (STCCWLS), 该实验用于说明子句加权策略和加强式格局检测策略结合的必要性; 第 4 个实验比较了标准格局检测与子句加权局部搜索算法 (SCCCWLS) 和加强式格局检测与子句加权局部搜索算法 (STCCWLS), 该实验用于说明加强式格局检测的优越性。表 2、表 3、表 4 和表 5 分别给出了以上 4 个实验的比较结果。在这 4 个表中, #var 表示实例中变量的个数, C/V 表示实例中子句数量与变量数量的比值, #cla 表示相应算法计算 50 个随机实例所得到的最小可满足子句数量的平均值, 所有算法的运行在近似解 1/6 h 内不发生变化时停止。

表 2 RWLS 和 CWLS 算法在随机实例上的对比

实例 #var C/V	RWLS #cla	CWLS #cla	实例 #var C/V	RWLS #cla	CWLS #cla
50 4.0	150.20	139.37	250 4.0	741.00	754.03
75 4.0	233.30	211.63	100 4.3	343.35	310.85
100 4.0	285.72	287.03	150 4.3	526.47	478.25
125 4.0	403.03	362.33	200 4.3	715.05	645.40
150 4.0	486.77	440.87	100 5.0	401.65	367.35
175 4.0	572.30	519.70	150 5.0	615.25	562.05
200 4.0	588.70	597.20	200 5.0	834.25	759.85
225 4.0	666.47	678.07			

表 3 RWLS 和 STCCLS 算法在随机实例上的对比

实例 #var C/V	RWLS #cla	STCCLS #cla	实例 #var C/V	RWLS #cla	STCCLS #cla
50 4.0	150.20	139.40	250 4.0	741.00	758.10
75 4.0	233.30	212.27	100 4.3	343.35	313.00
100 4.0	285.72	288.83	150 4.3	526.47	481.65
125 4.0	403.03	366.17	200 4.3	715.05	650.20
150 4.0	486.77	443.60	100 5.0	401.65	369.95
175 4.0	572.30	520.93	150 5.0	615.25	566.00
200 4.0	588.70	600.87	200 5.0	834.25	767.15
225 4.0	666.47	680.87			

表 4 CWLS、STCCLS 和 STCCCWLS 算法在随机实例上的对比

实例 #var C/V	CWLS #cla	STCCLS #cla	STCCCWLS #cla
50 4.0	139.37	139.40	139.33
75 4.0	211.63	212.27	210.67
100 4.0	287.03	288.83	285.55
125 4.0	362.33	366.17	360.40
150 4.0	440.87	443.60	435.60
175 4.0	519.70	520.93	512.03
200 4.0	597.20	600.87	588.10
225 4.0	678.07	680.87	667.13
250 4.0	754.03	758.10	741.93
100 4.3	310.85	313.00	309.45
150 4.3	478.25	481.65	474.15
200 4.3	645.40	650.20	638.45
100 5.0	367.35	369.95	365.90
150 5.0	562.05	566.00	556.60
200 5.0	759.85	767.15	750.20

表 5 SCCCWLS 和 STCCCWLS 算法在随机实例上的对比

实例 #var C/V	SCCCWLS #cla	STCCCWLS #cla	实例 #var C/V	SCCCWLS #cla	STCCCWLS #cla
50 4.0	139.37	139.33	250 4.0	747.87	741.93
75 4.0	211.30	210.67	100 4.3	310.60	309.45
100 4.0	286.97	285.55	150 4.3	476.15	474.15
125 4.0	362.40	360.40	200 4.3	642.30	638.45
150 4.0	437.97	435.60	100 5.0	366.80	365.90
175 4.0	514.67	512.03	150 5.0	558.70	556.60
200 4.0	593.03	588.10	200 5.0	753.35	750.20
225 4.0	672.13	667.13			

从表 2 可以看出,在随机行走局部搜索算法(RWLS)中加入了子句加权策略后得到的子句加权局部搜索算法(CWLS)在 15 个例子的实验结果上有 11 个实验结果比 RWLS 算法有提升,这也说明了子句加权策略起到了重要的作用.其原因是在使

表 6 RWLS、CWLS、SCCCWLS、STCCLS 和 STCCCWLS 算法在 RB 模型实例上的对比

实例	RWLS #cla	CWLS #cla	STCCLS #cla	SCCCWLS #cla	STCCCWLS #cla	实例	RWLS #cla	CWLS #cla	STCCLS #cla	SCCCWLS #cla	STCCCWLS #cla
frb30-15-1	30	30	30	30	30	frb50-23-1	50	50	50	50	50
frb30-15-2	30	30	30	30	30	frb50-23-2	50	50	50	50	50
frb30-15-3	30	30	30	30	30	frb50-23-3	50	50	50	50	50
frb30-15-4	30	30	30	30	30	frb50-23-4	50	50	50	50	50
frb30-15-5	30	30	30	30	30	frb50-23-5	50	50	50	50	50
frb35-17-1	35	35	35	35	35	frb53-24-1	53	53	53	53	53
frb35-17-2	35	35	35	35	35	frb53-24-2	53	53	53	53	53
frb35-17-3	35	35	35	35	35	frb53-24-3	53	53	53	53	53
frb35-17-4	35	35	35	35	35	frb53-24-4	53	53	53	53	53
frb35-17-5	35	35	35	35	35	frb53-24-5	53	53	53	53	53

用子句加权策略时,MinSAT 问题求解算法在搜索过程中一旦陷入局部最优就增加在当前解下可满足子句的权值,这使得每个变量的打分进行了更新,从而算法能够有效地跳出局部最优,进一步找到隐藏在局部最优附近更好的解.表 3 的实验结果显示了加强式格局检测局部搜索算法(STCCLS)的大部分实验结果要优于随机行走局部搜索算法(RWLS),这也说明了加强式格局检测策略的有效性.其原因是加强式格局检测策略考虑了变量的环境信息,尽可能地减少了局部搜索算法中的循环问题,进而提高了算法的性能.从表 4 可以看出,子句加强式格局检测与子句加权局部搜索算法(STCCCWLS)的实验结果要优于加权局部搜索算法(CWLS)和加强式格局检测局部搜索算法(STCCLS)的实验结果,对于这 3 个算法,在 15 个例子上 STCCCWLS 的结果都是最优的,实验结果表明加强式格局检测策略和子句加权策略的结合是有必要性的.从表 5 可以看出,加强式格局检测与子句加权局部搜索算法(STCCCWLS)的实验结果要优于标准格局检测与子句加权局部搜索算法(SCCCWLS),原因在于标准格局检测对变量的限制更为宽松,使得在搜索过程中容易漏掉一些较好的解,这也是本文采用加强式格局检测的原因.综上所述,通过表 2、表 3、表 4 和表 5 的实验结果对比,证明了 STCCCWLS 算法在随机实例上的有效性.

4.3.2 局部搜索算法在 RB 模型实例上的比较

本小节在 RB 模型实例上做了对比实验,实验中采用的对比算法有随机行走局部搜索算法(RWLS)、子句加权局部搜索算法(CWLS)、加强式格局检测局部搜索算法(STCCLS)、标准格局检测与子句加权局部搜索算法(SCCCWLS)和加强式格局检测与子句加权局部搜索算法(STCCCWLS).所有算法的运行在近似解 1/6 h 内不发生变化时停止.表 6 给出了对比实验的结果,从表 6 可以看出,

(续 表)

实例	RWLS	CWLS	STCCLS	SCCCWLS	STCCWLS	实例	RWLS	CWLS	STCCLS	SCCCWLS	STCCWLS
	#cla	#cla	#cla	#cla	#cla		#cla	#cla	#cla	#cla	#cla
frb40-19-1	40	40	40	40	40	frb56-25-1	56	56	56	56	56
frb40-19-2	40	40	40	40	40	frb56-25-2	56	56	56	56	56
frb40-19-3	40	40	40	40	40	frb56-25-3	56	56	56	56	56
frb40-19-4	40	40	40	40	40	frb56-25-4	56	56	56	56	56
frb40-19-5	40	40	40	40	40	frb56-25-5	56	56	56	56	56
frb45-21-1	45	45	45	45	45	frb59-26-1	59	59	59	59	59
frb45-21-2	45	45	45	45	45	frb59-26-2	59	59	59	59	59
frb45-21-3	45	45	45	45	45	frb59-26-3	59	59	59	59	59
frb45-21-4	45	45	45	45	45	frb59-26-4	59	59	59	59	59
frb45-21-5	45	45	45	45	45	frb59-26-5	59	59	59	59	59

对于 RB 模型实例来说随机行走局部搜索算法 (RWLS)、子句加权局部搜索算法 (CWLS)、加强式格局检测局部搜索算法 (STCCLS)、标准格局检测与子句加权局部搜索算法 (SCCCWLS) 和加强式格局检测与子句加权的局部搜索算法 (STCCWLS) 的求解效率相当, 都可以求得较好的解, 这几种算法都设置了相应的策略可以使算法跳出局部最优, 例如, RWLS 算法和 STCCLS 算法通过以 $1-p$ ($0 < p < 1$) 的概率随机地选择一个变量进行翻转的方式跳出局部最优, CWLS 算法、SCCCWLS 算法和 STCCWLS 算法通过采用子句加权策略跳出局部最优, 因此这 3 种算法可以得到较优的解。

4.3.3 局部搜索算法在工业化实例上的比较

本小节在工业化实例上也做了 4 个对比实验, 第 1 个实验比较了随机行走局部搜索算法 (RWLS)

和子句加权局部搜索算法 (CWLS), 该实验用于说明子句加权策略的有效性; 第 2 个实验比较了随机行走局部搜索算法 (RWLS) 和加强式格局检测局部搜索算法 (STCCLS), 该实验用于说明加强式格局检测策略的有效性; 第 3 个实验比较了子句加权局部搜索算法 (CWLS)、加强式格局检测局部搜索算法 (STCCLS) 和加强式格局检测与子句加权局部搜索算法 (STCCWLS), 该实验用于说明子句加权策略和加强式格局检测策略结合的必要性; 第 4 个实验比较了标准格局检测与子句加权局部搜索算法 (SCCCWLS) 和加强式格局检测与子句加权局部搜索算法 (STCCWLS), 该实验用于说明加强式格局检测的优越性。所有算法的运行在近似解 1/6 h 内不发生变化时停止。表 7、表 8、表 9 和表 10 分别给出了以上 4 个实验的比较结果。

表 7 RWLS 和 CWLS 算法在工业化实例上的对比

实例	RWLS #cla	CWLS #cla
s15850-bug-onevec-gate-0. dimacs. seq. filtered	39 499	38 976
c3_DD_s3_fl_e1_v1-bug-fourvec-gate-0. dimacs. seq. filtered	66 730	65 969
c3_DD_s3_fl_e1_v1-bug-onevec-gate-0. dimacs. seq. filtered	16 547	16 361
c5_DD_s3_fl_e1_v1-bug-onevec-gate-0. dimacs. seq. filtered	53 382	52 751
c5315-bug-gate-0. dimacs. seq. filtered	3827	3627
c6288-bug-gate-0. dimacs. seq. filtered	7165	5395
c7552-bug-gate-0. dimacs. seq. filtered	5312	5180
mot_comb1._red-gate-0. dimacs. seq. filtered	3783	3698
mot_comb2._red-gate-0. dimacs. seq. filtered	10 143	9902
mot_comb3._red-gate-0. dimacs. seq. filtered	22 470	20 840

表 8 RWLS 和 STCCLS 算法在工业化实例上的对比

实例	RWLS #cla	STCCLS #cla
s15850-bug-onevec-gate-0. dimacs. seq. filtered	39 499	39 023
c3_DD_s3_fl_e1_v1-bug-fourvec-gate-0. dimacs. seq. filtered	66 730	65 994
c3_DD_s3_fl_e1_v1-bug-onevec-gate-0. dimacs. seq. filtered	16 547	16 408
c5_DD_s3_fl_e1_v1-bug-onevec-gate-0. dimacs. seq. filtered	53 382	52 839
c5315-bug-gate-0. dimacs. seq. filtered	3827	3673
c6288-bug-gate-0. dimacs. seq. filtered	7165	5702
c7552-bug-gate-0. dimacs. seq. filtered	5312	5194
mot_comb1._red-gate-0. dimacs. seq. filtered	3783	3700
mot_comb2._red-gate-0. dimacs. seq. filtered	10 143	9918
mot_comb3._red-gate-0. dimacs. seq. filtered	22 470	21 020

表 9 CWLS、STCCLS 和 STCCWLS 算法在工业化实例上的对比

实例	CWLS #cla	STCCLS #cla	STCCWLS #cla
s15850-bug-onevec-gate-0. dimacs. seq. filtered	38976	39023	38386
c3_DD_s3_f1_e1_v1-bug-fourvec-gate-0. dimacs. seq. filtered	65969	65994	65467
c3_DD_s3_f1_e1_v1-bug-onevec-gate-0. dimacs. seq. filtered	16361	16408	16282
c5_DD_s3_f1_e1_v1-bug-onevec-gate-0. dimacs. seq. filtered	52751	52839	52659
c5315-bug-gate-0. dimacs. seq. filtered	3627	3673	2518
c6288-bug-gate-0. dimacs. seq. filtered	5395	5702	5068
c7552-bug-gate-0. dimacs. seq. filtered	5180	5194	4169
mot_comb1. _red-gate-0. dimacs. seq. filtered	3698	3700	2941
mot_comb2. _red-gate-0. dimacs. seq. filtered	9902	9918	7478
mot_comb3. _red-gate-0. dimacs. seq. filtered	20840	21020	19095

表 10 SCCCWLS 和 STCCWLS 算法在工业化实例上的对比

实例	SCCCWLS #cla	STCCWLS #cla
s15850-bug-onevec-gate-0. dimacs. seq. filtered	38798	38386
c3_DD_s3_f1_e1_v1-bug-fourvec-gate-0. dimacs. seq. filtered	66004	65467
c3_DD_s3_f1_e1_v1-bug-onevec-gate-0. dimacs. seq. filtered	16414	16282
c5_DD_s3_f1_e1_v1-bug-onevec-gate-0. dimacs. seq. filtered	52864	52659
c5315-bug-gate-0. dimacs. seq. filtered	2534	2518
c6288-bug-gate-0. dimacs. seq. filtered	4821	5068
c7552-bug-gate-0. dimacs. seq. filtered	4488	4169
mot_comb1. _red-gate-0. dimacs. seq. filtered	3043	2941
mot_comb2. _red-gate-0. dimacs. seq. filtered	7858	7478
mot_comb3. _red-gate-0. dimacs. seq. filtered	19028	19095

从表 7 可以看出子句加权局部搜索算法(CWLS)的实验结果明显优于随机行走局部搜索算法(RWLS)的实验结果,在 c6288-bug-gate-0. dimacs. seq. filtered 实例上所计算出的最小可满足子句的个数减少了将近 2000 个,这也说明了子句加权策略起到了重要的作用. 表 8 的实验结果显示了加强式格局检测局部搜索算法(STCCLS)的实验结果要优于随机行走局部搜索算法(RWLS),在 c6288-bug-gate-0. dimacs. seq. filtered 实例上计算出的最小可满足子句的个数也减少了将近 1500 个,这也说明了加强式格局检测策略的有效性. 从表 9 可以看出,子句加强式格局检测与子句加权局部搜索算法(STCCWLS)的实验结果要优于加权局部搜索算法(CWLS)和加强式格局检测局部搜索算法(STCCLS)的实验结果,实验结果表明加强式格局检测策略和子句加权策略的结合是有必要性的. 从表 10 可以看出,加强式格局

检测与子句加权局部搜索算法(STCCWLS)的实验结果要优于标准格局检测与子句加权局部搜索算法(SCCCWLS),这也是本文采用加强式格局检测的原因. 综上所述,通过表 7、表 8、表 9 和表 10 的实验结果对比,证明了 STCCWLS 算法在工业化实例上的有效性.

4.3.4 统计检验

为了对比随机行走局部搜索算法(RWLS)、子句加权局部搜索算法(CWLS)、加强式格局检测局部搜索算法(STCCLS)、标准格局检测与子句加权局部搜索算法(SCCCWLS)和加强式格局检测与子句加权局部搜索算法(STCCWLS)的性能,本小节使用了假设检验中的秩和检验(结果在表 11 中给出). 从表 11 可以看出,在给定的显著性水平下($\alpha=0.01$),我们所选的具有代表性的大部分实例中 STCCWLS 算法性能显著优于其它算法.

表 11 各算法的成对单边 Wilcoxon 秩和检验的 P 值

实例 #var C/V	STCCWLS_VS_RWLS #cla	STCCWLS_VS_CWLS #cla	STCCWLS_VS_STCCLS #cla	STCCWLS_VS_SCCCWLS #cla
100 4.0	9.17E-01	6.79E-04	4.77E-03	1.28E-01
150 4.0	2.80E-11	2.97E-07	2.86E-11	4.59E-02
200 4.0	5.99E-01	1.27E-10	1.03E-10	3.56E-04
100 4.3	2.61E-11	2.61E-11	2.61E-11	2.61E-11
150 4.3	2.69E-11	2.69E-11	2.69E-11	2.69E-11
200 4.3	9.17E-01	9.17E-01	9.17E-01	9.17E-01
100 5.0	2.61E-11	2.61E-11	2.61E-11	2.61E-11
150 5.0	2.69E-11	2.69E-11	2.69E-11	2.69E-11
200 5.0	9.17E-01	9.17E-01	9.17E-01	9.17E-01

5 结 论

本文提出了加强式格局检测与子句加权局部搜索算法来求解 MinSAT 问题, 格局检测策略和子句加权策略在求解 SAT 和最小顶点覆盖问题中已经表现出较好的求解效果, 本文在随机行走算法的基础上融入了子句加权策略, 并根据 MinSAT 问题本身的特征对格局检测策略进行了加强. 加强式格局检测策略可以减少局部搜索中的循环问题, 以此提高局部搜索算法的性能; 子句加权策略通过增加局部最优的代价, 使得算法可以进一步找到隐藏在局部最优附近的更好的解. 实验结果表明, 该算法能够有效求解 MinSAT 问题, 在大规模实例中具有良好的表现, 这也在一定程度上说明了加强式格局检测策略和子句加权策略的有效性.

参 考 文 献

- [1] Gu J, Purdom P W, Franco J, et al. Algorithms for the satisfiability (SAT) problem: A survey//Proceedings of the DIMACS Series in Discrete Mathematics and Theoretical Computer Science. Washington, USA, 1997: 19-152
- [2] Zhou Jun-Ping, Yin Ming-Hao, Gu Wen-Xiang, Sun Ji-Gui. Research on decreasing observation variables for strong planning under partial observation. *Journal of Software*, 2009, 20(2): 290-304(in Chinese)
(周俊萍, 殷明浩, 谷文祥, 孙吉贵. 部分可观察强规划中约减观察变量的研究. *软件学报*, 2009, 20(2): 290-304)
- [3] Yin Ming-Hao, Sun Ji-Gui, Lin Hai, Wu Xia. Possibilistic extension rules for reasoning and knowledge compilation. *Journal of Software*, 2010, 21(11): 2826-2837(in Chinese)
(殷明浩, 孙吉贵, 林海, 吴瑕. 可能性扩展规则的推理和知识编译. *软件学报*, 2010, 21(11): 2826-2837)
- [4] Li R, Hu S, Wang J. The community structure of the constraint satisfaction problem instances of model RB. *Journal of Computational and Theoretical Nanoscience*, 2015, 12(12): 6088-6093
- [5] Fu Z, Marhajan Y, Malik S. Zchaff sat solver. Princeton, New Jersey City in the United States: Department of Electrical Engineering, Princeton University, Technical Report, 2004
- [6] Braunstein A, Mézard M, Zecchina R. Survey propagation: An algorithm for satisfiability. *Random Structures and Algorithms*, 2005, 27(2): 201-226
- [7] Biaoletti M, Marcugini S, Milani A. An extension of SATPLAN for planning with constraints//Artificial Intelligence: Methodology, Systems, and Applications. Berlin Heidelberg, Springer, 1998: 39-49
- [8] Kautz H, Selman B. BLACKBOX: A new approach to the application of theorem proving to problem solving//Proceedings of the AIPS98 Workshop on Planning as Combinatorial Search. Pittsburgh, USA, 1998: 58-60
- [9] Selman B, Kautz H, Cohen B. Local search strategies for satisfiability testing//Proceedings of the DIMACS Workshop: Cliques, Coloring and Satisfiability, Washington, USA, 1993: 521-532
- [10] Braunstein A, Mézard M, Zecchina R. Survey propagation: An algorithm for satisfiability. *Random Structures and Algorithms*, 2005, 27(2): 201-226
- [11] Cook S A. An overview of computational complexity. *Communications of the ACM*, 1983, 26(6): 400-408
- [12] Kirkpatrick S, Selman B. Critical behavior in the satisfiability of random Boolean expressions. *Science*, 1994, 5163(264): 1297-1301
- [13] Mitchell D, Selman B, Levesque H. A new method for solving hard satisfiability problems//Proceedings of the 10th National Conference on Artificial Intelligence (AAAI). San Jose, USA, 1992: 440-446
- [14] Hirsch E A, Kojevnikov A. UnitWalk: A new SAT solver that uses local search guided by unit clause elimination. *Annals Math and AI*, 2005, 43(1): 91-111
- [15] Yin Ming-Hao, Zhou Jun-Ping, Sun Ji-Gui, Gu Wen-Xiang. Heuristic survey propagation algorithm solving QBF problem. *Journal of Software*, 2011, 22(7): 1538-1550(in Chinese)
(殷明浩, 周俊萍, 孙吉贵, 谷文祥. 求解 QBF 问题的启发式调查传播算法. *软件学报*, 2011, 22(7): 1538-1550)
- [16] Zhao Tong, Zhu Wen-Xing. An improved local search method for the Max-SAT problem. *Computer Engineering and Science*, 2008, 30(11): 50-52(in Chinese)
(赵同, 朱文兴. Max-SAT 问题一种改进的局部搜索算法. *计算机工程与科学*, 2008, 30(11): 50-52)
- [17] Li C M, Quan Z. An efficient branch-and-bound algorithm based on MaxSAT for the maximum clique problem//Proceedings of the 24th National Conference on Artificial Intelligence (AAAI). Atlanta, Georgia, USA, 2010: 128-133
- [18] Gramm J, Hirsch E A, Niedermeier R, et al. Worst-case upper bounds for MAX-2-SAT with an application to MAX-CUT. *Discrete Applied Mathematics*, 2003, 130(2): 139-155
- [19] Li C M, Manyà, F, Planes J. New inference rules for Max-SAT. *Journal of Artificial Intelligence Research*, 2007, 30: 321-359
- [20] Lin H, Su K, Li C M. Within-problem learning for efficient lower bound computation in Max-SAT solving//Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI). Chicago, USA, 2008: 351-356
- [21] Davies J, Cho J, Bacchus F. Using learnt clauses in MaxSAT//Proceedings of the International Conference on Principles and Practice of Constraint Programming. Berlin Heidelberg, Germany, 2010: 176-190

- [22] Fu Z, Malik S. On solving the partial MAX-SAT problem// Proceedings of the International Conference on Theory and Applications of Satisfiability Testing. Seattle, USA, 2006; 252-265
- [23] Heras F, Larrosa J, Oliveras A. Minimaxsat: An efficient weighted Max-SAT solver. *Journal of Artificial Intelligence Research*, 2008, 31: 1-32
- [24] Luo C, Cai S W, Wu W, et al. CCLS: An efficient local search algorithm for weighted maximum satisfiability. *IEEE Transactions on Computers*, 2015, 64(7): 1830-1843
- [25] Hoos HH. An adaptive noise mechanism for WalkSAT// Proceedings of the 18th National Conference on Artificial Intelligence (AAAI). Edmonton, Canada, 2002: 655-660
- [26] Stützle T, Hoos H H. *Stochastic local search: Foundations and applications*. San Francisco, USA: Morgan Kaufmann, 2003
- [27] Smyth K, Hoos H H, Stützle T. Iterated robust tabu search for MAX-SAT//*Advances in Artificial Intelligence*. Berlin Heidelberg: Springer, 2003; 129-144
- [28] Li C M, Zhu Z, Manyà F, et al. Minimum satisfiability and its applications//*Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. Barcelona, Spain, 2011; 605-610
- [29] Kohli R, Krishnamurti R, Mirchandani P. The minimum satisfiability problem. *Discrete Mathematics*, 1994, 7(2): 275-283
- [30] Marathe M V, Ravi S S. On approximation algorithms for the minimum satisfiability problem. *Information Processing Letters*, 1996, 58(1): 23-29
- [31] Avidor A, Zwick U. Approximating MIN 2-SAT and MIN 3-SAT. *Theory of Computing Systems*, 2005, 38(3): 329-345
- [32] Li C M, Manyà F, Quan Z, Zhu Z. Exact MinSAT solving // Strichman O, Szeider S, eds. *Proceedings of the 13th International Conference on Theory and Applications of Satisfiability Testing*. Edinburgh, UK, 2010; 363-368
- [33] Li C M, Zhu Z, Manyà F, et al. Optimizing with minimum satisfiability. *Artificial Intelligence*, 2012, 190: 32-44
- [34] Zhou Jun-Ping. *Research on the Upper Bounds and the Phase Transformation of the Automated Reasoning and the Planning* [Ph.D. dissertation]. Jinlin University, Changchun, 2011 (in Chinese)
(周俊萍. 自动推理与规划问题最小上界和相变规律研究[博士学位论文]. 吉林大学, 长春, 2011)
- [35] Shao Ming, Li Guang-Hui, Li Xiao-Wei. Survey propagation algorithm for SAT and its performance dominated by step length. *Chinese Journal of Computers*, 2005, 28(5): 849-855(in Chinese)
(邵明, 李光辉, 李晓维. 求解可满足问题的调查传播算法以及步长的影响规律. *计算机学报*, 2005, 28(5): 849-855)
- [36] Frieze A, Suen S. Analysis of two simple heuristics on a random instance of k-SAT. *Journal of Algorithms*, 1996, 20(2): 312-355
- [37] Zhou Jun-Ping, Yin Ming-Hao, Zhou Chun-Guang, et al. Minimized The upper bound of #3-SAT problem in the worst case. *Journal of Computer Research and Development*, 2011, 48(11): 2055-2063(in Chinese)
(周俊萍, 殷明浩, 周春光等. 最坏情况下#3-SAT问题最小上界. *计算机研究与发展*, 2011, 48(11): 2055-2063)
- [38] Yin M H, Lin H, Sun J G. Solving # SAT using extension rules. *Journal of Software*, 2009, 20(7): 1714-1725 (in Chinese)
(殷明浩, 林海, 孙吉贵. 一种基于扩展规则的#SAT求解系统. *软件学报*, 2009, 20(7): 1714-1725)
- [39] Cai S, Su K, Sattar A. Local search with edge weighting and configuration checking heuristics for minimum vertex cover. *Artificial Intelligence*, 2011, 175(9): 1672-1696
- [40] Cai S, Su K. Local search with configuration checking for SAT//*Proceedings of the IEEE 23rd International Conference on Tools with Artificial Intelligence (ICTAI)*. Boca Raton, USA, 2011: 59-66
- [41] Cai S, Su K. Comprehensive score: Towards efficient local search for SAT with long clauses//*Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAD)*. Beijing, China, 2013: 489-495
- [42] Cai S, Luo C, Thornton J, et al. Tailoring local search for partial MaxSAT//*Proceedings of the 28th National Conference on Artificial Intelligence (AAAI)*. Québec City, Canada, 2014: 2623-2629
- [43] Wang Y, Yin M, Ouyang D, et al. A novel local search algorithm with configuration checking and scoring mechanism for the set k-covering problem. *International Transactions in Operational Research*, 2017, 24(6): 1463-1485
- [44] Wang Y, Cai S, Yin M. Two efficient local search algorithms for maximum weight clique problem//*Proceedings of the 30th National Conference on Artificial Intelligence (AAAI)*. Phoenix, Arizona, 2016: 1-7
- [45] Li R, Hu S, Wang Y, Yin M. A local search algorithm with tabu strategy and perturbation mechanism for generalized vertex cover problem. *Neural Computing and Applications*, 2017, 28(7): 1775-1785
- [46] Cai S W, Su K L. Local search for Boolean satisfiability with configuration checking and subscore. *Artificial Intelligence*, 2013, 204: 75-98
- [47] Luo C, Su K, Cai S. More efficient two-mode stochastic local search for random 3-satisfiability. *Applied Intelligence*, 2014, 41(3): 665-680
- [48] Cai S, Luo C, Su K. Scoring functions based on second level score for k-SAT with long clauses. *Journal of Artificial Intelligence Research*, 2014, 51: 413-441
- [49] Drias H, Khabzaoui M. Scatter search with random walk strategy for SAT and MAX-W-SAT problems//*Proceedings of the Engineering of Intelligent Systems*. Berlin Heidelberg, Germany, 2001: 35-44



ZHOU Jun-Ping, born in 1981, Ph. D. candidate, lecturer. Her current research interests include intelligent planning and automated reasoning.

REN Xue-Liang, born in 1989, M. S. His current research interests include design and analysis of algorithms.

YIN Qian, born in 1990, M. S. Her current research interests include intelligent planning and automated reasoning.

LI Rui-Zhi, born in 1989, Ph. D. Her current research interests include design and analysis of algorithms.

YIN Ming-Hao, born in 1979, Ph. D., professor, Ph. D. supervisor. His current research interests include intelligent planning and automated reasoning.

Background

The maximum satisfiability problem (MaxSAT) has become a popular research field, which has a relative problem—minimum satisfiability problem (MinSAT). When solving some combinatorial optimization problems, converting them to MinSAT is faster than converting them to MaxSAT. Therefore, we propose a local search algorithm with strengthened configuration checking and clause weighting to solve MinSAT. The experimental results show that the strengthened configuration checking and clause weighting local search algorithm can solve MinSAT effectively, and it has good performance in the large-scale instances.

This paper is supported by the National Natural Science Foundation of China under Grant Nos. 61370156, 61403076, 61403077 and 61300099, the Research Fund for the Doctoral Program of Higher Education No. 20120043120017 and the Program for New Century Excellent Talents in University No. NCET-13-0724. The research project aims at discussing some combinatorial problems, the combinatorial problems can be solved by converting to MinSAT, so that new algorithms can be designed. Discussing the algorithm and complexity of MinSAT and its sub problems is one part of contents for the project.